

GEOMETRIC CONSTRAINT IMPOSITION ON TRIMMED NURBS PATCHES FOR ADJOINT OPTIMIZATION

MARIOS G. DAMIGOS^{1,2}, KYRIAKOS C. GIANNAKOGLOU² AND
EUGENE DE VILLIERS³

¹ ENGYS S.R.L.
Via del Follatoio 12, 34148, Trieste, Italy
mariosda@gmail.com

² National Technical University of Athens (NTUA), School of Mechanical Engineering, Parallel
CFD & Optimization Unit
9, Iroon Polytechniou Str., NTUA Zografou Campus, Athens 15780, Greece
kgianna@central.ntua.gr

³ ENGYS Ltd.
Studio 20, Royal Victoria Patriotic Building, John Archer Way, London SW18 3SX, United
Kingdom
e.devilliers@engys.com

Key words: NURBS, Constraints, Curvature, Volume, Adjoint, Shape, Optimization

Abstract. NURBS has been the leading mathematical standard to describe geometries contained in standard CAD files. NURBS surfaces have successfully been connected to adjoint optimization methods as they provide a robust and well defined deformation tool for the shapes to be designed. Quite often, the resulting optimal NURBS shapes are wrinkly with high-curvature areas that make the final model non-manufacturable. Furthermore, because NURBS surfaces are manipulated through control points, there is no direct control over the deformed shape's volume or other geometrical constraints. For instance, the optimal shape can have a volume smaller than the one required for it to be operational. For instance, the volume of an aircraft wing must be sufficiently large in order for a fuel tank to fit inside of it. In this work, we are addressing the solution to these two issues that may arise in various shape optimization scenarios. Without loss of generality, in what follows, the continuous adjoint technique is used to generate a sensitivity map, with the derivatives of the objective function w.r.t. the normal displacement of the surface points. The map is, then, projected onto the NURBS patches that are contained in a standard CAD file of the geometry under consideration. In general, these patches, are trimmed NURBS and the intersections between them are handled using a null space projection method [1]. The two constraints this paper is dealing with are defined as follows: keeping the curvature field on a NURBS-defined solid below a threshold value and the volume of that solid above a threshold value. The constraints are differentiated in order for them to be used within an optimization algorithm; tests are made using two optimization examples, namely a 3D wing and an S-bend duct.

1 Introduction

Manufacturing constraints in gradient-based optimization is an increasingly popular research topic but not yet exhaustively documented. A reason for this could be that manufacturability constraints are inherently connected with CAD parameterization. Connecting the latter to gradient-based optimization methods, such as those based on adjoints, is a challenge of its own. There is a high variety of CAD packages that are used for industrial design and each of them has a different parameterization which CAD vendors are very sensitive about. Therefore, in order to use a CAD parameterization inside the optimization loop, the corresponding CAD software (which is not necessarily an open-source tool) should be differentiated. Several options to avoid this impasse have been investigated [2, 3, 4]. In this article, standard CAD files, such as STEP or IGES, are used to transfer the model data and the null space projection method shown in [1] serves as a re-parameterization technique that can handle non-conforming trimmed NURBS geometry. Standard CAD files contain the Boundary Representation (BRep) of a CAD model which consists of surface patches. The geometry of the BRep may consist of elementary curves and surfaces for simpler shapes but, for more complex ones, the geometric entities are mainly trimmed NURBS curves and surfaces. Any relation to the design (feature) tree or the design intent is lost through the BRep and, therefore, all manufacturability constraints must be defined using surface representations. In this article, two manufacturability-related constraints are investigated:

- **Curvature:** Constraining the curvature of a surface (or an ensemble of surfaces) while searching for the optimal (according to an objective function) shape of the surface is a topic that has been covered mainly in non-CFD related optimization. Azariadis et al. [5] investigated the generation of planar development of 3D surfaces by taking into consideration the geodesic curvature, while Moreton et al. [6] focused on the construction of surfaces while blending, by minimizing curvature variation. In the attempt to constrain the curvature of a surface, one is obliged to choose which curvature (mean, Gauss, principal) should be constrained. In this paper, the curvatures in both principal directions are chosen to be constrained.
- **Volume:** Because of the NURBS free-form surface nature, the volume of a shape defined by a set of watertight NURBS surfaces can not be constrained in an intuitive manner. Therefore, the final volume can be smaller than a threshold defined by a designer. The volume of such a shape is computed using the Gauss divergence theorem for closed volumes. During the optimization, the volume of the design is constrained to remain greater than a threshold.

