Bombardieri, R., Cavallaro, R., Sanchez, R., & Gauger, N. R. (2021). Aerostructural wing shape optimization assisted by algorithmic differentiation. Structural and Multidisciplinary Optimization.

DOI: https://doi.org/10.1007/s00158-021-02884-5

ACCEPTED MANUSCRIPT. Paper is published on Structural and Multidisciplinary Optimization
https://doi.org/10.1007/s00158-021-02884-5

Aerostructural Wing Shape Optimization Assisted by Algorithmic Differentiation

Rocco Bombardieri · Rauno Cavallaro · Ruben Sanchez · Nicolas R. Gauger

Received: date / Accepted: date

Abstract With more efficient structures, last trends in aeronautics have witnessed an increased flexibility of wings, calling for adequate design and optimization approaches. To correctly model the coupled physics, aerostructural optimization has progressively become more important, being nowadays performed also considering higher-fidelity discipline methods, i.e., CFD for aerodynamics and FEM for structures. In this work a model for high-fidelity gradient-based aerostructural optimization of wings, assisted by algorithmic differentiation and including aerodynamic and structural nonlinearities, is presented. First, the model is illustrated: a key feature lies in its enhanced modularity. Each discipline solver, employing algorithmic differentiation for the evaluation of adjoint-based sensitivities, is interfaced at high-level by means of a wrapper to both solve the aerostructural primal problem and evaluate discrete-consistent gradients of the coupled problem. Second, to demonstrate the feasibility of the method, a framework is ad-hoc set up, within the open-source SU2 multiphysics suite, with the inclusion of a geometrically nonlinear beam FE and an interface module to deal with non-matching 3D surfaces. Finally, the framework is applied to perform aerostructural optimization of aeroelastic test cases based on the ONERA M6 and NASA CRM wings. Single-point optimizations, employing Euler or RANS flow models, are carried out to find wing optimal outer mold line in terms of aerodynamic efficiency. Results remark the importance of taking into account the aerostructural coupling when performing wing shape optimization.

Keywords Aerostructural Optimization · Open-source suite SU2 · Algorithmic Differentiation · Adjoint Sensitivities · Computational Fluid Dynamics · Aeroelasticity

Rocco Bombardieri
Department of Bioengineering and Aerospace Engineering,
Universidad Carlos III de Madrid (UC3M), Madrid, Spain.
ORCID: 0000-0001-8892-1936

Rauno Cavallaro
Department of Bioengineering and Aerospace Engineering,
Universidad Carlos III de Madrid (UC3M), Madrid, Spain.
E-mail: rauno.cavallaro@uc3m.es
ORCID: 0000-0001-8380-1388

Ruben Sanchez
SU2 Foundation, California, USA.
ORCID: 0000- 0003-2326-7739

Nicolas R. Gauger
Chair for Scientific Computing, TU Kaiserslautern, Germany.
ORCID: 0000-0002-5863-7384

1 Introduction

Multi-Disciplinary Optimization (MDO) applied to aerostructural wing design problems has been attracting the interest of both research and industry for the last fifty years and is still today a widely researched topic. Typically, when referring to aerostructural optimization, a system is considered, governed by two state (or discipline) systems of equations: one relative to steady aerodynamics and the other to solid mechanics. Commonly, Design Variables (DVs) consist in aerodynamic shape and/or structural parameters (e.g., thickness of the skin), while constraints consider the integrity of the structure and/or the trim of the aircraft (mainly in terms of lift to be generated). Very often, constraints and objective functions regard aircraft performances, such as, range or fuel burn, which in turn, depend on aircraft weight and aerodynamic efficiency.

A critical aspect is the coupling between the physics of the two considered disciplines: for a given flight condition, the deflected shape of the wing is tightly coupled with the generated aerodynamic forces in a two-way relation. Current trends to design more efficient and lighter structures end up having more flexible wings, showing significant deflections while in operation. This, in terms of modeling, exacerbates both the need to include such aeroelastic tight coupling and to employ nonlinear and higher-fidelity solvers.

One of the pioneers of aerostructural optimization was Haftka[25] who proposed a method for the optimization of flexible wings subjected to stress, strain and drag constraints. The adopted discipline tools reflected the computational availability of the time, and consisted in a Lifting Line Method (LLM) for aerodynamics and a Finite Element Method (FEM) for the structure. A penalty method formulation converted the constrained problem into an unconstrained one.

The work of Grossman *et al.*[24] applied an integrated aircraft design analysis and optimization on a sailplane, exploring two optimization strategies: a sequential one, in which each field was optimized independently, and an integrated one. The integrated approach demonstrated to be superior, capitalizing on the interaction between disciplines and finding better results with respect to sub-optimal ones given by the sequential optimization. This trend has been recently confirmed by Martins *et al.*[43] using higher-fidelity approaches.

In the last twenty years, researchers have started to employ higher-fidelity tools to perform aerostructural optimization. This increased level of fidelity came with several consequences, being the first one an increased computational demand. Furthermore, when simulating transonic flows, given the high sensitivity with respect to small geometric features,[57] a large number of DVs is needed to fully exploit the potentials of the optimization process. For such reasons, gradient-based optimization[55] is preferred, as it generally converges faster to a minimum, and comes with a reduced computational cost if adjoint-technique is used for sensitivity evaluation; however, chance of hitting a local minimum exists due to the possible non-convexity of the problem.[41]

One of the pioneering works in this direction is the one of Maute *et al.*,[45] featuring a gradient-based optimization framework with a three-field approach (structures, aerodynamics, mesh) to model the nonlinear coupled problem. A staggered (also known as partitioned or segregated) procedure was set up to solve the coupled fields; aerodynamics was modeled with CFD-Euler, structure with a linear FE model and fluid mesh deformation with a spring-analogy method. For the evaluation of sensitivities a direct approach was chosen, in which partial derivatives where analytically calculated and a staggered scheme was used to solve the system of equations. Direct approach is, anyway, only convenient when then number of design variables is small. The authors stated that three-field formulations are on average 25% slower than two-field ones (i.e., strategies that bypass fluid mesh deformation problem) but allow for a more robust handling of large structural displacements.

