






## CAD-based Adjoint Optimization Using Other Components in a CAD Model Assembly as Constraints

Dheeraj Agarwal<sup>1</sup> , Trevor T. Robinson<sup>2</sup>  and Cecil G. Armstrong<sup>3</sup> 

<sup>1</sup>The University of Liverpool, [dheeraj@liverpool.ac.uk](mailto:dheeraj@liverpool.ac.uk)

<sup>2</sup>Queen's University Belfast, [t.robison@qub.ac.uk](mailto:t.robison@qub.ac.uk)

<sup>3</sup>Queen's University Belfast, [c.armstrong@qub.ac.uk](mailto:c.armstrong@qub.ac.uk)

Corresponding author: Dheeraj Agarwal, [dheeraj@liverpool.ac.uk](mailto:dheeraj@liverpool.ac.uk)

**Abstract.** This paper presents a CAD-based shape optimization process which exploits the capabilities of modern CAD systems to enforce assembly constraints within the optimization process. The assembly constraints are imposed using adjacent components in the CAD model assembly, which the component being optimized is not allowed to overlap with. This is important in industrial workflows, where unwanted interference can often result during the final product assembly. Here, an optimization framework is presented where the parameters defining the features in a feature-based CAD model are used as design variables, and their gradients are computed by combining design velocities with sensitivities computed using adjoint methods. The benefits of this framework are three-fold: (1) the use of adjoint methods makes the computational cost essentially independent of the number of design variables, (2) the optimized geometry is available as a feature-based CAD model that can be easily used for downstream processes, (3) the optimized geometry respects space constraints imposed by other parts in the assembly. In this paper, the developed framework is demonstrated for the optimization of models created in CATIA V5, to be assembled with other components defined in the CATIA V5 assembly workbench.

**Keywords:** CAD, adjoint methods, shape optimization, product assembly, interference, constraints.

**DOI:** <https://doi.org/10.14733/cadaps.2023.749-762>

### 1 INTRODUCTION

With advances in the field of computers, and their progressive use within the industrial design process, the need for physical design prototypes has been extensively reduced and replaced with digital models which are constructed and analyzed using computers. Nowadays product design typically starts with the construction of a computer-aided design (CAD) geometry of an initial concept and the goal is to deliver an optimized geometry as a CAD model which can be used for manufacturing. In recent years, optimization has become an essential and integrated part of an

industrial design process. The need for optimizing designs led to the development of various stochastic methods like genetic and evolutionary algorithms [14],[18]. The main benefit of these methods is that the probability of obtaining a global optimum with a smaller design space is higher (but not guaranteed) compared to other methods such as gradient-based methods. The main limitation is the high computational cost associated with the large number of function evaluations required to reach the optimum. Gradient-based methods have advantages in terms of efficiency when optimizing designs with a large number of design variables. The choice of the optimization algorithm is problem dependent. In this paper, gradient-based optimization methods are used for the problem of shape optimization. This requires an efficient methodology for computing the gradient of an objective function as well as the constraints (if any) with respect to the design parameters.

Modern CAD systems like CATIA V5, SIEMENS NX, Solidworks etc. use feature-based modelling strategies to create a parametric CAD model. This enables the designers to create relationships between different CAD features, and sometimes between different parts or assemblies, to integrate the design intent for the product. The main advantage of the CAD-based optimization approaches is that the optimized model can be directly used for downstream applications including manufacturing and process planning. In this work, a CAD-centric adjoint optimization approach [5] is used, which enables the use of commercial feature-based CAD systems within the optimization.

In general, mechanical design processes are not only driven by performance but are also subjected to constraints. These constraints may include the size of geometric features like the trailing edge radius of the turbine blade, volume or mass constraints, constraints on cross-sectional area, constraints on flow fields to account for a minimum lift, fixed exit-flow angle etc. Mader and Martins [21] used constraints such as bending moment, static and dynamic stability to examine optimal wing shapes in subsonic and transonic flows. Walther and Siva [34] presented an adjoint-based shape optimization for a multistage turbine design, with the objective to maximize the efficiency while constraining the mass flow rate and the total pressure ratio. Kontoleon et al. [20] presented a constrained topology optimization approach for ducts with multiple outlets. The flow constraints are enforced at each outlet defining the volume flow rates, flow direction and/or mean temperature of the outgoing flow. In terms of geometrical constraints, Xu et al. [36] presented an approach employing a set of test points to impose the thickness and trailing edge radius constraint for the optimization of a nozzle guide vane.

When optimizing an industrial design, one of the important factors to consider is the packaging space in which the optimized component is expected to fit. This is typically constrained by other components in the product assembly which define the regions the optimized shape is not allowed to violate due to the existence of the other parts. Since individual components are designed and optimized by different designers, when these are assembled together, issues such as interference often occur, requiring engineering changes late in the product development cycle [11]. Thus, it is important for designers and manufacturers to devise methods to ensure that the optimized component can be assembled within the space available before the actual component is manufactured. With the advances in CAD systems and development of the Digital Mock-Up (DMU) for complex CAD model assemblies, it is now possible to replace the physical prototypes with virtual ones and do the assembly of components in a virtual environment before any prototype is built. The DMU is a product assembly workbench where different components are positioned in 3D space relative to each other. Interference checks can be made in the DMU. An obvious step to stop interference appearing in the first place would be to include constraints imposed by adjacent components in the assembly during the design optimization of individual components, or to apply clash detection and fix the interferences during the product assembly.