In the optimization examples shown in this paper, the adjoint technique [7, 8] is used for computing sensitivity derivatives. The adjoint technique in its discrete [9, 10], or continuous [11, 12, 13] form, is excellent for large scale optimization problems. In this article, the continuous adjoint is used.

This paper is structured as follows: In section 2, the basic ingredients of BRep-based optimization are explained and, in sections 3 and 4, the constraints are mathematically expressed. Then, in section 5, the proposed methods are tested in two cases.

2 BRep-Based Optimization

In this section, the manner in which standard CAD files are inserted in the optimization is presented. Initially, NURBS surfaces are discussed and, then, trimmed NURBS patches along with the method to handle their intersections are analyzed.

NURBS surfaces [14] are a generalization of tensor product B-Splines. Let us assume a NURBS surface with an $(n \times m)$ control grid, its control points $\vec{P}_{i,j} \in R^3$ and their respective weights $w_{i,j}$. Then, a point on the surface corresponding to parametric values (u, v) is

$$\vec{S}(u, v) = \frac{\sum_{i=1}^n \sum_{j=1}^m N_i^p(u) N_j^q(v) w_{i,j} \vec{P}_{i,j}}{\sum_{k=1}^n \sum_{l=1}^m N_k^p(u) N_l^q(v) w_{k,l}} = \sum_{i=1}^n \sum_{j=1}^m R_{i,j}(u, v) \vec{P}_{i,j} \quad (1)$$

where (p, q) are the polynomial degrees per parametric direction.

Since eq. 1 is linear w.r.t. the control points, its differentiation is straightforward. The result of the differentiation w.r.t. each control point is a diagonal rank 2 tensor,

$$\frac{d\vec{S}(u, v)}{d\vec{P}_{i,j}} = \text{diag}(R_{i,j}(u, v), R_{i,j}(u, v), R_{i,j}(u, v)) \quad (2)$$

In a similar manner, one can differentiate the first and second derivatives of the surface w.r.t. the parameters u and v .

In NURBS-based shape optimization on CAD models, ensuring smoothness and watertightness of the model is a non-trivial task. The parameterization of the different patches is non-conforming and the patches have trimmed parametric domains; therefore, this cannot be resolved by simply constraining the boundary control points of each patch. To resolve this issue, the method explained in [1] is used. This method initially identifies the intersecting trimmed NURBS patches of a CAD model. Then, for each intersection, it imposes watertightness (C_0 continuity) and smoothness (C_1 continuity) for a number of points on it. The C_0 and C_1 constraint equations can be handled to become linear w.r.t. the control points of both patches touching the intersection. Assuming the imposition of these constraints on n points along the intersecting curve and that the number of control points in both patches is m , then all the constraint equations can be put in the following matrix form

$$A_{n,m} P_{m,3} = 0_{n,3} \quad (3)$$

where A is the $n \times m$ coefficients matrix and P is the $m \times 3$ matrix that stores the coordinates of a control point in each row. The space which the solutions to eq. 3 lie on is the null space of A . In an optimization scenario in which eq. 3 must be satisfied, the

control point sensitivities must be projected onto this space to ensure that the update direction will not violate the constraints.

3 The Curvature Constraint

In the attempt to constrain the curvature field on NURBS surfaces, a prerequisite is the selection of the best curvature expression, which the constraint should be imposed to. Based on this, an inequality constraint must be imposed for all points on a NURBS surface.

The curvature at each point of a surface can be computed using the coefficients of its first and second fundamental forms [15], defining the tensors \mathcal{F}_1 and \mathcal{F}_2 respectively. At each point, the eigenvalues of the matrix $\mathcal{F}_1^{-1}\mathcal{F}_2$ provide the principal curvatures κ_1 and κ_2 and their eigenvectors provide the corresponding directions. Through them, the mean and Gaussian curvatures

$$\kappa_{mean} = \frac{\kappa_1 + \kappa_2}{2}, \quad \kappa_{Gauss} = \kappa_1\kappa_2 \quad (4)$$

can be defined. The principal curvatures represent the curvatures along perpendicular directions and, therefore, by constraining them to have an absolute value less than a threshold, a desired local "flatness" of the surface can be achieved. The expression chosen to be constrained, therefore, is