Martins *et al.*[42] proposed a framework for the calculation of coupled aerostructural sensitivities for cases in which aeroelastic interactions were significant. The method employed high-fidelity models for both aerodynamics (CFD-Euler) and structure (linear FEM). This work is considered, to the best of the authors' knowledge, one of the first efforts to employ, in aerostrucutral optimization, an adjoint method, which makes the time for calculating sensitivities almost independent of the number of DVs.[44] The proposed sensitivity evaluation framework was a lagged-coupled adjoint, in which the single discipline adjoint equations were lagged in a similar fashion to the primal solver solution strategy, and partial derivatives were evaluated analytically or by finite differences (FD).

In a following effort[43] a similar framework was applied to the design of a supersonic business jet; the selected objective function was a weighted sum of structural weight and drag coefficient evaluated for a design lift coefficient; Gradient evaluation time was observed to be almost independent of the number of DVs.

Following their previous effort,[45] Maute *et al.*[46] proposed a different method for aeroelastic optimization. For the same aerostructural problem, gradient calculation was achieved by analytically deriving the adjoint sensitivity equations: a staggered solution algorithm was implemented, where partial derivatives could be calculated analytically or by automatic differentiation. The paper highlighted the computational problems relative to the storage of Jacobians matrices of large-scale problems for adjoint-based sensitivity calculations.

The work of Barcelos *et al.*[3] presented an optimization methodology for fluid-structure interaction (FSI) problems in which aerodynamics took into account turbulence by means of RANS with Spalart-Allmaras (SA) turbulence model.[65] The formulation was based on the three-field strategy used in Maute *et al.*,[45,46] whereas the structure was modeled with geometrically nonlinear FEM. This counts, to the best of the authors' knowledge, as the highest level of fidelity adopted so far for aerostructural optimization of wings. Calculation of sensitivities followed the direct approach, using parallel matrix-free iterative solvers; all partial derivative contributions were evaluated analytically or via FD. With the direct approach, evaluation of total derivatives of the state variables for the coupled aerostructural problem needs to be repeated for each design variable.

Other literature works presented gradient-based aerostructural optimizations using adjoint methods. Brezillion *et al.*[8] proposed an articulated high-fidelity optimization framework wrapping DLR's TAU code for CFD (RANS-SA) which includes a discrete-adjoint model with non-frozen turbulence and ANSYS for linear structural FEM.

A similar approach was pursued by Ghazlane *et al.*[22] In the presented aerostructural framework, aerodynamics (Euler flow model) was simulated by ONERA's *elsA* code,[10] which used an iterative fixed-point scheme for the solution of the aerodynamic adjoint; for the structure, a linear structural FEM module was analytically differentiated.

In effort[33] Kenway *et al.* introduced a high-fidelity framework that could perform aerostructural optimization with respect to thousands of multidisciplinary DVs, thanks to an improved parallel scalability of the method; a fully-coupled Newton-Krylov approach was employed for the solution of the aerostructural and the relative adjoint systems. The aerostructural solver was based, for the aerodynamic part, on *SUmb*[70] code, featuring automatic differentiation (*ADjoint*), and, for the structural part, on *TACS*,[30] also able to evaluate adjoint-based sensitivities. In the cited effort, Euler flow model was used together with a linear detailed FEM model of the structure. To solve the aerostructural adjoint equations a combination of analytic, forward, and reverse AD methods was adopted. The method was demonstrated on an aerostructural CRM test case.[67]

Similarly, in another study[35] Kenway *et al.* presented a multipoint high-fidelity aerostructural optimization of a CRM aeroelastic model employing a linear structural model and Euler-based aerodynamics, augmented with a low-fidelity viscous drag estimate.

Works of Kennedy *et al.*[31,34] focused on the aerostructural optimization of the CRM introducing composite materials. A linear structural model (TACS) was used together with low-fidelity aerodynamics (panel method) and Euler/RANS flow models.

Kenway *et al.*[32] and Brooks *et al.*[9] performed a similar RANS-based aerostructural optimization on variations of the NASA CRM (undeflected and higher aspect ratio versions) using a similar computational infrastructure as the one mentioned above.

The study of Hoogervorst *et al.*[28] proposed an MDO architecture based on an Individual Discipline Feasible (IDF) approach:[14,16] the disciplines of the aerostructural problem were decoupled and convergence was ensured imposing additional equality constraints on the interdisciplinary state variables, which became additional surrogate design variables. This allowed to lower the computational cost of the problem, especially in cases of strong nonlinearities, but only when the number of DVs was kept small. In the study, flow was modeled with the Euler equation using the open-source code *SU2*[20] while the structural solver FEMWET was used to solve the linear structural equations. A combination of continuous adjoint approach and FD was used for gradient evaluation.

In the field of rotorcraft analysis and optimization, the work of Mishra *et al.*[47,48] presented a sensitivity analysis of fully-coupled time-dependent aeroelastic problems. Both forward and adjoint sensitivity formulations were derived, levering on the same iterative solution strategy used for the primal problem. NSU3D software and a nonlinear in-house beam were used to solve the RANS flow-model and the structure. Both solvers featured a discrete adjoint method for sensitivities evaluation. Wang *et al.*[69] used a sensitivity analysis procedure based on a discretely consistent adjoint-based method in the fluid-dynamic solver (FUN3D) and a complex variable finite difference approach for structural sensitivities with respect to aerodynamic loads, by means of the nonlinear multibody structural solver DYMORE.

Kiviaho *et al.*[36] focused on the modularity of the framework for the steady aeroelastic optimization of wings. A nonlinear block Gauss-Seidel procedure was used to solve the primal and the discrete adjoint problems. A Python-based framework interfaced FUN3D (Euler/RANS flow model and mesh deformation), FUNtoFEM (inteface module) and TACS (linear FEM). With the reformulation of the Lagrangian, high sparsity of the system matrix was guaranteed for the adjoint problem, allowing for a staggered solution procedure, without passing copies of transposed discipline Jacobians between the several modules.

