Aerodynamic shape optimization based on discrete adjoint and RBF

Luca Abergo *, Myles Morelli, Alberto Guardone

Department of Aerospace Science and Technology, Politecnico di Milano, Building B12, Via La Masa 34, Milano, 20156, Italy

ABSTRACT

This paper presents a new framework for Aerodynamic Shape Optimization (ASO) implemented inside the open-source software SU2. A parametrized surface is iteratively morphed to improve a desired aerodynamic coefficient. At each loop, the computational grid must be updated. The radial basis function (RBF) mesh deformation technique is introduced to extend the capability to explore the design space. RBF is implemented with some state-of-the-art data reduction systems to lower the computational cost. The Discrete Adjoint is adopted to compute the sensitivity in combination with Automatic Differentiation to calculate the required Jacobians. RBF is differentiated as well, resulting in a method-dependent surface sensitivity. The gradient-based algorithm "Sequential Least Squares Programming" drives the research of the minimum. The study demonstrates that the presented combination is more robust than an ASO, including linear elasticity analogy (ELA) as mesh deformation method. It can handle complex geometries and apply larger displacements, making possible the optimization of a wing-winglet configuration and a rotating wind turbine. Results are presented in two and three-dimensions for compressible and incompressible flows, showing a stronger reduction of the drag without affecting the lift.

© 2023 Elsevier Inc. All rights reserved.

1. Introduction

Automatic Aerodynamic Optimization based on Computational Fluid Dynamics (CFD) is a powerful tool able to improve the performance of an aero-body. It applies to a wide range of geometries, including airfoils, wings and rotors. There is an instinct highly sensitivity to geometric modifications. Even slight changes in the shape can have a significant influence on the final design's performance. A recent review of the state of the art regarding shape optimization [43] identifies 304 remarkable papers, which appeared in more than 120 conferences and journals, highlighting the increasing interest of the CFD community and the aerospace industry. As announced by the agenda of the Advisory Council for Aeronautics Research in Europe, the target is to reduce by 75% the CO_2 emissions per passenger kilometer and the noise by 65%. The airframe contribution is estimated at around 20%, besides automatic shape optimization (ASO) could play a key role.

Optimization is intended as the research of the best configuration where, especially concerning gradient-based ASO, the shape obtained is not a global minimum although clearly superior to the original one. In the early 1980s, the CFD community began to include in their codes the capability to perform a sensitivity analysis [44], which consists in the evaluation of the first derivatives of an aerodynamic quantity, which has a clear dependence both on geometry and the flow, with re-

---

spect to some design variables (DVs). The DVs are the control parameters of functions that describe the surface of the body. Originally, the gradients of interest were calculated by Finite Differences (FD). Firstly introduced by Reneaux and Thibert [34], FD requires no modification of the solver itself; however the computational cost scales up with the number of design parameters. Later, with the famous article of Jameson [19] (1988), in which the continuous adjoint method for aerodynamics was introduced, shape optimization became more attractive and affordable. For the first time, it is possible to compute the objective sensitivity with respect to DVs at the cost scaling with the number of objective functions. It requires a deep manipulation of the flow equations and the boundary conditions, only at the end the equations are discretized to pursue the numerical solution. The complexity of the mathematical formulation was partially overtaken by Shubin and Frank [41]. They proposed a discrete version of the adjoint method; in this case, the control theory is applied directly to the set of discrete flow equations. The major advantage is the possibility to easier introduce different turbulence models and deal with multiphysics problems such as fluid-structure interaction (FSI) or noise propagation [53] [52]. Moreover, there is the possibility to choose a different spatial discretization for the adjoint PDE however, the adjoint theory decay since consistency is lost therefore the convergence of the adjoint is no more guaranteed. In principle, the primal flow discretization does not dictate the choice since the solution of the adjoint system is consecutive to the direct simulation and only the converged flow state is shared.

1.1. Optimization problem and chain

In general, an optimization problem can mathematically be formulated as a minimization of an objective function J, usually it is an aerodynamic coefficient:

$$\min_{\alpha} \; J(U(\alpha), X(\alpha)) \tag{1}$$

$$\text{subject to} \;\; R(U(\alpha), X(\alpha)) = 0 \tag{2}$$

$$X(\alpha) = M(\alpha). \tag{3}$$

where U are the flow variables and α is the design variables' vector. The surface's body and the flow volume are discretized with a grid called $X(\alpha) = X_{vol}(X_{surf}(\alpha))$. The two optimization constraints Eq. (2) and Eq. (3) respectively requires that the flow's residual R is null, so U^* is a converged solution, and M explicits the method used to modify the grid. It is also possible to run an optimization with multiple objective functions combined and multiplied by a specific weight decided by the user. Typically, concerning a wing or an airfoil, it is required to minimize the drag and reduce the moment with respect to a specific axis simultaneously. The user can also impose some aero-constraints, such as requiring the conservation of the lift or the pitch moment inside a range. However, the computational cost and time required for the sensitivity computation grow linearly. Geometrical constraints can be introduced without significantly increasing the CPU time, such as a maximum thickness of the airfoil can be prescribed, or the wing volume must not decrease.

The exploration of the design space generating different shapes is entrusted to a gradient-based optimization algorithm based on the sensitivity computed with the adjoint. This choice has attached some intrinsic limitations, the gradient approach will likely converge to the nearest local minimum, where "nearest" is intended with respect to the starting point. This means that the software cannot drastically modify the initial object's shape. Concerning SU2, the Sequential Least Squares Programming (SLSQP) drives the optimization [21], the process ends when the Karush-Kuhn-Tucker (KKT) conditions are satisfied [22] or the maximum number of design loops is matched. The SLSQP treats the CFD solver as a black-box, it has no consciousness of the physic involved. It just requires as inputs the value of the objective function J, the DVs α_t, and the sensitivity $\frac{dJ}{d\alpha}$. It provides as output a new point of the design space that needs to be evaluated, which correspond to a new vector of DVs α_{t+1} and should be associated with a better value of J. Fig. 1 clearly shows what happens inside a single design loop, each brick is associated to a specific mathematical method selected for its characteristics. They all together will determine the general properties of the optimization chain.

The entire ASO chain presented in this work is coded in the open-source multiphysics solver SU2 [29], where the discrete adjoint code was implemented by Albring, Saugeman and Gauge [3]. The choice is driven by the possibility of using Automatic Differentiation to calculate the numerous Jacobians involved, taking advantage of a library named CodiPack [35]. The exact Jacobian is obtained with automatic differentiation, like manually differentiating all the terms. Often, the resulting system is poorly conditioned, some improvements can be obtained by introducing approximations like frozen viscosity or constant JST coefficients, but validity limit must be considered [15]. Software structure is conceptually straightforward, it combines automatic differentiation (AD) and discrete adjoint. However, it is significantly more complex from the coding point of view. Once the objective function J is computed, solving the proper RANS equations and surface sensitivity is obtained by the adjoint theory. To explore the entire design field, the software must have the capability to morph the body and update the mesh. The proposed ASO is Cad free, any type of mesh can be handled by the solver and the connectivity of the grid is never modified. To have the capability of morphing the body, first the wall boundaries need to be mathematically described. The relation between the body's surface and the DVs is provided by the shape parametrization method. In this work, the free-form deformation method (FFD) is applied. FFD directly parametrizes the nodes locations, meaning that it
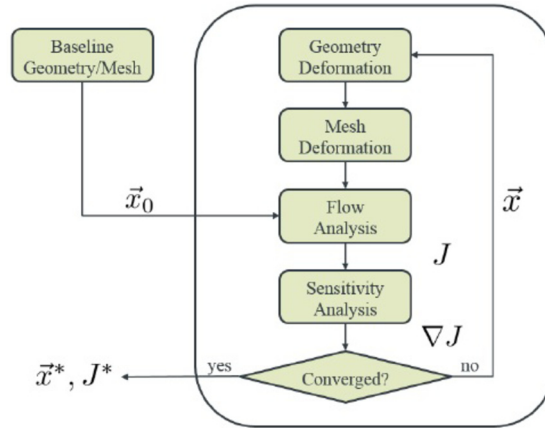
Fig. 1. Optimization design chain, from [3].

does not need the geometry's abstraction, although is able to handle only small or medium geometry changes. The basic idea is to embed the body and the mesh inside a box of flexible plastic; when the block is deformed, the surface grid is consistently shifted [36]. Mathematically, FFD consists in a mapping from $\mathbb{R}^3$ to $\mathbb{R}^3$ through a tensor product Bernstein polynomial. First, a local coordinate system is set inside the delimited volume, any grid point X inside the control box has coordinates (s, t, u), also called lattice coordinates:

$$X = X_0 + sS + tT + uU$$

The box is divided into l x m x k sub-control volumes, which coordinates of the vertex (i, j, k) in respect to a global reference system are contained in the matrix $P_{i,j,k}$. Some of these points or all of them can be used as design variable for the optimization process. The required displacement $\Delta P_{i,j,k}$ is obtained through the discrete adjoint and then the projection of the surface sensitivity into the design space. The movement of any point in local coordinates is calculated by:

$$x(s, t, u) + \Delta x(s, t, u) = \sum_{i=1}^{l} \sum_{j=1}^{m} \sum_{k=1}^{n} \left[B_{l-1}^{i-1}(s) B_{m-1}^{j-1}(t) B_{n-1}^{k-1}(u) \right] * \left[P_{i,j,k} + \Delta P_{i,j,k} \right] \tag{4}$$

where the Bernstein polynomial of degree l-1, also called the blending function here chosen for the clarity of the discussion, is determined as follows:

$$B_{l-1}^{i-1}(s) = \frac{(l-1)!}{(i-1)!(l-1)!} s^{i-1} (1-s)^{l-i} \tag{5}$$

SU2 uses Bezier's curves with global support and also B-Sline with local support are available but not used in this work [8]. The vertices' positions of the sub-control volumes are the design variables of the optimization problem.

The routine to update the fluid grid is differentiated inside the adjoint computation, accordingly it directly influences the sensitivity field. Besides, the possibility of the mesh deformation techniques of handling arbitrary geometry and applying significant displacements to the nodes strongly impacts the possibility of properly exploring the design space. A loss of quality properties in the output grid could cause the divergence of the following RANS simulation. Moreover, the computational cost, especially this step's physical memory consumption is not irrelevant. Many methodologies have been developed, mainly they can be split into two classes: physical analogy and interpolation technique [38]. The first one creates a similarity with a physical process that can be modeled using numerical methods, the spring analogy introduced by Batina [5] and linear elastic equations (ELA) first presented by Baker and Cavallo [4] belong to this category. This method was implemented to avoid that a node's crosses over an element face during the deformation process. The robustness is increased, however at the same time, it is computationally prohibitive and remains unable to handle large displacements [27,14]. In this case the grid is described as an elastic continuum, a natural mechanism prevents the generation of a negative volume. We could call this technique a FEM model whose target is to compute the node displacement considering the mesh element as a homogeneous and isotropic material. The constitutive law can be expressed as:

$$\varepsilon_{ij} = D_{ijks} \sigma_{ks} \tag{6}$$

Two independent variables are present in matrix (6): the Young's modulus E and Poisson's ratio ν. As they are assigned will strongly affect the quality of the output mesh. The standard approach called "inverse volume" is used for setting the Young modulus in this work. To ν is assigned a fixed value, instead E depends on the cell: $E = \frac{1}{V_i}$, where V_i is the element

volume. Instead, mesh deformation using interpolation does not need the connectivity information, therefore can be applied to any mesh that contains general polyhedral elements or hanging nodes [25]. The two main techniques are Inverse Distance Weighting (IDW) and Radial Basis Function (RBF).

The critical contribution of this work is the introduction of Radial Basis Function mesh deformation into a gradient-based discrete adjoint design optimization framework implemented in SU2 [26]. Radial Basis Function mesh deformation has been successfully used in the past as part of design optimization frameworks [18,1]. However, the significant computational cost of standard Radial Basis Function mesh deformation has meant alternative mesh deformation techniques have been preferable for three-dimensional problems. Moreover, Radial Basis Function mesh deformation is yet to be used alongside a discrete adjoint framework. This work harnesses state-of-the-art data reduction schemes to address the major weakness of radial basis function mesh deformation for large-scale three-dimensional problems. The robustness and high mesh quality properties of radial basis function mesh deformation allow the design optimization of topologically complex geometries, including transonic wings and winglets. Non-planar geometry optimization like fuselage-wing configuration has been already optimized with an adjoint-CFD method [13,20] but also using wind tunnel experiments [6], zero gradient method [12], or codes based on vortex lattice methods [48].

