International Forum on Aeroelasticity and Structural Dynamics
IFASD 2022
13-17 June 2022, Madrid, Spain

AERO-STRUCTURAL SENSITIVITY ANALYSIS WITH COUPLED ADJOINT METHOD

Claudio Conti[1] **and Andrea da Ronch**[1]

[1]University of Southampton, SO17 1BJ, UK
c.conti@soton.ac.uk
a.da-ronch@soton.ac.uk

Abstract: An efficient and well established technique for Multi Disciplinary Optimization is the adjoint based one. In particular, it is well-suited to aero-structural optimisation problems, since the computational cost for the evaluation of the gradient is independent on the number of design variables. The challenge is in solving the coupled adjoint equations in computational frameworks involving different source codes for each subproblem. This work aims at realizing a coupled adjoint sensitivity analysis tool within a Fluid Structure Interaction interface which enables the communication between SU2 and Nastran for the aerodynamic and structural analysis respectively. Moreover, the work proposes a geometry control methodology where a user defined shape change can be integrated in a shape optimisation with no modifications to the pre-existing codes. This expands the space where the optimal design is being searched, allowing to define a wider variety of optimisation problems where the interpretability of the design variables is more intuitive for designers.

1 INTRODUCTION

The coupling between a flexible structure and the surrounding fluid problem is a challenging task due to the substantial differences in the constitutive equations of each discipline. Much effort has been spent in the past decades in the development of FSI (Fluid-Structure Interface) techniques to tackle this issue, aiming at the realization of computationally efficient and accurate tools computing the aero-structural response. This field also gained momentum thanks to the advancement in computers power and development of computational techniques capable of reducing the computational burden. The level of maturity that has been reached has allowed FSI analysis to be progressively more adopted across the design stages of several industrial applications. However, the great cost of a single evaluation of the coupled solution represents a major limitation when it comes to gradient-based optimization processes, which usually imply a large number of subsequent evaluations before convergence to a solution for the optimisation problem. This is why surrogate models, like [1] normally find wide adoption in this kind of aero-structural frameworks, whose accuracy is normally dependent on the specific problem. Alternatively, low fidelity methods might be chosen [2, 3] decreasing the cost of each single evaluation of the objective function. The number of evaluations can be reduced to improve efficiency and to get to a converged solution more quickly, by improving the accuracy of the gradient evaluation. Techniques like complex step, or direct methods [4, 5] are normally chosen to achieve a highly accurate estimation of the optimisation gradient, which allow tackling some of the issues related to finite differences approaches [6]. Nevertheless, when a large number of design variables is involved, evaluating the gradient can be demanding and prohibitive. This is a typical scenario when it comes to aerodynamic shape or aero-structural optimisation, where for large models either a large number of structural properties or geometry morphing parameters is involved. Adjoint methods provide highly accurate gradients and, most importantly, break the link between computational cost and number of design variables. Their adoption in the aero-structural optimisation field was initiated by [7], and later followed by [8], and [9]. The aim of the work here presented is to develop a sensitivity analysis tool for a high–fidelity aero–structural optimisation. A coupled adjoint technique is being adopted, allowing an accurate and efficient evaluation of the gradient regardless of the number of design variables. The computational framework consists in an FSI interface between SU2, [10] an open–source software for CFD and multi–physics simulations, and MSC Nastran, a well established industrial Finite Element Analysis (FEA). The work is built around the following objectives:

- Realization of a FSI interface for computation of the aeroelastic response
- Realization of a custom shape control morphing tool, capable of changing consistently the shape of both the structural and aerodynamic geometry
- Implementation of the coupled adjoint equations and solution algorithm

What is presented here is a state of the art of this activity, showing some preliminary results suggesting validation of the approach being pursued. A shape morphing tool has been realized, enabling a mutual control of CFD mesh and corresponding Finite Element Model (FEM). A modification of the pre-existing SU2 adjoint based optimisation workflow is here proposed. This allows an alternative aerodynamic shape optimisation to the Free Form Deformation based one currently implemented in SU2, enabling the definition of non standard optimisation problems.

2 COUPLED ADJOINT SENSITIVITY ANALYSIS

Optimisation problems normally rely on the definition of an objective function, which could be representative of whatever physical response one intends to either minimise or maximise. Assume that the constitutive equation of a general problem is expressed as

$$\mathcal{G}(\mathbf{w}(\alpha), \alpha) = \mathbf{0} \tag{1}$$

then, if $J = J(\mathbf{x}(\alpha), \alpha))$ is the function of interest (or objective function), the optimisation problem can be formulated as

$$\begin{aligned} &\min_{\alpha} J \\ \text{s.t.} \quad &\mathcal{G}(\mathbf{w}, \mathbf{w}(\alpha)) = \mathbf{0} \end{aligned} \tag{2}$$

where α collects the design variables while $\mathbf{w}$ indicates the state variables. Note that Eq. 1 underlines the dependencies of these latter on the design variables themselves. In absence of further conditions, fulfillment of the governing equation is the only constraint that every design represented by an assignment of α has to meet. Let then introduce an associated Lagrangian, defined as

$$\mathcal{L} = J(\mathbf{w}, \alpha) + \tilde{\mathbf{w}}^{\mathbf{T}}\mathcal{G}(\mathbf{w}, \alpha) \tag{3}$$

with $\tilde{\mathbf{w}}$ defined as the adjoint or dual variable of the problem. Imposition of Karush Kuhn Tucker (KKT) conditions ([11]) on Eq. 3 leads to a rearrangement of the governing equation as

$$\frac{\partial \mathcal{L}}{\partial \tilde{\mathbf{w}}^T} = \mathcal{G}(\mathbf{w}, \alpha) = \mathbf{0} \tag{4}$$

A local minimum, solution to Eq. 1, has to satisfy stationary condition, expressed as

$$\frac{\partial \mathcal{L}}{\partial \alpha} = 0 \tag{5}$$

which, on the base of Eq. 3, leads to

$$\frac{\partial J(\mathbf{w},\alpha)}{\partial \alpha} + \tilde{\mathbf{w}}^T \frac{\partial \mathcal{G}(\mathbf{w},\alpha)}{\partial \alpha} = 0 \tag{6}$$

while Eq. 1 brings to the adjoint equation, that is

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \frac{\partial J(\mathbf{w},\alpha)}{\partial \mathbf{w}} + \tilde{\mathbf{w}}^T \frac{\partial \mathcal{G}(\mathbf{w},\alpha)}{\partial \mathbf{w}} = 0 \tag{7}$$

Finally, noting that the derivative of the residual of the governing equation with respect to the state variable can be rewritten introducing the problem Jacobian, that is $\mathbf{K}_{\mathcal{G}}^T \tilde{\mathbf{w}} = -\frac{\partial J^T}{\partial \mathbf{w}}$, and recombining Eqs. 3 and 7, one can finally get

$$\frac{dJ}{d\alpha} = \frac{\partial J}{\partial \alpha} + \tilde{\mathbf{w}}^T \frac{\partial \mathcal{G}}{\partial \alpha} \tag{8}$$

Eq. 8 provides an estimation of the optimisation gradient, based on the computation of the adjoint vector, $\tilde{\mathbf{w}}$, which has to be worked out from Eq.7. Hence, Eqs. 7 and 8 make up the so called dual problem, a system of equations which leads to a computation of the gradient whose cost is independent on the number of design variables.

