

Article

Aerodynamic Shape Optimisation Using Parametric CAD and Discrete Adjoint

Dheeraj Agarwal ^{1,*} , Simão Marques ^{2,†}  and Trevor T. Robinson ^{1,†} ¹ School of Mechanical and Aerospace Engineering, Queen's University Belfast, Belfast BT9 5AH, UK² School of Mechanical Engineering Sciences, The University of Surrey, Guildford GU2 7XH, UK

* Correspondence: d.agarwal@qub.ac.uk

† These authors contributed equally to this work.

Abstract: This paper presents an optimisation framework based on an open-source CAD system and CFD solver. In this work, the high-fidelity flow solutions and surface sensitivities are obtained using the primal and discrete adjoint formulations of SU^2 . This paper shows the direct use of CAD models for optimisation by developing a CAD system application programming interface and creating a link between the CAD-MESH-CFD analysis. A methodology to obtain geometric sensitivities is introduced, enabling the calculation of accurate gradients with respect to CAD variables and the deformation of the analysis mesh during the optimisation process. This methodology guarantees that the new surface mesh lies exactly on the CAD geometry. The optimisation framework is applied to a rectangular wing and a three section high-lift aerofoil configuration derived from the NASA CRM-HL configuration. Both geometries are created using FreeCAD. The performance objectives are to decrease the drag while constraining the lift to be above a desired value. The twist distribution of the wing was parameterised within the CAD system, allowing the minimisation of the induced drag by obtaining a nearly elliptical lift distribution. For the high-lift configuration, the position and rotation of the flap and slat were parameterised with respect to the original section; the final optimised positions yield a drag reduction of approximately 16.5%. These results show that the CAD parameterisation can be reliably used to obtain efficient optimums while operating directly on the CAD geometries.

Keywords: shape optimisation; CAD; parameterisation; aerodynamics; discrete adjoint



Citation: Agarwal, D.; Marques, S.; Robinson, T.T. Aerodynamic Shape Optimisation Using Parametric CAD and Discrete Adjoint. *Aerospace* **2022**, *9*, 743. <https://doi.org/10.3390/aerospace9120743>

Academic Editor: Giuseppe Pezzella

Received: 18 October 2022

Accepted: 19 November 2022

Published: 23 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

To enable the use of CAD systems within gradient-based design optimisation frameworks, it is essential to develop a link between the parameterisation defined within CAD systems and the high-fidelity Computational Fluid Dynamics (CFD) analysis, which can provide the gradients of objective functions with respect to CAD design parameters. Adjoint methods [1–8] have been extensively developed over the last two decades and can provide the gradient of objective functions with respect to the movement of surface mesh node points on the geometry. Adjoint methods have been mainly classified into two categories: (a) continuous adjoint and (b) discrete adjoint. This categorisation is based on the mathematical formulations used to obtain the adjoint equations from the flow field equations. The adjoint method allows the computations of adjoint surface sensitivities which relate how the objective function changes for an infinitesimally small movement of each surface mesh node.

The parameters used to create a feature-based CAD model within a mechanical CAD system (e.g., CATIA V5, FreeCAD, etc.) can be accessed by using a suitable CAD system application programming interface (API). Modifying these parameters by small amounts typically causes movements of the boundary proportional to, and of the same order as, the size of the parameter perturbation. If the desire is to use the parametric CAD model

within the optimisation framework, this requires the computation of geometric sensitivities, or design velocities, which quantify the boundary movement with respect to a change in the individual parameter values [9]. The geometric sensitivities computed for different parameters combined with adjoint surface sensitivities can be used to compute the gradients required to drive a gradient-based optimisation algorithm.

In the past, authors have developed CAD-based optimisation processes using non-uniform rational B-splines (NURBS) patches using the NURBS control-point locations (the x , y and z coordinates) as the design variables. The advantage of this approach is that the NURBS represent a richer design space; however, the downside is that sometimes the NURBS control net may be too coarse in certain regions and would require a process to enrich the design space by adding more control points before it can be used for optimisation. This was addressed in Ref. [10] where an adaptive parameterisation approach was presented where the NURBS control net was refined by using a knot insertion and was subsequently used for the purpose of optimisation. The downside of these approaches is that, as they do not work directly on the parametric CAD model created in a feature-based CAD system, the design intent and the parametric associativity captured in the choice of features used to build the model is lost. Another efficient way to compute geometric sensitivities is by directly differentiating the mathematical expressions coded within the CAD systems by using automatic differentiation (AD). The idea behind this approach is to analytically differentiate all the mathematical operations performed by a computer program and accumulate the derivative values using the chain rule. Some authors have tried to differentiate different in-house CAD tools [11,12] and also the open-source CAD system Open-CASCADE [13]. These approaches to calculate geometric sensitivities have significant advantages as they do not require a geometry or mesh to be recomputed; however, the downside is that they require access to the underlying source code of the CAD system, which is highly unlikely to be available for the commercial CAD systems.

In previous work by the authors [9,14–16], CAD system APIs were used to access the parameters defined within a commercial CAD system, CATIA V5, and they were used within a gradient-based optimisation framework. However, the analysis mesh was regenerated at each optimisation step, limiting the applicability of the optimisation framework. To further improve the efficiency, this work proposes (a) a methodology for gradient evaluations using design velocities, which are a measure of the boundary movement with respect to a change in the parameter value, wherein the boundary movements are calculated in three axes directions (x , y and z), thus enabling the computation of gradients by linking them with adjoint sensitivities, and (b) to automatically, and rigorously, deform the analysis mesh while the CAD model is updated by the parameter changes during the optimisation. One of the limitations of the previous approach [9] was that the design velocities computed were not suitable to deform the analysis mesh, particularly in the areas with a high surface curvature and in the presence of rigid body translations and rotations.

This paper presents a method for the computation of the geometric sensitivities for the CAD parameters defined within an open-source CAD system, FreeCAD [17]. These geometric sensitivities complete the chain of derivatives required to evaluate the objective function gradient and can also compute the required displacements to deform the analysis mesh during the optimisation process. The results in this paper show that the approach is an effective way to move the surface mesh nodes during the optimisation process, providing a gateway to a fully automatic CAD-based optimisation framework using open-source software, FreeCAD and SU^2 , for aerodynamic shape optimisation problems.

The paper is structured as follows: Section 2 provides the details of the computation of the adjoint and geometric sensitivities and linking them to evaluate the objective function gradient. Section 3 elaborates the optimisation framework. The results are presented in Section 4, including the application of the methodology to optimise the performance of a rectangular wing and a high-lift aerofoil configuration. Section 5 further discusses the proposed methodology and the paper ends with the conclusions drawn in Section 6.

2. Methodology

2.1. Adjoint Sensitivities

The introduction of adjoint methods has been critical to enable optimisation using high-fidelity computational models by largely decoupling the computational cost of computing the derivative of the objective function with respect to the design variables. For aerodynamic shape optimisation, the solution of the adjoint equations allows the calculation of the surface sensitivities, i.e., what is the effect of a small perturbation to the surface mesh nodes on the objective function. In this paper, the surface sensitivities are computed using the discrete adjoint formulation in SU^2 [7], which provides adjoint sensitivities in the x, y and z directions as ϕ_x, ϕ_y and ϕ_z . The adjoint equations are formulated using the following methodology:

Consider a semi-discrete system of fluid conservation laws described as

$$\frac{d\mathbf{U}}{dt} = \mathbf{R}(\mathbf{U}, \mathbf{X}). \quad (1)$$

In Equation (1), $\mathbf{U}$ is the vector of the fluid system variables, $\mathbf{R}$ is the residual of the fluid equations that are a function of the system variables and mesh nodes, $\mathbf{X}$. For steady-state problems, the solution of Equation (1), also known as the *primal* solution, involves driving the non-linear residual $\mathbf{R}$ to zero. For aerodynamic shape optimisation, the objective function, J , will typically depend upon the flow variables $\mathbf{U}$ and a set of parameters, θ , that control the shape and therefore the mesh nodal positions:

$$J = J(\mathbf{U}, \mathbf{X}(\theta)). \quad (2)$$

The gradient of the objective function with respect to the design variables can be defined by applying the following chain rule:

$$\frac{dJ}{d\theta} = \frac{dJ}{d\mathbf{X}} \frac{d\mathbf{X}}{d\mathbf{X}_s} \frac{d\mathbf{X}_s}{d\theta}. \quad (3)$$