The new optimization chain is compared with an ASO of SU2 that uses a linear elastic analogy (ELA) as mesh deformation method. The comparison is made with the 2D and 3D wing benchmark optimization problems proposed to the CFD community by AIAA Aerodynamic Design Optimization Group (ADODG). Moreover, an incompressible test case of a Wind Turbine is performed with a rotating frame. Last, a 3D wing-winglet configuration obtained from the Onera M6 is optimized. Each test case is performed both with RBF and ELA for updating the grid, the results are compared also in terms of virtual memory consumption.

2. Sensitivity calculation

The research inside D_α of the best shape is driven by a gradient-based algorithm, therefore the sensitivity of the functional with respect to the design variables is required. As mentioned before in the ASO proposed the discrete adjoint is used to obtain it. The discrete adjoint equations if solved exactly provide a perfect gradient of an inexact cost function and the derivatives are fully consistent with complex-step gradients independent of the mesh size. This guarantees that the optimization process can converge. Meanwhile, the accuracy of the continuous adjoint increases as the mesh is refined, although there is a slight inconsistency between the discrete objective functions and the computed gradient. Indeed, the continuous approach yields to a discrete approximation of the gradient of the analytic objective function with respect to each of the DVs. It is not perfectly identical to the gradient of a discrete approximation of the objective function. Moreover, the turbulence model can be differentiated avoiding the frozen turbulence hypothesis which in case of strong separations can be limiting.

2.1. Discrete adjoint

The computation of $\frac{dJ}{d\alpha}$ in reality can be divided in three terms: flow, shape and grid sensitivity. The latter has a computational cost that strongly impact the optimization chain, especially concerning large unstructured meshes. In the mid 2000s it became clear that computing $\frac{\partial X}{\partial \alpha}$ was computationally restrictive, limiting in practice the cost independence of the adjoint from the length of the α vector. All costs scaling with the design parameters must be avoided. One elegant way to remove the volume mesh sensitivity from the discrete adjoint was proposed by Nielson and Park [28], it is called "double adjoint" or "adjoint mesh". For the first time, it allows the discrete adjoint computation to be really free from any cost related to the number of design variables involved. It must be highlighted that the surface sensitivity still depends on the grid deformation method selected even if the volume mesh sensitivity is eliminated. The total mesh can be considered as the sum of two different components, $X = X_{\mathrm{vol}} + X_{\mathrm{surf}} = M(\alpha)$ where $X_{\mathrm{vol}} = X_{\mathrm{vol}}(X_{\mathrm{surf}})$. It means that the volume grid depends explicitly on the wall mesh, that depends smoothly on the design variables. No assumptions on the structure of M are considered, except that is differentiable. The solution process of Eq. (2) can be transformed into a fixed point equation $U^{n+1} =: G(U^n)$. Therefore, the optimization problem can also take the form of:

$$
\begin{aligned}
\min_\alpha \quad & J(U(\alpha), X(\alpha)) \\
\text{subject to} \quad & U(\alpha) = G(U(\alpha), X(\alpha)) \\
& X(\alpha) = M(\alpha).
\end{aligned}
\tag{7}
$$

The easiest way to approach adjoint formulation is through Lagrangian multipliers. A unique expression associated to this problem can be created:

$$
L(\alpha, U, X, \Delta_f, \Lambda_g) = J(U, X, \alpha) + [G(U, X) - U]^T \Lambda_f + [M(\alpha) - X]^T \Lambda_g
\tag{8}
$$

where Λ_f represents the adjoint variables linked with the flow state, meanwhile Λ_g is a new set of adjoint variables multiplying the residual of the grid movement problem. Both vectors can assume arbitrary values. L has to be differentiated

with respect to the design variables using the chain rule. Taking advantage of the free value of the adjoint vectors, the terms $\frac{\partial U}{\partial \alpha}$ and $\frac{\partial X}{\partial \alpha}$ can be eliminated. This leads to two adjoint systems that need to be solved:

$$\Lambda_f = \frac{\partial J^T}{\partial U} + \frac{\partial G^T}{\partial U} \Lambda_f \tag{9}$$

$$\Lambda_g = \frac{\partial J^T}{\partial X} + \frac{\partial G^T}{\partial X} \Lambda_f \tag{10}$$

The total derivative of J is reduced to:

$$\frac{dL}{d\alpha}^T = \frac{dJ}{d\alpha}^T = \frac{dM(\alpha)^T}{d\alpha} \Lambda_g. \tag{11}$$

For the sake of clarity, if the linear elasticity approximation is selected as grid deformation strategy:

$$K X = X_{\text{surf}} \tag{12}$$

it provides the explicit relation between volume and surface mesh. Therefore, equations Eq. (8) and Eq. (10) can be specialized to this case:

$$L(\alpha, U, X, \Lambda_f, \Lambda_g) = J(U, X, \alpha) + [G(U, X) - U]^T \Lambda_f + [X_{\text{surf}} - K X]^T \Lambda_g \tag{13}$$

$$K^T \Lambda_g = -[\frac{\partial J}{\partial X} + \left(\frac{\partial R}{\partial X}\right)^T \Lambda_f] \tag{14}$$

The final form of the sensitivity vector turns out to be:

$$\frac{dL}{d\alpha} = \frac{\partial J}{\partial \alpha} + \Lambda_f^T \frac{\partial R}{\partial \alpha} - \Lambda_g^T \frac{\partial X_{surf}}{\partial \alpha} \tag{15}$$

The term $\Lambda_g^T \frac{\partial X_{\text{surf}}}{\partial \alpha}$ is cheap to be calculated, and it only requires an explicit inner product. Its size is linked to the length of the α vector and to the number of surface nodes and not to the entire grid extension as usually is with the standard discrete adjoint formulation.

2.2. Practical reverse automatic differentiation

The Jacobians involved in the formula (15) are obtained automatically differentiating different the SU2 code using the library CoDiPack. A detail description can be found in [10]. In practice, to obtain the evaluation chain, all the variables are substitute by a new datatype that stores at the same time the value and the elementary operations involved. All the simplest operations like sum or multiplications are hidden modified using the operator overloading method based on the expression template programming technique in C++. This way, the concatenation and the primal evaluation are performed simultaneously without touching the core of the code. The methods "RegisterOutput()" and "RegisterInput()" tape the independent variables and the outputs respect to which the partial derivatives are requested. The methods "StartRecording()" and "StopRecording()" delimits the routines that connect inputs and outputs and register all the operations involved. In the end, using "ComputeAdjoint()" the register sequence is traversed in reverse order to compute the gradients based on the chain rule.
For example, the required Jacobian $\frac{\partial X_{surf}}{\partial \alpha}$ is obtained differentiating the surface parametrization and surface deformation part of the code. Concerning the described optimization chain of this paper, the inputs variables are the positions of the FFD box's vertices, meanwhile the output are the position of each surface's cell center.

3. Radial basis functions

RBF is used to transfer the known displacement of a certain boundary to the fluid grid. This scheme generates an output high-quality mesh with the orthogonality well preserved near the morphing profile. It can be applied to any kind of mesh, both structural and unstructured hybrid mesh, since the connectivity is not required. Furthermore, the system of equations generated is linear, meaning that a large and well established amount of efficient schemes to solve it is available in literature. The size of the system is strongly linked to the mesh dimensions and if no expedient is coded the system matrix is strictly $N_s \times N_v$, where N_s are the surface nodes and N_v the volume points. Remarkable are the works of Rendall and Allen [33] and the study of Sheng [39]. They introduced the use of data reduction schemes, in particular the multilevel greedy-algorithm for the selection of surface nodes and the volume reduction method, both presented here.
The general theory of RBF, extensively displayed by Wendland [9] and Buhmann [49] is based on a series of functions whose value is linked to the distance between the selected position and a supporting point named "control point" [7]. The

displacement of a collection of nodes in the flow volume is described by an interpolation function $F(r)$, which is a sum of basis functions multiplied by a scalar which is unknown. The interpolation firstly introduced in [32] can be expressed as:

$$F(r) = \sum_{i=1}^{N} \alpha_i \phi(||r - r_i||) \tag{16}$$

where r_i is the radial basis center and the distance is intended as a Euclidean distance, meaning the spatial length between two nodes. To compute the weight coefficients, an exact recovery of the assigned function values at the control points has to be performed. The entire surface grid displacement is assumed to be known and collected in the vector ΔS for the three Cartesian direction:

$$\Delta S = \Delta X_s \hat{x} + \Delta Y_s \hat{y} + \Delta Z_s \hat{z} \tag{17}$$

In analogy also the weight coefficients are collected in a vector:

$$\alpha_x = \left[\alpha_{x,s_1}, \alpha_{x,s_2}, \dots, \alpha x, N_s \right]^T \tag{18}$$

The y and z coefficients are analogous. Following, the weights can be extracted by solving the linear system:

$$\Delta S = \Phi_{s,s} \alpha \tag{19}$$

where Φ is the universal basis matrix, it is generated with the radial basis function evaluated at each surface nodes, meaning that the matrix has size of N_s^2. The compact form of the universal basis function is expressed as:

$$\Phi_{s_j,s_i} = \phi||r_{s_i} - r_{s_j}|| \tag{20}$$

The next step is to compute the volume base matrix $\Phi_{v,s}$ of size $N_v \times N_s$, where "v" indicates a volume point. Finally, the volume displacement can be interpolated multiplying the above-mentioned matrix with the weights previously computed:

$$\Delta V = \Phi_{v,s} \alpha \tag{21}$$

The behavior of the interpolation between points or outside the dataset (extrapolation) is linked to the kind of radial function selected. The most used for complex applications are the Wendland [9] compact support functions.

Up to now, N_s surface nodes were used to generate Φ, resulting in a cost of solving the linear system proportional to N_s^3 and for updating the entire volume grid of $N_s \times N_v$. A process of selection of the surface nodes to obtain a subset set P^c with limited dimension is introduced. The algorithm is created to obtain a set of sample points according to the error generated by describing the entire surface displacement with a reduced RBF interpolation. The scheme start with a single point, then an extra surface node is added where the difference between the interpolated value and the exact one is maximum. This loop is repeated until the interpolation meets a selected tolerance ϵ multiplied with the maximum displacement ΔS. The selected nodes are collected in a vector X_c of increasing size N_c where the subscript "c" denotes the control point. The error vector E is computed using:

$$E = \Delta S - \Phi_{s,c} \alpha \tag{22}$$

where the matrix $\Phi_{s,c}$ is now of size $N_s \times N_c$ and where E is always of size N_s. Each time that a node is selected, the linear system to obtain the weights must be solved, so the CPU cost of the greedy algorithm is of the order of N_c^4. A multi-level subspace radial basis function interpolation, firstly introduced by Wang [47], is adopted. The object for the second level of interpolation is set equal to the error of the first step $E^{(0)}$. In a general form, it can be expressed as:

$$\Delta S_{l+1} = E^{(l)} \tag{23}$$

where the next step of the multi-level selection process is indicated by the subscript "$l + 1$". The residual of Eq. (19) at the second level can be expressed as:

$$\Delta S^{(1)} = \Delta S^{(0)} - \Phi W^{(1)} = \Delta S - \Phi(\alpha^{(0)} + \alpha^{(1)}) \tag{24}$$

the size of the displacement is strongly reduced $\Delta S_{l+1} << \Delta S_l$. The computational cost for the multilevel greedy algorithm is now of order of $N_l \times N_c^4$ instead of $(N_l \times N_c)^4$ for the single step. The overall process can be summarized:

$$\Delta S = \sum_{i=0}^{i=N_l-1} \Delta S^{(i)} = \sum_{i=0}^{i=N_l-1} \Phi_{s,c}^{(i)} \alpha^{(i)}$$

$$\Delta V = \sum_{i=0}^{i=N_l-1} \Delta V^{(i)} = \sum_{i=0}^{i=N_l-1} \Phi_{v,c}^{(i)} \alpha^{(i)} \tag{25}$$
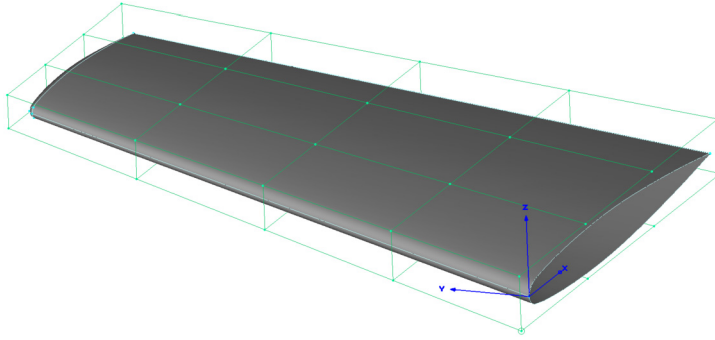
Fig. 2. Gradient verification FFD box.