He *et al.*[27] presented an open-source, object-oriented MDO framework which provides a high-level interface to implement discrete adjoint method for existing or new steady-state primal solvers of OpenFOAM[26] with minimum impact on the source code. To do so, partial derivatives were calculated by finite differences accelerated by an advanced graph coloring algorithm.

1.1 Contributions of the present study

In this work, a modular strategy for gradient-based, high-fidelity aerostructural optimization, assisted by algorithmic differentiation is presented. The formulation is based on the work of Sanchez *et al.*[63] in which an integrated AD-based method, levering on a fixed-point iteration procedure, for the calculation of coupled aerostructural sensitivities was presented and implemented within the environment of the SU2 multiphysic suite.[2,23,49,54,56,73] In this paper, the adjoint problem is reformulated and extended in order to decouple the discipline solvers, which now only communicate by means of an external Python wrapper for the solution of both the primal and the relative adjoint problems. Such an approach provides increased flexibility: different solvers, ad-hoc tailored on the physics to be modeled, can be interfaced to perform aerostructural analysis and/or optimization. Moreover, the use of AD within the entire sensitivities evaluation workflow, delivers discrete-consistent sensitivities, if proper convergence of the primal and dual problems is achieved.

Exploitation of the herein proposed formulation allowed, with respect to the work of Sanchez *et al.*,[63] the introduction of a spline module, the interface of a geometrically nonlinear beam-based structural solver, and their integration into the sensitivities evaluation workflow. Such generalized layout is thought for aeronautical applications, typically featuring slender structures and non-matching 3D structural/fluid interfaces.

The framework implementing the proposed formulation has been released as *open-source* within the SU2 suite. SU2 dedicated solvers are used for CFD and fluid mesh deformation, while structure is handled by the beam FE solver *pyBeam* (ad-hoc rearranged in fixed-point form to exploit AD for sensitivities evaluation). PyBeam has been adapted as a model code to show how simple the integration of the chosen AD library (CoDiPack[60]) would be on existing codes. This can avoid manual implementations of adjoint algorithms which are usually complex and time-consuming to perform, despite recent progresses in this direction.[27] Being open-source, this framework provides an easy access to an aerostructural optimization tool, tailored for aircraft applications to a potentially large user audience.

Application of the method is carried out on aeroelastic test cases of potential industrial interest, based on the ONERA M6 and NASA CRM wings and featuring relevant structural deflection. Several levels of fidelity are employed in the analyses: together with the geometrically nonlinear structural model, both Euler and RANS-SA flow models are used for aerodynamics. For RANS-SA based applications, particular attention has been payed to solve the fluid primal and relative adjoint problems following the full-turbulence (or non-frozen-turbulence[3,41]) approach. Relevance of the aerostructural coupling on the optimization results is highlighted, showing how neglecting it can lead to a less performing design with respect to the initial nonoptimized configuration. Moreover, an alternative strategy is proposed, in which aerostructural coupling is considered only in the primal problem, and a comparison with the fully-coupled approach is provided.

Basic benchmarks of the implementation, in terms of speed, scalability and memory usage are also shown.

1.2 Organization of the paper

The remainder of this paper is organized as follows: in Section 2 the theoretical background for the solution of the aerostructural problem and for sensitivities evaluation is detailed; in Section 3 the aeroelastic test cases are introduced (i.e., aeroelastic models based on the ONERA M6 wing and on the CRM) whereas in Section 4 aerostructural sensitivities verification is presented and results of the optimization are shown and discussed. Section 5 concludes the paper and provides recommendations for future works.

2 Theoretical background

A more in-depth overview of the theoretical background and the solvers is presented in this section. First, the primal problem, i.e., the static aeroelastic equilibrium of a flexible wing subjected to a given flow, is formulated; each of the discipline solvers is briefly reviewed and their coupling and interfacing within the framework is discussed. Later on, the same primal problem is reformulated in the form of fixed-point iterations, which is the most suitable one for the implementation of the used AD-based adjoint method. State and design variables are introduced and the complete set of equations of the adjoint problem is shown, together with the reverse computational path followed by the algorithm for the evaluation of sensitivities. It is relevant to stress out that structural, aerodynamic and mesh solvers are implemented in independent modules, each one featuring its own AD-based sensitivity capability. Hence, the approach to be described is suitable for different combinations of solvers provided with the adequate primal/adjoint interfaces.

Last topic covered in this section is the aerostructural wing shape optimization formulation.

2.1 Primal problem

Structural FEM solver The structural in-house solver pyBeam relies on a 6-dof geometrically nonlinear beam model following the work of Levy.[39] The Euler-Bernoulli beam kinematic assumption is considered; the formulation follows an Updated Lagrangian approach with a corotational[5] frame to extract the strains from large displacements. Nonlinear rigid elements are employed for a correct transfer of displacements and forces between the structural and fluid meshes (see Section 3). The implementation is based on the penalty method proposed by Belytschko.[5]

In its FE discretized form the governing equation is:

$$\mathscr{S}(\mathbf{u_s}) = \mathbf{f_s} - \mathbf{f_{int}}(\mathbf{u_s}) - \mathbf{f_{rig}}(\mathbf{u_s}) = \mathbf{0} \tag{1}$$

where, $\mathbf{u_s}$, $\mathbf{f_s}$, $\mathbf{f_{int}}$ and $\mathbf{f_{rig}}$ are, respectively, the nodal generalized displacements (measured from the unloaded initial configuration), external and internal nodal forces vector, and the contribution of rigid elements to the residual. Equation (1) is solved iteratively with a Newton-Raphson method:

$$\mathbf{K}\,\Delta\mathbf{u_s} = -\mathscr{S}(\mathbf{u_s}) \tag{2}$$

with the Jacobian $\mathbf{K} = \frac{\partial \mathscr{S}(\mathbf{u_s})}{\partial \mathbf{u_s}}$ retaining the nonlinear contributions of both beam and rigid elements. A load-stepping strategy, i.e., a progressive application of the external loads, is used in equation (1) to facilitate convergence.