The volumetric sensitivity term $dJ/d\mathbf{X}$ can be obtained by differentiating Equation (2) with respect to $\mathbf{X}$ as

$$\frac{dJ}{d\mathbf{X}} = \frac{\partial J}{\partial \mathbf{X}} + \frac{\partial J}{\partial \mathbf{U}} \frac{d\mathbf{U}}{d\mathbf{X}}. \quad (4)$$

The solution of Equation (4) using finite differences requires the solution of Equation (1) for each design variable. Alternatively, it can be shown that by selecting an arbitrary vector ψ , Equation (4) can be re-written as [18]:

$$\frac{dJ}{d\mathbf{X}} = \frac{\partial J}{\partial \mathbf{X}} + \psi^T \frac{\partial \mathbf{R}}{\partial \mathbf{X}}. \quad (5)$$

The critical advantage of this method is that only one additional PDE system of equations needs to be solved for each objective or constraint function to obtain the volumetric sensitivities with respect to all design parameters. These sensitivities can then be combined with the mesh deformation algorithm to produce the surface sensitivities Φ as

$$\Phi = \frac{dJ}{d\mathbf{X}_s} = \frac{dJ}{d\mathbf{X}} \frac{d\mathbf{X}}{d\mathbf{X}_s}, \quad (6)$$

The sensitivity Φ is a vector quantity with components $[\phi_x, \phi_y, \phi_z]$ along the three axes (x, y, z). The end result of using the adjoint method to compute sensitivities is a method that does not scale with the number of design variables, hence significantly reducing the computational cost required to evaluate gradients.

2.2. Geometric Sensitivities

In this paper, the geometric sensitivities are the change in the position of a point lying on the CAD model boundary following a unit change in the individual parameter value. Chen and Torterelli [19] used the parametric position of points on the boundary of the baseline and perturbed CAD model to compute the geometric sensitivities. The authors analysed a structural optimisation problem using the approach and concluded that the accuracy of the computed results relied on the quality of the finite element mesh. Truong et al. [20] outlined an approach for deforming the surface mesh, where the tessellation of CAD surfaces was performed, and parametric definitions of mesh node points were obtained with respect to these tessellated surfaces. The baseline and perturbed models were compared using finite differences to obtain the geometric sensitivities. Sun et al. [21] applied colour to each face to create a unique tag to facilitate the mapping process and used virtual topology-based procedure to create an analysis topology representation of the baseline and perturbed model. The virtual boundary entities in the model were parameterised to compute both the parametric position of a node in the original model and in the perturbed model.

In this work, the geometric sensitivities are computed in the three Cartesian directions, i.e., x , y and z , and are based on the models created within the open-source CAD system FreeCAD, but the approach can be extended to other CAD systems. As a first step, a CAD system API is created to interact with FreeCAD and assigns a unique colour to each face in the model. Secondly, a surface mesh representation of the boundary is created using the open-source meshing system GMSH [22], which is then used to obtain the surface face colour in the CAD model on which the mesh nodes lie. The normalised parametric location (u, v) of these mesh nodes (x_b, y_b, z_b) is obtained with respect to the faces in the CAD model within FreeCAD, such that $u, v \in [0, 1]$. After a parametric perturbation, the mapping of the faces is completed by comparing the colour of the faces in the baseline and perturbed model. The normalised (u, v) coordinates of the mesh nodes in the baseline model are then scaled using the new parametric range to find the new (x_p, y_p, z_p) coordinates of the mesh nodes in the perturbed model. Thereafter, the design velocities are computed using Equation (7) for each mesh node as

$$V_x = \frac{x_p - x_b}{d\theta}, \quad V_y = \frac{y_p - y_b}{d\theta}, \quad V_z = \frac{z_p - z_b}{d\theta} \quad (7)$$

where (x_b, y_b, z_b) is the position of the surface mesh node on the original or unperturbed geometry; (x_p, y_p, z_p) defines the position of surface mesh node on the geometry obtained by perturbing a parameter by $d\theta$; V_x, V_y, V_z are the computed geometric sensitivities in the x, y and z directions, respectively. It is acknowledged that this approach is only applicable for parametric perturbations where there is no change in the boundary topology of the model. In addition, this work assumes that the colour assignment will not change due to the parametric change. The complete algorithm for the computation of geometric sensitivities is described in Algorithm 1.

Figure 1 shows the geometric sensitivities computations for a box model, where the parametric perturbation changes the geometry from the baseline to the perturbed shape shown in Figure 1a. Figure 1b shows the design velocities computed in the direction normal to the surface (V_n) following the approach described in Ref. [9], while Figure 1c,d present the geometric sensitivities computed using the approach described in Algorithm 1. Both approaches produce the same result for the two faces (top and right) that are moving in the normal direction; however, the former approach gives a zero value for the design velocities for all other faces (as they are moving in the tangential direction). The latter approach, introduced in this work, gives the information about the movement in the direction of the three axes, which is essential to apply smooth deformations on the analysis mesh during the optimisation cycle.

$$\nabla J = \frac{\Delta J}{\Delta \theta} \quad (8)$$

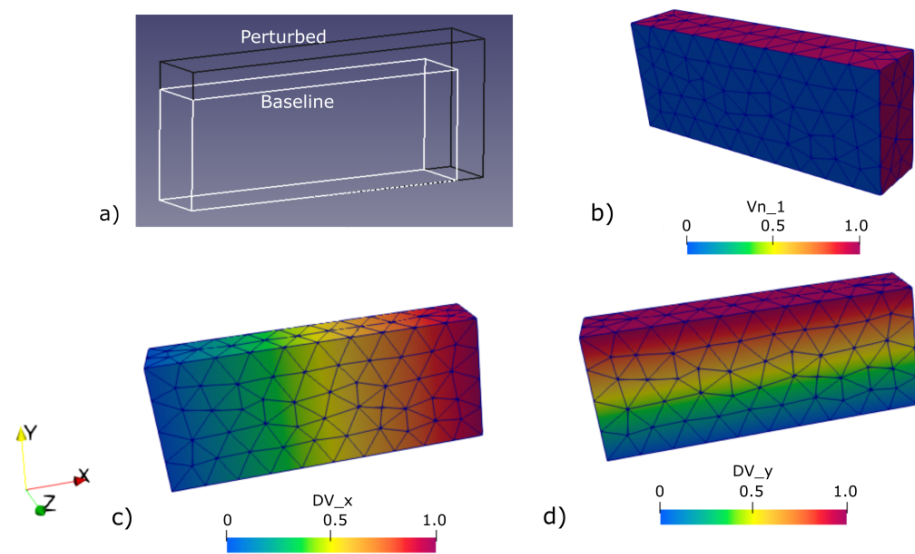


Figure 1. Computation of geometric sensitivities or design velocity: (a) box model showing baseline and perturbed geometry; (b) design velocity computed in normal direction (V_n) using approach described in Ref. [9]; geometric sensitivities in (c) x direction and (d) y direction.