Some of the early works in the field of interference detection between two solids are found in [6],[8]. Recent developments in this field include [27], which enabled interference detection directly using CAD models. Zubairi et al. [37] developed a sensitivity approach to eliminate interference (if present) in a 3D CAD assembly, by identifying parameters defining the CAD features which needs to be modified and calculating the amount of change required to eliminate interference. The approach is effective in eliminating interferences but the effect of the resulted shape change on the performance of the individual components was not considered, meaning that the process of

eliminating interference could reduce the performance of a product, or even make it unsuitable for its role. Recently, a 2D shape optimization of a NACA0012 aerofoil based on a polynomial response surface model was presented in [19], where product assembly constraints were formulated to consider the presence of a fuel-box in the interior of the wing profile. The constraints were enforced as constant upper and lower bounds for the design parameters. Although the approach was successfully applied for optimization, using only the parameter bounds may be highly restrictive when the number of design variables increase.

In this paper, research has been set out to propose a novel optimization framework which interacts with the CAD assembly design to obtain the information of the available packaging space constrained by the presence of other components in the CAD model assembly. The developed approach is demonstrated on 2D and 3D parametric CAD models built in CATIA V5 and assembled with other components in CATIA V5 assembly workbench. Here, SU2 [26] is used for flow simulations, and Python 3.5 is used as the programming interface.

## 2 BACKGROUND

### 2.1 Adjoint Methods

The primary attraction of adjoint methods is their ability to compute gradient information at a computational cost which is essentially independent of the number of design parameters. This, in turn, opens the possibility to explore significantly larger design spaces than those possible with traditional approaches, in time-scales which are acceptable for industrial design.

The development of adjoint methods started with the works of Pironneau [31] in the field of optimal shape design, and have followed two different paths, i.e. continuous adjoint [7],[10],[12],[13],[25],[28-30],[35], and discrete adjoint [9],[16],[22],[23],[32]. These formulations are based on the process followed to mathematically formulate the adjoint equations from the flow field equations. In the continuous adjoint formulation, the adjoint equations are derived directly from the governing partial differential equations and then discretized, while in the discrete adjoint formulation the governing partial differential equations are discretized first and then the adjoint equations are formulated. In [24] the continuous and discrete adjoint approaches for aerodynamic optimization were compared and it was found that the discrete adjoint gradients are in closer agreement with the gradients computed using finite differences than those computed using continuous adjoint methods. But the difference is not significant, and it is further reduced as the mesh resolution increases. Also, it was commented that the computational cost of deriving the discrete adjoint is greater than the continuous adjoint. Full details on the use of the adjoint approach to design are given by Giles and Pierce [17]. In this work, the adjoint surface sensitivities produced using the Open-source adjoint solver SU2 are integrated within a CAD-based gradient optimization framework by calculating the parametric design velocity, i.e. the boundary shape movement resulting from a change in CAD parameter.

### 2.2 Parametric Design Velocity

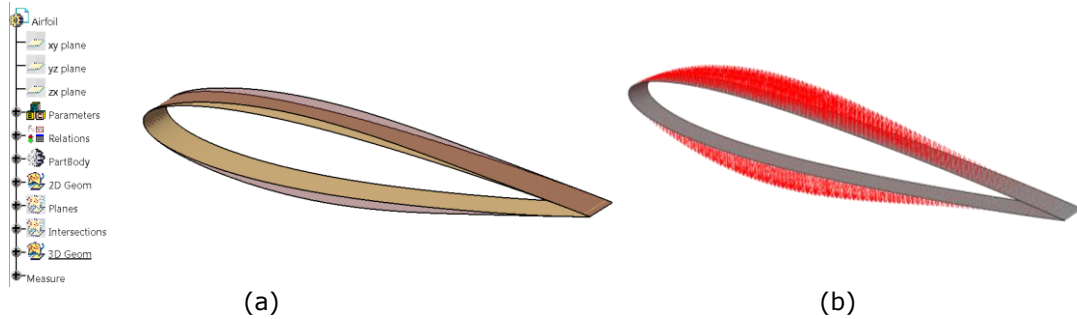
Parametric design velocity ( $V_n$ ) quantifies the boundary movement with respect to a change in the parameter value. In this work, design velocity is defined as the normal component of shape displacement on the boundary of the model caused by a parametric perturbation. Where the motive is to use a parametric CAD model in an optimization framework, the availability of a robust and efficient way of calculating parametric design velocity is of utmost importance. This work is concerned with computing

$$V_n = \delta X_s \cdot \hat{n}, \quad (1)$$

where,  $\delta X_s$  is the movement of surface nodes and  $\hat{n}$  is the outward unit normal.

In Figure 1, the arrows represent the design velocities as the boundary changes from the solid brown to the translucent airfoil. The convention adopted throughout this work is that a positive design velocity represents an outward movement of the boundary, and negative is inward. The

approach used in this work for computing design velocity for CAD-based design parameters is developed by [5], and is applicable to any feature-based CAD modelling package. The approach places no constraints on boundary topology changes which often occurs after parametric changes and which hampers the alternative approaches. Also, it does not require any special access to the source code of the CAD modeler.



**Figure 1:** (a) CAD model of the original and perturbed geometry, (b) design velocity vectors.

### 2.3 Gradient Computation

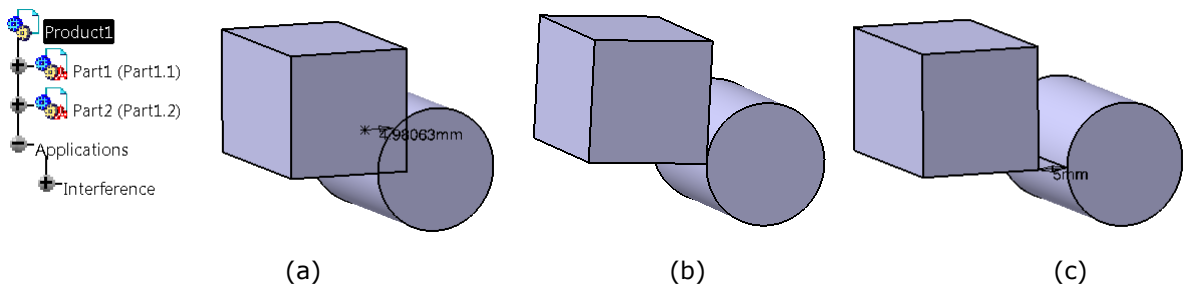
For the optimizer to establish a new search direction it is necessary for the gradient to be evaluated with respect to each design variable. In this case, it means evaluating the change in each objective function and each constraint due to unit perturbation of a CAD parameter. This can be achieved using