For coupled problems, like the FSI ones, the governing equation has the form of a system, where each component represents the governing equation of each subproblem (either structural or fluid-dynamics). Then, by assuming that

$$\mathcal{G} = \begin{Bmatrix} \mathcal{F} \\ \mathcal{S} \end{Bmatrix} \tag{9}$$

with $\mathcal{F}$ and $\mathcal{S}$ being the residuals of the fluid and structural problem, respectively, the entire formulation can be specialized for the coupled case, leading the following formulation of the dual problem

$$\begin{gathered} \begin{pmatrix} \frac{\partial \mathcal{F}}{\partial \mathbf{w}} & \frac{\partial \mathcal{F}}{\partial \mathbf{w}} \\ \frac{\partial \mathcal{S}}{\partial \mathbf{u}} & \frac{\partial \mathcal{S}}{\partial \mathbf{u}} \end{pmatrix} \begin{Bmatrix} \Psi \\ \Phi \end{Bmatrix} = - \begin{pmatrix} \frac{\partial J}{\partial \mathbf{w}} \\ \frac{\partial J}{\partial \mathbf{u}} \end{pmatrix} \\ \frac{\partial J}{\partial \alpha} = \frac{dJ}{d\alpha} + \Psi^{\mathsf{T}} \frac{\partial \mathcal{F}}{\partial \alpha} + \Phi^{\mathsf{T}} \frac{\partial \mathcal{S}}{\partial \alpha} \end{gathered} \tag{10}$$

where $\mathbf{w}$ and $\mathbf{u}$ represent the aerodynamic and structural state variables respectively.

Solution of Eq. 10 is a challenging task. Specifically, the evaluation of the derivatives appearing in the system matrix, that is $\partial\mathcal{F}/\partial\mathbf{w}$, $\partial\mathcal{F}/\partial\mathbf{u}$, $\partial\mathcal{S}/\partial\mathbf{w}$ and $\partial\mathcal{F}/\partial\mathbf{u}$, is not trivial, especially because the solution technique depends on the specific formulation adopted for each subproblem and on the software architecture. Several are the approaches available in literature for the solution of Eq. 10 [12]. SU2 already provides a tool implementing the aerodynamic shape optimization with adjoint method. This can be executed in two different fashions, that is the continuous and discrete adjoint solver. The former, proposes a strategy where the problem Jacobian in Eq. 7 is worked out analytically, depending on the flow problem one wants to solve (Euler, Navier-Stokes or RANS). The discrete adjoint, instead, exploits the capabilities of the Automatic Differentiation (AD) through the `CoDiPack` library [13], which allows the solution of the aerodynamic adjoint equation without an explicit formulation of the problem Jacobian. This is achieved through the application of the AD reverse mode sequence to the consecutive iterations of the CFD solver, as illustrated in [14], which allows a quicker, accurate and independent on the flow model solution of the adjoint equation. The challenge, however, is how to include the structural field. A well established approach to solve the equation is the *Lagged Coupled Adjoint* (LCA) method, as proposed by [15]. It relies on a decoupling of the two disciplines, which is particularly convenient for those partitioned FSI architectures which do not allow modifying much the pre-existing workflow built in each subproblem software. From this perspective, Eq. 7 is rearranged as

$$\begin{aligned}
&\underbrace{\frac{\partial \mathcal{F}}{\partial \mathbf{w}}\Psi = -\frac{\partial J}{\partial \mathbf{w}}}_{\text{Aerodynamic adjoint equation}} -\frac{\partial \mathcal{S}}{\partial \mathbf{w}}\tilde{\Phi} \\
&\underbrace{\frac{\partial \mathcal{S}}{\partial \mathbf{u}}\Psi = -\frac{\partial J}{\partial \mathbf{u}}}_{\text{Structural adjoint equation}} -\frac{\partial \mathcal{F}}{\partial \mathbf{u}}\tilde{\Phi}
\end{aligned} \tag{11}$$

Eq. 11 suggests that the off-diagonal terms appearing in the Jacobian sensitivity matrix of Eq. 7 can be treated as external forcing terms on the mono-disciplinary adjoint equations. Vectors $\tilde{\Phi}$ and $\tilde{\Psi}$ denote the lagged structural and aerodynamic adjoints, respectively. In an iterative approach to solution of Eq. 11, these are evaluated at the previous step, while the algorithm is carried on until convergence.

The derivative $\partial\mathcal{F}/\partial\mathbf{w}$ represents the dependency of the aerodynamic Jacobian on the state variables. It is normally a large matrix (dimensions depending on the size of the CFD mesh), prohibitive to store. That's why it is normally matrix –free form approaches are adopted. The off-diagonal term $\partial\mathcal{F}/\partial\mathbf{u}$ in the LCA, Eq. 11, represents the effect of structural displacements on the CFD residuals through the perturbation of the mesh. When a given structural node moves, both the surface and the interior of the CFD grid must be perturbed, thus affecting a large number of CFD mesh points. Even though the flow variables are constant in the calculation of this partial derivative, the change in the mesh geometry affects the sum of the fluxes, whose variation is easily obtained by recalculating the residuals for the perturbed cells. For the structural side, instead, the derivative $\partial\mathcal{S}/\partial\mathbf{u}$ represents the dependency of the structural residual on the structural displacements. The computation of this term depends mainly on the structural properties (stiffness and mass), depending on whether a static or dynamic formulation is considered. On the other hand, the off-diagonal derivative appearing in the first equation, $\partial\mathcal{S}/\partial\mathbf{w}$ represents the impact of the aerodynamic state variables on the structural residuals. Structural properties are uncorrelated to the aerodynamics and the further assumption that structural displacements are not directly affected by these, can be done. Further details are provided in the upcoming sections.

3 SHAPE CONTROL FOR OPTIMISATION

A fundamental aspect of optimisation problems is the choice of design variables. This strictly depends on the nature of the problem in study and on the type of objective function one wants to optimise. For a structural optimization, for instance, design variables could be the properties of the Finite Elements defining the numerical model (thicknesses of panels, cross-section areas of beam-like elements, ...). Concerning an aerodynamic shape optimisation, the morphing of the geometry is a function of prescribed design variables. The formulation of the geometrical morphing is defined by the shape control that one wants to achieve and determines the space of the available solutions and the conditioning of the constraints. Assessing the choice of shape control functions highly affects the robustness and final result of the optimisation problem. For a given discretisation of the geometry, say $\mathbf{x} = [x_1, ..., x_N]^{\mathsf{T}}$, its generalised dependency on the design variables can be expressed as

$$\mathbf{x}(t) = \sum_i \phi_i(t)\alpha_i \tag{12}$$

where $\Phi_i(t)$ is the i-th basis function defining the shape design space, t the shape control parameter and α_i the value of the corresponding design variable. If the objective shape gradient, that is $\mathbf{g}_x = dJ/d\mathbf{x}$ is provided, the optimisation gradient can then be expressed as the following inner product

$$\mathbf{g}_{\alpha_i} = \frac{dJ}{d\alpha_i} = \int \phi_i(t) \cdot \frac{dJ}{dx(t)} dt \tag{13}$$

which for a fixed discretisation, where the topology is unchanged, Eq. 13 can be linearly expressed as

$$\mathbf{x} = \mathbf{\Phi}^{\mathsf{T}}\alpha \tag{14}$$

being $\mathbf{\Phi}$ the $m \times n$ linear shape control matrix. Through an optimisation cycle, the update of the design variable at the k-th iteration can be expressed as

$$\alpha^{k+1} = \alpha^k + \mathbf{D}^k \mathbf{g}_\alpha \tag{15}$$

where $\mathbf{D}^k$ is the operator identifying the descent direction towards the local minimum. Combining of Eq. 15 with 12 leads to

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{\Phi}\mathbf{D}^{\mathsf{k}}\mathbf{\Phi}^{\mathsf{T}}\mathbf{g}_{\mathsf{x}} \tag{16}$$

Such an expression underlines, as pointed out by [16], how the choice of the shape control matrix can act as a preconditioner of the descent for a gradient-based optimisation. This clearly suggests how the shape control strategy directly affects the search for an optimal design. A very typical classification of shape morphing techniques splits them in two categories, which are the constructive and deformative. In this latter, each new shape is generated as a deformation of a baseline geometry. On the other hand, a constructive approach aims at generating from scratch such a geometry, with the advantage of not being restricted to a fixed topology, which is the main limitation for deformative methods. However, these find their application where an initial shape cannot be generated from a set of shape control functions, Φ.