Instead, to decrease N_v, one method was proposed by Xie and Liu [50]. They introduced a function which value is based on the distance from the closest wall:

$$\psi = \psi \left(\frac{d(r)}{D} \right).$$

(26)

where $d(r)$ is the space distance and D a support value imposed. We define the ratio between the two distances as ξ. The function decays in value when the distance increase and is zero outside the supported distance, so it is a compact support function. It can be expressed as:

$$\psi(\xi) = \begin{cases} (1 - \xi) & 0 \le \xi < 1 \\ 0 & \xi \ge 1 \end{cases}$$

(27)

The value of support distance D depends on the maximum surface displacement multiplied for a volume reduction factor $k > 1$ imposed by the used $D = k(\Delta S_l)^{\max}$. The interpolation Eq. (16) is modified in order to include the wall distance correction:

$$F(r) = \psi \left(\frac{d(r)}{D} \right) \sum_{i=1}^{N} \alpha_i \varphi(||r - r_i||)$$

(28)

When the volume points reduction method is combined with a multilevel greedy algorithm, as it is done in SU2, at the first level N_c is relative small meanwhile N_v remains high to absorb the large deformation. At each level, the support distance is updated with the new maximum displacement. Since, as explained previously, $\Delta S_{l+1} << \Delta S_l$ the range of influence of the basis function is strongly reduced, at the same time also the number of volume points is decreased $N_{v,l+1} << N_{v,l}$. The computational cost descends with the number of steps.

4. Verification

The verification of the adjoint solver is conducted, comparing the obtained gradients with a second order centered finite difference method:

$$f'(x) = \frac{f(x + h) - f(x - h)}{2h}$$

(29)

where h is the step size. An accurate verification of the SU2 adjoint solver without RBF has been already performed by T. Albring et al. [2] with 2D test cases. Since the current work is more focused on three-dimensional optimization, the verification is performed considering the Onera M6 wing at transonic flow conditions (M = 0.84, $\alpha = 3.06$ and Re = 14.6E6). The surface is parametrized with a single FFD box using Bézier functions with 4 degree of freedom in x direction, 5 for y and 2 for z. The control points at the root are kept fixed for a total of 32 design variables. See Fig. 2.

The convergence of the mesh is shown in Sec. 5.2, for the verification the denser hybrid grid with 5 million cells is used and $y+ < 1$ everywhere. RANS equations are solved with SA turbulence model [45], 4th order JST is used to calculate the fluxes [46]. It is required to have at least six order reduction of the relevant residuals and the finals lower than 10^{-13}. Concerning RBF, Wendland C2 functions are selected, maximum 2000 control points with a single level selection process, the final relative error has to be lower than 0.01%. A reasonable value for the step size for finite differences was found to be $5x10^{(-7)}$. Very good agreement is found between the sensitivities calculated with finite differences and with the discrete adjoint method with a relative l2-norm distance of 0.59% for the Cd gradient. See Figs. 3 and 4.
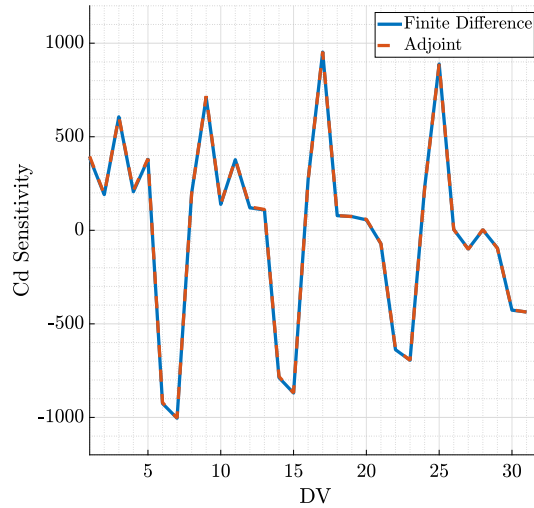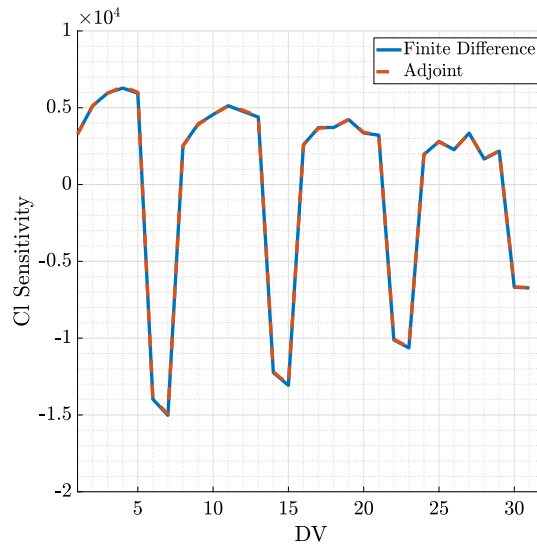
Fig. 3. Verification of drag sensitivity.

Fig. 4. Verification of lift sensitivity.

5. Results

In order to verify the capability of the presented ASO to optimize in different situations, some bi and three-dimensional test cases are presented. All the simulations are performed using JST as numerical scheme, with 0.5 and 0.02 as 2nd and 4th order artificial dissipation coefficients. A consistent approach is used for the adjoint simulation, also the viscosity model is differentiated. Each optimization is performed both with RBF and ELA as mesh deformation algorithm. Concerning ELA, the Young modulus E is always set with an inverse volume criteria. The geometry is parametrized with FFD using Bezier functions, each test case has a different number of degree of freedom. The displacements of the FFD vertices are kept in a range between 10^{-3} m and 10^{-2} m. The first benchmark test case is proposed by AIAA Aerodynamic Design Optimization Group (ADODG) so the results are comparable with the other tests present in the literature [24,40,17]. Each simulation is performed on 48 cores, distributed on 4 nodes, with a total RAM of 128 Gb. This strongly limits for the 3D simulations the resolution of the mesh.

5.1. NACA0012 drag minimization

The optimization of the classical airfoil NACA0012 has been performed, in addition to what suggested by ADODG the direct simulation here is a RANS instead of Euler. A hybrid mesh is generated in order to have high-quality mesh properties. To guarantee the convergence of the RANS simulation with SA as turbulence model, the $y+ < 1$ everywhere. The cells close

Table 1
Free stream conditions NACA0012.

Mach	0.76
AoA	$2°$
Re	$6.04E6$
Temperature	215.38 K
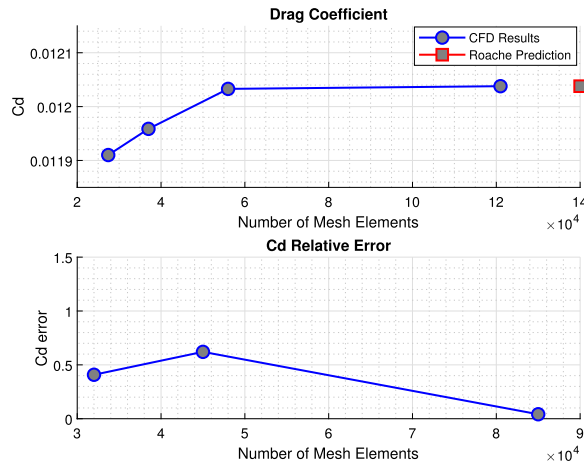
Fig. 5. NACA0012: C_D convergence.
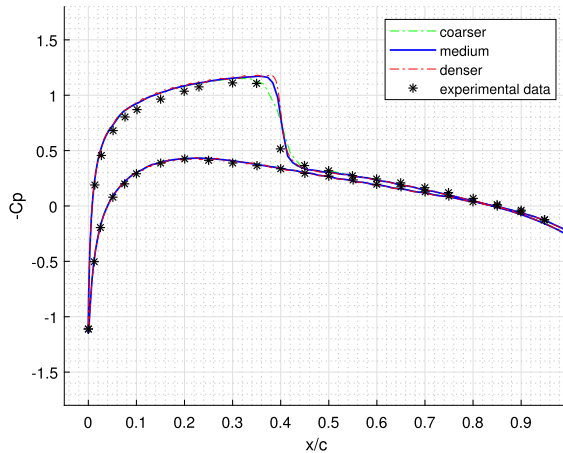
Fig. 6. NACA0012: C_P distribution.

to the airfoil are structured rectangles, the total height of this region has to contain the entire boundary layer. Meanwhile, the rest of the grid is unstructured. The free stream conditions selected are shown in Table 1.

Experimental data are provided by NASA [23], and the mesh convergence is performed according to the method proposed by Roache [11]. The value of the drag coefficient estimated for an infinitely dense mesh is shown in Fig. 5, while the C_P convergence is displayed in Fig. 6.

The goal of the optimization is to reduce the drag by modifying the shape without decreasing the maximum thickness of the airfoil. A constraint within the optimization is to maintain the C_l. Therefore, two adjoint simulations are performed, one for the drag and the second regarding the lift sensitivity. The angle of attack is kept fixed. The NACA0012 airfoil is inserted inside an FFD box and split in a certain number of vertical rectangles, which vertices are selected as design variables. See Fig. 7. The approximation of frozen viscosity is used in this test case. Since the output of the optimization is strongly related to the length of α vector, the first analysis shown in Fig. 8 is to find how many DVs are required to obtain the best performance. Concerning the case with sixteen design variables, the variation of the C_d with the number of deformations is shown in Fig. 9.
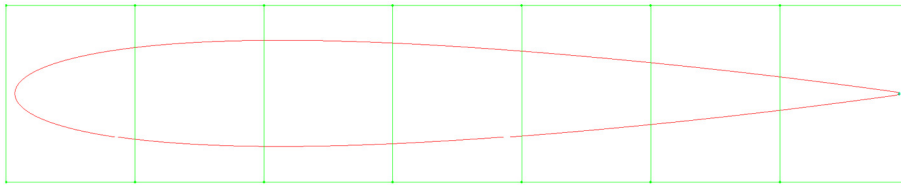
Fig. 7. NACA0012 best case FFD box.
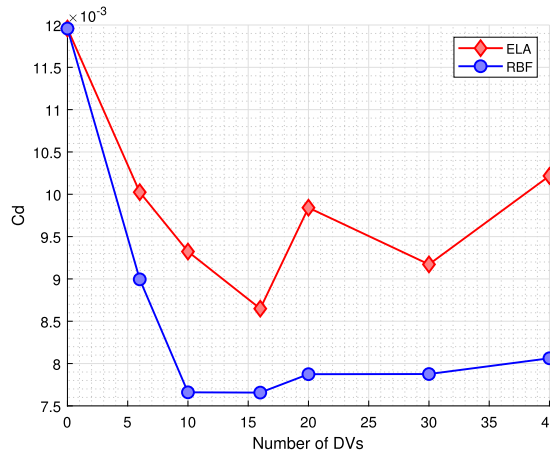
Fig. 8. NACA0012: C_d respect to DVs.
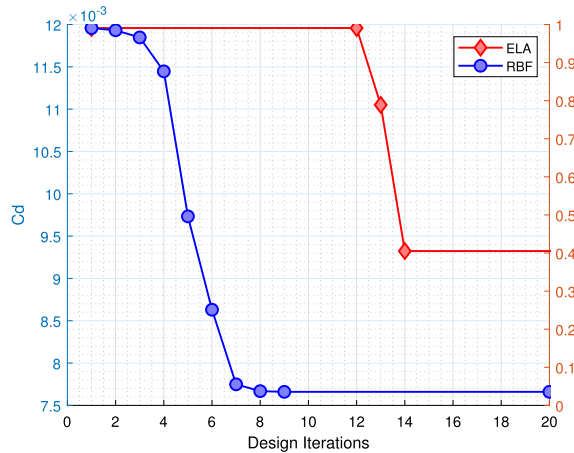

Fig. 9. NACA0012: C_d V_s design loops.

It is evident from Figs. 8 & 9 that the optimization process with RBF finds a deeper minimum of the C_d and it is also achieved quicker than with ELA. The C_d obtained using the RBF technique is 11.47% lower than the coefficient obtained with elastic analogy. Moreover, the reduction with respect to the original coefficient is 56%. In this specific case the large difference between the two methods is due to the computed surface sensitivities which are dissimilar in both sign and amplitude as shown by Fig. 10. Owing to this, the optimization algorithm reaches two distant minima. It is not possible to know, especially with a gradient based method, if a global minimum is reached or probably both are local ones. This highlight the complexity of proper explore the design space without getting stuck. Therefore, the shape of the profile undergoes to two disjointed movements: ELA makes thicker the upper part, RBF augment the surface area under the leading edge zone. Different sensitivity guide the optimization algorithm to obtain different pressure distribution as it can be seen in Fig. 11.