$$\begin{bmatrix} \frac{\partial f}{\partial \theta_1} \\ \frac{\partial f}{\partial \theta_2} \\ \vdots \\ \frac{\partial f}{\partial \theta_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial x_1}{\partial \theta_1} & \dots & \frac{\partial x_m}{\partial \theta_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_1}{\partial \theta_n} & \dots & \frac{\partial x_m}{\partial \theta_n} \end{bmatrix} \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_m} \end{bmatrix} \quad (2)$$

and provides flexibility for the use of different parameterization methods and function evaluations. Here  $n$  and  $m$  are the number of design variables and surface mesh points, respectively;  $f$ , represents the current function of interest (the objective or constraint functions). The variables  $x_i$  represent the normal displacement of discrete points on the surface. The second term  $\partial x/\partial \theta$  is known as the geometric sensitivity matrix and measures the influence that each design variable ( $\theta_i$ ) has on the position of each grid point on the surface mesh ( $V_n$ ) computed using Eqn. 1. The third term represents the surface sensitivities for a change in the function of interest with respect to the change in the position of surface grid points.

### 3 INTERFERENCE DETECTION

Interference occurs when some components in the product assembly violate others by occupying the same physical space. The interference detection system in CATIA V5 provides capabilities to obtain the penetration depth between the interfering components. This is the minimum distance required to translate a product to avoid interference. In addition, the clearance distance between two components can be obtained.



**Figure 2.** Interference between two boxes as in CATIA (a) interference, (b) contact, and (c) clearance.

Figure 2 displays the part-to-part interference detection in CATIA V5. It shows examples where the selected parts are in clash (Figure 2(a)), are in contact (Figure 2(b)), or have a clearance between them (Figure 2(c)). A CAD system application programming interface (API) is developed in this work to automatically calculate the interference between the component being optimized and other assembly components. In this work, a positive value of interference defines the amount of penetration distance between the components, while the negative value defines the clearance distance.

From the perspective of optimization, the objective is to optimize a model such that the optimized shape has no interference with other components. It is therefore necessary to automatically compute the amount of interference between the CAD model being optimized and other components in the CAD model assembly. This is achieved by using a CAD system API which is configured to detect the components (other than the component being optimized) in the CATIA V5 product assembly module (i) and use the interference tool to compute the individual interferences with the initial CAD model ( $\delta_i^{initial}$ ). At each optimization step, the CAD system API records the name of different components in the CAD product assembly and computes their interference values (used as assembly constraints). The other requirement is the computation of gradients of each assembly constraint with respect to the parameters used to define the initial CAD model. CATIA V5 offers capabilities to access the part model's parameterization through the assembly workbench. So, to compute the gradients of constraints, each parameter of the CAD model (j) is perturbed by a small amount ( $\Delta\theta_j$ ), and the interference tool is used to obtain the new interference values ( $\delta_i^j$ ) for all components. In practice this is done at the same time as the design velocity computations. The respective gradients of the constraints are then obtained using a finite difference method as

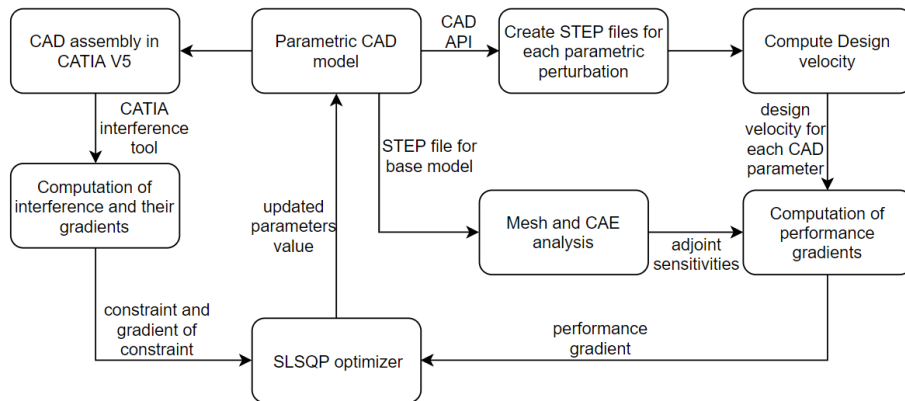
$$gradient = \frac{\delta_i^j - \delta_i^{initial}}{\Delta\theta_j} \quad (3)$$

#### 4 OPTIMIZATION FRAMEWORK

In this work, a gradient based optimization technique is used to guide the design towards a local optimum over multiple optimization steps. Within each optimization step, the design variables are set to new values, causing a change in the objective function. A general optimization with product assembly constraints can be defined as:

$$\begin{aligned} \text{Minimize:} & \quad \text{objective function,} \\ \text{Subject to:} & \quad \delta_i^j < 0 \\ \text{design variables:} & \quad \text{vector of CAD parameter values} \end{aligned} \quad (4)$$

The flow diagram for the optimization process is shown in Figure 3. The constraints due to interference with the adjacent components in the product assembly are enforced through an inequality constraint for the optimizer such that the value of interference is less than zero.