$$\kappa = \frac{1}{2}(\kappa_1^2 + \kappa_2^2) \quad (5)$$

Minimizing the quantity given by eq. 5 is the means to control curvature. The expression in eq. 5 refers to the curvature at a single surface point and is, therefore, local. To fully constrain the curvature on a NURBS surface, a grid of u, v parameters must be created, which should be fine enough to capture the surface details. Then, an inequality constraint for κ must be imposed at each grid point. This is impractical as it can result in a number of constraints in the order of millions. An alternate approach is to reduce all the inequality constraints to a single equality constraint. This can be done by employing a filtering function at each grid point. The filtering function receives the curvature metric at a grid point (eq. 5) and returns a zero value and derivative if this metric is less than the threshold. A function like this can take the form

$$F_{filter}(\kappa) = \begin{cases} 0 & , \kappa < \kappa_m \\ \alpha(\kappa - \kappa_m)^4 + \beta(\kappa - \kappa_m)^3 & , \kappa_m \leq \kappa < \kappa_s \\ \kappa + (y_s - \kappa_s) & , \kappa \geq \kappa_s \end{cases} \quad (6)$$

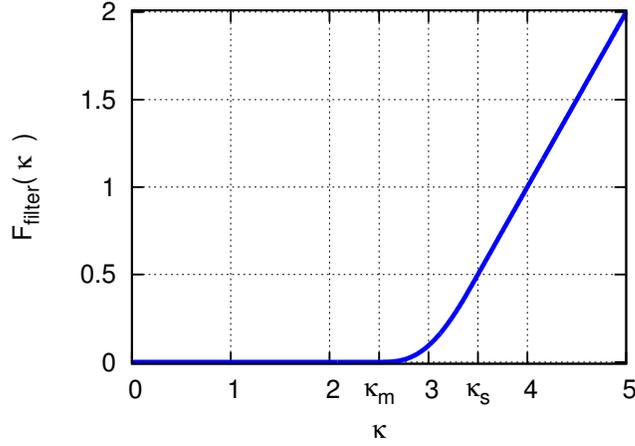
Terms involved in eq. 6 are as in table 1, ensuring second-order continuity throughout its domain of definition. κ_m denotes the curvature threshold while κ_s ($\kappa_s > \kappa_m$) denotes a metric value at which $F'_{filter}(\kappa)$ becomes maximum.

By integrating the filter function over the patches of a NURBS defined model yields

$$C_1 = \iint_S F_{filter}(\kappa) dS \quad (7)$$

Table 1: Quantities in eq. 6.

Variable	Value
κ_m	Curvature upper bound
κ_s	A value at which $F'_{filter}(\kappa)$ becomes maximum
α	$-\frac{1}{2(\kappa_s - \kappa_m)^3}$
β	$-2\alpha(\kappa_s - \kappa_m)$
y_s	$\alpha(\kappa_s - \kappa_m)^4 + \beta(\kappa_s - \kappa_m)^3$


Figure 1: The filtering function for $\kappa_m = 2.5$, $\kappa_s = 3.5$.

Nullifying the constraint C_1 leads to the satisfaction of all the constraints.

As an example of the effectiveness of the constraint, a simple test on a semi-cylindrical NURBS surface is presented. Its radius is $\rho = 0.5$ which leads to a uniform $\kappa = 2$ at all of its points. Eq. 7 is formulated for three different values of κ_m and equation $C_1 = 0$ is solved iteratively by selecting the control point positions. For all three cases $\kappa_s = \kappa_m + 0.1$ has been set for consistency. Solving $C_1 = 0$, leads the curvature metric of each surface point to become less than κ_m . The resulting shapes as well as the convergence history of all three cases, are shown in fig 2.

Finally, in order to demonstrate how κ_s affects convergence rate and/or stability the solution of $C_1 = 0$ was attempted for the same surface with $\kappa_m = 1.0$ but for three different values of κ_s (fig. 3).