Several morphing methods have been proposed in literature, like for instance the Hicks-Henne Bumps (HHB), where the baseline geometry is perturbed with a set of smooth sinusoidal bump functions, whose width and location can be controlled by two parameters for each bump. This provides the ability to concentrate in those regions of the geometry where the objective function is more sensitive to shape changes, while leaving unchanged the shape where the objective function is not expected to be much affected. However, these functions are not orthogonal and do not make up a basis which can guarantee the existence of an optimal solution, as underlined by [17]. Free Form Deformation (FFD) describes methods in which a continuous displacement field is defined over a volume based on the movement of a set of control points. The evaluation of the displacement field at the surface points provides the surface deformations. The method is completely independent of the particular representation of the body to deform and inherently maintains smoothness. Furthermore, the parameterisation is simple to setup, allowing the user to specify degrees of local and global control. The volume transformation can be defined using a variety of methods including multivariate Beziers and B-splines, and RBFs. These are already implemented and available for users in SU2 for shape aerodynamic optimisation problems, both for finite differences or adjoint method. Although this latter tackles the issue of handling a large number of design parameters, which is normally the case especially when FFD strategy is implemented on a large geometry, these approaches struggle to find their application in industrial frameworks. This is normally because they potentially can lead to optimal shapes which, because of manufacturing requirements and constraints, are not realizable. The difficult interpretability of those parameters controlling these shape control techniques from the perspective of a designer, can make them unattractive for industrial purposes and motivates our effort in implementing a new interpretable parametrization.

4 IMPLEMENTATION

Here further details on the implementation of the method described above are described. First, an overview of the aerodynamic adjoint optimisation is provided, then the workflow for the structural analysis is introduced.

4.1 Aerodynamic Adjoint based optimisation

The aerodynamic optimisation adopted in this study exploits the SU2 built–in discrete adjoint solver described in the previous paragraph. However, it is in the author's interest to develop a customized shape control tool which allows an external user to define a generic shape parameter as design variable of the optimisation problem. This study proposes the implementation of a tool which allows to do so without interfering with the SU2 internal structure. This is realized by exploiting the python wrappers currently available in SU2. These are based on the `SWIG` library [18], which allows control of the solvers workflow, written in C++, through high level python scripting. A schematic representation of the architecture of how low level SU2 solvers are integrated in a python environment is illustrated in Fig. 1. This makes it possible to access mesh deformation routines, overwriting the coordinates of the surface mesh and propagating the geometry displacement across the volume mesh. This latter follows the deformation of the moving boundary as a perturbed elastic net, complying with elasticity-like equations, similar to the Finite Element problem. In regards to the geometry shape control, two approaches have been implemented, one based on the RBF interpolation method and a direct approach. While the former uses structural nodes as control points, whose imposed displacement is then interpolated onto the surface CFD mesh, the latter applies the prescribed displacement straight onto both CFD mesh and FE model. This raises the question on how to use a user-defined geometrical shape control functions, as design variable for the aerodynamic optimisation, which means computing a gradient representing the derivative of the objective function. Surface sensitivity analysis is already available in SU2, providing a tool for the estimation of the gradient of the

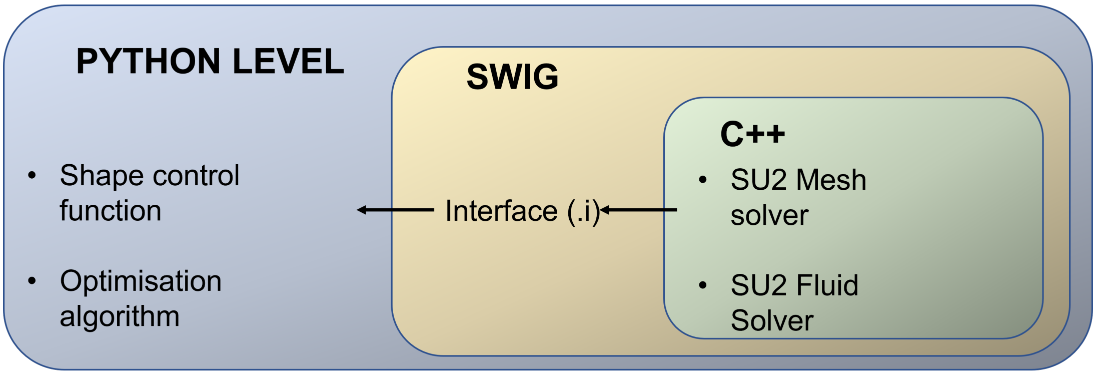


Figure 1: Wrapping schematic architecture

objective function with respect to the position of the surface mesh points, which will be here indicated as $dJ/d\mathbf{p}$ (where $\mathbf{p}$ collects the coordinates of the surface points). Therefore, the optimisation gradient can be worked out by means of the chain rule, as

$$\frac{dJ}{d\alpha} = \frac{dJ}{d\mathbf{p}} \cdot \frac{d\mathbf{p}}{d\alpha} \tag{17}$$

where $d\mathbf{p}/d\alpha$ can be computed within the morphing script. This is a vector with the same dimension as the number of surface mesh points. For large 3D fine meshes, the computation of this quantity can be prohibitive with classic finite differences schemes. As long as simple shape morphing functions are involved, derivation of this partial derivative can be pretty simply achieved analytically, but for more sophisticated shape changes this can become a bit more challenging. To deal with this aspect of the implementation, an Automatic Differentiation (AD) approach has been pursued, exploiting the capabilities of the `Algopy` open source library [19]. Specifically, the forward mode of AD has been adopted, which is much more computationally efficient for those cases where the number of outputs (variables to be differentiated) is much higher than the number of inputs [20] (variable with respect to which we are differentiating).