**Figure 3.** CAD-based optimization using constraints from product assembly.

The optimization starts with a parametric CAD model, which is exported in a neutral CAD format i.e. STEP for creating the mesh and performing the CFD and adjoint analysis. A series of CAD system APIs are created to modify the CAD model parameters and compute the design velocities using the approach described in [5]. The design velocities and adjoint sensitivities are then used to obtain the performance gradients with respect to CAD design parameters. Moreover, CAD system's API are also created to modify the CAD parameters and subsequently use the modified part model in the assembly workbench to compute the interference from other components. This helps to compute the geometric interference and its gradient to be used within an optimization algorithm SLSQP (sequential least square programming) implemented in Scipy [4]. SLSQP is a gradient based optimization algorithm which minimizes a function with any combination of bounds, equality and inequality constraints. In this work, the CAD models are created in CATIA V5, while optimization framework and CAD system API's are created in Python. Note, that while the STEP files are used in the computation of gradients, optimization is based on the native CAD file.

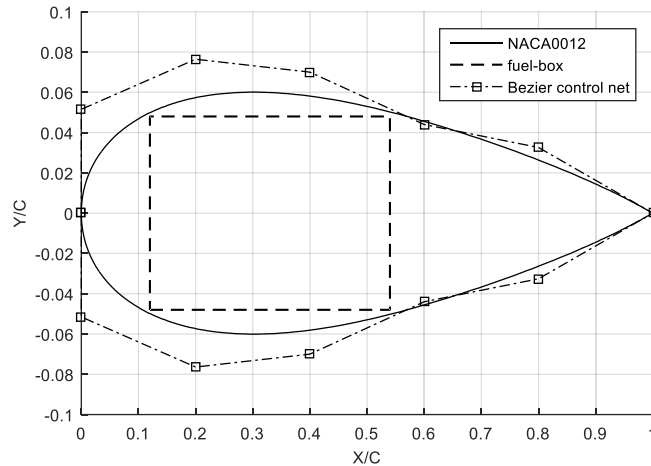
## 5 RESULTS

In this work, the use of product assembly constraints during optimization is demonstrated for two test cases with increasing complexity.

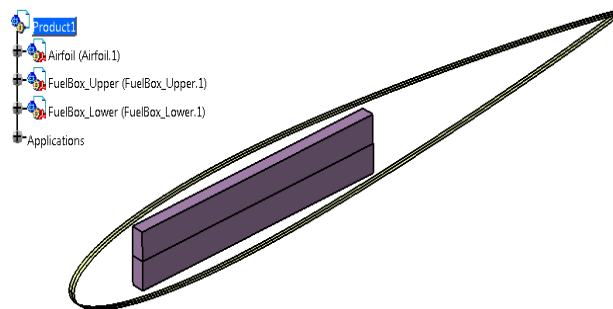
### 5.1 NACA0012 Optimization

The first test case considered is the benchmark two-dimensional NACA0012 aerofoil [1],[33]. NACA0012 represents a symmetrical airfoil with zero chambers and 12% thickness to chord ratio. Here, the aerofoil is constructed using two Bézier curves, one defining the upper surface and one defining the lower surface. Each Bézier curve is defined by seven control points. The design variables are the  $Y$ -coordinates of the five-control point defining the upper and lower curves with the following constraints: the leading edge and trailing edge points are fixed, and the control points on each curve after the leading edge are constrained to move in equal and opposite directions, vertically offset from the leading-edge point. These constraints ensure C2 continuity at the leading edge. The initial profile of Bézier control points and resulting airfoil profile are shown in Figure 4. Here, the airfoil is optimized (at zero angle of attack) using five design variables resulting in a symmetrical flow around the airfoil. The flow conditions are defined as:

- Freestream Temperature = 273.15 K
- Freestream Mach number = 0.80
- Angle of attack (AOA) = 0 deg
- Objective Function =  $\min(C_d)$
- Design variables = 5



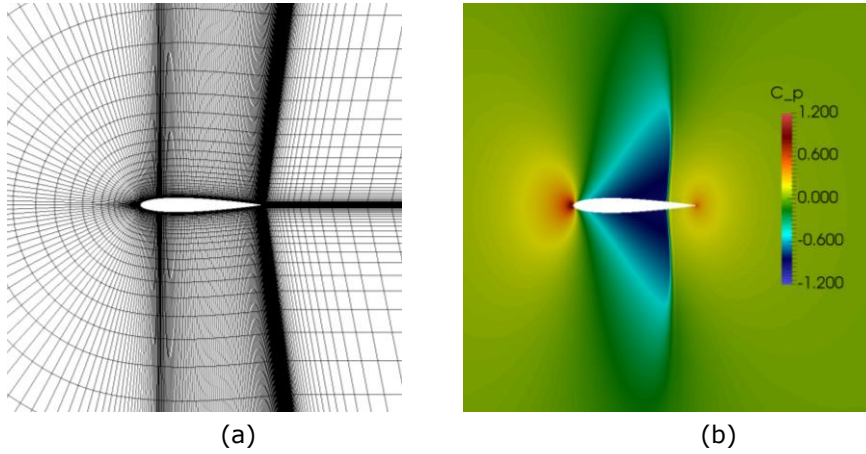
**Figure 4.** NACA0012 airfoil with Bézier control points and fuel-box.



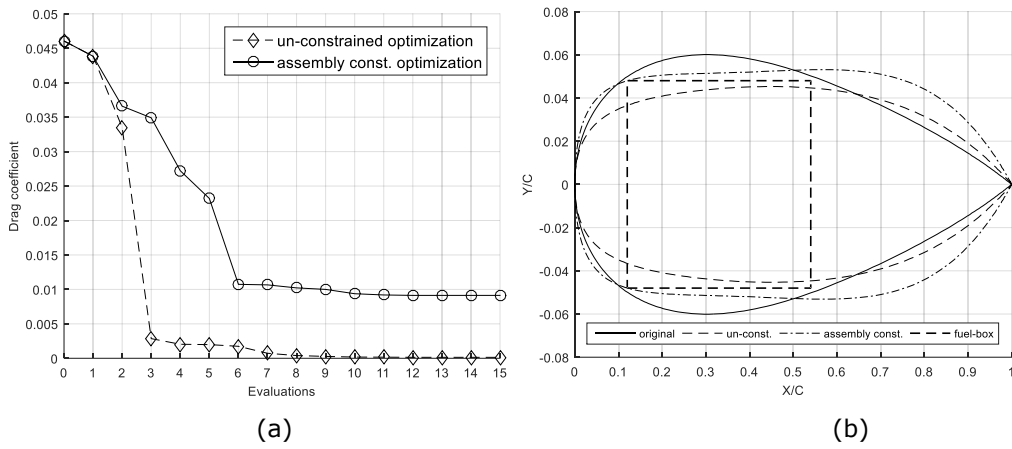
**Figure 5.** NACA0012 product assembly in CATIA V5.