The shock positioned on the upper surface of the airfoil in the second case is moved more forward, this means that the Mach one bubble has a shorter radius. The shock is subsequently less intense and the consequent jump of pressure is reduced. See Figs. 12 and 13.
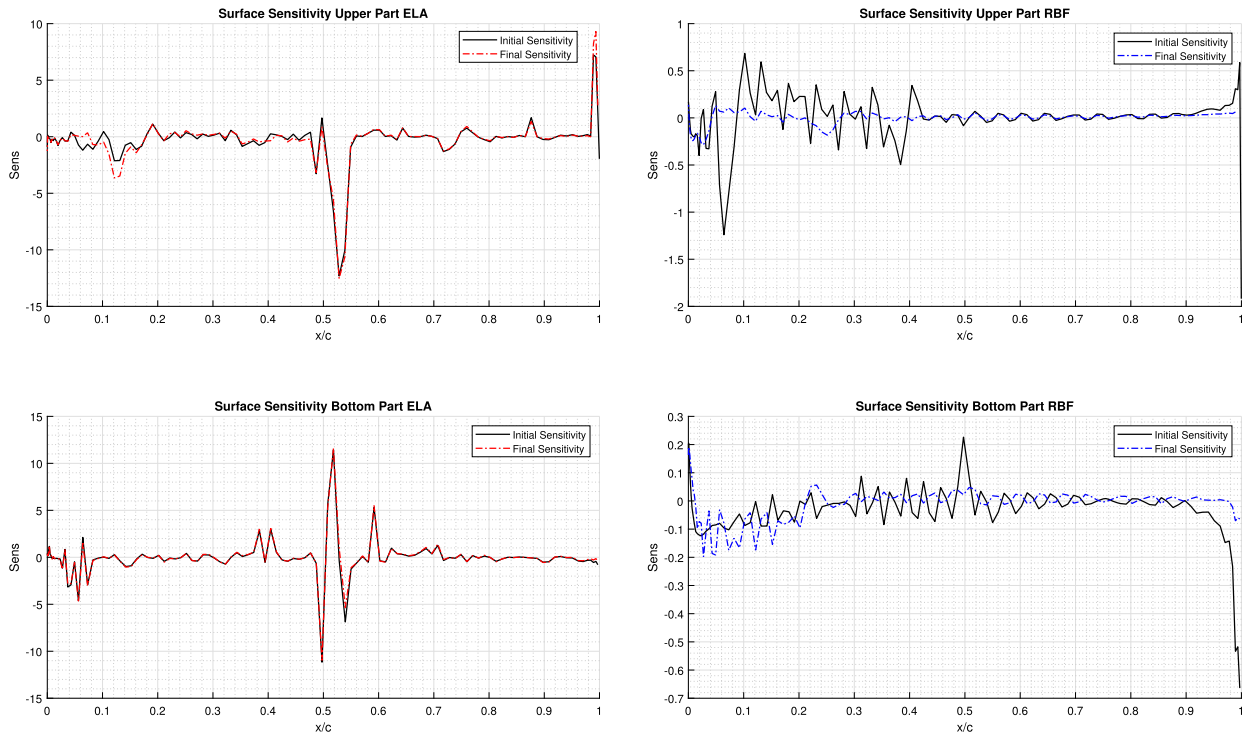
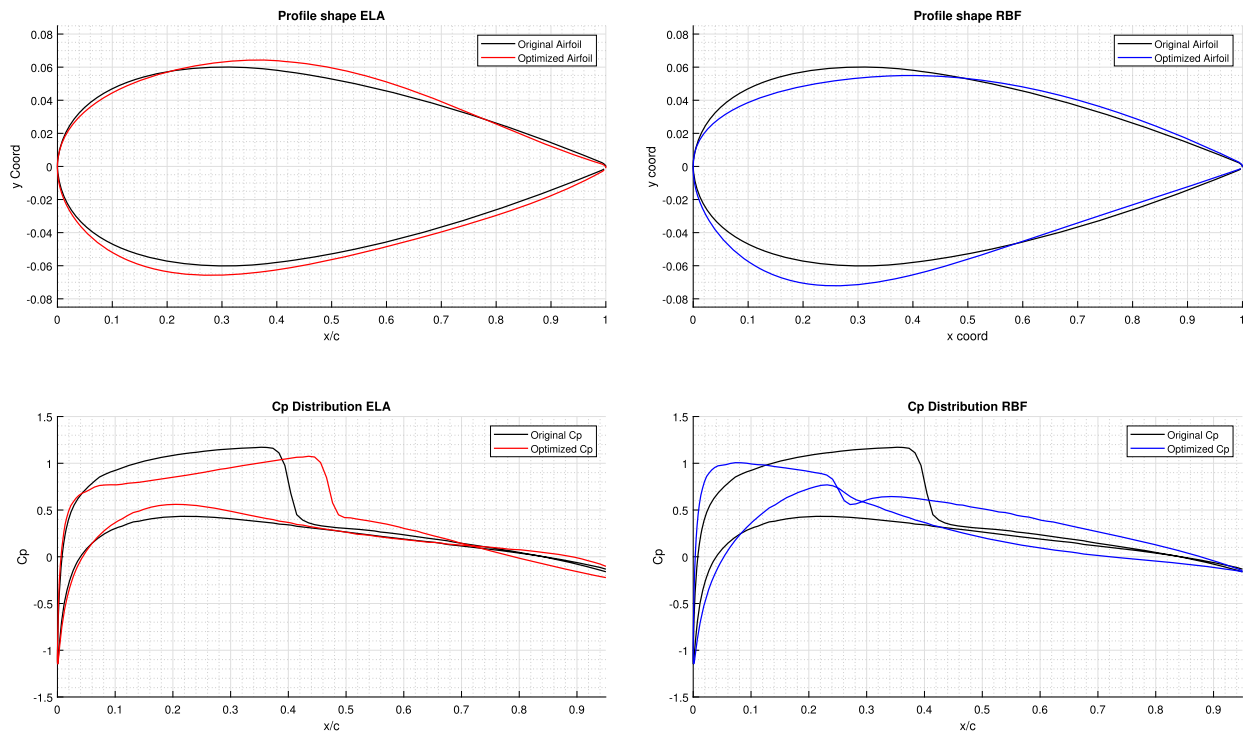**Fig. 10.** NACA0012: initial and final surface sensitivity.

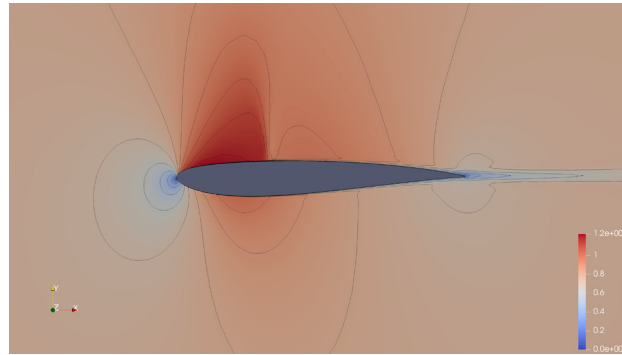**Fig. 11.** NACA0012: airfoil shape and C_p distribution.

Fig. 12. NACA0012: Final Mach with ELA. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)
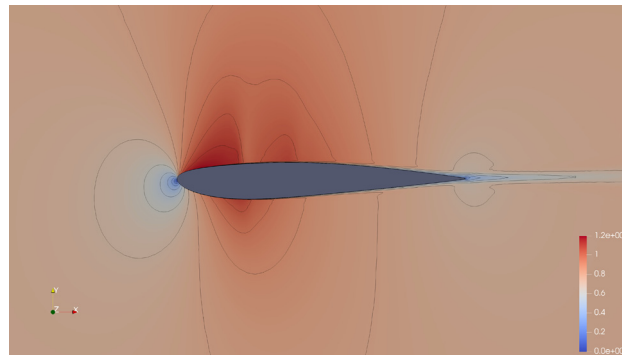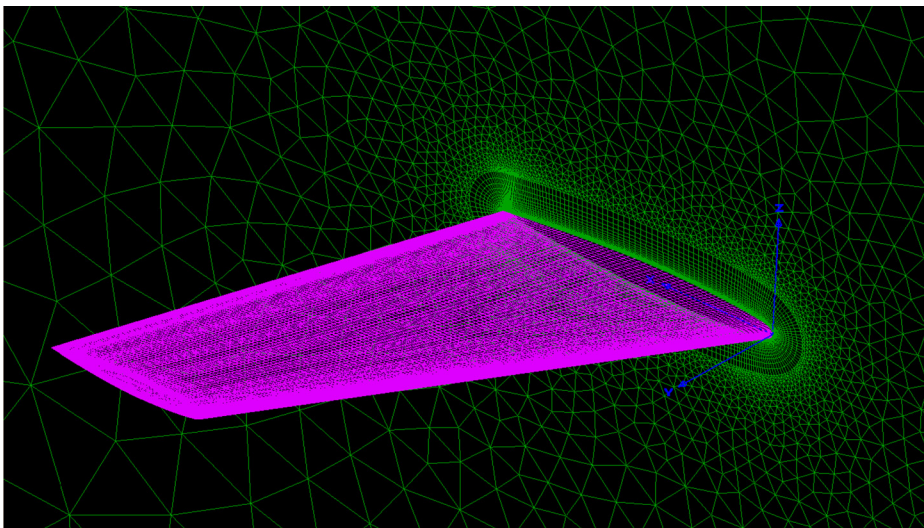

Fig. 13. NACA0012: final Mach with RBF.

Fig. 14. Onera M6: medium mesh.

5.2. Onera M6 drag minimization

The ONERA M6 can be described as a swept, semi-span wing with no twist. The symmetric ONERA D section is used as an airfoil. It is a typical test case for turbulence flow over a transonic wing, widely adopted for CFD validation. Experimental data for the comparison of the pressure distribution are provided in [37]. The flight conditions are chosen to deal with a strong shock on the upper part of the wing, collocated close to the 25% of the chord. See Fig. 14 and Table 2.

Firstly, mesh convergence has been performed, four hybrid meshes are generated. The smallest one, even if it does not perfectly predict the intensity of the shock and the pressure distribution, is used for optimization. The first three layers of the mesh have constant height than the growth ratio decrease from the coarse to the finer grid. Since the surface grid is

Table 2
Free stream conditions Onera M6.

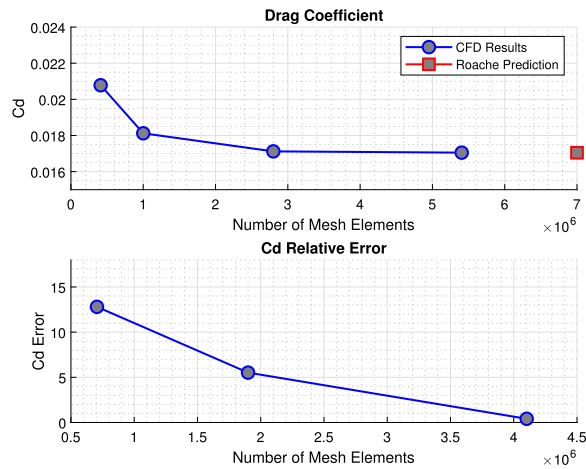| Mach	0.84
AoA	3.06
Re	14.6E6
Temperature	300 K

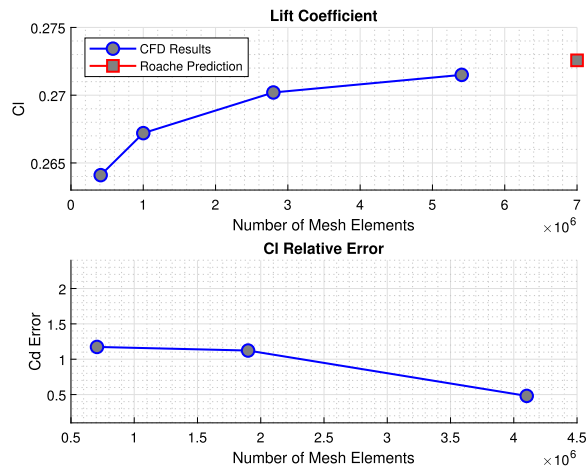**Fig. 15.** Onera M6: C_d convergence.

Fig. 16. Onera M6: C_l convergence.

almost everywhere structured, the number of points in the x, y direction are easily multiplied by a factor of 1.5 for the convergence. The following figures report the aerodynamic coefficients obtained and the relative error, in accordance to the bi-dimensional case, also the predicted value for an infinity dense mesh is marked. Moreover, the C_p distribution at two different stations is monitored. See Figs. 15, 16, 17 and 18.