4.2 Structural Adjoint based optimisation

In regards to the structural optimisation, recall first the governing equation for the static Finite Element response, in a residual form, that is a $N_{dof} \times N_{dof}$ [1] system expressed as

$$\mathcal{S} = \mathsf{K}\mathbf{u} - \mathsf{f} = 0 \tag{18}$$

where K is the stiffness matrix, while $\mathbf{u}$ and f represent the field of structural displacements and applied forces respectively. The adjoint equation for this problem corresponds to the second component of Eq. 25, without the contribution of the aerodynamic Jacobian $\partial\mathcal{F}/\partial u$, therefore

$$\frac{\partial\mathcal{S}}{\partial\mathbf{u}}\Phi = -\frac{\partial I}{\partial\mathbf{u}} \tag{19}$$

Based on Eq. 18, with the assumption that the forces do not depend on the structural deformation, being a linear structural operator self-adjoint (Betti's theorem), the only contribution to the structural Jacobian comes from the stiffness matrix. Hence

$$\frac{\partial\mathcal{S}}{\partial\mathbf{u}} = \mathsf{K} \tag{20}$$

The structural adjoint equation can then be rewritten as

$$\mathsf{K}\Phi = \frac{\partial J}{\partial\mathbf{u}} \tag{21}$$

The derivative at the right hand side of Eq. 19 depends on the choice of the objective function. For typical aeronautical problems, structural optimisation aims at the reduction of either weight (or volume equivalently) or structural stresses, like in load alleviation problems. Assuming a simple example where one wants to minimise the static wing root bending moment (or a structural stress in any location of the FEM), this can be worked out from the stiffness properties, once a field of displacement under an applied load is computed. Therefore the structural adjoint vector will be uniquely dependent on intrinsic structural properties.

[1] N_{dof} indicates the number of Finite Element degrees of freedom

If in a coupled aeroelastic problem the objective function is a purely aerodynamic quantity, like for instance the drag, the interpretation of the equation changes. The structural stiffness, as well as the field of displacement are not directly linked to an aerodynamic force, which is just an integration of a field of pressure distribution. Therefore, $\partial J/\partial \mathbf{u}$ will evaluate how a given field of displacement, or a structure deformation, warps the geometry on which a given pressure field is applied.

The computation of the adjoint vector supports the evaluation of the optimisation gradient, as in Eq. 8. Then, a formulation for the derivative $\partial \mathcal{S}/\partial \alpha$ has to be defined. This quantifies the sensitivity of the structural residual to a chosen design variable. As long as a shape optimisation is involved, the only contribution of a shape control parameter to the structural residual is through the applied forces, since neither the Finite Element stiffness or the field of displacements depend on a geometrical parameter. Therefore,

$$\frac{\partial \mathcal{S}}{\partial \alpha} = \frac{\partial \mathbf{f}}{\partial \alpha} \tag{22}$$

which allows to recast the gradient expression as

$$\frac{dI}{d\alpha} = \Phi_l^{\mathsf{T}} \frac{\partial \mathbf{f}}{\partial \alpha} \tag{23}$$

Within a FSI interface, where the applied forces are evaluated by standalone CFD software, the derivative in the right hand side of Eq. 23 represents the derivation of the operator integrating a pressure field, which is assumed to be independent on the displacement field itself. This has a similar meaning to $\partial J/\partial \alpha$ when the function of interest if an aerodynamic force, as previously mentioned.

5 NUMERICAL RESULTS

Here some numerical results of the work conducted so far are illustrated. First is a simple problem, where the trivial solution validates the approach for the computation of the gradient and its integration in the optimisation workflow. This is proposed both as an unconstrained and constrained problem. Second comes the shape optimisation of a finite-span wing, where the effect of the planform shape change is highlighted.

5.1 Unconstrained NACA 0012 Aerofoil

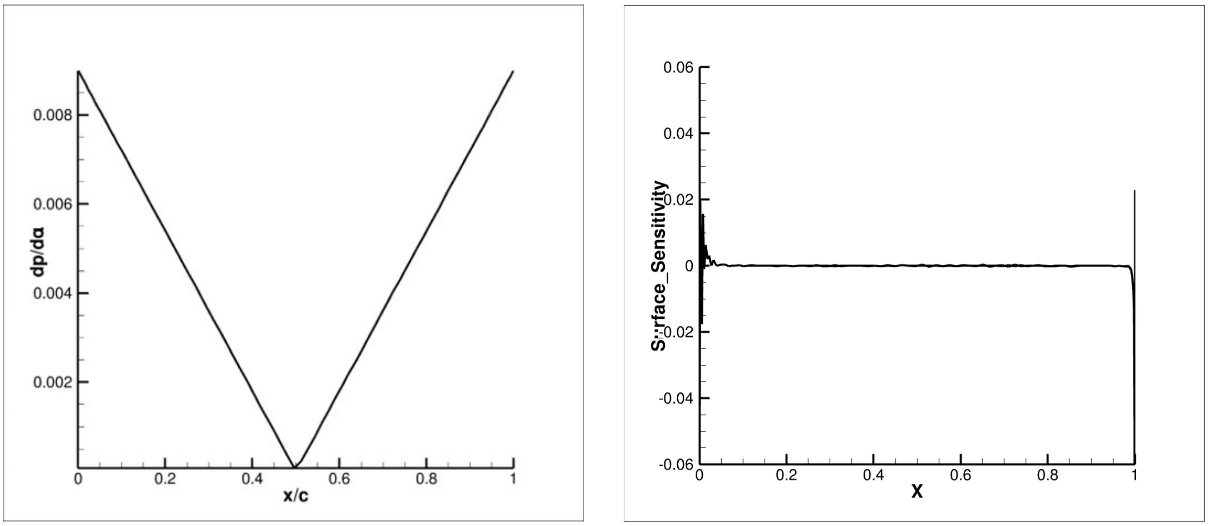
The first test case to be presented is a NACA 0012, where the optimisation problem is defined as follows

$$\min_{0<\alpha<10} \quad C_l \tag{24}$$

This is a prototype aerodynamic problem which has been used to validate the whole procedure involving morphing of the geometry and gradient estimation with respect to custom design variables.

Some preliminary analyses have been carried out for the purpose. Here, the objective function is the lift coefficient, C_l, while the design variable defining the morphing problem is the twist rotation angle, which coincides for a symmetric aerofoil with the angle of attack. Therefore, numerically the gradient should coincide with the lift coefficient slope, as

$$\frac{dC_l}{d\alpha} = C_{l,\alpha} \tag{25}$$



(a) $d\mathbf{p}/d\alpha$ (b) Lift surface sensitivity

Figure 2: NACA 0012 aerofoil: geometric sensitivities

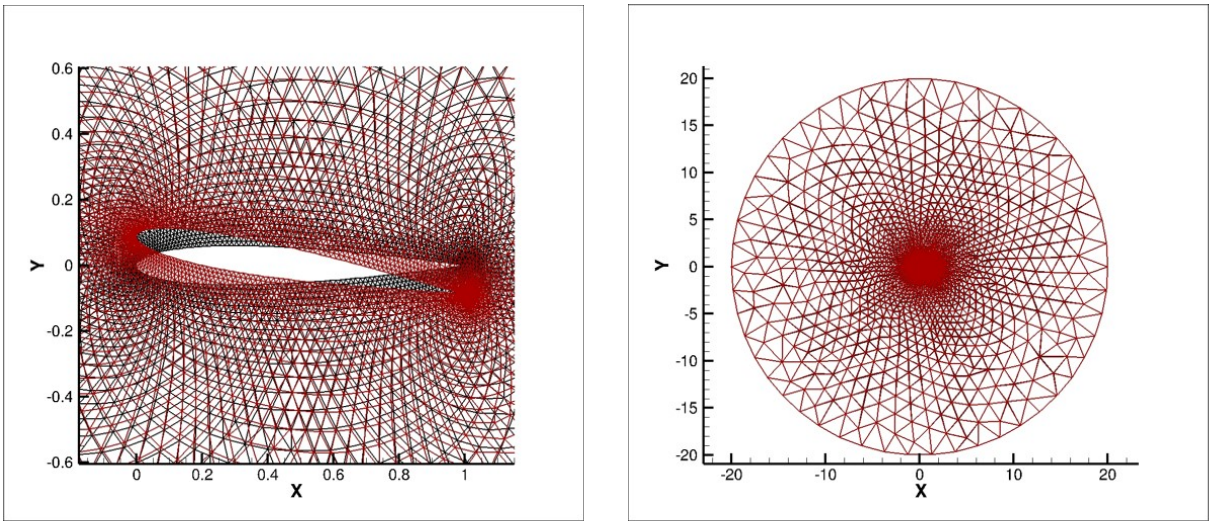
The gradient is worked out through the specialization of Eq. 17 to the presented test case, where the lift surface sensitivity and partial derivative $d\mathbf{p}/d\alpha$ are illustrated in Fig. 2, which highlights how the leading and trailing edge are the regions where a shape perturbation affects most the lift coefficient. The lift slope was obtained interpolating between two points in proximity of the value of the twist angle at which the gradient was evaluated. Results were obtained with the Euler solver, at Mach number $M = 0.3$. The grid consists of 5233 points, with a farfield at 20 chords from the aerofoil.