The product assembly constraint in this example is a rectangular fuel-box around which the optimized airfoil must fit. Here, the 2D model was extruded by  $\pm 1\text{mm}$  to create 3D geometry which was assembled with a solid model of the fuel-box in the CATIA V5 product assembly workbench (Figure 5). The CFD mesh is created in ICEM-CFD [2] using a multi-block strategy with 300 points on the airfoil and 51 points in the direction normal to the airfoil. A detailed view of the mesh around the aerofoil is shown in Figure 6(a). The contour plot of the pressure coefficient ( $C_p$ ) is shown in Figure 6(b), where it can be seen that a strong shock-wave is formed at the upper surface of the airfoil, which contributes to increased drag on the airfoil. The automatic optimization framework described in section 4 is used to minimize the aerodynamic drag on the airfoil.

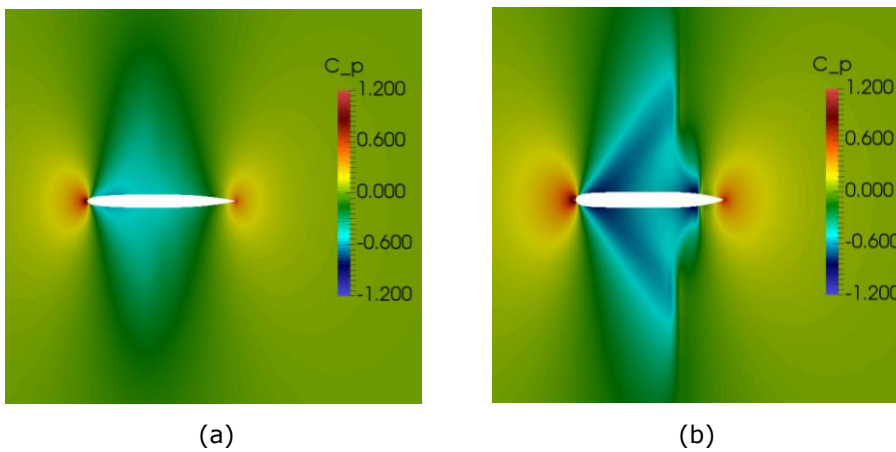
For each optimization step, a new CFD mesh is created in ICEM-CFD using an automated blocking script. The optimization history of the aerofoil with the product assembly constraint is shown in Figure 7(a), where it is compared with the optimization when no such constraint exists. The drag coefficient is reduced from 0.04605 to 0.0091 when subjected to the product assembly constraint compared to 0.00013 for an un-constrained optimization (with parameter bound). The optimized geometries are compared in Figure 7(b), where the Y-axis is amplified to enhance visual comparison. The un-constrained optimization results in a thinner aerofoil, compared to that obtained in the presence of product assembly constraints, but it clearly violates the space occupied by the fuel-box. The pressure distributions around the aerofoil are shown in Figure 8. It should be noted that the ideal solution for the un-constrained optimization problem is a very thin airfoil but using the parameter bounds restricts the optimizer to result in such geometry.



**Figure 6.** (a) Mesh around NACA0012 airfoil. (b)  $C_p$  distribution at the start of optimization.



**Figure 7.** NACA0012 optimization with and without assembly constraints. (a) function evaluations, (b) geometry comparison.

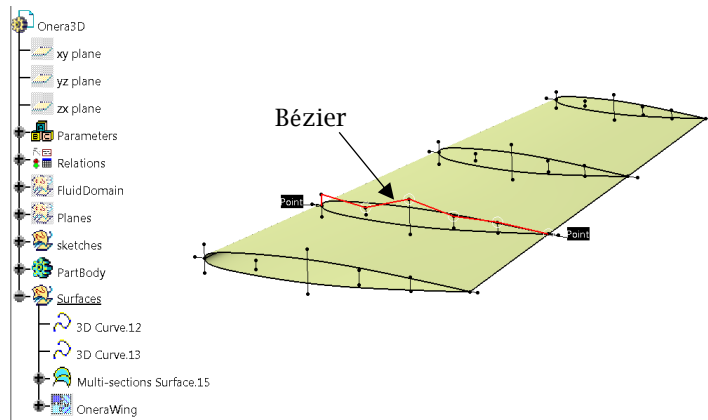


**Figure 8.**  $C_p$  contours on optimized NACA0012. (a) un-constrained, (b) with product assembly constraint.



## 5.2 ONERA M6 Wing Optimization

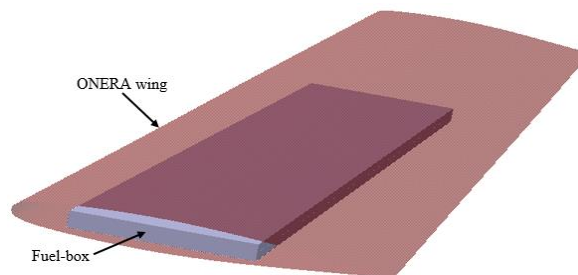
The second test case is the ONERA M6 wing model [3], which is a swept, semi-span wing with no twist. A parametric CAD model for the wing is constructed using four different cross-sections along the wing span. Each cross-section is defined using the same parameterization strategy as used for NACA0012. In order to keep the shape of the wing tip fixed only the first three sections are parameterized, resulting in a total of 27 CAD parameters. The 3D wing is then constructed by sweeping a surface through the section curves as shown in Figure 9. Here, the optimization is performed with product assembly constraint of a 3D rectangular fuel-box that is to be contained within the wing volume as shown in Figure 10.