Algorithm 1 Computation of Geometric Sensitivities

```

1: Input: Parametric FreeCAD model
2: Output: Design Velocity in  $x$ ,  $y$  and  $z$  directions
3: tag each unique CAD model surface faces as a colour (using CAD system API)
4: compute the design velocities at each surface mesh node created using the baseline CAD model
5: for mesh node  $i \leftarrow 1$  to  $N$  do
6:   store information of the face in baseline model
7:   compute in which face each mesh node ( $x_b^i, y_b^i, z_b^i$ ) lies and its corresponding parametric coordinate ( $u_i, v_i$ )
8: end for
9: for parameters  $j \leftarrow 1$  to  $n$  do
10:  perturb  $j^{th}$  parameter by small value ( $d\theta$ )
11:  for mesh node  $i \leftarrow 1$  to  $N$  do
12:    find the face with same tag (on which  $i$ th node lies in baseline model) in the perturbed model
13:    use the parametric coordinates ( $u_i, v_i$ ) to get the  $x_p^i, y_p^i, z_p^i$  location of mesh node in perturbed model
14:    calculate geometric sensitivities  $V_{xi}, V_{yi}, V_{zi}$  for each mesh node  $i$ 
15:  end for
16: end for

```

2.3. Gradient Evaluation

For the optimiser to establish a new search direction, it is necessary to evaluate the gradient with respect to each design variable. In this particular case, it means evaluating the change in objective function due to unit perturbation of a CAD parameter. This can be achieved through the use of the chain rule, as shown:

$$\begin{bmatrix} \frac{\partial J}{\partial \theta_1} \\ \frac{\partial J}{\partial \theta_2} \\ \vdots \\ \frac{\partial J}{\partial \theta_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial x_1}{\partial \theta_1} & \cdots & \frac{\partial x_m}{\partial \theta_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_1}{\partial \theta_n} & \cdots & \frac{\partial x_m}{\partial \theta_n} \end{bmatrix} \begin{bmatrix} \frac{\partial J}{\partial x_1} \\ \frac{\partial J}{\partial x_2} \\ \vdots \\ \frac{\partial J}{\partial x_m} \end{bmatrix} \quad (9)$$

where J represents the function of interest (objective or constraint); n and m are the number of design variables and surface mesh points, respectively; x_i represents the displacement of a discrete grid point on the surface. The third term represents the surface sensitivities obtained using the adjoint solution, i.e., the change in the function of interest with respect to a change in the surface grid points. The Jacobian $\frac{\partial x}{\partial \theta}$ is known as the geometric sensitivity matrix and is used to compute the influence of each design variable (θ_i) on the position of each grid point of the surface mesh. These geometric sensitivities or design velocities are computed using the methodology outlined in Section 2.2.

Once the adjoint surface sensitivity (ϕ) and geometric sensitivities ($\mathbf{V}$) are obtained, the total change in objective function can be predicted as the summation over the boundary:

$$\Delta J = \sum_s (\phi \cdot \mathbf{V}) \quad (10)$$

Following a parametric perturbation, the objective function changes and the gradient ∇J can be computed by normalising the change computed in Equation (10) with respect to the size of the parameter perturbation used to perturb the boundary as:

3. Optimisation Framework

In general, optimisation framework can be formulated as:

$$\begin{aligned} &\text{Minimise } \theta : J(\theta) \\ &\text{subject to : } g(\theta) \geq 0 \\ &h(\theta) = 0, \end{aligned} \quad (11)$$

where $J(\theta)$ is the objective function to be minimised (maximised), $g(\theta)$ gives the inequality constraints and $h(\theta)$ represents equality constraints. In this work, a gradient-based optimisation framework is used where the gradients of the objective/constraints are computed and used to drive the optimisation to reach the optimum. The first step is to define a search direction (using the computed gradients) for the optimisation to proceed. The second step is the line-search operation which defines the step size to be taken in the search direction to minimise (maximise) the objective function. The most commonly used gradient-based optimisation algorithm is the gradient descent or steepest descent method, which minimises the objective function $J(\theta)$ by changing the parameters proportional to their individual sensitivities with respect to the objective function (ΔJ). In this work, the optimisation method is based on the sequential least square programming (SLSQP) algorithm, which is part of a class of Sequential Quadratic Programming algorithms and is highly efficient in the line-search process, typically resulting in faster convergence than the aforementioned method. The flowchart of the optimisation framework created as part of this work is shown in Figure 2.

The optimiser starts with a baseline parametric CAD model created in FreeCAD, surface and volumetric mesh which are used for the CFD primal and adjoint analysis. A python CAD system API is configured such that it perturbs the model parameters within FreeCAD and links them with the surface mesh nodes to obtain the geometric sensitivities in the x , y and z directions using Equation (7), i.e., (V_x, V_y, V_z) . These geometric sensitivities are then linked with the adjoint surface sensitivities (ϕ_x, ϕ_y, ϕ_z) obtained from the CFD adjoint analysis to compute the performance gradient with respect to the CAD model

parameters. These gradients are used with the gradient-based optimisation algorithm SLSQP to reach the optimised geometry. If the optimisation criteria are not met, the CAD system API updates the CAD model parameters using the values provided by the optimiser. Another CAD system API is configured to use the information from the baseline surface mesh to obtain the updated locations such that the mesh nodes are snapped to the boundary of the updated CAD model. The updated surface mesh nodes are then used to deform the volumetric mesh in SU^2 and subsequently used for primal and adjoint analysis, while the CAD model is used to compute the geometric sensitivities which are then used to compute the gradients to drive the optimisation process.

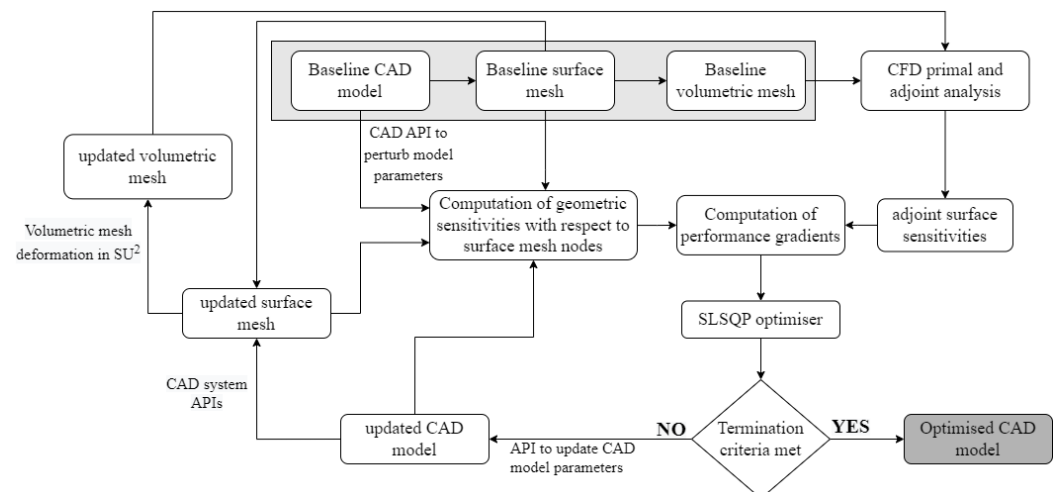


Figure 2. Optimisation flowchart.

3.1. Flow and Adjoint Solver

SU^2 provides an open-source suite of computational analysis tools, including a state-of-the-art flow and adjoint solver suitable for aerodynamic design optimisation. The main flow solver uses a node-centred finite-volume method to discretise the fluid flow equations, with levels of fidelity ranging from inviscid to Detached Eddy Simulations. The spatial discretisation schemes available include the Jameson–Schmidt–Tukel (JST) and Roe schemes, among others. Both flow and adjoint solutions can be marched forward in time using a Runge–Kutta explicit scheme or a backward Euler implicit method, with or without multi-grid acceleration [23]. The SU^2 suite is able to solve the discrete and continuous adjoint Euler/RANS equations. The discrete adjoint problem and associated gradients are obtained by algorithmic differentiation of the primary solver, including the turbulence model equations. The volume mesh deformation is obtained by solving a linear elasticity problem, whose gradients are also obtained by algorithmic differentiation; further details on the discrete adjoint method in SU^2 are given in ref. [7].

3.2. Mesh Deformation

In any optimisation framework, there is a requirement to update the model geometry and respective mesh as the optimisation progresses. In the field of high-fidelity CFD simulations, it is of utmost importance to have a good quality mesh to obtain accurate flow predictions. In general, a substantial effort goes into producing a high-quality mesh for the baseline model, and it is extremely challenging to automatically re-mesh a complex geometry at each optimisation step. Thus, the mesh deformation becomes an important step in which the new coordinates are computed for the mesh nodes whilst maintaining the mesh connectivity.