PyBeam is coded in C++ and its top-level functions are wrapped in Python using SWIG,[4] to be accessible by external solvers. Moreover, like SU2 CFD solver,[2] it employs AD by means of CoDiPack library for sensitivites evaluation.

CFD solver Focus is on transonic flows around aerodynamic bodies governed by the compressible Navier-Stokes equations. For this purpose, the flow solver available in the open-source multiphysics suite SU2 is chosen. Following the work of Economon *et al.*,[20] the governing equations formulated in conservative form including the energy equation can be written as:

$$\mathscr{F}(\mathbf{w}) = \frac{\partial \mathbf{w}}{\partial t} + \nabla\cdot\mathbf{F}^c(\mathbf{w}) - \nabla\cdot\mathbf{F}^v(\mathbf{w}) - \mathbf{Q}(\mathbf{w}) = \mathbf{0} \tag{3}$$

where $\mathbf{w} = (\rho, \rho\mathbf{v}, \rho E)$ is the vector of conservative variables, ρ the flow density, $\mathbf{v}$ the flow velocity and E the total energy per unit mass. $\mathbf{Q}(\mathbf{w})$ is a generic source term, $\mathbf{F}^c(\mathbf{w})$ and $\mathbf{F}^v(\mathbf{w})$ are, respectively, the convective and viscous fluxes, and can be written as

$$\mathbf{F}^c(\mathbf{w}) = \begin{pmatrix} \rho\mathbf{v} \\ \rho\mathbf{v}\otimes\mathbf{v} + p\mathbf{I} \\ \rho E\mathbf{v} + p\mathbf{v} \end{pmatrix} \tag{4}$$

$$\mathbf{F}^v(\mathbf{w}) = \begin{pmatrix} 0 \\ \tau \\ \tau\cdot\mathbf{v} + \mu^* C_p \nabla T \end{pmatrix} \tag{5}$$

where C_p is the specific heat at constant pressure and T is the temperature, calculated using the ideal gas model. The viscous stress tensor is:

$$\tau = \mu_{tot}\left(\nabla\mathbf{v} + \nabla\mathbf{v}^T - \frac{2}{3}\mathbf{I}(\nabla\cdot\mathbf{v})\right) \tag{6}$$

where, based on the Boussinesq hypothesis,[72] the total viscosity μ_{tot} is modelled as a sum of a laminar component which satisfies Sutherland's law[71] (μ_{lam}) and a turbulent component (μ_{turb}) which is obtained from the solution of a turbulence model. Finally,

$$\mu^* = \frac{\mu_{lam}}{Pr_l} + \frac{\mu_{turb}}{Pr_t} \tag{7}$$

where Pr_l and Pr_t are the laminar and turbulent Prandtl numbers, respectively. In this investigation, when viscous flows are considered, μ_{turb} is calculated by means of the one-equation Spalart-Allmaras turbulence model.[65]

SU2 core is written in C++ and top-level functions are wrapped in Python using SWIG. Both continuous and discrete adjoint capabilities are provided; in particular, discrete adjoint implementation features CoDiPack for AD-based sensitivities evaluation.

Fluid mesh deformation solver For problems involving moving boundaries it is important to account for the modification of the fluid domain. Among the several strategies proposed in the literature,[18] it has been decided to rely on a linear elastic volume deformation method, which performs well in case of large deformations. Such strategy is implemented in the SU2 dedicated mesh deformation solver; it supports AD for gradient evaluation and its top level functions are wrapped in Python using SWIG.

In order to find the new position of the nodes in the fluid domain, the mesh deformation problem can be treated as a pseudo-elastic linear problem,[19]

$$\mathscr{M}(\mathbf{z}, \mathbf{u_f}) = \mathbf{K_m}\cdot\mathbf{z} - \tilde{\mathbf{f}}(\mathbf{u_f}) = 0 \tag{8}$$

where $\mathbf{K_m}$ is a fictitious stiffness matrix and $\tilde{\mathbf{f}}$ are fictitious forces which ensure the boundary displacements $\mathbf{u_f}$.

In problems involving moving mesh boundaries, equation (3) needs to be rewritten with the inclusion of the domain grid points position $\mathbf{z}$, following the Arbitrary Lagrangian-Eulerian (ALE) formulation:[17,18,52,53]

$$\mathscr{F}(\mathbf{w},\mathbf{z}) = \frac{\partial \mathbf{w}}{\partial t} + \nabla\cdot\mathbf{F}^c(\mathbf{w},\mathbf{z}) - \nabla\cdot\mathbf{F}^v(\mathbf{w},\mathbf{z}) - \mathbf{Q}(\mathbf{w}) = \mathbf{0} \tag{9}$$

Interfacing method To transfer information between the non-conformal structural FEM and CFD grids, an in-house Moving Least Square algorithm is implemented,[13] based on Radial Basis Functions (RBF)[59] and *ANN* library.[50] Briefly, given $\mathbf{x_s} \in \mathbb{R}^{N_s}$, the position of the structural nodes and $\mathbf{x_f} \in \mathbb{R}^{N_f}$, the position of the fluid nodes on the moving boundary, it is possible to build a so-called spline matrix $\mathbf{H}_{MLS} = \mathbf{H}_{MLS}(\mathbf{x_s}, \mathbf{x_f}) \in \mathbb{R}^{N_f \times N_s}$ such that:

$$\mathbf{u_f} = \mathbf{H}_{MLS}\cdot\mathbf{u_s}, \tag{10}$$

$$\mathbf{f_s} = \mathbf{H}_{MLS}^T\cdot\mathbf{f_f} \tag{11}$$

where N_f and N_s are, respectively, the dimensions of the aerodynamic moving surface and structural degrees of freedom, while $\mathbf{u_f}$, $\mathbf{f_f} \in \mathbb{R}^{N_f}$ and $\mathbf{u_s}$, $\mathbf{f_s} \in \mathbb{R}^{N_s}$ are, respectively, the displacements/forces defined on the aerodynamic/structural mesh. Aerodynamic forces, i.e., $\mathbf{f_f} = \mathbf{f_f}(\mathbf{w},\mathbf{z})$, are provided by the fluid solver for the converged solution of equation (9). As already stated in the work of Quaranta *et al*,[58] employing the transpose of the spline matrix in equation (11) ensures the energy conservation.