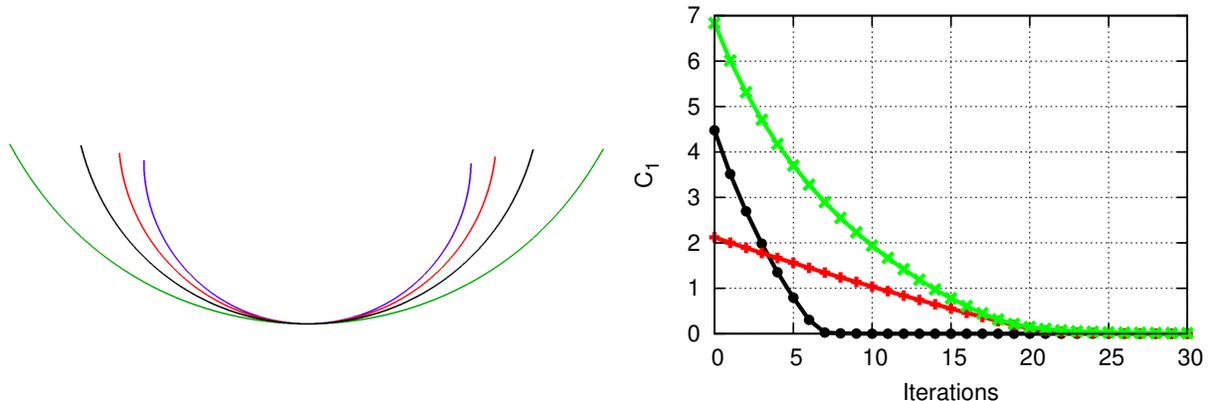


Figure 2: Left: The initial cross-section of a semi-cylindrical surface (blue) and the cross-sections resulting from the solution of $C_1 = 0$ for $\kappa_m = 1.5$ (red), $\kappa_m = 1.0$ (black) and $\kappa_m = 0.5$ (green). Right: The convergence history of the three iterative procedures leading to the results presented on the left. Curves for $\kappa_m = 1.5$ (red), $\kappa_m = 1.0$ (black) and $\kappa_m = 0.5$ (green) are shown.

4 The Volume Constraint

For a watertight solid model, its volume can be computed using only the nodal coordinates of its boundary patches. Assume such a model and let V be its volume and S its total surface. From the Gauss divergence theorem, it is known that the volume of a watertight shell of surface S can be computed using its boundary surfaces. It is obvious that the total surface of a solid is a union of all its trimmed patches $\vec{\sigma}_i$ and, therefore, the volume is given by

$$V = \frac{1}{3} \sum_i \int_u \int_v \vec{\sigma}_i(u, v) \cdot \vec{n}(u, v) \|\vec{\sigma}_u(u, v) \times \vec{\sigma}_v(u, v)\| dudv \quad (8)$$

The volume constraint is an inequality that enforces the volume to be lower (in cases where a model must fit inside an assembly) or greater (in cases where an assembly must fit inside a model i.e. fuel tank) than a threshold. For the latter case, the constraint has the expression

$$C_2 = V - V_{min} \geq 0 \quad (9)$$

This constraint is handled using slack variables within the Augmented Lagrangian Method (ALM) algorithm [16].

5 Examples

In this section, the constraints are tested on two test problems. For the first case, (the S-bend duct shown in fig. 4) the effectiveness of the curvature constraint is tested,

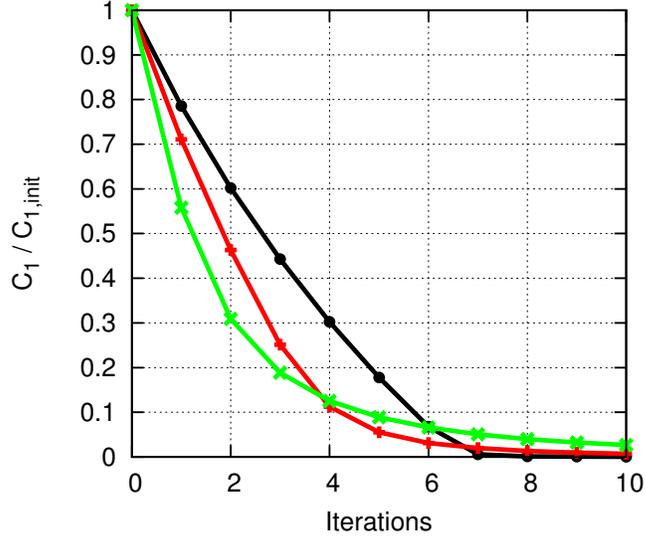


Figure 3: The convergence of the iterative solution of $C_1 = 0$ for the semi-cylindrical test case. Curves for $\kappa_s = 1$ (black), $\kappa_s = 0.5$ (red) and $\kappa_s = 0.1$ (green) are shown.

while for the second, (an extruded NACA0012-based wing) the effectiveness of the volume constraint is examined.