The goal of the optimization is to reduce the drag without decreasing the lift and the volume occupied by the body. The angle of attack is free to change, the variation of the torque is only monitored. Only six geometrical constraints are imposed: the final maximum thickness at different span stations cannot be lower than the 75% of the initial value. Both ASO presented are used. Regarding RBF, the maximum number of control points selectable is the 10% of the surface's nodes, the volume reduction factor k is set at 5 and the Wendland $C2$ is selected as the interpolation function. This configuration makes RBF really robust, fast, and computationally cheap. Instead, considering ELA a final residual of $10^{(-10)}$ is required for the solution of the linear system and the stiffness of the cells is computed inversely with respect to their volume. The FFD box has 7 degrees of freedom in x direction, 9 for y and 2 in z direction for a total of 126 DVs. Results are reported in Figs. 19 and 20.
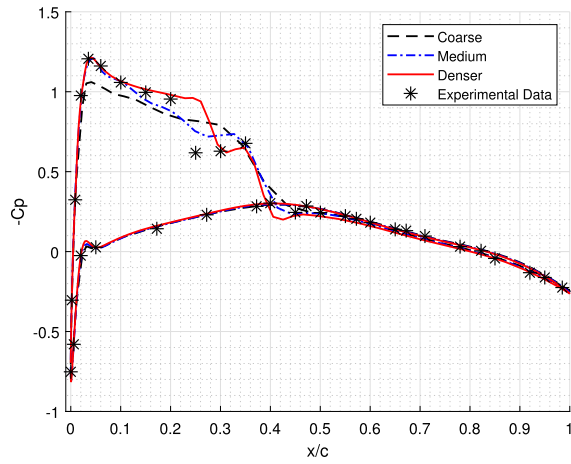
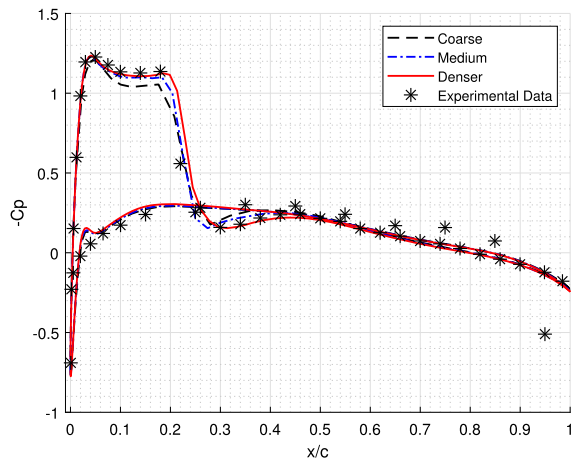**Fig. 17.** Onera M6: C_p convergence y = 80%b.

Fig. 18. Onera M6: C_p convergence y = 95%b.
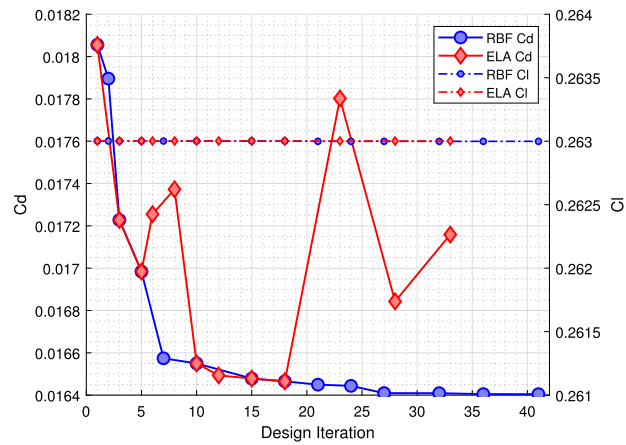

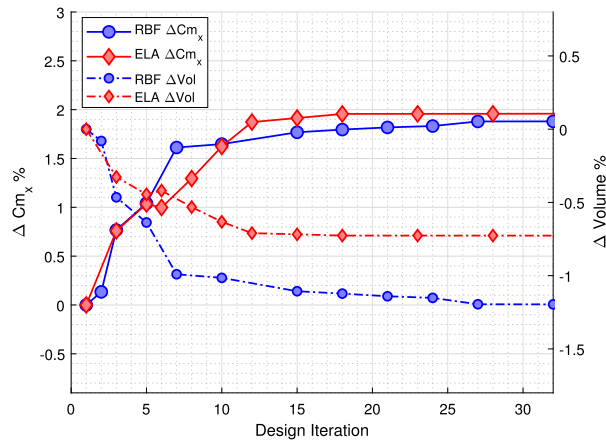**Fig. 19.** Onera M6: C_d variation.

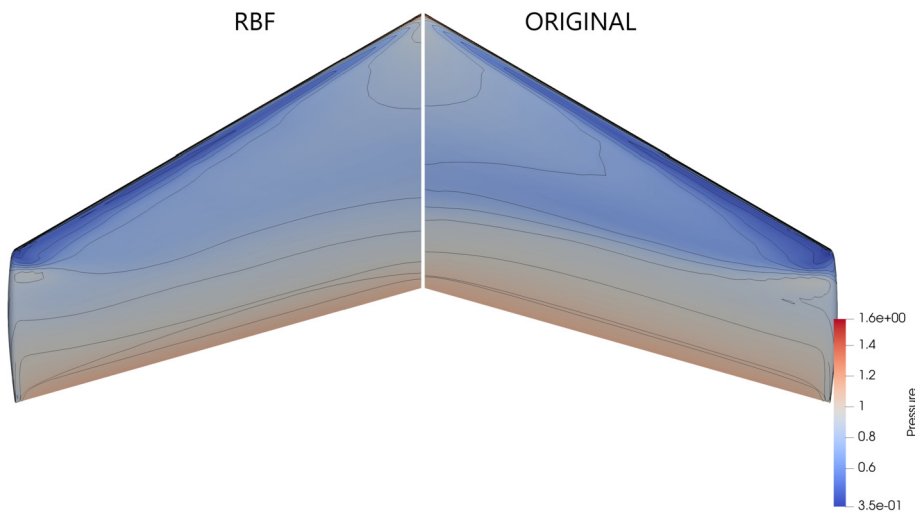**Fig. 20.** Onera M6: volume and torque.

Fig. 21. Onera M6: isopressure lines upper surface.

Fig. 19 clearly shows that both methods are able to conserve the lift and reduce the drag, however the optimization process seems to be way more robust using RBF with a more clear convergence. Regarding RBF, the overall drag reduction is around 9.15%, particularly the drag reduction between the last two loops is insignificant, underlying the achievement of a minimum. Instead, ELA obtains a slightly smaller reduction of the C_d around 8.57%. The evolution of the drag with respect to the deformation loops is oscillating, this behavior has been observed also by other researchers [51]. Both methods increase the torque of the 2% and slightly reduce the volume, even if a clear geometrical constraint is imposed. How the shape of the sections and thus the pressure distribution are modified after the optimization is shown in the next pics. The sections close to the root are more morphed, instead the tip of the wing is just more twisted. Especially close to the wing's tip, the peak of suction is decreased and this results in a less intense shock wave, accordingly a lower jump of pressure in Fig. 22. See also Figs. 21 and 23.

Moreover, Fig. 24 shows that the control points selected on the upper part of the wing, the distribution is strictly linked to the deformation wanted. In this case, the movement field was the result of the first optimization's loop, it is evident that the tip is not involved instead the leading edge and the central part of the wing are more affected by the deformation. The region of greater displacement should correspond to a zone of the surface with higher sensitive. It is confirmed by Fig. 25 where is shown the value of the adjoint variables for the last equation of the adjoint system and the magnitude of the three momentum equations combined.

5.3. Wing-winglet optimization

The optimization of a more complex geometry like a non-planar wing is presented in this section. Regarding SU2, the task has been already faced by prof. Cavallaro in the project NERONE [31]. In that work, the flow is modeled with Euler equations, neglecting the viscous effects. Inspired by that project, a new high fidelity ASO is presented in this section, where
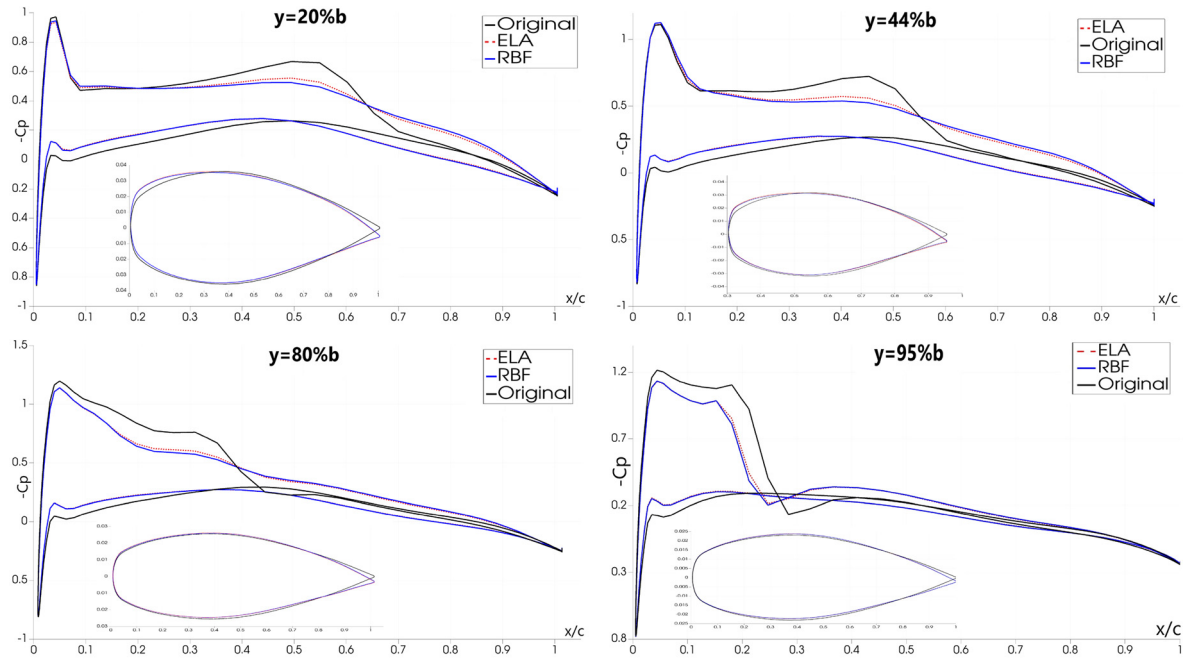
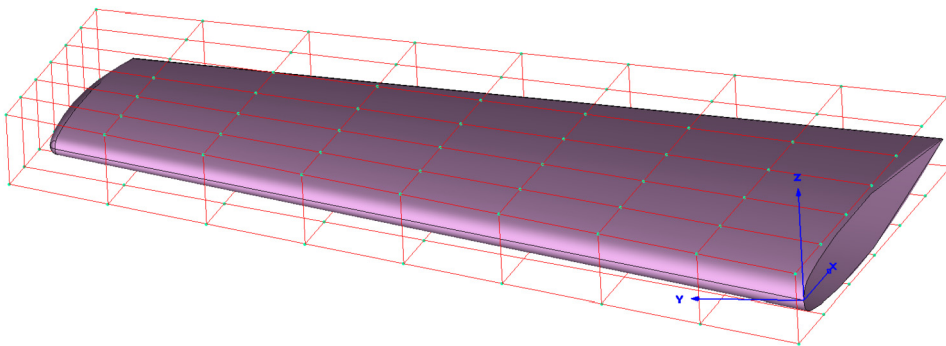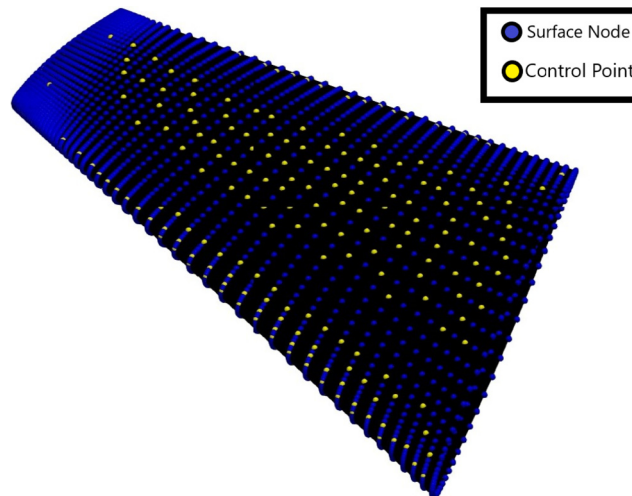**Fig. 22.** Onera M6: C_p optimization.

Fig. 23. Onera M6: FFD box.