The spline tool has been already successfully applied to a large variety of challenging cases, such as wings with mobile surfaces, free-flying aircraft models[12] and other cases in which interpolation of information between 1D (structural) and 3D (aerodynamic) topologies had to be performed.[6] The module, coded in C++ into an independent library, has also been wrapped in Python.

Coupling method A partitioned approach is employed for the FSI system solution, following a three-field formulation.[3,21] This approach, according to Maute *et al.*,[46] is suitable for problems featuring large structural deformations.

Recalling the three fields under investigation, namely, structural $\mathscr{S}$, fluid $\mathscr{F}$ and mesh $\mathscr{M}$, the whole FSI system $\mathscr{G}$ can be expressed as a function of the state variables $\mathbf{u_s}$, $\mathbf{w}$

and $\mathbf{z}$ which are, respectively, structural displacements, aerodynamic state variables and fluid mesh nodes displacements. Hence, following Sanchez,[62] the primal problem is:

$$\mathscr{G}(\mathbf{u_s},\mathbf{w},\mathbf{z}) = \begin{cases} \mathscr{S}(\mathbf{u_s},\mathbf{w},\mathbf{z}) = 0, \\ \mathscr{F}(\mathbf{w},\mathbf{z}) = 0, \\ \mathscr{M}(\mathbf{u_s},\mathbf{z}) = 0, \end{cases} \tag{12}$$

Due to the non matching structural and fluid interfaces, the above system is closed by means of interfacing equations (10,11).

Due to the nonlinear nature of the FSI problem, an iterative approach based on Newton method is sought. At each iteration, the corresponding linear system is solved using a Block Gauss-Seidel (BGS) strategy, which suits well the the selected partitioned approach. The approximated linear system reads:

$$\begin{bmatrix} \frac{\partial \mathscr{F}}{\partial \mathbf{w}} & 0 & 0 \\ 0 & \frac{\partial \mathscr{S}}{\partial \mathbf{u_s}} & 0 \\ 0 & \frac{\partial \mathscr{M}}{\partial \mathbf{u_s}} & \frac{\partial \mathscr{M}}{\partial \mathbf{z}} \end{bmatrix} \begin{Bmatrix} \Delta \mathbf{w} \\ \Delta \mathbf{u_s} \\ \Delta \mathbf{z} \end{Bmatrix} = - \begin{Bmatrix} \mathscr{F}(\mathbf{w},\mathbf{z}) \\ \mathscr{S}(\mathbf{u_s},\mathbf{w},\mathbf{z}) \\ \mathscr{M}(\mathbf{u_s},\mathbf{z}) \end{Bmatrix} \tag{13}$$

in which the upper-right part of the Jacobian has been set to 0.[3] It is underlined how, although linear, the mesh deformation problem of equation (8) has been recast into Newton-Rapson-like form to fit into the BGS representation shown above.

In the work of Degroote *et al.*[15] slow convergence or divergence of the BGS approach in case of strong FSI interactions (e.g., strong geometrical nonlinearities) is observed. To ensure the stability of the method, a relaxation parameter α is applied to the boundary displacements:[29]

$$\mathbf{u_f}^{*} = \alpha \mathbf{u_f}^{n} + (1-\alpha)\mathbf{u_f}^{n-1}. \tag{14}$$

where n and $n-1$ are, respectively, the current and previous BGS subiterations.

Concerning the implementation, a Python orchestrator links the wrapped libraries and allows the sequential solution of each discipline within a single FSI iteration. Structural displacements $\mathbf{u_s}$ are accessible from pyBeam module; they are interpolated into the fluid boundary using equation (10) after the spline matrix has been assembled by the interface module. The fluid boundary displacements are then transferred to the mesh solver in SU2 via its Application Programming Interface (API).[64] A new value of the aerodynamic forces on the boundary is obtained after a CFD simulation in SU2 and interpolated back into the structural model using equation (11). The primal solver layout is shown in Figure 1; its algorithm is resumed in Algorithm 1.

Algorithm 1: Aerostructural primal solver

```
Initialize $N = 1, (\mathbf{w}, \mathbf{z}, \mathbf{u_s}, \mathbf{u_f}, \mathbf{f_s}, \mathbf{f_f}) = \mathbf{0}$
while $N \leq N_{FSI}$ do
    Run CFD solver: $\mathbf{z} \rightarrow \mathbf{w}, \mathbf{f_f}, C_D, C_L$
        while $\left\| \mathbf{w}_{k_f} - \mathbf{w}_{k_f+1} \right\| \leq \varepsilon_f$ do: iterate $k_f$
    Transfer loads (Spline): $\mathbf{f_f} \rightarrow \mathbf{f_s}$
    Run structural FEM solver: $\mathbf{f_s} \rightarrow \mathbf{u_s}$
        while $\left\| \mathbf{u_s}_{k_s} - \mathbf{u_s}_{k_s+1} \right\| \leq \varepsilon_s$ do: iterate $k_s$
    Transfer displacements (Spline): $\mathbf{u_s} \rightarrow \mathbf{u_f}$
    Run mesh deformation solver: $\mathbf{u_f} \rightarrow \mathbf{z}$
end
```

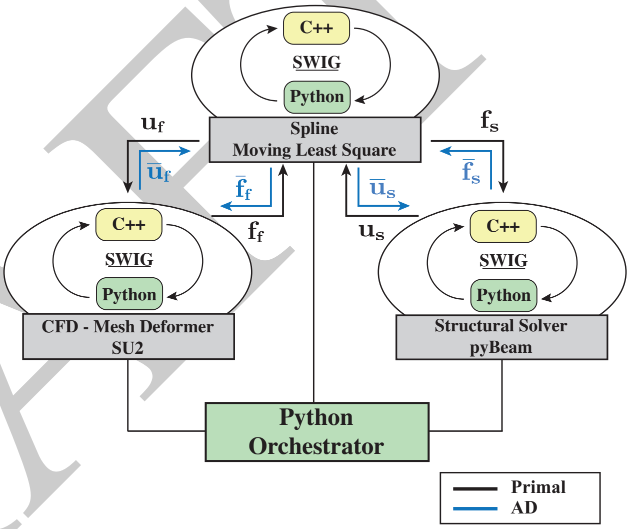


Fig. 1 Framework layout for Primal and AD modes.