5.1 The S-bend Duct

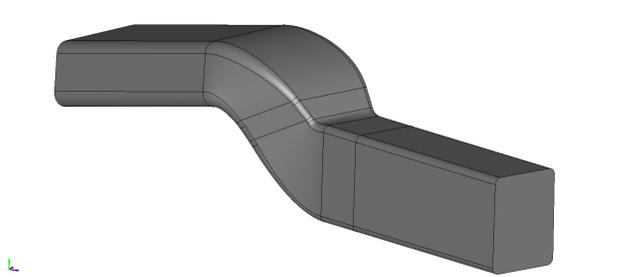


Figure 4: The S-bend climate duct.

The S-bend climate duct is a test case provided by VolksWagen AG. The goal is to minimize the viscous losses of the flow subject to the deformation of the S-section of the duct. The objective function has the following form

$$J = \sum_{inlet} Q(p + \frac{1}{2}\rho U^2)dA - \sum_{outlet} Q(p + \frac{1}{2}\rho U^2)dA \quad (10)$$

where p is the pressure, U is the velocity and ρ is the density of the fluid while Q is the flow rate.

The geometry consists of 46 trimmed NURBS patches, among which 28 belong to the S-section. These 28 patches are high-degree surfaces and this results to a total number of 5830 control points (17490 degrees of freedom) on the S-section. The flow analysis is done using a mesh of 700,000 cells. The flow is laminar with $Re \approx 400$.

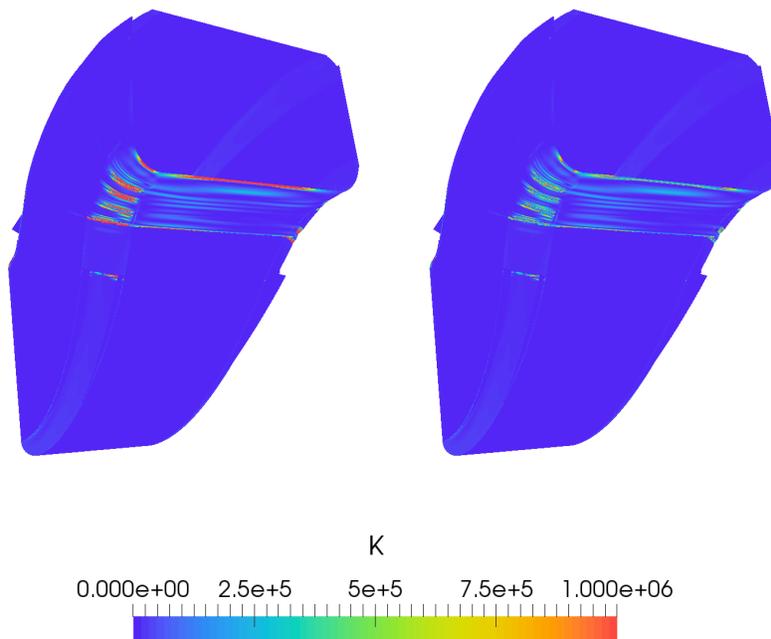


Figure 5: Curvature iso-areas on the optimal shape resulting from the unconstrained (left) and the constrained optimization (right) of the S-bend duct.

Initially, an unconstrained adjoint optimization is performed on the duct that leads to a total viscous losses drop of 9.1 % after 40 iterations. However, the resulting shape, due to the high number of control points, becomes wavy with areas of very high curvature (fig. 5). Then, a constrained optimization is performed with $\kappa_m = 10^6$. Using the constrained optimization, a drop of 3.9 % is achieved. The curvature map on the optimal solution of the constrained and unconstrained optimization are shown in figure 5 and the final CAD surfaces are shown in fig. 7. It is noticeable from the curvature map that the shape attempts to form the wavy surface but the constraint function halts this process. Finally, the convergence history of the objective function and the constraint, can be seen in fig. 6. The optimization was performed using steepest descent updates and the search direction was computed using ALM.