Fig. 24. Onera M6: RBF Case 1 control points.
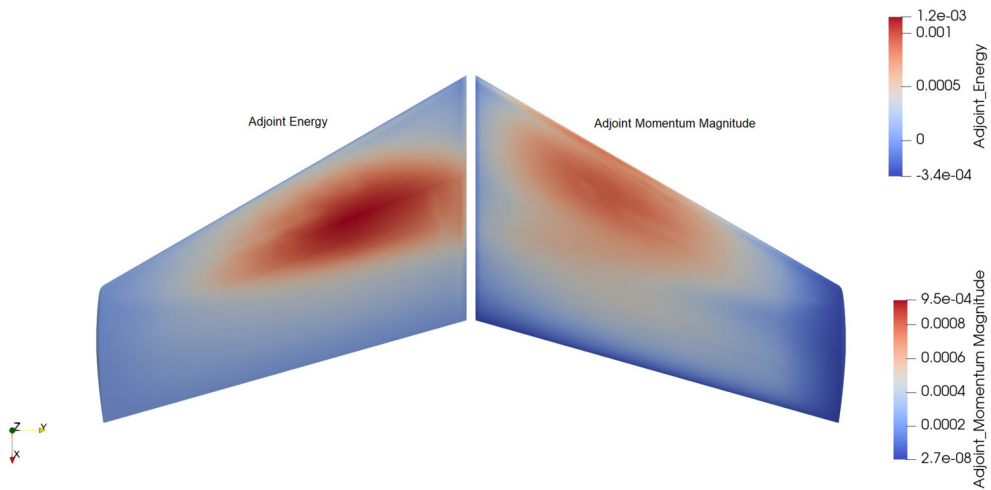
Fig. 25. Onera M6: adjoint variable.

Table 3
Free stream conditions Onera
M6 with winglet.

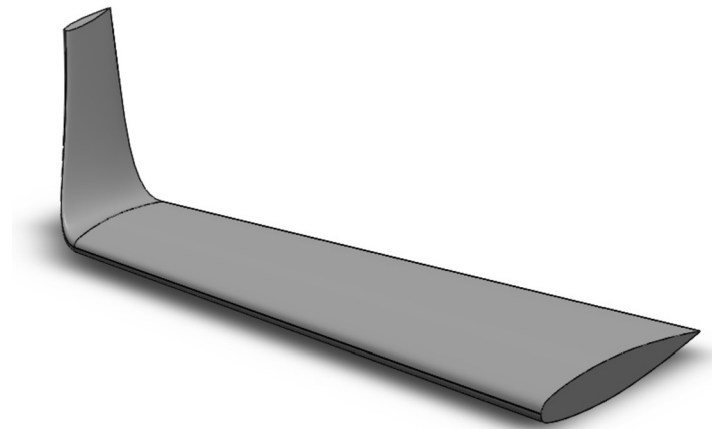| Mach	0.84
AoA	3.06
Re	14.6E6
Temperature	300 K
Target C_l	0.263

Fig. 26. Onera M6 with winglet attached CAD.

RANS equations with SA turbulence model are resolved and RBF is applied to deform the hybrid mesh. The Onera M6 wing is used, to which a winglet has been added. The winglet is drawn with a CAD software, generating a loft by a gradual contraction of the tip's airfoil until it halves the initial chord. The free stream conditions are the same used in the previous section, the target C_l is recomputed for the new geometry shown in Table 3. See also Fig. 26.

The convergence of the mesh is conducted at fixed C_l, since the surface grid is structured the number of points in the three directions is multiplied for 1.5. The coarser grid is used for the optimization to respect the virtual memory available. Therefore, the optimization starts from an inaccurate value of the drag, which is still highly mesh dependent. The results should be considered as a demonstration of the capability to improve the aero-performance of a non-planar wing, and not as a benchmark to match. See Fig. 27.

Regarding the optimization, the drag is selected as the objective function, the lift and the volume must not decrease. The angle of attack is free to change, the y-momentum is just monitored, while the root shape is fixed. The area of five sections on the wing and three on the winglet cannot become lower than the 75% of the initial value. A hybrid mesh of 0.7 million cells is generated, with the first two layers of constant height proving a starting $y+ < 0.9$. A single level RBF is used with
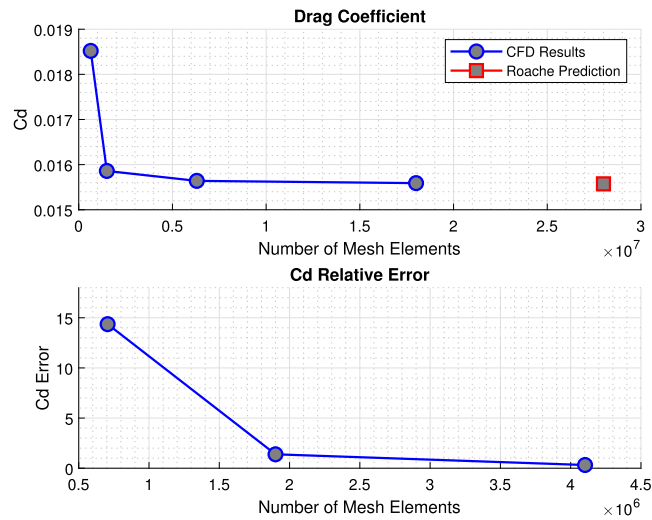
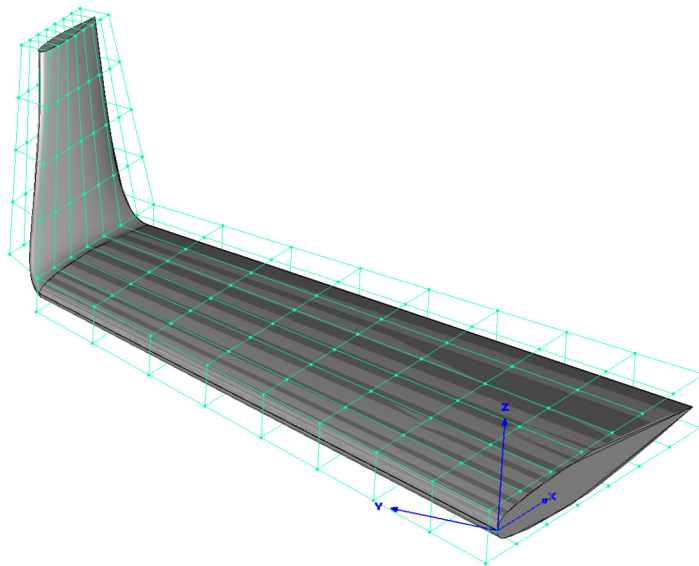**Fig. 27.** Onera M6 with winglet mesh convergence.

Fig. 28. Onera winglet: FFD box.

Wendland C0 as the basis function and maximum 10% of the surface nodes selectable as control points. This configuration provides for all the design loops a maximum error of the displacement lower than 0.5%. The optimization process is split in two stages. First, two FFD boxes are used, one containing the wing with 136 DVs (8x9x2) that can translate only in z-direction and a second around the winglet with 72 DVs (8x2x5) free to shift in y,x direction. The control points in common between the two boxes cannot translate and are not counted as Dvs. This way, the issue related to the continuity of two adjacent FFD boxes is avoided, it is not part of this work to tackle the multi-boxes problem. The sensitivity computed for wing DVs is one order of magnitude higher, accordingly the nodes of the horizontal part of the body are more shifted than the rest of the wing. When the drag reduction starts to be too low, the ASO restarts from the last output grid, this time with only FFD on the winglet. The number of DVs is increased to 160, this way the winglet is proper morphed and the difference is visible also with the naked eye. See Figs. 28, 29 and 30.

As can be seen in the previous figure, the drag is reduced from 185.34 counts to 163.43 which correspond to a decrease of 11.82%. The volume is perfectly preserved, instead the y-momentum remarkably increases from -0.112 to -0.148. The shape of the section and the C_p distribution are reported for four positions, two related to the winglet and two to the horizontal part. The optimization has also been tested with an elastic analogy method for the mesh deformation. The orthogonality of the mesh after the first design loop is chosen as indicator of the grid quality. Concerning ELA, if the stiffness is set with an inverse volume criteria, already the second direct simulation on the morphed grid diverges. See Table 4 and Fig. 31.
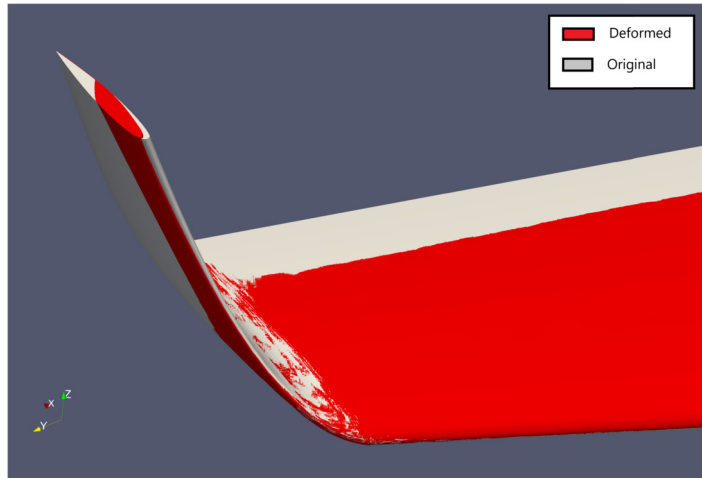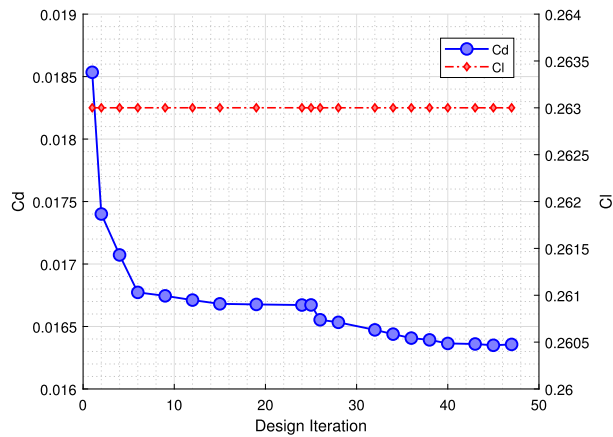
Fig. 29. Onera M6: winglet shape.

Fig. 30. Onera M6: winglet C_d optimization.

Table 4
Mesh orthogonality.

Method	N_c	N_v	Min Orthogonality
RBF C0	1100	392619	6.32
ELA inv volume	12676	645590	-43.92

5.4. NREL phase VI wind turbine

To further demonstrate the capability of the proposed methodology to optimize complex geometry, the NREL Phase VI Wind Turbine is selected. It is a two blades turbine generated with an S809 airfoil section and a radius of 5.029 m. It rotates at $\omega_z = 72$ RPM with a freestream wind of 7 m/s. This test case has been widely been adopted for unsteady aero CFD since experimental data are available [30,42]. A steady rotation is imposed to the turbine, therefore it is convenient to change the reference system of the RANS equations with a rotating frame in solidarity with the blades. Accordingly, the problem can be treated as steady, avoiding a very expensive unsteady optimization. Details can be found in [16], where also the grid convergence study has been already conducted for SU2. In this work a coarser mesh of 1.2 million of elements is used since the target is just to compare the two ASO chains. See Fig. 32.