**Figure 9.** ONERA M6 CAD model showing Bézier control points for section profiles.

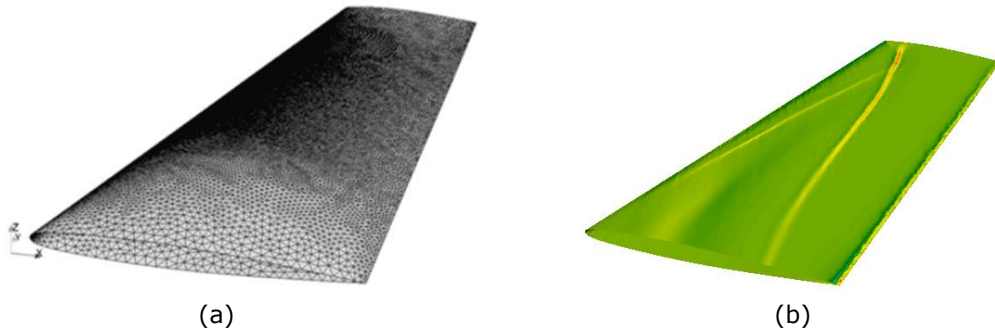
The following flow conditions are defined for CFD analysis:

- Freestream Temperature = 288.15 K
- Freestream Mach number = 0.8395
- Angle of attack (AOA) =  $3.06^\circ$
- Objective Function =  $\min(C_d)$
- Design variables = 27



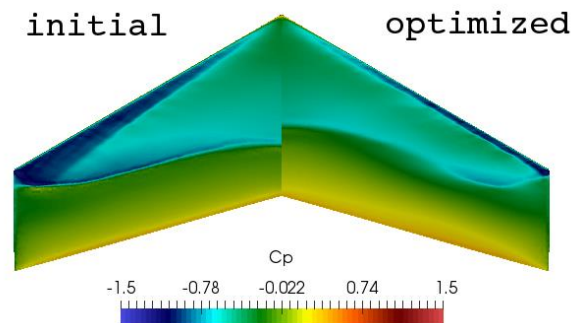
**Figure 10.** ONERA M6 wing with fuel-box.

An unstructured mesh was created in GMSH [15] with 154,617 nodes and 707,115 tetrahedral elements, the respective surface mesh is shown in Figure 11(a) and used for both primal and adjoint analysis. The mesh density near the leading and trailing edge of the wing is controlled by implementing a background mesh field with refinement boxes. The adjoint sensitivity for minimizing aerodynamic drag over the ONERA M6 wing is shown in Figure 11(b), which indicates how the mesh nodes on the wing's upper surface should move to minimize the drag.

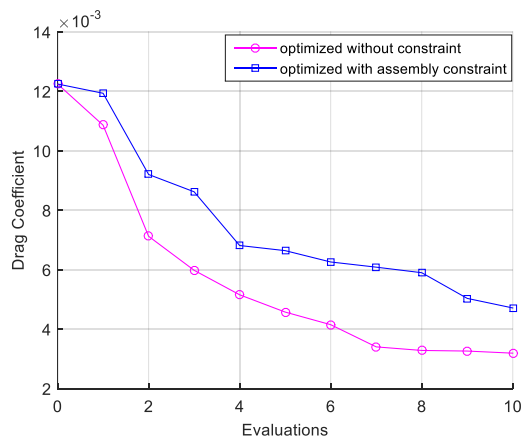


**Figure 11.** (a) ONERA M6 CFD mesh, and (b) adjoint sensitivity map.

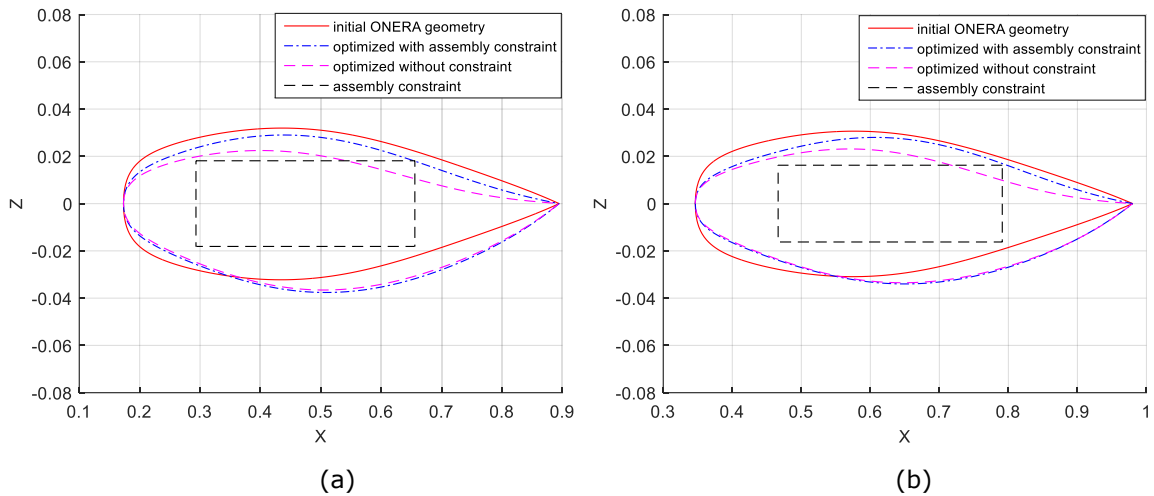
For each optimization step, a new mesh was created in GMSH using density box and background field functionality in GMSH. The pressure distribution on the initial and ONERA M6 wing model optimized in the presence of product assembly constraints is shown in Figure 12. During the optimization the drag coefficient was reduced from 0.01232 to 0.0045, compared to 0.0032 for an un-constrained optimization (with parameter bounds) as shown in Figure 13. A comparison between the initial and optimized geometry at two different cross-sections are shown in Figure 14, where the  $Z$ -axis is amplified to enhance visual comparison. In each case, the optimized result for the constrained optimization is very close to the contact with the fuel-box, while the un-constrained optimization result violates the space occupied by the fuel-box.