2.2 Coupled aerostructural adjoint method

It has already been mentioned that each of the described modules features CoDiPack for adjoint AD-based sensitivities evaluation, being one of the contributions of the present study to show how different AD-based modules can be interfaced, by means of a high-level Python wrapper, for the purpose of coupled sensitivities evaluation.

Following the work of Sanchez *et al.*[63] let's define the Objective Function (OF) and DVs for the current optimization problem. The OF that will be considered in this effort is the aerodynamic drag coefficient:

$$J = C_D(\mathbf{w},\mathbf{z}) \tag{15}$$

Table 1 summarizes the set of state and design variables. Notice that the displacement of the wing surface $\mathbf{u_{tot}}$ is expressed as the sum of the displacements due to the jig-shape redesign $\mathbf{u}_{\mathbf{F}_\alpha}$ and the displacements due to the deflection (aerostructural coupling). Even though both OF and DVs

Table 1 Complete set of state and design variables for aerostructural shape optimization.

	State variables
$\mathbf{u_s}$	Structural displacements
$\mathbf{w}$	Flow conservative variables
$\mathbf{z}$	Volume mesh displacements
$\mathbf{f_f}$	Fluid loads
$\mathbf{f_s}$	Structural loads
$\mathbf{u_f}$	Displacements of wing surface due to deflection
$\mathbf{u_{tot}}$	Cumulative displacements of wing surface
	Design variables
$\mathbf{u}_{\mathbf{F}_\alpha}$	Variation of the jig-shape

are relative to aerodynamics, the problem is still multidisciplinary due to the aerostructural coupling of the governing equations. The proposed optimization framework is flexible and general, and allows to freely choose OF, constraints and DVs for any of the considered disciplines with minor modifications of the source code and of the sensitivity evaluation workflow.

Let the complete set of equations of the primal problem, illustrated in section 2.1, be rewritten in the form of *fixed-point* iterators:[1,2,37]

$$\mathbf{F}(\mathbf{w},\mathbf{z}) - \mathbf{w} = 0 \tag{16a}$$

$$\mathbf{F_f}(\mathbf{w},\mathbf{z}) - \mathbf{f_f} = 0 \tag{16b}$$

$$\mathbf{M}(\mathbf{u_{tot}}) - \mathbf{z} = 0 \tag{16c}$$

$$\mathbf{H}_{MLS}^T \cdot \mathbf{f_f} - \mathbf{f_s} = 0 \tag{16d}$$

$$\mathbf{S}(\mathbf{u_s},\mathbf{f_s}) - \mathbf{u_s} = 0 \tag{16e}$$

$$\mathbf{H}_{MLS} \cdot \mathbf{u_s} - \mathbf{u_f} = 0 \tag{16f}$$

$$\mathbf{u_{tot}} - \mathbf{u_f} - \mathbf{u}_{\mathbf{F}_\alpha} = 0 \tag{16g}$$

In system (16) equation (16a) is the fixed-point version of equation (9) and, together with fluid loads and objective function evaluation (equations (16b,15) respectively) represents the core of the aerodynamic solver. Equation (16c) is the fixed-point form of mesh deformation problem (equation (8)), whereas equation (16e) is the fixed-point version of the structural problem (equation (1)). $\mathbf{F}$, $\mathbf{F_f}$, $\mathbf{M}$ and $\mathbf{S}$ are the fixed-point operators of the corresponding residuals of the primal problem (see system of equations (12)).

Equations (16a-16c,15) are handled by the SU2 suite while equation (16e) is handled by pyBeam. Cross-dependencies (displacements $\mathbf{u_s}/\mathbf{u_f}$ and forces $\mathbf{f_s}/\mathbf{f_f}$) are handled by the orchestrator at high-level after the spline matrix has been assembled.

The optimization problem is formulated as:

$$\begin{aligned} \min_{\mathbf{u}_{\mathbf{F}_\alpha}} \quad & J(\mathbf{w},\mathbf{z}) \\ \text{subject to:} \quad & \mathbf{F}(\mathbf{w},\mathbf{z}) - \mathbf{w} = 0 \\ & \mathbf{F_f}(\mathbf{w},\mathbf{z}) - \mathbf{f_f} = 0 \\ & \mathbf{M}(\mathbf{u_{tot}}) - \mathbf{z} = 0 \\ & \mathbf{H}_{MLS}^T \cdot \mathbf{f_f} - \mathbf{f_s} = 0 \\ & \mathbf{S}(\mathbf{u_s},\mathbf{f_s}) - \mathbf{u_s} = 0 \\ & \mathbf{H}_{MLS} \cdot \mathbf{u_s} - \mathbf{u_f} = 0 \\ & \mathbf{u_{tot}} - \mathbf{u_f} - \mathbf{u}_{\mathbf{F}_\alpha} = 0 \end{aligned} \tag{17}$$