The entire blade is immersed in a single FFD box and optimized with 64 Dvs with one degree of freedom each, also 100 and 150 DVs have been tested but performed worst. It is required to maximize the couple that produce the rotating movement M_z, also the in plane force T and the blending moment M_y are monitored. The maximum thickness of 8 sections per blade, equally distributed, cannot decrease more than 20% in respect to the starting value. ELA and RBF are configured as in the other three-dimensional test cases. Both cases presented a very oscillating path inside the D_α. Concerning the old methodology the minimum is found after 8 loops, instead the new one requires four optimization cycles. It is evident that
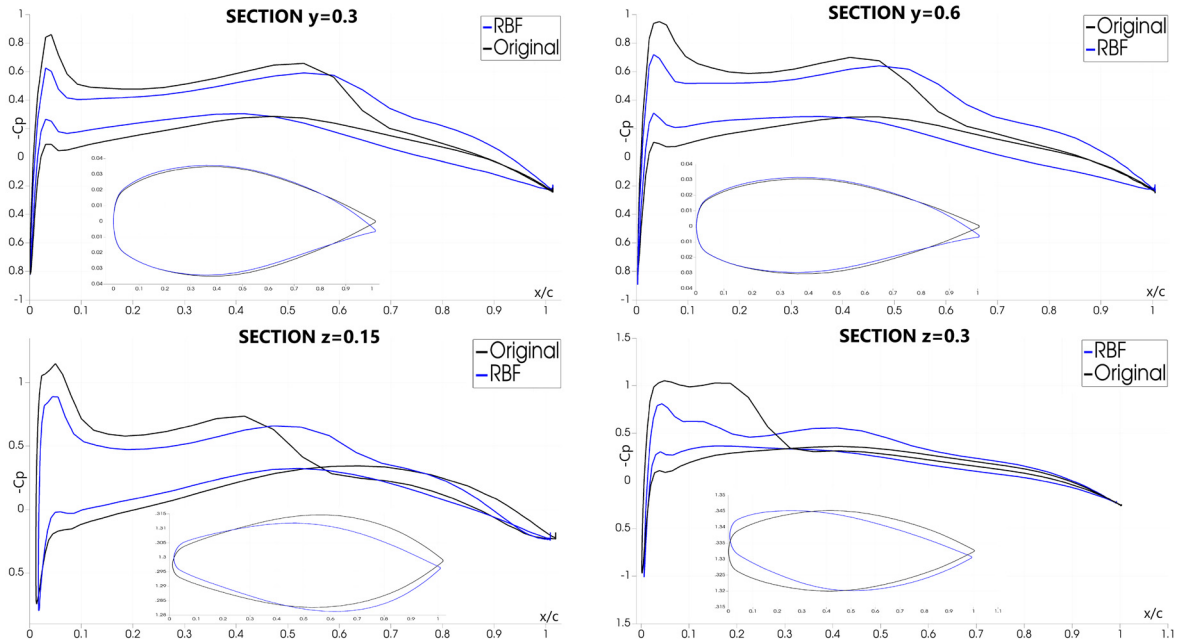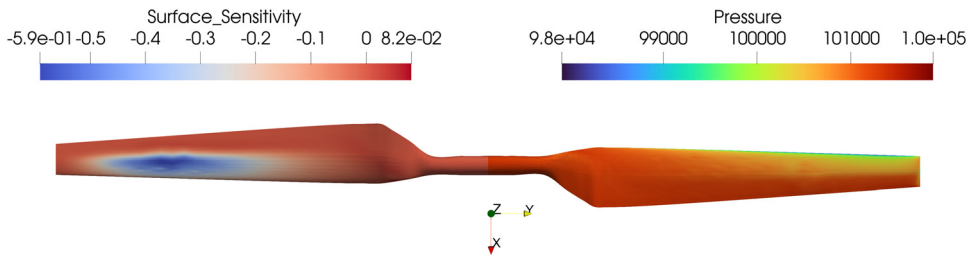
Fig. 31. Onera winglet: C_p optimization.

Fig. 32. NREL: initial pressure and sensitivity distribution.

Table 5
NREL: optimization results.

	RBF	ELA
M_z	+2.7%	+1.54%
T	+2.9%	+1.7%
M_y	+4.05%	3.08%

RBF provides a greater improvement of the objective function at the cost of a more severe blending moment. The variation of the force and momentum is reported in the next table. The change of shape is almost totally concentrated closer to the tip, the morphed section and the pressure distribution are shown for $r/R = 0.9$. See Table 5 and Figs. 33–35.

5.5. Mesh deformation performance

The planar Onera M6 test case has been used to compare the wall time and the ram consumption of the two mesh deformation considered in this text. The grid considered is hybrid, with 7319 surface nodes and 446685 cells. The performances of RBF are measured considering different settings: interpolation function, k, number of levels, amount of control points. The mesh deformation processes are executed on a single core of an AMD EPYC^{TM} of 2.4 GHz clock. The latter are imposed as a percentage of the total surface points. Table 6 reports the combination of the parameters tested and the results obtained.

Three outputs are of our interest: the maximum virtual memory allocated, the wall time excluded the deallocation of the data and the error due to the surface interpolation. Regarding ELA, as said in the previous section, the stiffness of the cells is set with an inverse volume logical. The final linear system has to be solved with a final residual of $10^{(-10)}$ in maximum
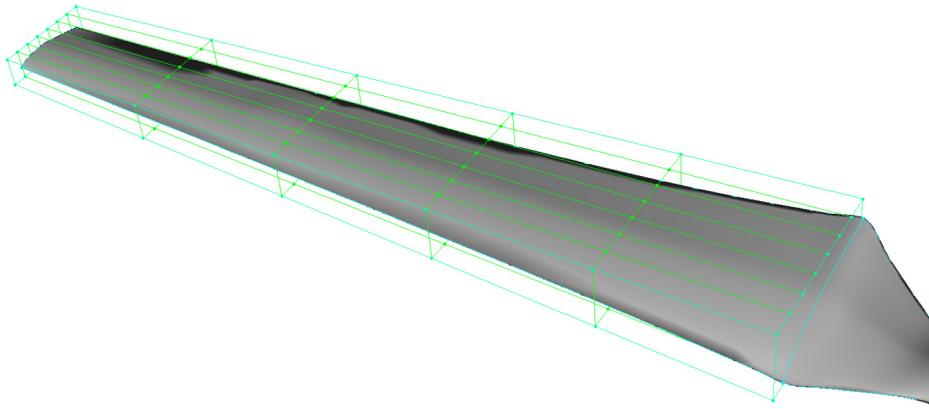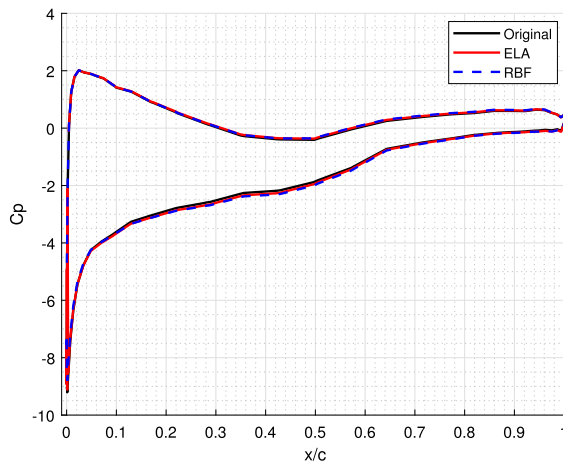
Fig. 33. NREL: FFD box.

Fig. 34. NREL: C_p distribution optimized at r/R=0.9.
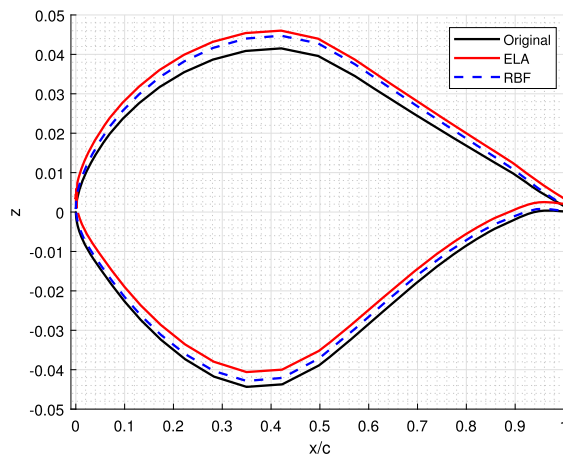
Fig. 35. NREL: section shape optimized at r/R=0.9.

800 iterations which are largely sufficient, this is the classical setting for the linear elasticity analogy proposed in the SU2 tutorials. The results are reported in Table 7.

The comparison between ELA and the RBF number one, which is used for the optimization, shows remarkable results. The RAM usage is almost six times lower and the time employed is one quarter. Considering only RBF, in order to have a very low interpolation error maintaining excellent performance, the comparison shows that is better to increase the order

Table 6
RBF parameters.

N.	Wendland	Control Points	K	N. Levels
0	C0	5%	5	one
1	C0	10%	5	one
2	C0	15%	5	one
3	C0	10%	10	one
4	C0	10%	5	two
5	C2	10%	5	one

Table 7
Performance results.

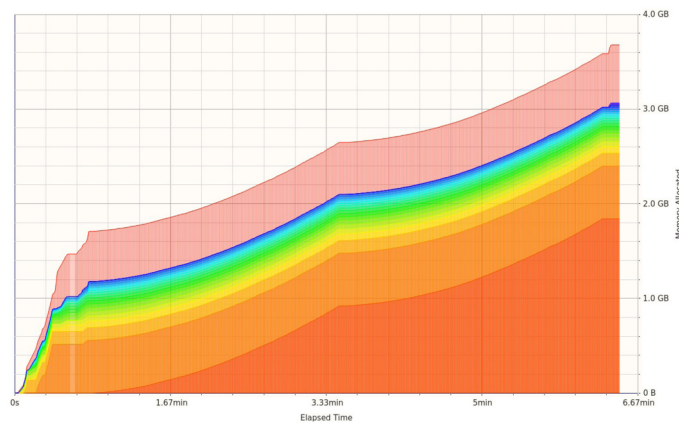| N.	RAM (G_b)	Interpolation Error	Cpu Time (min)
RBF.0	1.94	2.18%	1.81
RBF.1	2.55	1.07%	3.33
RBF.2	5.41	0.51%	7.89
RBF.3	2.55	0.98%	3.4
RBF.4	3.68	0.47%	6.46
RBF.5	2.74	0.15%	6.53
ELA	14.8	0%	13.66

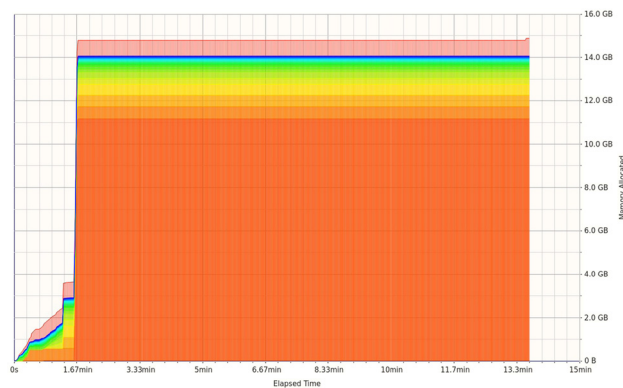**Fig. 36.** Onera M6: performance case 4 Rbf.

Fig. 37. Onera M6: performance Ela.

of the interpolation function instead of the number of levels of the greedy algorithm. The maximum number of control points selectable should be the 10% of surface nodes. The following figures show the allocation of the data in the virtual memory with respect to the cpu time. It can be noticed that the behavior of RBF and ELA are completely different, RBF is progressive, instead ELA quickly allocate all necessary information, then the ram consumption remains constant until the linear system is solved. For sake of clarity the legend is omitted in the next figures, each color correspond to the memory allocated by a specific C++ class, this information is irrelevant for the discussed comparison. See Figs. 36 and 37.
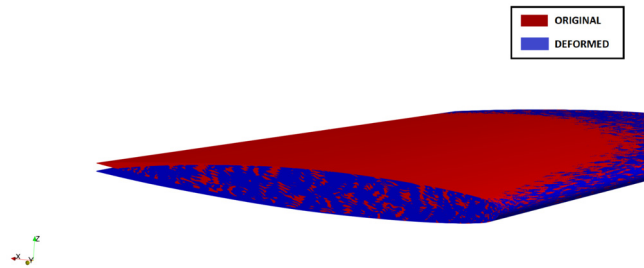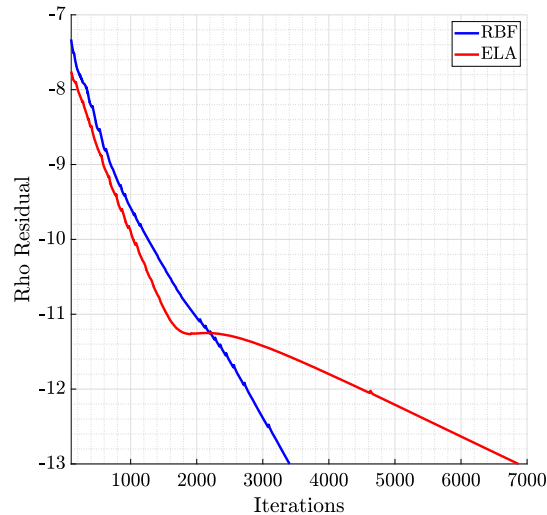
Fig. 38. Onera M6: mesh deformation.

Fig. 39. Onera M6: convergence history.

Moreover, the same deformation is applied to the OneraM6 grid of the verification Sec. 4. The control point closest to the tip and the trailing edge is shifted in order to obtain a maximum 1 cm displacement of the surface, which is the upper bound of the allowed range during the optimization process. The deformation is performed with ELA and RBF, the latter perfectly preserves the minimum orthogonality with value 11.64. Instead, with ELA the minimum ortho becomes 6.86. To demonstrate the impact on convergence, two RANS simulations are performed with SA and JST, the CFL is free to locally change. As linear solver, FGMRES is used with ILU preconditioner and 15 iterations. It is required to obtain a final residual of the density of 10^{-13}. ELA converges more slowly, the bad impact on the convergence due to the mesh deformation method suggests that the range of displacement should be limited with ELA. This can impact the capability of exploring the design space. See Figs. 38 and 39.