Twist (deg)	Gradient	Lift curve slope
5	0.1256	0.1253

Table 1: NACA 0012 aerofoil: comparison between gradient $dC_l/d\alpha$ and lift curve slope $C_{l,\alpha}$

Table 1 summarises the data. The error appears in the fourth digit only.

Then, a preliminary investigation on the mesh deformation had to be carried out. Indeed, throughout the optimization cycle, new design variables and configurations are assigned, and the objective function is evaluated at each of these iterations. The morphing of the geometry, as previously anticipated, determines a deformation of the mesh boundary (surface mesh of a wing or an aerofoil), which has to be propagated across the volume mesh. In the elasticity-like solver offered by SU2, the local stiffness of the volume mesh can be modeled with a wall-distance or inverse-volume approach. This latter works out the stiffness as a function inversely proportional to the volume of the mesh cell, meaning higher flexibility in proximity of the moving boundary. In the other case, mesh stiffness linearly increases with the distance from the moving boundary, tending to infinity at the farfield, which in both approaches is kept fixed. The effect of a variable stiffness on the mesh deformation is illustrated in Fig. 3. Specifically, Fig. 3ashows how the deformation of the volume cells conforms with the imposed displacement of the boundary, while Fig. 3b highlights how the farfield remains fixed. For this background validation, CFD simulations have been executed with a prescribed change of the geometric twist angle of 5 and 10 degrees, and the results have been compared to the estimated lift coefficient of a benchmark



(a) Aerofoil closeup

(b) Farfield view

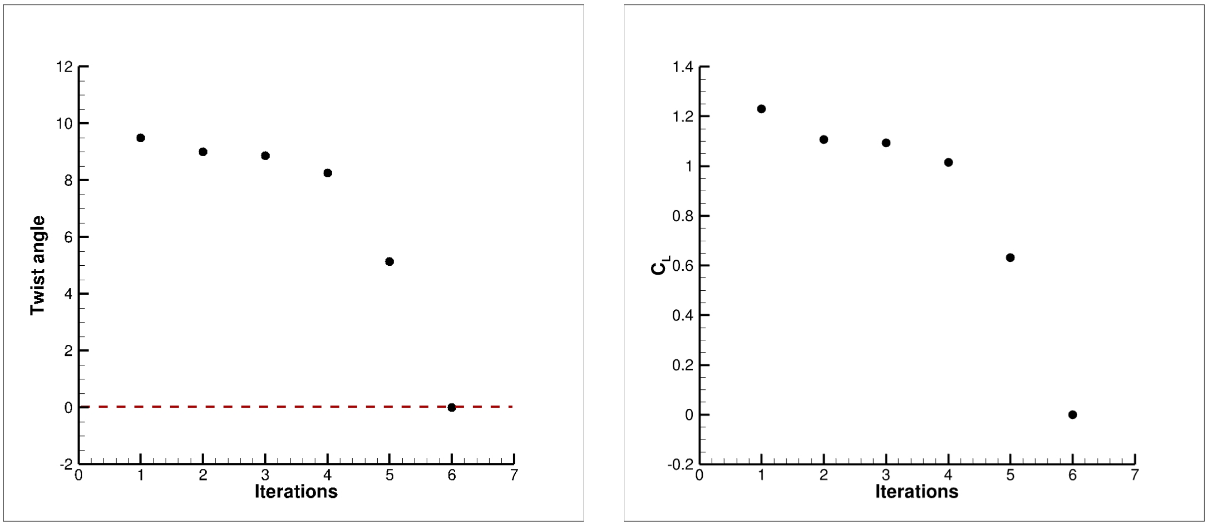
Figure 3: NACA 0012 aerofoil: volume mesh warping due to twist angle

simulation on the baseline undeformed mesh, where the angle of attack is set to these values. Results are showed in Tab. 2, which underlines how the quality of the mesh isn't affected by such a deformation.

Twist angle (deg)	Morphed	Undeformed	Relative error (%)
5	0.6268	06142	2.01
10	1.2307	1.2452	1.16

Table 2: C_l comparison between twisted geometry and benchmark SU2 estimation at equivalent AOA

Results of the optimisation problem defined in Eq. 24 are illustrated in Fig. 4. The convergence of the optimizer to the lower boundary (for a symmetric aerofoil the solution is trivial as the C_l linearly decreases with the angle of attack) proves the correctness of the approach.



(a) Twist angle

(b) Lift Coefficient

Figure 4: NACA 0012 aerofoil: history of optimisation convergence M = 0.3

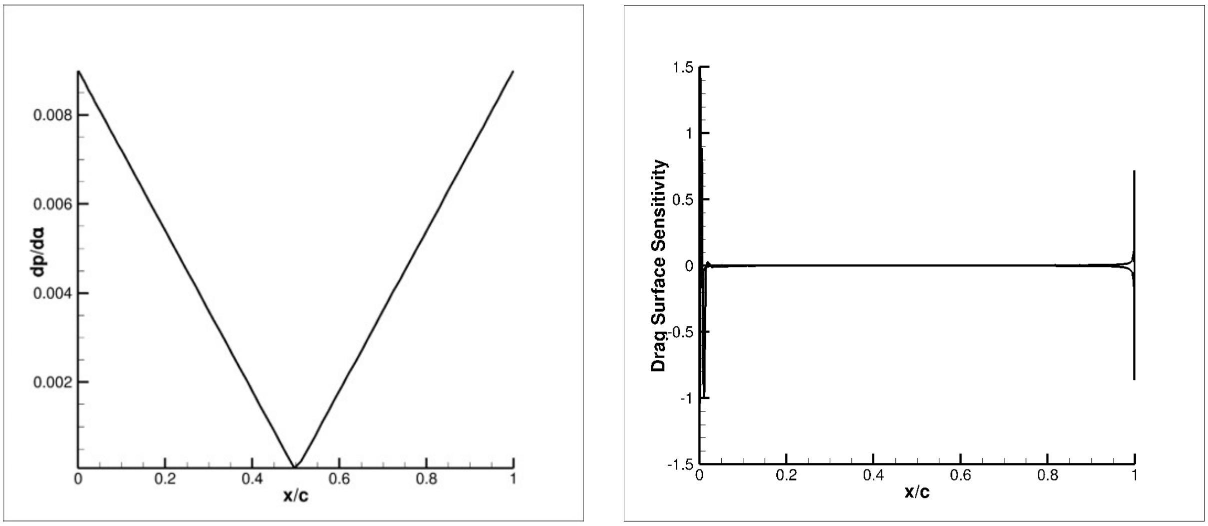
These results were obtained both with the direct surface mesh morphing and with the RBF interpolation from a FE model, with no numerical difference being observed. However, the interpolation approach relies on the computation of a large matrix mapping the two different domains, increasing the computational cost and time of execution both for the geometrical morphing and for the gradient computation, leading to a longer time of execution of the optimization workflow.