Such problem can be reformulated in the equivalent unconstrained optimization problem defined with the Lagrangian $\mathscr{L}$:

$$\begin{aligned} \mathscr{L}(\mathbf{w},\bar{\mathbf{w}},\mathbf{z},\bar{\mathbf{z}},\mathbf{u_s},\bar{\mathbf{u}}_\mathbf{s},\mathbf{u_f},\bar{\mathbf{u}}_\mathbf{f},\mathbf{f_s},\bar{\mathbf{f}}_\mathbf{s},\mathbf{f_f},\bar{\mathbf{f}}_\mathbf{f},\mathbf{u_{tot}},\bar{\mathbf{u}}_\mathbf{tot},\mathbf{u}_{\mathbf{F}_\alpha}) = \\ J(\mathbf{w},\mathbf{z}) + \bar{\mathbf{w}}^T \left[\mathbf{F}(\mathbf{w},\mathbf{z}) - \mathbf{w}\right] + \bar{\mathbf{z}}^T \left[\mathbf{M}(\mathbf{u_{tot}}) - \mathbf{z}\right] + \\ \bar{\mathbf{u}}_\mathbf{tot}^T \left[\mathbf{u_{tot}} - \mathbf{u_f} - \mathbf{u}_{\mathbf{F}_\alpha}\right] + \bar{\mathbf{u}}_\mathbf{f}^T \left[\mathbf{H}_{MLS} \cdot \mathbf{u_s} - \mathbf{u_f}\right] + \\ \bar{\mathbf{u}}_\mathbf{s}^T \left[\mathbf{S}(\mathbf{u_s},\mathbf{f_s}) - \mathbf{u_s}\right] + \bar{\mathbf{f}}_\mathbf{s}^T \left[\mathbf{H}_{MLS}^T \cdot \mathbf{f_f} - \mathbf{f_s}\right] + \\ \bar{\mathbf{f}}_\mathbf{f}^T \left[\mathbf{F_f}(\mathbf{w},\mathbf{z}) - \mathbf{f_f}\right] \end{aligned} \tag{18}$$

in which the Lagrangian multipliers $\bar{\mathbf{w}}$, $\bar{\mathbf{z}}$, $\bar{\mathbf{u}}_\mathbf{tot}$, $\bar{\mathbf{u}}_\mathbf{f}$, $\bar{\mathbf{u}}_\mathbf{s}$, $\bar{\mathbf{f}}_\mathbf{s}$ and $\bar{\mathbf{f}}_\mathbf{f}$, corresponding to the adjoint of the state variables, are introduced.

Imposing the Karush-Kuhn-Tucker (KKT) conditions it is possible to: retrieve the state equations (16) by differentiation of the Lagrangian with respect to the adjoint variables; obtain the set of adjoint equations differentiating the Lagrangian with respect to the state variables:

$$\frac{\partial \mathscr{L}}{\partial \mathbf{w}} = \frac{\partial J}{\partial \mathbf{w}} + \bar{\mathbf{w}}^T \left[\left.\frac{\partial \mathbf{F}}{\partial \mathbf{w}}\right|_{\mathbf{w}^*,\mathbf{z}^*} - 1 \right] + \bar{\mathbf{f}}_\mathbf{f}^T \left.\frac{\partial \mathbf{F_f}}{\partial \mathbf{w}}\right|_{\mathbf{w}^*,\mathbf{z}^*} = \mathbf{0} \tag{19a}$$

$$\frac{\partial \mathscr{L}}{\partial \mathbf{z}} = \frac{\partial J}{\partial \mathbf{z}} + \bar{\mathbf{w}}^T \left.\frac{\partial \mathbf{F}}{\partial \mathbf{z}}\right|_{\mathbf{w}^*,\mathbf{z}^*} + \bar{\mathbf{f}}_\mathbf{f}^T \left.\frac{\partial \mathbf{F_f}}{\partial \mathbf{z}}\right|_{\mathbf{w}^*,\mathbf{z}^*} + \bar{\mathbf{z}}^T = \mathbf{0} \tag{19b}$$

$$\frac{\partial \mathscr{L}}{\partial \mathbf{u_{tot}}} = \frac{\partial J}{\partial \mathbf{u_{tot}}} + \bar{\mathbf{z}}^T \left.\frac{\partial \mathbf{M}}{\partial \mathbf{u_{tot}}}\right|_{\mathbf{u}_\mathbf{tot}^*} + \bar{\mathbf{u}}_\mathbf{tot}^T = \mathbf{0} \tag{19c}$$

$$\frac{\partial \mathscr{L}}{\partial \mathbf{u_f}} = \frac{\partial J}{\partial \mathbf{u_f}} - \bar{\mathbf{u}}_\mathbf{f}^T - \bar{\mathbf{u}}_\mathbf{tot}^T = \mathbf{0} \tag{19d}$$

$$\frac{\partial \mathscr{L}}{\partial \mathbf{u_s}} = \frac{\partial J}{\partial \mathbf{u_s}} + \bar{\mathbf{u}}_\mathbf{s}^T \left[\left.\frac{\partial \mathbf{S}}{\partial \mathbf{u_s}}\right|_{\mathbf{u}_\mathbf{s}^*,\mathbf{f}_\mathbf{s}^*} - 1 \right] + \bar{\mathbf{u}}_\mathbf{f}{}^T \, \mathbf{H}_{MLS} = \mathbf{0} \tag{19e}$$

$$\frac{\partial \mathscr{L}}{\partial \mathbf{f_s}} = \frac{\partial J}{\partial \mathbf{f_s}} + \bar{\mathbf{u}}_\mathbf{s}^T \left.\frac{\partial \mathbf{S}}{\partial \mathbf{f_s}}\right|_{\mathbf{u}_\mathbf{s}^*,\mathbf{f}_\mathbf{s}^*} - \bar{\mathbf{f}}_\mathbf{s}^T = \mathbf{0} \tag{19f}$$

$$\frac{\partial \mathscr{L}}{\partial \mathbf{f_f}} = \frac{\partial J}{\partial \mathbf{f_f}} - \bar{\mathbf{f}}_\mathbf{f}^T + \bar{\mathbf{f}}_\mathbf{s}^T \, \mathbf{H}_{MLS}^T = \mathbf{0}, \tag{19g}$$

and, finally, retrieve the optimality condition, differentiating the Lagrangian with respect to the DVs. For a local minimum, it holds that:

$$\frac{\mathrm{d}J}{\mathrm{d}\mathbf{u}_{\mathbf{F}_\alpha}} = \frac{\partial \mathscr{L}}{\partial \mathbf{u}_{\mathbf{F}_\alpha}} = \frac{\partial J}{\partial \mathbf{u}_{\mathbf{F}_\alpha}} - \bar{\mathbf{u}}_{\mathbf{tot}}^T = \mathbf{0} \tag{20}$$