The first step in the optimisation process is the creation of the CFD mesh from the CAD model, which ascertains that the initial mesh conforms to the CAD model boundary. This is checked as part of the optimisation framework by formulating a CAD system API for FreeCAD which computes the distances of the CFD mesh nodes from the CAD model

boundary to be within a tolerance of 10^{-6} mm. Thereafter, a method similar to that outlined in Section 2.2 is used to compute the parametric location of the CFD mesh nodes with respect to the faces in the baseline CAD model. Subsequently, the parameters in the CAD model are modified to obtain the optimised CAD model followed by mapping the faces by comparing the colour information in the baseline and perturbed model. The parametric coordinates of the mesh nodes (in the baseline model) are then used to obtain the new coordinates of the mesh nodes which conform to the perturbed model boundary. Because this methodology directly uses the CAD model boundary information for computing the new surface mesh points, it ensures that the deformed CFD mesh nodes lie directly on the CAD model boundary. The surface deformation is propagated to the volumetric mesh during the optimisation process by using the *SU2_DEF* module within *SU2* as shown in Figure 3. This approach can also be configured to snap the CFD mesh to the CAD model boundary if the CFD mesh nodes are at an offset distance that exceeds a tolerance limit.

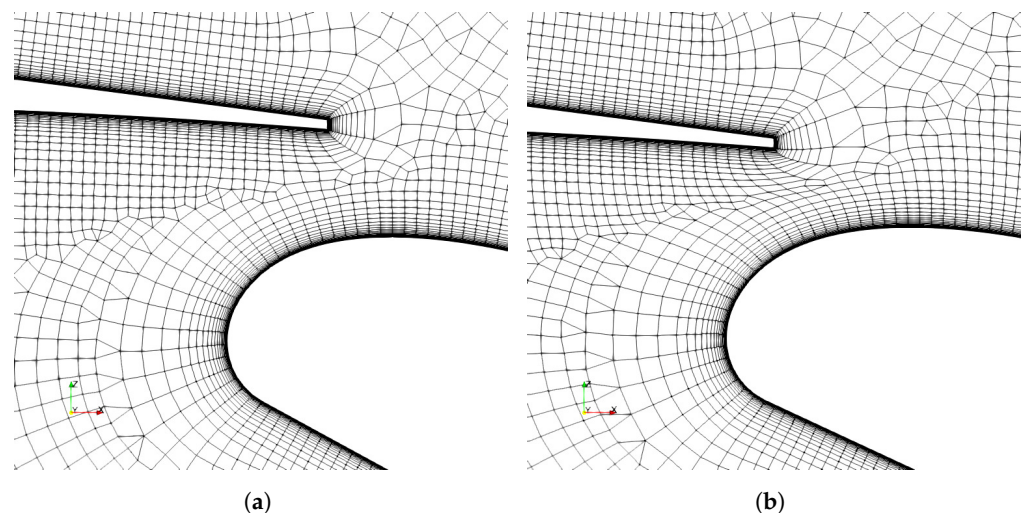


Figure 3. Mesh deformation due to the movement of flap for a three-element aerofoil. (a) Initial mesh. (b) Deformed mesh.

4. Test Cases and Results

4.1. Case 1: Twist Optimisation of Rectangular Wing

The first test case considered is the optimisation of the twist distribution along the span of a rectangular wing (with NACA0012 sections) to minimise the induced drag in an inviscid subsonic flow. This test case is one of the benchmark test cases defined by the AIAA Aerodynamic Design Optimization Discussion Group, with no wing-tip cap. The purpose of this test case is to obtain an elliptic lift distribution on the wing and a span efficiency factor approximately equal to one.

4.1.1. Problem Definition

The problem consists of the minimisation of the drag coefficient, C_D , with the coefficient of lift, C_L , acting as a constraint by requiring it to be equal or greater than 0.375. The angle of attack is selected to allow the problem to start at a feasible point, i.e., making the C_L marginally larger than 0.375. The flow conditions and optimisation problem are defined as:

- The freestream Mach number = 0.5.
- The angle of attack (AOA) = 4.3° .
- The objective function = $\min(C_D)$.
- The constraint: $C_L \geq 0.375$.

4.1.2. Geometry and Mesh Description

The baseline model is created in FreeCAD by defining six equally spaced sections along the wing span defined by the B-spline curves, each describing the NACA0012 aerofoil. The sections are parameterised to allow rotation about the leading edge of the wing. In total, there are six parameters controlling the twist of the wing along the span, as shown in Figure 4a. A lofted surface is created through the profile using a cubic B-spline interpolation to create the wing surface, as shown in Figure 4b. The semi-span length and semi-span area of the wing are defined as $b/2 = 3c$ and $S = 3c^2$, where c is the wing chord ($c = 1$ m).

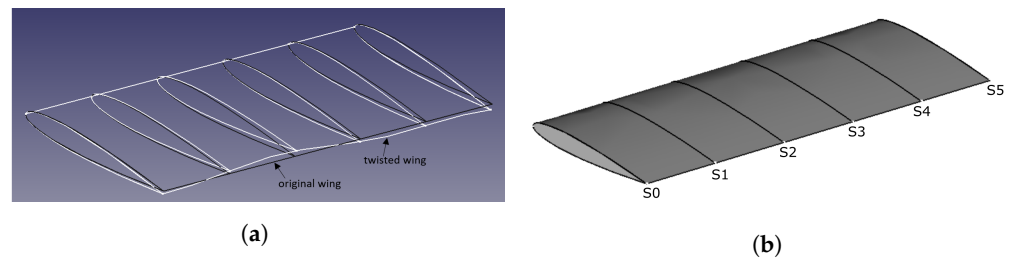


Figure 4. Wing design: (a) original and twisted wing profiles; (b) CAD model.

The analysis mesh is generated using the ICEM-CFD, as shown in Figure 5. The far-field is located at a distance of $25c$ away from the wing's leading edge, tip, upper and lower surface and is at $50c$ in the downstream direction, away from the trailing edge. The grid points are clustered near the leading and trailing edges and also clustered at the wing tip to capture the main flow features. A grid independence study was conducted using four different meshes, as shown in Table 1. The fine density mesh was selected for the optimisation, consisting of nearly 1.3 million cells.

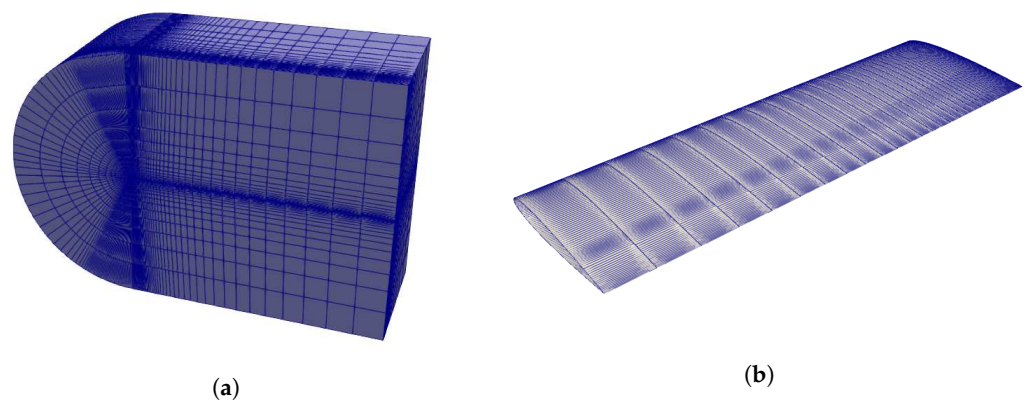


Figure 5. Mesh for the rectangular wing geometry. (a) Far field view. (b) Wing surface mesh.

Table 1. Grid independence study for the wing.

Mesh Level	Elements	C_D	C_L
very coarse	29,369	0.028987	0.35504
coarse	124,526	0.011381	0.373843
medium	526,747	0.008691	0.382739
fine	1,297,134	0.008500	0.383678
very fine	7,212,505	0.008354	0.38500

4.1.3. Validation of Gradients

Because this is a constrained optimisation problem, it requires the computation of the adjoint sensitivities using the drag and lift as the objective functions. The convergence of

the residuals of the primal and adjoint conservation of the mass equations for the baseline geometry are shown in Figure 6.