5.1.1 Constrained NACA 0012 Aerofoil

Similarly to the previous one, a constrained optimization test case is here shown. The problem consists in finding the twist angle rotation minimizing the drag coefficient, subject to a constraint on a target lift coefficient. The problem can be formulated as

$$\begin{aligned} &\min_{0<\alpha<10} \quad C_d \\ &\text{s.t.} \quad C_l - C_l^* > 0 \end{aligned} \tag{26}$$

For this case, the target lift coefficient was set to $C_l^* = 0.32$, exemplary. The optimiser has to be fed with both an analytical expression for the inequality constraint equation, that is $C_l - C_l^* > 0$, and with its slope, which is the derivative with respect to the design variable. In this case, this slope has the same meaning as for the gradient of the previous case, which is basically $C_{l,\alpha}$. Its computation follows then the same approach. For the optimisation gradient, instead, the drag surface sensitivity to be projected on the design variable, as of Eq. 17, is displayed in Fig. 5

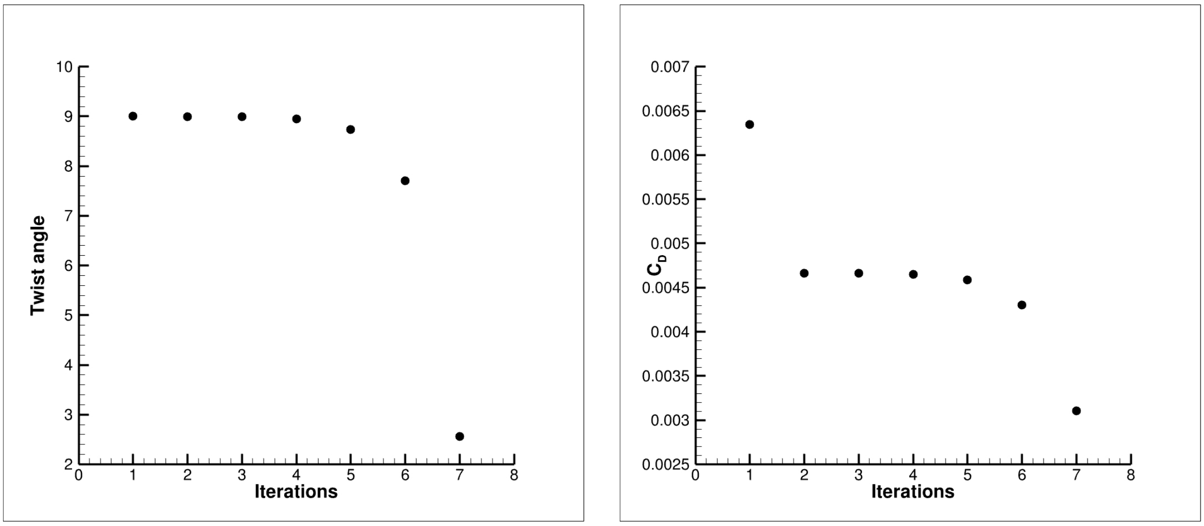


(a) $dp/d\alpha$ (b) Drag surface sensitivity

Figure 5: NACA 0012 aerofoil: geometric sensitivities

This trend shows how the regions in proximity of the leading and trailing edge are those who mostly influence the aerofoil drag. Small shape changes around these locations will have a much higher impact on the objective function than on any other part of the aerofoil.

Fig. 6 shows the trend of the objective function (drag coefficient) and design variable (geometric twist angle) at each optimisation iteration. The numerical results of the optimization are summarized in Tab. 3, showing a net drop in the C_d



(a) Twist angle (b) Drag coefficient

Figure 6: Values of C_d and Twist angle at each optimisation iteration

Initial C_d	Final C_d	Final Twist Angle
0.0063	0.0031	2.56

Table 3: Results of constrained C_d optimisation

5.2 Finite–span Wing

Here a 3D test case is proposed. The model is the same adopted in the investigation carried out by [21], consisting in a rectangular wing, with a NACA 0012 section. Wingspan is 3.06 m, chord is 1.0 m. The optimisation problem is formulated as

$$\min_{-15<\Lambda<0} \quad C_d \tag{27}$$

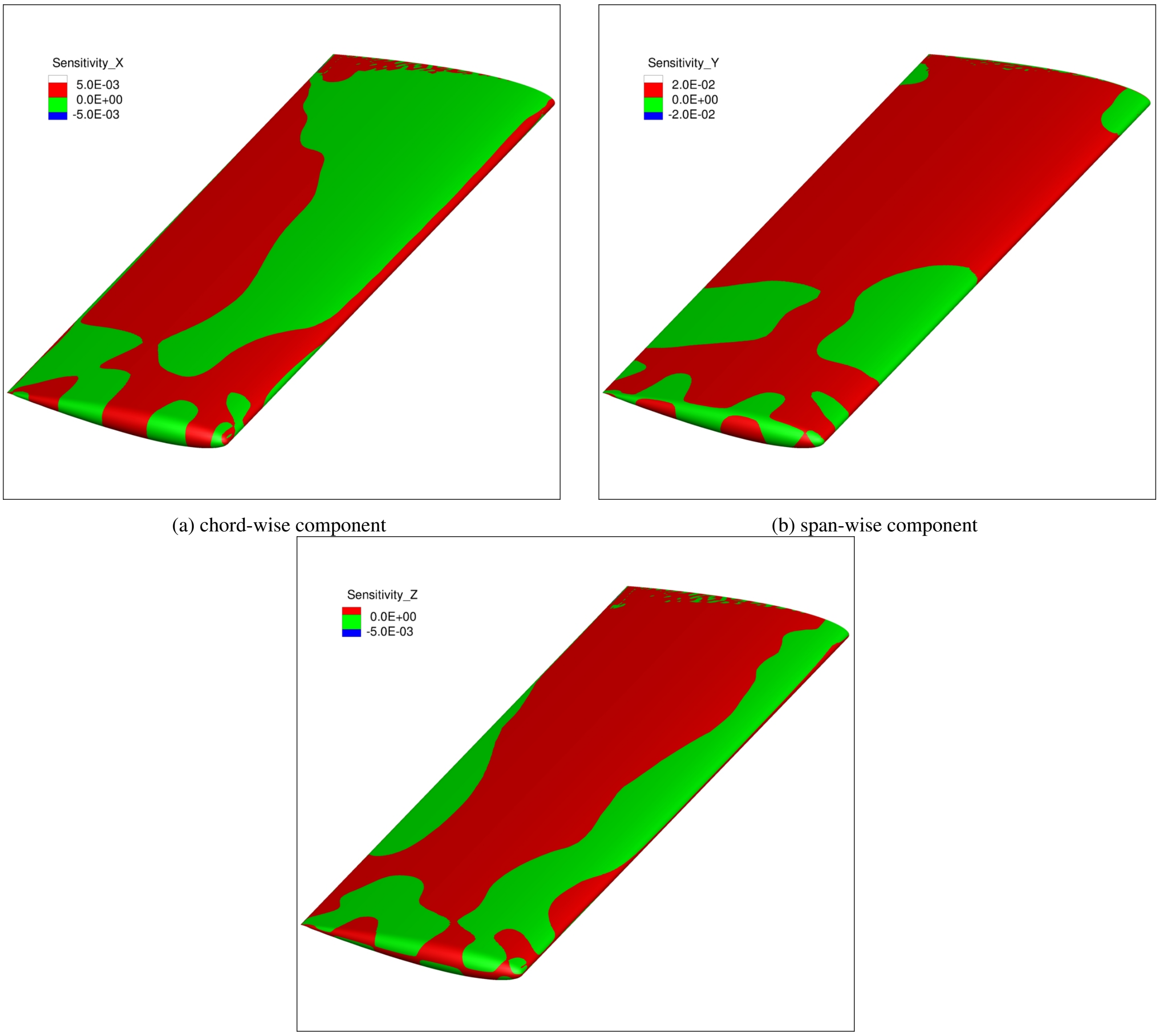
that is a search of the sweep angle (Λ) which minimizes of the drag coefficient. The expectation here is for a beneficial effect of a backward swept wing, as it contributes positively to the reduction of the induced drag. The numerical setup is summed up in Tab. 4.