**Figure 12.**  $C_p$  distribution on ONERA M6 wing, initial (left) and optimized with product assembly constraint (right).



**Figure 13.** Optimization history for ONERA M6 with product assembly constraint.



**Figure 14.** ONERA M6 wing optimized section at (a) 30% span, (b) 60% span.

## 6 DISCUSSION AND CONCLUSIONS

In this work, a constrained shape optimization framework was successfully demonstrated which uses the prior information from product assembly components. The objective function was optimized without violating the space available for storing other components in the assembly. This was achieved by a series of Python CAD system APIs which interact with the product assembly module of CATIA V5 through its API to extract interference distances between the component being optimized and other components to be assembled together. In this paper, the models created in CATIA V5 were used as the product assembly constraint, but the framework can also be used for models created in other CAD systems (e.g. SIEMENS NX, SolidWorks etc.) to define these constraints.

For the test cases analyzed in this paper, the CFD analysis was performed using the open-source CFD solver SU2, and the objective was to minimize the aerodynamic drag. The product assembly constraint for the two cases was imposed using a rectangular fuel-box. In a broader perspective, these constraints could represent any region of space the CAD model is not allowed to occupy. The developed optimization framework successfully optimized the component without introducing interference during the optimization. It was interesting to note that for all the examples, optimizing the models without considering adjacent components, resulted in optimized shapes which would have caused fit issues when the product assembly would have been attempted.

Since the optimization algorithm SLSQP uses a least-square formulation to compute the Lagrangian multipliers to incorporate the constraints in the optimization, the efficiency of this constrained optimization framework depends on the number and types of constraints. Here, the complexity arises from the perspective of optimizer but not from the methodology described in this paper for the computations of product assembly constraint and their respective gradients. In future, the applicability of the developed framework would be tested for the optimization of an industrial component in the presence of significantly larger number of other components in the product assembly.

## ACKNOWLEDGMENTS

This work has been conducted within the IODA project (<http://ioda.sems.qmul.ac.uk>), funded by the European Union HORIZON 2020 Framework Programme for Research and Innovation under Grant Agreement No. 642959.

Dheeraj Agarwal, <https://orcid.org/0000-0001-5340-5851>  
 Trevor T. Robinson, <https://orcid.org/0000-0002-6595-6308>  
 Cecil G. Armstrong, <https://orcid.org/0000-0001-8695-5016>

## REFERENCES

- [1] Aerodynamic Design Optimization: Drag Minimization of the NACA 0012 in Transonic Inviscid Flow.
- [2] Ansys ICEM CFD, <https://www.ansys.com/training-center/course-catalog/fluids/introduction-to-ansys-icem-cfd>, Accessed 08/03/2022.
- [3] ONERA M6, <https://www.grc.nasa.gov/www/wind/valid/m6wing/m6wing.html>, Accessed 08/03/2022.
- [4] Scipy Optimize, <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html>, Accessed 08/03/2022.
- [5] 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, *Engineering with Computers*, 34(2), 2018, 225-239, <https://doi.org/10.1007/s00366-017-0534-x>.
- [6] Ahuja, N.; Chien, R.T.; Yen, R.; Bridwell, N.: Interference Detection and Collision Avoidance Among Three Dimensional Objects, *AAAI-80 Proceedings*, pp. 44-48.
- [7] Anderson, W.K.; Venkatakrisnan, V.: Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation, *Computers and Fluids*, 28(1999), 443-480, [https://doi.org/10.1016/S0045-7930\(98\)00041-3](https://doi.org/10.1016/S0045-7930(98)00041-3).
- [8] Boyse, J.W.: Interference detection among solids and surfaces, *Communications of the ACM*, 22(1), 1979, 3-9, <https://doi.org/10.1145/359046.359048>.
- [9] Burgreen, G.W.; Baysal, O.: Three-dimensional aerodynamic shape optimization using discrete sensitivity analysis, *AIAA Journal*, 34(9), 1996, 1761-1770, <https://doi.org/10.2514/3.13305>.
- [10] Castro, C.; Lozano, C.; Palacios, F.; Zuazua, E.: Systematic continuous adjoint approach to viscous aerodynamic design on unstructured grids, *AIAA Journal*, 45(9), 2007, 2125-2139, <https://doi.org/10.2514/1.24859>.
- [11] Chang, K.-H.; Silva, J.; Bryant, I.: Concurrent design and manufacturing for mechanical systems, *Concurrent Engineering*, 7(4), 1999, 290-308, <https://doi.org/10.1177/1063293X9900700403>.
- [12] Economon, T.D.; Palacios, F.; Alonso, J.J.: A viscous continuous adjoint approach for the design of rotating engineering applications, *AIAA Journal*, 2580(2013), 24-27, <https://doi.org/10.2514/6.2013-2580>.
- [13] Fang, L.; Li, X.: Design optimization of unsteady airfoils with continuous adjoint method, *Applied Mathematics and Mechanics-English Edition*, 36(10), 2015, 1329-1336, [10.1007/s10483-015-2010-9](https://doi.org/10.1007/s10483-015-2010-9).
- [14] Fonseca, C.M.; Fleming, P.J.: An overview of evolutionary algorithms in multiobjective optimization, *Evolutionary computation*, 3(1), 1995, 1-16,
- [15] Geuzaine, C.; Remacle, J.F.: GMSH: A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities, *International Journal of Numerical methods in Engineering*, 79(2009), 1309-1331, <https://doi.org/10.1002/nme.2579>.
- [16] Giles, M.B.; Duta, M.C.; Muller, J.-D.; Pierce, N.A.: Algorithm developments for discrete adjoint methods, *AIAA Journal*, 41(2), 2003, 198-205,
- [17] Giles, M.B.; Pierce, N.A.: An introduction to the adjoint approach to design, *Flow, turbulence and combustion*, 65(3-4), 2000, 393-415,
- [18] Goldberg, D.E.: *Genetic algorithms in search, optimization, and machine learning*, 1989, Reading: Addison-Wesley, 1989,