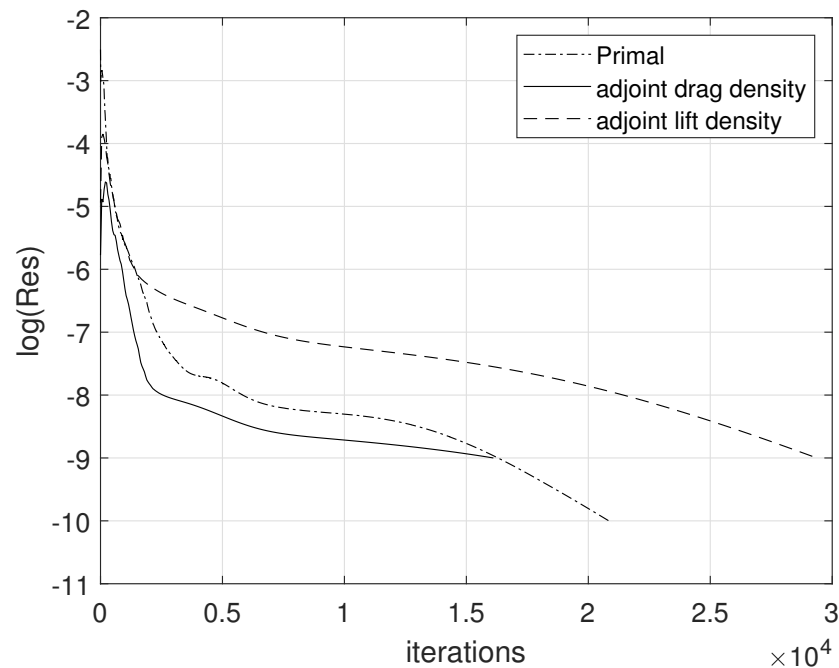


Figure 6. Convergence of the primal and adjoint CFD for the baseline rectangular wing geometry.

The performance gradients obtained by linking the CAD parameterisation with adjoint surface sensitivities are validated using the values obtained by the following finite-difference approximation:

$$\frac{\Delta f}{\Delta \theta} = \frac{f(\theta + \varepsilon) - f(\theta)}{\varepsilon}, \quad (12)$$

where ε is a small perturbation, f is the performance function and θ is the design variable. The CFD flow solutions are obtained using the geometries with small perturbations (0.1° twist), and then the finite-difference gradients are computed using Equation (12). For the perturbed model, the mesh deformation strategy outlined in the previous section has been used to obtain the mesh for the CFD simulation. These gradients are then compared with those obtained using the methodology proposed in Section 2.3, using the geometric sensitivities and adjoint surface sensitivities (see Table 2). The gradients for the drag and lift computed using the proposed meshes are within 1.5% of the values obtained using the finite differences, giving the necessary confidence in the developed framework.

Table 2. Validation of performance gradients computed using adjoint + CAD and finite difference.

Station	Adjoint + CAD (A)		Finite Difference (B)		A-B (%)	
	∇C_D	∇C_L	∇C_D	∇C_L	∇C_D	∇C_L
S0	0.000282243	0.00771650	0.00028398	0.00770876	0.62	0.10
S1	0.000975287	0.02596906	0.00098722	0.02597942	1.22	0.04
S2	0.000709438	0.01744390	0.00071825	0.01746158	1.24	0.10
S3	0.000726939	0.01817444	0.00073523	0.01818534	1.14	0.06
S4	0.000879464	0.01750809	0.00088931	0.01753582	1.12	0.16
S5	0.000227776	0.00391706	0.0002303	0.00394805	1.11	0.79

4.1.4. Optimisation Results

To assist with the analysis of the optimisation framework, the span efficiency (e) factor for the wing designs is computed as

$$e = \frac{C_L^2}{\pi AR C_D}, \quad (13)$$

where $AR = b^2/(2S) = 6$ is the aspect ratio. The evolution of C_D and the span efficiency factor throughout the optimisation are plotted in Figure 7a,b, respectively. Figure 8a shows the lift distribution across the wing span for the initial and optimised geometries compared to the theoretical elliptical lift distribution. It can be seen that the optimiser is able to achieve the lift constraint whilst recovering a near elliptical span-load distribution up until the wing tip. The obtained twist distribution is shown in Figure 8b, where the twist angles are measured with respect to the angle of attack. As expected, the inboard wing produces a positive twist to increase the lift while the outboard sections produce a negative twist to unload the wing.

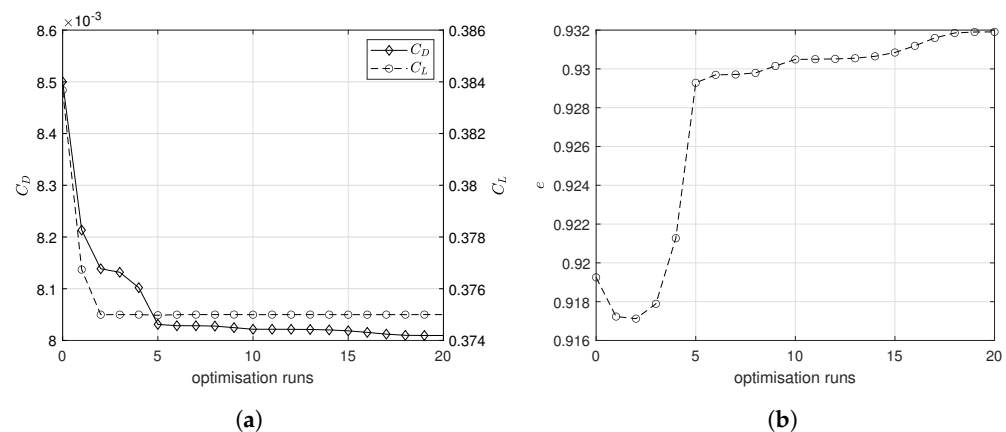


Figure 7. Twist optimisation of rectangular wing. (a) Coefficient of drag and lift. (b) Span efficiency factor.

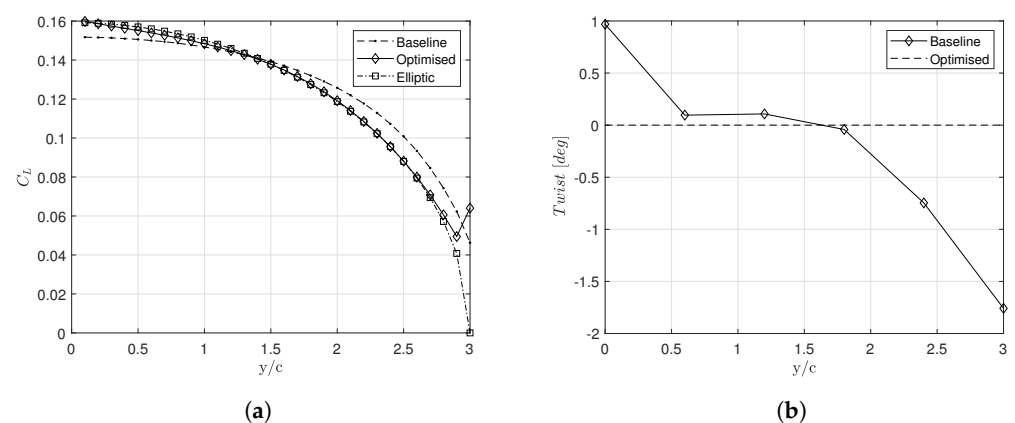


Figure 8. Lift and twist distributions for the baseline and optimised wing geometries. (a) Lift distribution. (b) Twist distribution.

Figure 9 shows the deviation of the surface mesh points from the CAD model during the optimisation process. As it can be seen, the baseline block-structured mesh created in the ICEM-CFD has a maximum and average deviation of the order of 10^{-6} mm, which has been subsequently reduced to a value of the order of 10^{-12} mm, guaranteeing that the new surface mesh points lie very close to the CAD geometry.

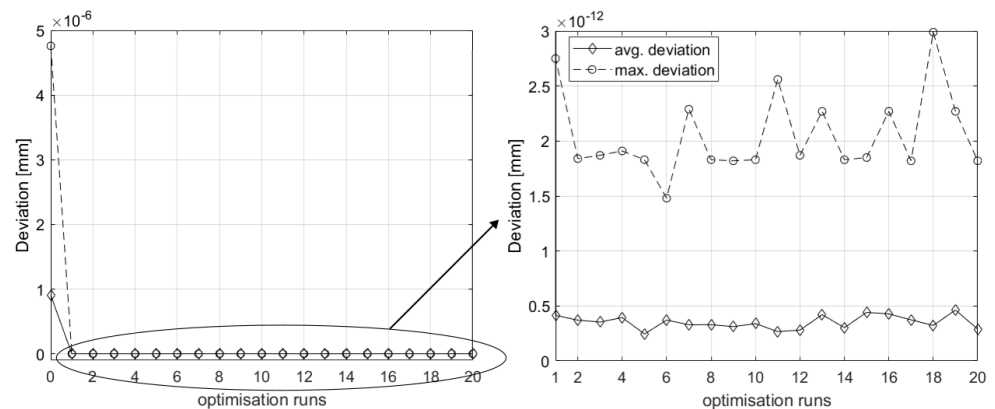


Figure 9. Deviation of surface mesh from CAD during optimisation.