Name	Value
Mach	0.75
Flow model	Euler
AOA (deg)	1.0
CFL	5.0

Table 4: Numerical setup of CFD simulation

As for the previous test cases, a surface sensitivity has to be generated first. Then, the computation of $d\mathbf{p}/d\alpha$ allows to evaluate the internal product of Eq. 17 for the optimisation gradient estimation. The distribution of the surface sensitivity is illustrated in Fig. 7, which shows it in its three components in the reference system.

Among them, the vertical component appears to be slightly dominating, meaning that a surface displacement in such a direction (like a thickness increment) affects more the drag coefficient.



(a) chord-wise component

(b) span-wise component

(c) vertical component

Figure 7: Rectangular wing from [21]: surface drag sensitivity at M = 0.3

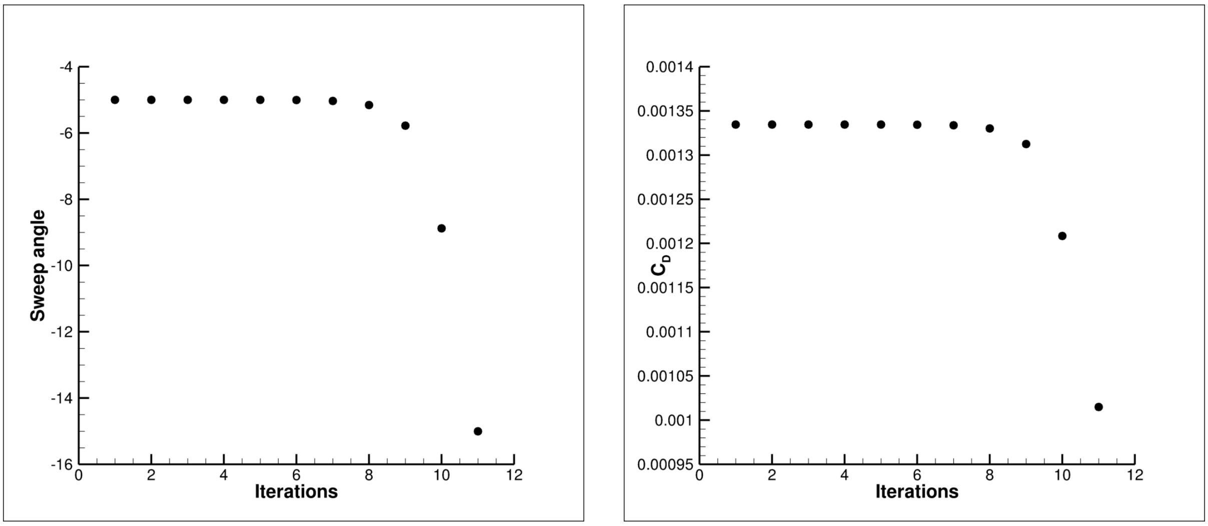
In regards to the chord-wise trend, both the x and z component of the sensitivity suggest a higher value around the leading and trailing edge areas. Each section of the wing, indeed, is a NACA 0012 aerofoil which, net of 3D aerodynamic effects, has the same characteristics shown in Fig. 5b. This trend changes towards the wing tip, as for the onset of 3D aerodynamic effects.

Results of this optimisation problem are summarized in Fig. 8, showing the value of the drag coefficient and sweep angle at each optimisation iteration. For a more generalised formulation of the problem, the baseline wing has been first morphed to an initial sweep angle $\Lambda_0 = -5.0$, which is assumed as the initial configuration where the algorithm starts.

Fig. 9 shows a comparison between the pressure coefficient (C_p) distribution on the initial wing and the optimized one. In particular, Fig. 9c underlines how the influence of the sweep angle is beneficial in reducing the intensity of the shock wave which on the baseline geometry appears around 30% of the chord. It also contributes to moving the shock wave upstream.

6 CONCLUSIONS

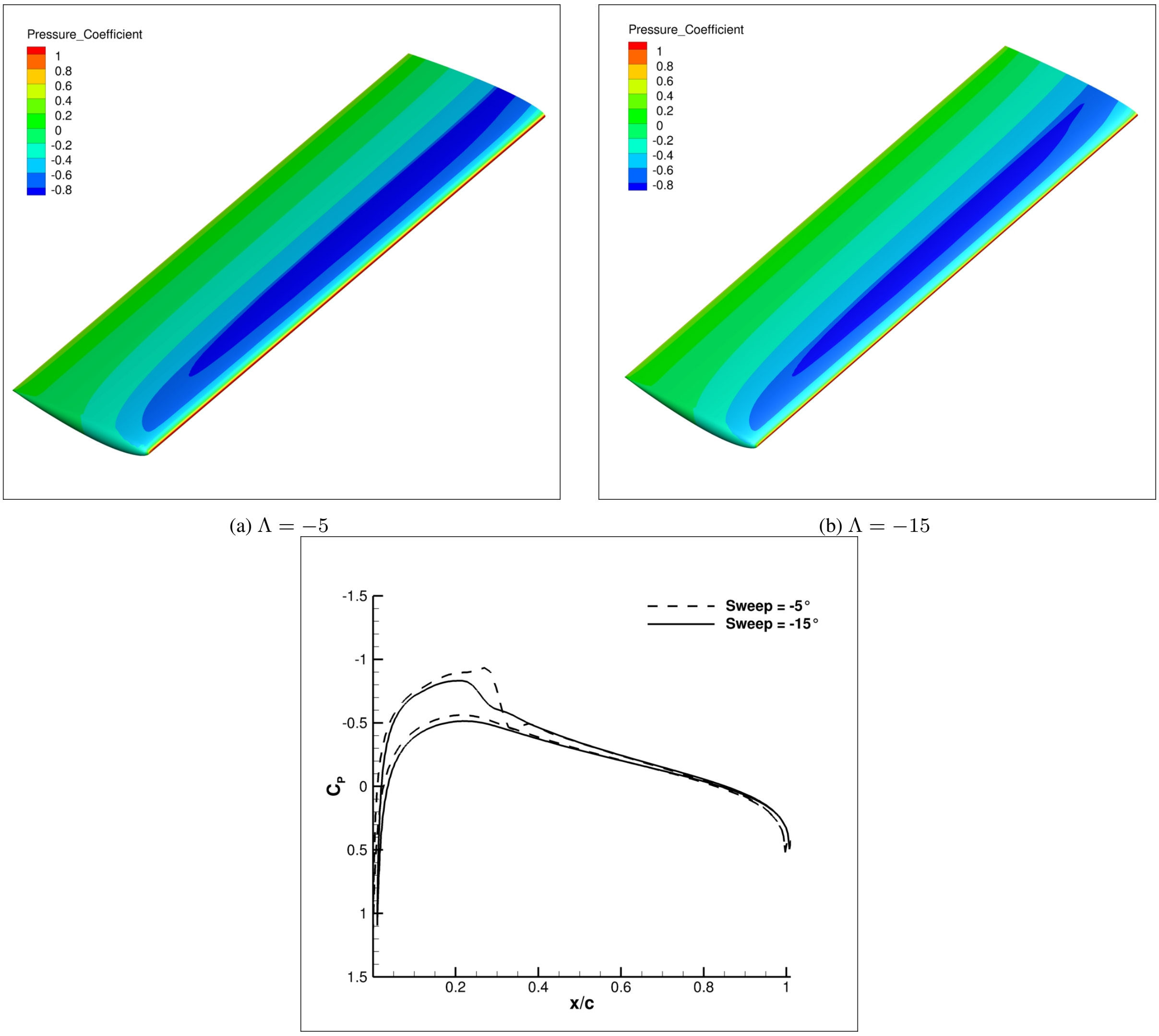
The work presented here is a state of the art of a much wider research activity which will lead to the realization of a coupled adjoint optimization tool, in an FSI interface which couples SU2