6. Conclusion

This works shows the potential of an ASO, implemented in SU2, by the combination of the double discrete adjoint for computing the sensitivity and the Radial Basis Function method for morph the grid. RBF is differentiated taking advantage of the Automatic Differentiation technique. Parametrizing the body with FFD, the ASO results completely cad-free, can handle any type of mesh and does not change its connectivity. The data-reduction schemes present in RBF makes it computationally affordable while maintaining a high quality mesh. This characteristic enlarges the possibility to properly explore the design space and handle more complex geometry. The entire ASO results more robust than the one implemented with ELA. Moreover, this work underlines the dependency of the surface sensitivity on the mesh deformation method selected. This can drive the SLSQP, which is a gradient based optimization algorithms, to a completely new stationary point.

CRediT authorship contribution statement

The corresponding Author Luca Abergo and the second author Myles Morelli developed and implemented in the software RBF and took care of the verification and the simulations reported in the paper that they wrote and revised. The Full Professor Alberto Guardone supervised the project and helped in the critical moments with his precious advice. Moreover, it provides the funding necessary to complete the project leading to this publication.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: The software used and developed inside this work, SU2, is totally open source and licensed under GNU Lesser General Public License v2.1.

Data availability

Data will be made available on request.

References

[1] L. Abergo, M. Morelli, A. Guardone, Aerodynamic optimization based on a discrete adjoint framework and radial basis function mesh deformation in SU2, in: AIAA Aviation 2021 Forum, 2021, p. 3034.

[2] T. Albring, M. Sagebaum, N.R. Gauger, A consistent and robust discrete adjoint solver for the SU2 framework—validation and application, in: New Results in Numerical and Experimental Fluid Mechanics X, Springer, 2016, pp. 77–86.

[3] T.A. Albring, M. Sagebaum, N.R. Gauger, Efficient aerodynamic design using the discrete adjoint method in SU2, in: 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2016, p. 3518.

[4] T. Baker, P. Cavallo, Dynamic adaptation for deforming tetrahedral meshes, in: 14th Computational Fluid Dynamics Conference, 1999, p. 3253.

[5] J.T. Batina, Unsteady Euler airfoil solutions using unstructured dynamic meshes, AIAA J. 28 (1990) 1381–1388, https://doi.org/10.2514/3.25229.

[6] S. Belferhat, S. Meftah, T. Yahiaoui, B. Imine, Aerodynamic optimization of a winglet design, in: EPJ Web of Conferences, EDP Sciences, 2013, p. 01010.

[7] M.E. Biancolini, Fast Radial Basis Functions for Engineering Applications, Springer, 2017.

[8] W. Böhm, G. Farin, J. Kahmann, A survey of curve and surface methods in cagd, Comput. Aided Geom. Des. 1 (1984) 1–60.

[9] M.D. Buhmann, Radial Basis Functions: Theory and Implementations, vol. 12, Cambridge University Press, 2003.

[10] O. Burghardt, N.R. Gauger, P. Gomes, R. Palacios, T. Kattmann, T.D. Economon, Coupled discrete adjoints for multiphysics in SU2, in: AIAA Aviation 2020 Forum, 2020, p. 3139.

[11] I.B. Celik, U. Ghia, P.J. Roache, C.J. Freitas, Procedure for estimation and reporting of uncertainty due to discretization in cfd applications, J. Fluids Eng. 130 (2008), https://doi.org/10.1115/1.2960953.

[12] U. Cella, D. Romano, Assessment of optimization algorithms for winglet design. EnginSoft-Newsletter Year 7nl, https://www.esteco.com/modefrontier/assessment-optimization-algorithms-winglet-design, 2010.

[13] S. Chen, Z. Lyu, G.K. Kenway, J.R. Martins, Aerodynamic shape optimization of common research model wing–body–tail configuration, J. Aircr. 53 (2016) 276–293.

[14] R.P. Dwight, Robust mesh deformation using the linear elasticity equations, in: Computational Fluid Dynamics 2006, Springer, 2009, pp. 401–406.

[15] R.P. Dwight, J. Brezillon, Effect of approximations of the discrete adjoint on gradient-based optimization, AIAA J. 44 (2006) 3022–3031.

[16] T.D. Economon, F. Palacios, J.J. Alonso, A viscous continuous adjoint approach for the design of rotating engineering applications, in: 21st AIAA Computational Fluid Dynamics Conference, 2013, p. 2580.

[17] X. He, J. Li, C.A. Mader, A. Yildirim, J.R. Martins, Robust aerodynamic shape optimization—from a circle to an airfoil, Aerosp. Sci. Technol. 87 (2019) 48–61, https://doi.org/10.1016/j.ast.2019.01.051.

[18] S. Jakobsson, O. Amoignon, Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization, Comput. Fluids 36 (2007) 1119–1136, https://doi.org/10.1016/j.compfluid.2006.11.002.

[19] A. Jameson, Aerodynamic design via control theory, J. Sci. Comput. 3 (1988) 233–260, https://doi.org/10.1007/BF01061285.

[20] P.W. Jansen, R.E. Perez, J.R. Martins, Aerostructural optimization of nonplanar lifting surfaces, J. Aircr. 47 (2010) 1490–1503.

[21] T.A. Johansen, T.I. Fossen, S.P. Berge, Constrained nonlinear control allocation with singularity avoidance using sequential quadratic programming, IEEE Trans. Control Syst. Technol. 12 (2004) 211–216, https://doi.org/10.1109/TCST.2003.821952.

[22] H.W. Kuhn, A.W. Tucker, Nonlinear programming, in: Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, Calif., 1951, pp. 481–492, https://projecteuclid.org/euclid.bsmsp/1200500249.

[23] C.L. Ladson, A.S. Hill, W. Johnson, Pressure distributions from high Reynolds number transonic tests of an naca 0012 airfoil in the langley 0.3-meter transonic cryogenic tunnel, https://ntrs.nasa.gov/citations/19880009181, 1987.

[24] D. Li, R. Hartmann, Adjoint-based airfoil optimization with discretization error control, Int. J. Numer. Methods Fluids 77 (2015) 1–17, https://doi.org/10.1002/fld.3971.

[25] E. Luke, E. Collins, E. Blades, A fast mesh deformation method using explicit interpolation, J. Comput. Phys. 231 (2012) 586–601, https://doi.org/10.1016/j.jcp.2011.09.021.

[26] M. Morelli, T. Bellosta, A. Guardone, Efficient radial basis function mesh deformation methods for aircraft icing, J. Comput. Appl. Math. (2021) 113492, https://doi.org/10.1016/j.cam.2021.113492.

[27] E.J. Nielsen, W.K. Anderson, Recent improvements in aerodynamic design optimization on unstructured meshes, AIAA J. 40 (2002) 1155–1163, https://doi.org/10.2514/2.1765.

[28] E.J. Nielsen, M.A. Park, Using an adjoint approach to eliminate mesh sensitivities in computational design, AIAA J. 44 (2006) 948–953, https://doi.org/10.2514/1.16052.

[29] F. Palacios, M.R. Colonno, A.C. Aranake, A. Campos, S.R. Copeland, T.D. Economon, A.K. Lonkar, T.W. Lukaczyk, T.W. Taylor, J.J. Alonso, Stanford university unstructured (SU2): an open-source integrated computational environment for multi-physics simulation and design, AIAA paper 287, 2013, https://doi.org/10.2514/6.2013-287, 2013.

[30] M. Potsdam, D. Mavriplis, Unstructured mesh cfd aerodynamic analysis of the nrel phase vi rotor, in: 47th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2009, p. 1221.

[31] L. Pustina, R. Cavallaro, G. Bernardini, Nerone: an open-source based tool for aerodynamic transonic optimization of nonplanar wings, Aerotec. Missili Spaz. 98 (2019) 85–104, https://doi.org/10.1007/s42496-019-00007-4.

[32] T.C. Rendall, C.B. Allen, Unified fluid–structure interpolation and mesh motion using radial basis functions, Int. J. Numer. Methods Eng. 74 (2008) 1519–1559, https://doi.org/10.1002/nme.2219.

[33] T.C. Rendall, C.B. Allen, Efficient mesh motion using radial basis functions with data reduction algorithms, J. Comput. Phys. 228 (2009) 6231–6249, https://doi.org/10.1016/j.jcp.2009.05.013.

[34] J. Reneaux, J.J. Thibert, The use of numerical optimization for airfoil design, in: 3rd Applied Aerodynamics Conference, Colorado Springs, 1985, p. 5026, https://arc.aiaa.org/doi/pdf/10.2514/6.1985-5026.

[35] M. Sagebaum, T. Albring, N.R. Gauger, High-performance derivative computations using codipack, ACM Trans. Math. Softw. 45 (2019) 1–26, https://doi.org/10.1145/3356900.

[36] J. Samareh, Aerodynamic shape optimization based on free-form deformation, in: 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2004, p. 4630.

[37] V. Schmitt, Pressure distributions on the onera m6-wing at transonic Mach numbers, experimental data base for computer program assessment, AGARD AR-138, https://www.sto.nato.int/publications/AGARD/AGARD-AR-138/AGARD-AR-138.pdf, 1979.

[38] M. Selim, R. Koomullil, et al., Mesh deformation approaches–a survey, J. Phys. Math. 7 (2016), https://doi.org/10.4172/2090-0902.1000181.

[39] C. Sheng, C.B. Allen, Efficient mesh deformation using radial basis functions on unstructured meshes, AIAA J. 51 (2013) 707–720, https://doi.org/10.2514/1.J052126.

[40] Y. Shi, C.A. Mader, S. He, G.L. Halila, J.R. Martins, Natural laminar-flow airfoil optimization design using a discrete adjoint approach, AIAA J. 58 (2020) 4702–4722, https://doi.org/10.2514/1.J058944.

[41] G.R. Shubin, Obtaining "cheap" optimization gradients from computational aerodynamics codes, Applied Mathematics and Statistics Technical Report AMS, Boeing Computer Services, https://ntrs.nasa.gov/api/citations/19920004715/downloads/19920004715.pdf?attachment=true, 1991.

[42] D. Simms, S. Schreck, M. Hand, L.J. Fingersh, NREL unsteady aerodynamics experiment in the NASA-Ames wind tunnel: a comparison of predictions to measurements, Technical Report, National Renewable Energy Lab, Golden, CO (US), 2001.

[43] S.N. Skinner, H. Zare-Behtash, State-of-the-art in aerodynamic shape optimisation methods, Appl. Soft Comput. 62 (2018) 933–962, https://doi.org/10.1016/j.asoc.2017.09.030.

[44] J. Sobieszczanski-Sobieski, The case for aerodynamic sensitivity analysis, in: H.M. Adelman, R.T. Haftka (Eds.), Sensitivity Analysis in Engineering, Feb 1987, pp. 77–96, compiled by H.M. Adelman and R.T. Haftka, NASA CP 2457, https://ntrs.nasa.gov/citations/19870009428.

[45] P. Spalart, S. Allmaras, A one-equation turbulence model for aerodynamic flows, in: 30th Aerospace Sciences Meeting and Exhibit, 1992, p. 439.

[46] R.C. Swanson, E. Turkel, On central-difference and upwind schemes, J. Comput. Phys. 101 (1992) 292–306, https://doi.org/10.1016/0021-9991(92)90007-L.

[47] G. Wang, H.H. Mian, Z.Y. Ye, J.D. Lee, Improved point selection method for hybrid-unstructured mesh deformation using radial basis functions, AIAA J. 53 (2015) 1016–1025, https://doi.org/10.2514/1.J053304.

[48] J. Weierman, J. Jacob, Winglet design and optimization for UAVs, in: 28th AIAA Applied Aerodynamics Conference, 2010, p. 4224.

[49] H. Wendland, Scattered Data Approximation, vol. 17, Cambridge University Press, 2004.

[50] L. Xie, H. Liu, Efficient mesh motion using radial basis functions with volume grid points reduction algorithm, J. Comput. Phys. 348 (2017) 401–415, https://doi.org/10.2514/6.2008-305.

[51] Y. Yu, Z. Lyu, Z. Xu, J.R. Martins, On the influence of optimization algorithm and initial design on wing aerodynamic shape optimization, Aerosp. Sci. Technol. 75 (2018) 183–199, https://doi.org/10.1016/j.ast.2018.01.016.

[52] B. Zhou, T.A. Albring, N.R. Gauger, C. Ilario, T.D. Economon, J.J. Alonso, Reduction of airframe noise components using a discrete adjoint approach, in: 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2017, p. 3658.

[53] B.Y. Zhou, T.A. Albring, N.R. Gauger, T.D. Economon, F. Palacios, J.J. Alonso, A discrete adjoint framework for unsteady aerodynamic and aeroacoustic optimization, in: 16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2015, p. 3355.