4.2. Case 2: Optimisation of Three-Section Aerofoil Surface

The second test case is the optimisation of the position and orientation of the slat and flap of a three-element aerofoil with the fixed section of the main wing. This aerofoil configuration is derived from a section of the NASA CRM-HL configuration [24]. It is a “kink cut” from the 3D configuration (perpendicular to the leading edge up to the 25% chord, then chordwise going aft), with the chordwise part located at $y = 16.26$ m of the full-scale NASA CRM-HL configuration. This geometry was used in a “Mesh Effects for CFD Solutions” special session at AIAA Aviation 2020, sponsored by the Geometry and Mesh Generation Workshops (GMGW) committee [25].

4.2.1. Problem Definition

The problem is formulated as a minimisation objective with C_D acting as the objective function and C_L constrained to remain equal or larger than 99% of the baseline configuration. The flow conditions and optimisation problem are defined as:

- The freestream Mach number = 0.2.
- The angle of attack (AOA) = 8° .
- The Reynolds number = 5×10^6 .
- The objective function = $\min(C_D)$.
- The constraint: $C_L \geq 2.54$.
- The turbulence model: Spalart–Allmaras.

4.2.2. Mesh Description and Geometry Parameterisation

The mesh used to start the optimisation process was obtained from the AIAA CFD High-Lift Prediction Workshop (<https://turbmodels.larc.nasa.gov/multielementverification.html> (accessed on 17 October 2022)) and is shown in Figure 10. The mesh consisted of mainly quadrilateral elements, conforming to the aerofoil surfaces, containing a total of 1528 nodes along the wing surface, 500 nodes along the surface of the flap and 250 nodes on the slat surface. For the baseline configuration, the values of C_D and C_L are obtained as 0.044842 and 2.55789, respectively.

The CAD model of the aerofoil configuration is created within FreeCAD using the information from the CFD mesh. A python CAD system API is configured to read the surface mesh nodes of the main wing, slat and flap sections and form the two-dimensional three-section aerofoil. The position and orientation of the flap and slat section are configured to change in the x and y directions with respect to the user-defined parameters, as shown in Figure 11. This results in three design variables for each slat and flap (Δx , Δy and $\Delta \alpha$), respectively. The rotation centres are located at the midpoints of the individual chords of each element. This type of parameterisation allows to optimise the positioning of the high-lift devices to maximise the performance, without altering the shape of the aerofoil, which is often designed based on the cruise condition of the aircraft.

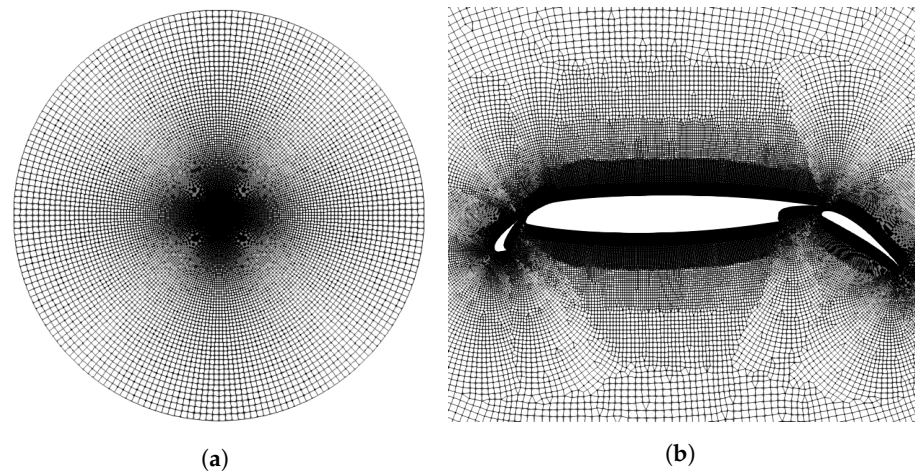


Figure 10. Mesh for the high-lift aerofoil configuration, along with slat and flap. (a) Far-field view. (b) Close-up view.

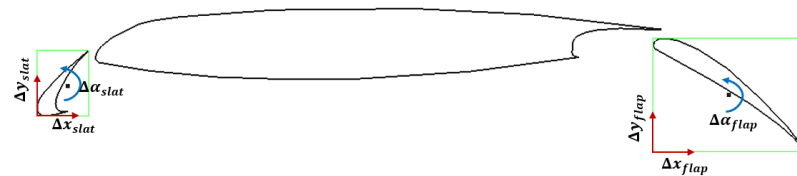


Figure 11. Parameterisation of a three-section aerofoil high-lift configuration.

The local chord (measured with the flap and slat stowed) is $c = 1$ m, and the CAD model created in FreeCAD is shown in Figure 12. The design parameters are bounded and are allowed to move within a range as shown below:

$$\begin{aligned}
 -0.015c &\leq \Delta x_{slat} \leq 0.015c \\
 -0.015c &\leq \Delta y_{slat} \leq 0.015c \\
 -0.015c &\leq \Delta x_{flap} \leq 0.015c \\
 -0.015c &\leq \Delta y_{flap} \leq 0.015c \\
 -5 &\leq \Delta \alpha_{slat} \leq 5 \\
 -5 &\leq \Delta \alpha_{flap} \leq 5
 \end{aligned} \tag{14}$$

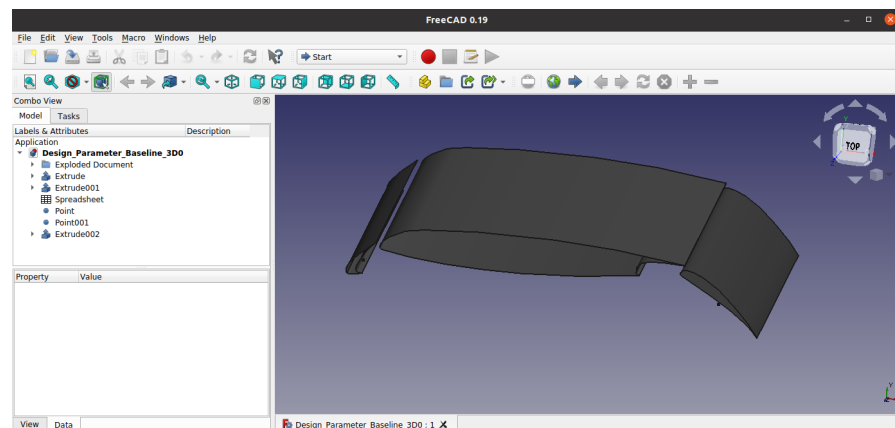


Figure 12. Three-dimensional view of the high-lift configuration aerofoil (extruded in z direction).

4.2.3. Optimisation Results

The problem is set up as a constrained optimisation problem, and this requires the computation of both the objective function (C_D) as well as the constraint function (C_L); hence, it was necessary to run one primal and two adjoint CFD simulations (with C_D and

C_L as the objectives). A typical convergence of the residuals of the adjoint density for the baseline geometry is shown in Figure 13.