(a) Sweep angle

(b) Drag Coefficient

Figure 8: Iterations of the optimisation algorithm



(a) $\Lambda = -5$

(b) $\Lambda = -15$

(c) Comparison at 50% of span

Figure 9: C_P distribution on initial and final geometry

with Nastran. The preliminary results here illustrated prove the validity of the approach in the integration of a custom geometry morphing tool with a pre-existing adjoint aerodynamic solver, given the correctness of the gradient estimation and the consistency of the optimisation results presented.

7 ACKNOWLEDGEMENTS

This project has received funding from the Clean Sky 2 Joint Undertaking (JU) under grant agreement No 885052. The JU receives support from the European Union's Horizon 2020 research and innovation programme and the Clean Sky 2 JU members other than the Union.

Any dissemination of results must indicate that it reflects only the author's view and that the JU is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] Alexander I.J. Forrester and Andy J. Keane. "Recent advances in surrogate-based optimization". In: *Progress in Aerospace Sciences* 45.1 (2009), pp. 50–79. ISSN: 0376-0421. DOI: https://doi.org/10.1016/j.paerosci.2008.11.001. URL: https://www.sciencedirect.com/science/article/pii/S0376042108000766.

[2] Shamsheer Chauhan and Joaquim Martins. "Low-Fidelity Aerostructural Optimization of Aircraft Wings with a Simplified Wingbox Model Using OpenAeroStruct". In: Sept. 2018, pp. 418–431. ISBN: 978-3-319-97772-0. DOI: 10.1007/978-3-319-97773-7_38.

[3] Leifur Leifsson, Slawomir Koziel, and Stanislav Ogurtsov. "Low-Fidelity Model Mesh Density and the Performance of Variable-Resolution Shape Optimization Algorithms". In: *Procedia Computer Science* 9 (Dec. 2012), pp. 842–851. DOI: 10.1016/j.procs.2012.04.090.

[4] Sebastian Ruder. "An overview of gradient descent optimization algorithms". In: (Sept. 2016).

[5] Gerhard Venter. "Review of Optimization Techniques". In: Dec. 2010. ISBN: 9780470686652. DOI: 10.1002/9780470686652.eae495.

[6] Hao-Jun Michael Shi et al. *On the Numerical Performance of Derivative-Free Optimization Methods Based on Finite-Difference Approximations*. 2021. DOI: 10.48550/ARXIV.2102.09762. URL: https://arxiv.org/abs/2102.09762.

[7] Ted Belytschko. "Fluid-structure interaction". In: *Computers Structures* 12.4 (1980), pp. 459–469. ISSN: 0045-7949. DOI: https://doi.org/10.1016/0045-7949(80)90121-2. URL: https://www.sciencedirect.com/science/article/pii/0045794980901212.

[8] Yuri Bazilevs, Kenji Takizawa, and Tayfun Tezduyar. *Computational Fluid-Structure Interaction: Methods and Applications*. Feb. 2013. ISBN: 9780470978771. DOI: 10.1002/9781118483565.

[9] stefan keye stefan. "Fluid-Structure-Coupled Analysis of a Transport Aircraft and Comparison to Flight Data". In: June 2009. DOI: 10.2514/6.2009-4198.

[10] Thomas Economon et al. "SU2: An Open-Source Suite for Multiphysics Simulation and Design". In: *AIAA Journal* 54 (Dec. 2015), pp. 1–19. DOI: 10.2514/1.J053813.

[11] William Karush and Albert W. "Karush–Kuhn–Tucker conditions". In: (Jan. 2008), pp. 1–3.

[12] Gaetan Kenway, Graeme Kennedy, and Joaquim Martins. "Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Adjoint Derivative Computations". In: *AIAA Journal* 52 (May 2014), pp. 935–951. DOI: 10.2514/1.J052255.

[13] M. Sagebaum, T. Albring, and N. R. Gauger. "Expression templates for primal value taping in the reverse mode of algorithmic differentiation". In: *Optimization Methods and Software* 33.4-6 (2018), pp. 1207–1231. DOI: 10.1080/10556788.2018.1471140. eprint: https://doi.org/10.1080/10556788.2018.1471140. URL: https://doi.org/10.1080/10556788.2018.1471140.

[14] Tim Albring, Max Sagebaum, and Nicolas Gauger. "Development of a Consistent Discrete Adjoint Solver in an Evolving Aerodynamic Design Framework". In: June 2015. DOI: 10.2514/6.2015-3240.

[15] Joaquim Martins, Juan Alonso, and James Reuther. "A Coupled-Adjoint Sensitivity Analysis Method for High-Fidelity Aero-Structural Design: Special Issue on Multidisciplinary Design Optimization (Guest Editor: Natalia Alexandrov)". In: *Optimization and Engineering* 6 (Mar. 2005). DOI: 10.1023/B:OPTE.0000048536.47956.62.

[16] F. Beux and Alain Dervieux. "A Hierarchical approach for shape optimisation". In: *Engineering Computations* 11 (Dec. 1994), pp. 25–48. DOI: 10.1108/02644409410799191.

[17] James Reuther and Antony Jameson. "A comparison of design variables for control theory based airfoil optimization". In: (Aug. 1995).

[18] David Beazley. "Automated scientific software scripting with SWIG". In: *Future Generation Comp. Syst.* 19 (July 2003), pp. 599–609. DOI: 10.1016/S0167-739X(02)00171-1.

[19] Sebastian F. Walter and Lutz Lehmann. "Algorithmic differentiation in Python with AlgoPy". In: *Journal of Computational Science* 4.5 (2013), pp. 334–344. ISSN: 1877-7503. DOI: http://dx.doi.org/10.1016/j.jocs.2011.10.007. URL: http://www.sciencedirect.com/science/article/pii/S1877750311001013.

[20] Charles Margossian. "A Review of automatic differentiation and its efficient implementation". In: (Nov. 2018).

[21] Andrea Da Ronch Guangda Yang. "Wing twist optimisation using aerodynamic solvers of different fidelity". In: (Apr. 2018).