- [19] Immonen, E.: 2D shape optimization under proximity constraints by CFD and response surface methodology, *Applied Mathematical Modelling*, 41(2017), 508-529, <https://doi.org/10.1016/j.apm.2016.09.009>.
- [20] Kontoleonos, E.A.; Papoutsis-Kiachagias, E.M.; Zymaris, A.S.; Papadimitriou, D.I.; Giannakoglou, K.C.: Adjoint-based constrained topology optimization for viscous flows, including heat transfer, *Engineering Optimization*, 45(8), 2013, 941-961, <https://doi.org/10.1080/0305215X.2012.717074>.
- [21] Mader, C.A.; Martins, J.R.R.A.: Stability-constrained aerodynamic shape optimization of flying wings, *Journal of Aircraft*, 50(5), 2013, 1431-1449, <https://doi.org/10.2514/1.C031956>.
- [22] Mader, C.A.; Martins, J.R.R.A.; Alonso, J.J.; Van Der Weide, E.: Adjoint: an approach for the rapid development of discrete adjoint solvers, *AIAA Journal*, 46(4), 2008, 863-873, <https://doi.org/10.2514/1.29123>.
- [23] Nadarajah, S.K.: The discrete adjoint approach to aerodynamic shape optimization. stanford university Stanford, USA, 2003.
- [24] Nadarajah, S.K.; Jameson, A.: A comparison of the continuous and discrete adjoint approach to automatic aerodynamic optimization, 38th Aerospace Sciences Meeting and Exhibit, p. 667.
- [25] Othmer, C.: Continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows, *International Journal of Numerical Methods in Fluids*, 58(2008), 861-877, <https://doi.org/10.1002/flid.1770>.
- [26] Palacios, F.; Alonso, J.J.; Duraisamy, K.; Colonno, M.; Hicken, J.; Aranake, A.; Campos, A.; Copeland, S.R.; Economon, T.D.; Lonkar, A.: 2013. Stanford University Unstructured (SU2): an open-source integrated computational environment for multi-physics simulation and design, 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition. AIAA, Grapevine, Texas.
- [27] Pan, C.; Smith, S.S.F.; Smith, G.C.: Determining interference between parts in CAD STEP files for automatic assembly planning, *Journal of Computing and Information Science in Engineering*, 5(1), 2005, 56-62, <https://doi.org/10.1115/1.1861473>.
- [28] Papadimitriou, D.I.; Giannakoglou, K.C.: Total pressure loss minimization in turbomachinery cascades using a new continuous adjoint formulation, *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, 221(6), 2007, 865-872, <https://doi.org/10.1243/09576509JPE463>.
- [29] Papoutsis-Kiachagias, E.M.; Giannakoglou, K.C.: Continuous adjoint methods for turbulent flows, applied to shape and topology optimization: industrial applications, *Archives of Computational Methods in Engineering*, 23(2), 2016, 255-299, <https://doi.org/10.1007/s11831-014-9141-9>.
- [30] Papoutsis-Kiachagias, E.M.; Magoulas, N.; Müller, J.D.; Othmer, C.; Giannakoglou, K.C.: Noise reduction in car aerodynamics using a surrogate objective function and the continuous adjoint method with wall functions, *Computers & Fluids*, 122(2015), 223-232, <https://doi.org/10.1016/j.compfluid.2015.09.002>.
- [31] Pironneau, O.: On optimum design in fluid mechanics, *Journal of Fluid Mechanics*, 64(1), 1974, 97-110, <https://doi.org/10.1017/S0022112074002023>.
- [32] Roth, R.; Ulbrich, S.: A discrete adjoint approach for the optimization of unsteady turbulent flows, *Flow, Turbulence and Combustion*, 90(2013), 763-783, <https://doi.org/10.1007/s10494-012-9439-3>.
- [33] Vassberg, J.C.; Harrison, N.A.; Roman, D.L.; Jameson, A.: 2011. A systematic study on the impact of dimensionality for a two-dimensional aerodynamic optimization model problem, 29th AIAA Applied Aerodynamics Conference, Honolulu, HI.
- [34] Walther, B.; Nadarajah, S.: Constrained adjoint-based aerodynamic shape optimization of a single-stage transonic compressor, *Journal of Turbomachinery*, 135(2), 2013, 021017, <https://doi.org/10.1115/1.4007502>.
- [35] Wolosz, K.J.; Wernik, J.: Three-dimensional flow optimization of a nozzle with a continuous adjoint, *International Journal of Nonlinear Sciences and Numerical Simulation*, 16(3-4), 2015, 151-156, <https://doi.org/10.1515/ijnsns-2014-0011>.

- [36] Xu, S.; Radford, D.; Meyer, M.; Müller, J.D.: CAD-based adjoint shape optimisation of a one-stage turbine with geometric constraints, ASME Turbo Expo.
- [37] Zubairi, M.S.; Robinson, T.T.; Armstrong, C.G.; Soban, D.S.: A sensitivity approach for eliminating clashes from computer aided design model assemblies, Journal of Computing and Information Science in Engineering, 14(3), 2014, 031002, <https://doi.org/10.1115/1.4027345>.