5.2 The Extruded NACA0012 Wing

The NACA0012 airfoil which lies on the $X - Y$ plane, is fitted by a NURBS curve. This NURBS curve is then extruded along the Z axis to create a wing (fig. 8).

The volume of the wing spanning between the planes at $Z = 0m$ and $Z = 0.5m$ is $V_{init} = 0.0412$. The flow analysis around the wing is performed using a mesh of

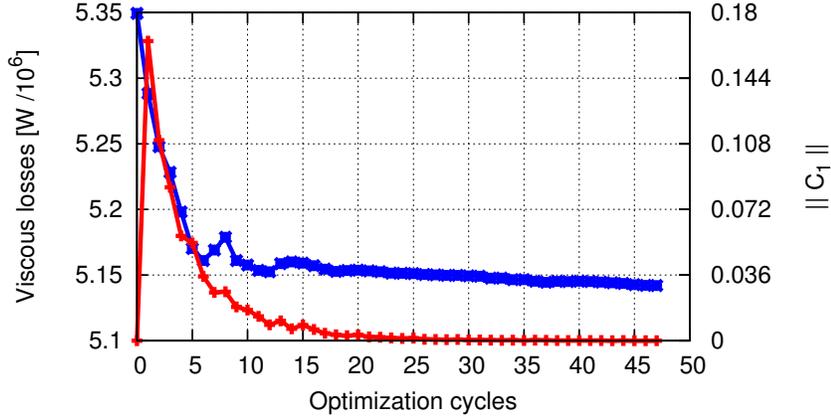


Figure 6: The convergence history of the viscous losses objective function (blue) and the curvature constraint (red) for the S-bend duct.

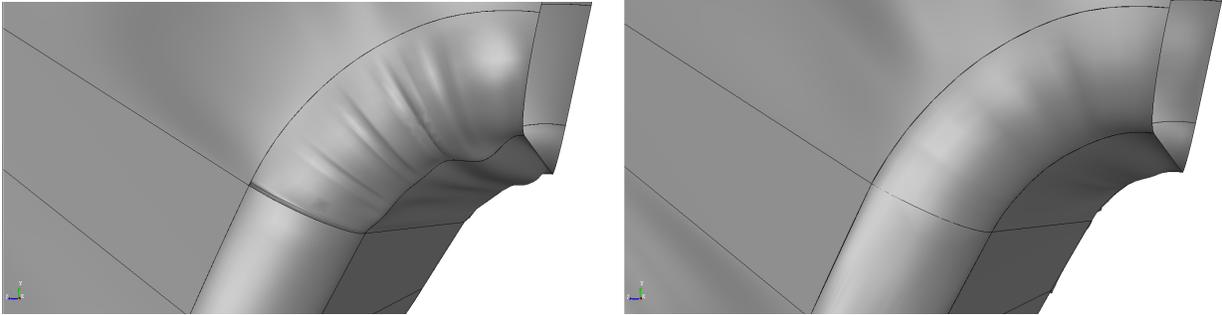


Figure 7: Comparison of the optimal shape resulting from the constrained (right) and the unconstrained (left) optimization of the S-bend duct.

approximately 250,000 cells. The flow is turbulent ($Re = 10^6$) with a free stream velocity $U = 60m/s$ and an angle of attack at 0° . The target of the constrained optimization is to reduce the drag force on the wing while making sure that $V \geq V_{min} = 0.8V_{init} = 0.033$. As shown in fig. 9, the optimization procedure starts with the objective function dominating the constraint but, as the penalty factor of the constraint increases, the optimization converges to a 4.3% drop in the drag force.

6 Conclusions

In this article, the implementation and testing of two geometric constraints is presented. Firstly, a method to constrain curvature was shown. Using this constraint, an optimal NURBS shape can refrain from becoming oscillatory (with high curvature regions) and can have a much smoother surface. This is highly advantageous, since high curvature regions make a model hard to manufacture (or even non-manufacturable). Secondly, a

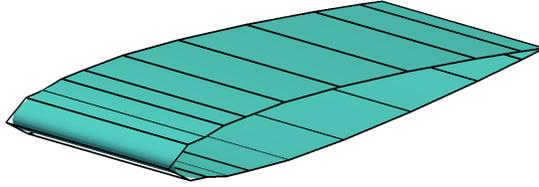


Figure 8: The NACA0012 extruded wing (light blue) and its NURBS control grid (black wireframe).