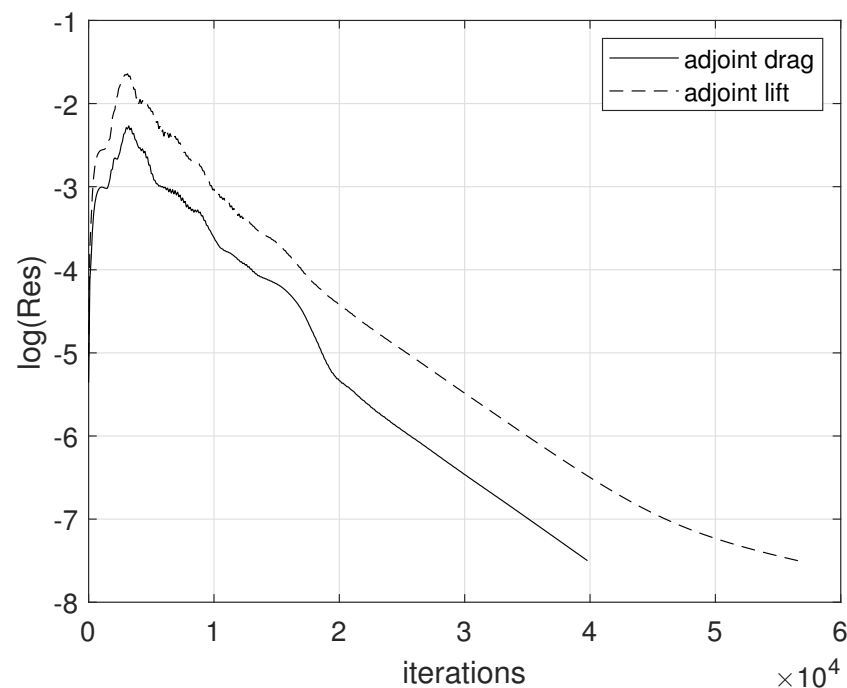


Figure 13. Convergence of the adjoint for the baseline aerofoil geometry.

The optimisation runs for 30 iterations, and the optimum location and orientation of slat and flap found are shown in Table 3. The C_D and C_L evolution during the optimisation process is plotted in Figure 14a, where the C_D and C_L values are plotted for each optimisation step. The optimised geometry resulted in a reduced drag value of $C_D = 0.037465$ and a lift of $C_L = 2.54016$, i.e., an approximately 16.5% decrease in C_D with respect to the baseline configuration. A comparison of the baseline and optimised locations of the slat and flap with respect to the main wing of the aerofoil is shown in Figure 14b. The Mach number contours and surface C_p values for the initial and optimised high-lift aerofoil geometries are shown in Figures 15 and 16, respectively. Figure 15 shows the new location of the slat shifting the slat wake away from the surface of the main element and mitigating the interaction with the main element boundary layer; the new position of the flap, together with the aforementioned changes, allows a more efficient pressure recovery of the flow over the flap, as indicated by Figure 16.

Table 3. Optimised values of the design parameters.

	Δx	Δy	$\Delta \alpha$ [deg]
slat	−0.00743c	0.00366c	5
flap	0.01144c	0.00705c	−4.011

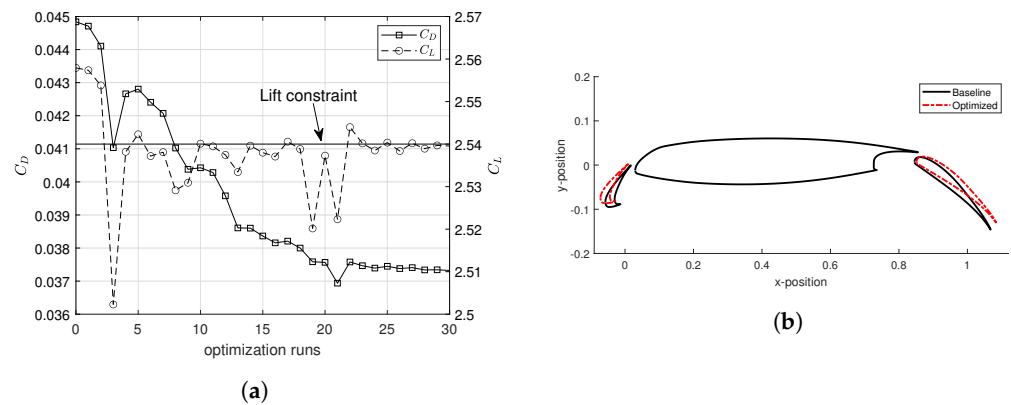


Figure 14. Optimisation of position and orientation of the slat and flap. (a) Coefficient of drag and lift. (b) Baseline and optimised designs.

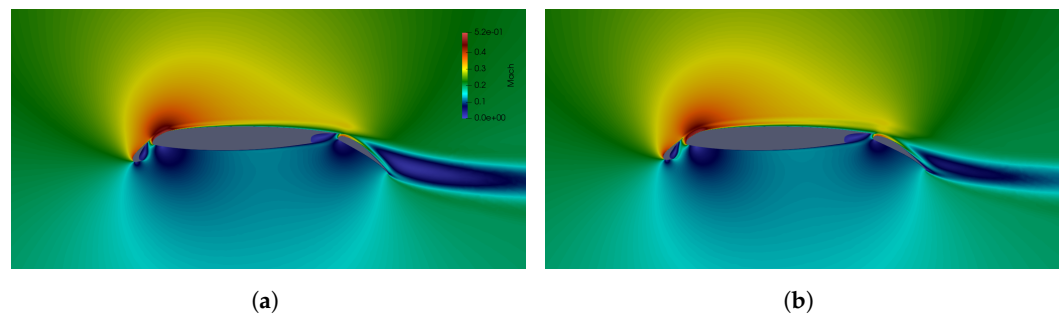


Figure 15. Mach no. contours for the baseline and optimised aerofoils. (a) Baseline geometry. (b) Optimised geometry.

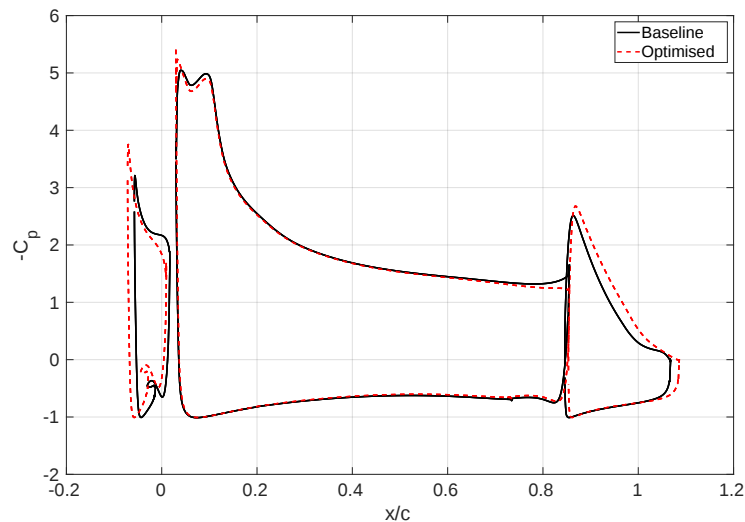


Figure 16. Surface C_p distribution for the baseline and optimised aerofoils.

5. Discussion

This paper describes a geometric method that is used to: (i) combine the geometric sensitivity with the adjoint sensitivity and (ii) deform the surface and volumetric mesh for use within the optimisation framework. For the computation of the geometric sensitivities in this work, a CAD system API was written to directly interact with the CAD model in FreeCAD to assign and read the colour attribute of each face before and after the parametric perturbation, rather than extracting this information from the STEP file as has been performed in Ref. [21]. This helps to avoid an additional step of exporting and reading the STEP file to obtain the topology information of the CAD model. While colour was used here, any method for assigning an individual identifier to a face in a model

can be used. The adjoint surface sensitivities extracted from the discrete adjoint solver gives the information about the performance changes with respect to the movement of the surface mesh nodes in the three axes directions (x , y and z). The geometric sensitivities are also computed in the these axes directions using the methodology outlined in Section 2.2, enabling the computation of the performance gradients with respect to the CAD parameters as described in Section 2.3.

This work is an advancement over the previous approaches used for computing geometric sensitivities [9,14], where the baseline and perturbed model were discretised using facets and the accuracy of the computations was dependent on the facet sizing. In this work, the geometric sensitivities are directly computed within the CAD system without any approximations using facets. This helps to improve the accuracy of the sensitivity computations, particularly in areas with a high curvature. This has been analysed in Section 4.1, where the twist distribution of the rectangular wing was optimised to minimise the drag on the wing. In addition, this approach also enables the computations of accurate geometric sensitivities in the scenario of rigid body motions (translation and rotation), which the previous approach [9,14] was not able to cater for, as the geometric sensitivities were computed by using projections in the direction normal to the surface of the baseline model. This has been analysed in Section 4.2, where the translation and rotation of the slat and flap section of the wing were used as the optimisation parameters to enhance the performance. The surface mesh deformation methodology demonstrated in this work computes the required displacements in the x , y and z directions by directly querying the CAD model within the CAD system, thus guaranteeing the deformed mesh to lie directly on the CAD geometry. This has been demonstrated for the 3D wing geometry case in Figure 9, where the maximum deviation of the surface mesh points is in the range of 10^{-12} mm.

For the examples in this paper, the computations of the geometric sensitivities and surface mesh deformations were carried out on a laptop workstation with 16 GB of RAM. For the test cases shown in this paper, the time for computing the geometric sensitivities at each mesh node and the required mesh deformation was in the range of 20–30 s, which is negligible with respect to the cost of determining how to move the mesh manually, or the cost of the primal and adjoint analysis.

6. Conclusions

In this paper, an efficient methodology is presented for computing the geometric sensitivities with respect to CAD parameters which is also used for the movement of surface mesh during the optimisation process, with the guarantee that the new surface points lie on the CAD geometry. In addition, the design features of an open-source CAD system, FreeCAD, are explored along with the adjoint capabilities of an open-source CFD solver, SU^2 . This methodology is generic and can be extended to work with other CAD systems and CFD solvers. The applicability of the developed framework is tested in the optimisation of the twist distribution of a rectangular wing and the position of the slat and flaps for a high-lift aerofoil configuration. It is shown that the methodology is robust for a range of different types of CAD-based geometry parameterisations and delivers significant performance improvements.

Author Contributions: Methodology, D.A., S.M. and T.T.R.; Software, D.A.; Investigation, D.A., S.M. and T.T.R.; Writing—original draft, D.A.; Writing—review & editing, D.A., S.M. and T.T.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: The CFD computations for the test cases were undertaken on Barkla, part of the High-Performance Computing facilities at the University of Liverpool, UK.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Giles, M.B.; Pierce, N.A. An introduction to the adjoint approach to design. *Flow Turbul. Combust.* **2000**, *65*, 393–415. [\[CrossRef\]](#)
2. Giles, M.B.; Duta, M.C.; Muller, J.D.; Pierce, N.A. Algorithm developments for discrete adjoint methods. *AIAA J.* **2003**, *41*, 198–205. [\[CrossRef\]](#)
3. Brezillon, J.; Gauger, N. 2D and 3D aerodynamic shape optimisation using the adjoint approach. *Aerosp. Sci. Technol.* **2004**, *8*, 715–727. [\[CrossRef\]](#)
4. Mader, C.A.; Martins, J.R.; Alonso, J.J.; Van Der Weide, E. ADjoint: An approach for the rapid development of discrete adjoint solvers. *AIAA J.* **2008**, *46*, 863–873. [\[CrossRef\]](#)
5. Roth, R.; Ulbrich, S. A discrete adjoint approach for the optimization of unsteady turbulent flows. *Flow Turbul. Combust.* **2013**, *90*, 763–783. [\[CrossRef\]](#)
6. Zhou, B.Y.; Albring, T.A.; Gauger, N.R.; Economon, T.D.; Palacios, F.; Alonso, J.J. A discrete adjoint framework for unsteady aerodynamic and aeroacoustic optimization. In Proceedings of the 16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Dallas, TX, USA, 22–26 June 2015; p. 3355.
7. Albring, T.A.; Sagebaum, M.; Gauger, N.R. Efficient aerodynamic design using the discrete adjoint method in SU2. In Proceedings of the 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Washington, DC, USA, 13–17 June 2016; p. 3518.
8. Kaya, H.; Tuncer, İ.H. Discrete Adjoint-Based Aerodynamic Shape Optimization Framework for Natural Laminar Flows. *AIAA J.* **2022**, *60*, 197–212. [\[CrossRef\]](#)
9. Agarwal, D.; Robinson, T.T.; Armstrong, C.G.; Marques, S.; Vasilopoulos, I.; Meyer, M. Parametric design velocity computation for CAD-based design optimization using adjoint methods. *Eng. Comput.* **2018**, *34*, 225–239. [\[CrossRef\]](#)
10. Jesudasan, R.; Zhang, X.; Mueller, J.D. Adjoint optimisation of internal turbine cooling channel using NURBS-based automatic and adaptive parametrisation method. In Proceedings of the Gas Turbine India Conference. American Society of Mechanical Engineers, Bangalore, India, 7–8 December 2017; Volume 1, V001T02A009.
11. Xu, S.; Jahn, W.; Müller, J.D. CAD-based shape optimisation with CFD using a discrete adjoint. *Int. J. Numer. Methods Fluids* **2014**, *74*, 153–168. [\[CrossRef\]](#)
12. Sanchez Torreguitart, I.; Verstraete, T.; Mueller, L. Optimization of the LS89 axial turbine profile using a cad and adjoint based approach. *Int. J. Turbomach. Propuls. Power* **2018**, *3*, 20. [\[CrossRef\]](#)
13. Banović, M.; Mykhaskiv, O.; Auriemma, S.; Walther, A.; Legrand, H.; Müller, J.D. Algorithmic differentiation of the Open CASCADE Technology CAD kernel and its coupling with an adjoint CFD solver. *Optim. Methods Softw.* **2018**, *33*, 813–828. [\[CrossRef\]](#)
14. Robinson, T.T.; Armstrong, C.G.; Chua, H.S.; Othmer, C.; Grahs, T. Optimizing parameterized CAD geometries using sensitivities based on adjoint functions. *Comput.-Aided Des. Appl.* **2012**, *9*, 253–268. [\[CrossRef\]](#)
15. Agarwal, D.; Robinson, T.T.; Armstrong, C.G.; Kapellos, C. Enhancing CAD-based shape optimisation by automatically updating the CAD model's parameterisation. *Struct. Multidiscip. Optim.* **2019**, *59*, 1639–1654. [\[CrossRef\]](#)
16. Agarwal, D.; Kapellos, C.; Robinson, T.; Armstrong, C. Using parametric effectiveness for efficient CAD-based adjoint optimization. *Comput.-Aided Des. Appl.* **2019**, *16*, 703–719. [\[CrossRef\]](#)
17. FreeCAD 0.19.2. Available online: <https://www.freecadweb.org/> (accessed on 5 November 2021).
18. Vasilopoulos, I.; Agarwal, D.; Meyer, M.; Robinson, T.T.; Armstrong, C.G. Linking parametric CAD with adjoint surface sensitivities. In Proceedings of the ECCOMAS Congress 2016—Proceedings of the 7th European Congress on Computational Methods in Applied Sciences and Engineering, Crete, Greece, 5–10 June 2016; Volume 2, pp. 3812–3827.
19. Chen, S.; Tortorelli, D. Three-dimensional shape optimization with variational geometry. *Struct. Optim.* **1997**, *13*, 81–94. [\[CrossRef\]](#)
20. Truong, A.H.; Zingg, D.W.; Haimes, R. Surface mesh movement algorithm for computer-aided-design-based aerodynamic shape optimization. *AIAA J.* **2016**, *54*, 542–556. [\[CrossRef\]](#)
21. Sun, L.; Yao, W.; Robinson, T.T.; Armstrong, C.G.; Marques, S.P. Automated mesh deformation for computer-aided design models that exhibit boundary topology changes. *AIAA J.* **2020**, *58*, 4128–4141. [\[CrossRef\]](#)
22. Geuzaine, C.; Remacle, J.F. Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. *Int. J. Numer. Methods Eng.* **2009**, *79*, 1309–1331. [\[CrossRef\]](#)
23. Palacios, F.; Alonso, J.; Duraisamy, K.; Colonno, M.; Hicken, J.; Aranake, A.; Campos, A.; Copeland, S.; Economon, T.; Lonkar, A.; et al. Stanford University Unstructured (SU 2): An open-source integrated computational environment for multi-physics simulation and design. In Proceedings of the 51st AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, Grapevine, TX, USA, 7–10 January 2013; p. 287.
24. Turbulence Modelling Resource, NASA. Available online: https://turbmodels.larc.nasa.gov/multielementverif_grids.html (accessed on 12 October 2021).
25. GMGW AIAA. Available online: <http://www.gmgworkshop.com/gmgw25.shtml> (accessed on 6 November 2021).