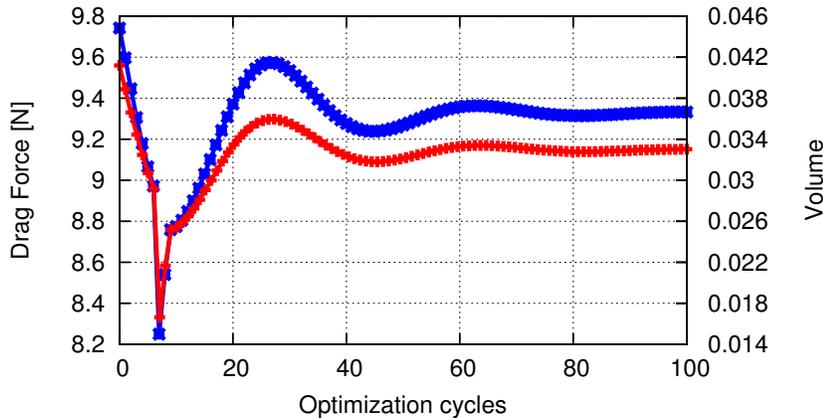


Figure 9: The convergence history of the drag objective function (blue) and the constraint function (red) for the NACA0012 wing.

method to constrain the volume of a watertight NURBS shape was shown. This can be very useful in optimization scenarios, in which the objective function continuously reduces the volume of the shape (similarly to the one shown in section 5.2). Finally, the constraints were tested for effectiveness using a climate duct and an extruded NACA0012 wing.

Acknowledgements

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

REFERENCES

- [1] M. G. Damigos, E. de Villiers Adjoint Shape Optimization using model Boundary Representation. EUROGEN 2017, 12th International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems, Madrid, Spain, September 13–15, 2017.
- [2] D. Brujic, M. Ristic, M. Mattone, P. Maggiore, G. P. Poli CAD based shape optimization for gas turbine component design, *Structural and Multidisciplinary Optimization*, 41(4), 647–659 (2009)
- [3] E. Hardee, K. Chang, J. Tu, K. K. Choi, I. Grindeanu, X. Yu, A CAD–based design parameterization for shape optimization of elastic solids, *Advances in Engineering Software*, 30(3), 185–199 (1999)
- [4] G. Yu, J. Mller, D. Jones, F. Christakopoulos, CAD–based shape optimisation using adjoint sensitivities, *Computers and Fluids*, 46(1), 512–516 (2011)
- [5] P.N. Azariadis, N.A. Aspragathos, Geodesic curvature preservation in surface at-tening through constrained global optimization, *Computer–Aided Design* 33 (2001) 581–591
- [6] H. P. Moreton, C. H. Squin, (1992). Functional optimization for fair surface design, *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*
- [7] O. Pironneau, On optimum design in fluid mechanics. *Journal of Fluid Mechanics* 64(1), 970–110 (1974).
- [8] Jameson, A. Aerodynamic design via control theory, *Journal of Scientific Computing* 3(3), 233–260 September (1988)
- [9] M. B. Giles, M. C. Duta, J.D. Mller, N. A. Pierce, Algorithm developments for discrete adjoint methods, (2001)
- [10] R. Vishnampet, D. J. Bodony, J. B. Freund, A practical discrete–adjoint method for high–fidelity compressible turbulence simulations. *Journal of Computational Physics* 285(C), 173–192 March (2015)
- [11] C. Othmer, Implementation of a continuous adjoint for topology optimization of ducted flows (2007)
- [12] K. C. Giannakoglou, D. I. Papadimitriou, Adjoint Methods for Shape Optimization, 79–108. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
- [13] K. C. Giannakoglou, D. I. Papadimitriou, E. M. Papoutsis–Kiachagias, I. S. Kavvadias, Aerodynamic Shape Optimization Using Turbulent Adjoint And Robust Design in Fluid Mechanics, 289–309. Springer International Publishing, Cham (2015)

- [14] L. Piegl, and W. Tiller, The NURBS book, Springer, Berlin, 1997.
- [15] M. M. Lipschutz, (1969). Differential geometry. New York: McGraw–Hill Education–Europe
- [16] J. Nocedal, S. J. Wright, Numerical Optimization, Springer, New York, 2006.
- [17] Industrial Optimal Design using Adjoint CFD, URL: <http://ioda.sems.qmul.ac.uk/>