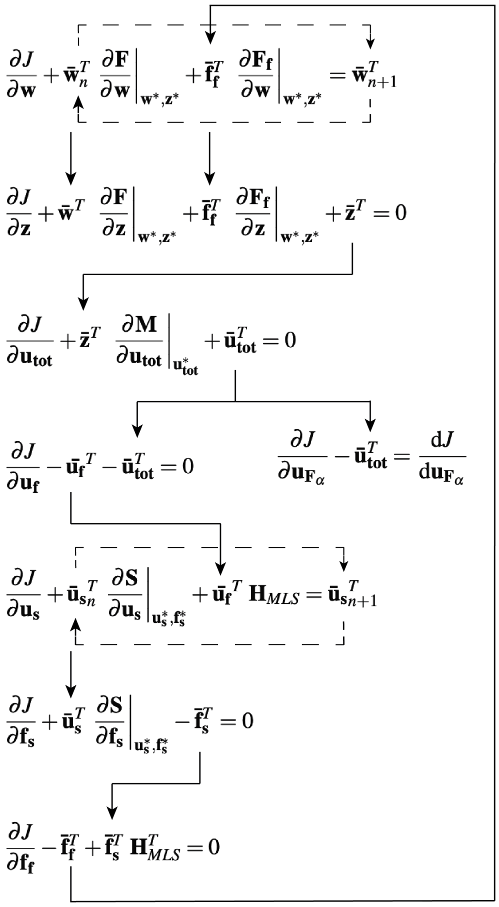
The adjoint variables can be computed solving the system (19), and are then used for gradient evaluation in equation (20). The matrix-vector products in the form $\bar{\mathbf{y}}^T \left.\frac{\partial \mathbf{A}}{\partial \mathbf{x}}\right|_{\mathbf{x}^*}$ in equations (19) are evaluated using the AD tool CoDiPack about the solution of the aerostructural primal problem (i.e., at $\mathbf{w}^*$, $\mathbf{z}^*$, $\mathbf{u_{tot}}^*$, $\mathbf{u_f}^*$, $\mathbf{u_s}^*$, $\mathbf{f_s}^*$ and $\mathbf{f}_\mathbf{f}^*$).

To solve the general problem $\bar{\mathbf{x}}^T = \bar{\mathbf{y}}^T \left.\frac{\partial \mathbf{A}}{\partial \mathbf{x}}\right|_{\mathbf{x}^*}$, given the solution of the primal problem $\mathbf{x}^*$, the primal solver $\mathbf{A}$ is advanced for one iteration which is recorded with CoDiPack. Once the solution is recorded, CoDiPack evaluates $\bar{\mathbf{x}}$ for a given value of $\bar{\mathbf{y}}$, provided in input by the user. This strategy allows to iteratively solve in parallel the adjoint system of equations (19) and the optimality condition together, without the need to store and operate directly on large-scale matrices such as $\frac{\partial \mathscr{F}}{\partial \mathbf{z}}$ and $\frac{\partial \mathscr{F}}{\partial \mathbf{w}}$. This is one of the main bottlenecks that needs to be addressed, for example by means of matrix-free approaches,[3,36] when implementing discrete adjoint methods.

The reverse computational path for sensitivities calculation is summarized in Algorithm 2 and Figure 2. Within the context of the current modular framework, AD is applied to every module and cross-term adjoints are propagated backward to each discipline by the top-level orchestrator. The procedure, repeated till convergence, is conceptually similar to the BGS staggered solution used in the primal problem (see Figure 1).

Algorithm 2: Aerostructural adjoint problem

```
Initialize $N = 1, (\bar{\mathbf{w}}, \bar{\mathbf{z}}, \bar{\mathbf{u}}_{\mathbf{tot}}, \bar{\mathbf{u}}_\mathbf{f}, \bar{\mathbf{u}}_\mathbf{s}, \bar{\mathbf{f}}_\mathbf{s}, \bar{\mathbf{f}}_\mathbf{f}) = \mathbf{0}$
while $N \le N_{ADJ}$ do
    Run fluid adjoint (eq. (19a)): $\bar{\mathbf{f}}_\mathbf{f} \to \bar{\mathbf{w}}$
        while $\left\|\bar{\mathbf{w}}_{k_f} - \bar{\mathbf{w}}_{k_f+1}\right\| \le \varepsilon_f$ do: iterate $k_f$
    Evaluate $\bar{\mathbf{z}}$ (eq. (19b)): $\bar{\mathbf{f}}_\mathbf{f}, \bar{\mathbf{w}} \to \bar{\mathbf{z}}$
    Run mesh adjoint (eq. (19c)): $\bar{\mathbf{z}} \to \bar{\mathbf{u}}_{\mathbf{tot}}$
    Evaluate $\bar{\mathbf{u}}_\mathbf{f}$ (eq. (19d)): $\bar{\mathbf{u}}_{\mathbf{tot}} \to \bar{\mathbf{u}}_\mathbf{f}$
    Run struct. adjoint (eq. (19e)): $\bar{\mathbf{u}}_\mathbf{f} \to \bar{\mathbf{u}}_\mathbf{s}$
        while $\left\|\bar{\mathbf{u}}_{\mathbf{s}k_s} - \bar{\mathbf{u}}_{\mathbf{s}k_s+1}\right\| \le \varepsilon_s$ do: iterate $k_s$
    Evaluate $\bar{\mathbf{f}}_\mathbf{s}$ (eq. (19f)): $\bar{\mathbf{u}}_\mathbf{s} \to \bar{\mathbf{f}}_\mathbf{s}$
    Evaluate $\bar{\mathbf{f}}_\mathbf{f}$ (eq. (19g)): $\bar{\mathbf{f}}_\mathbf{s} \to \bar{\mathbf{f}}_\mathbf{f}$
end
```



$\frac{\partial J}{\partial \mathbf{w}} + \bar{\mathbf{w}}_n^T \left.\frac{\partial \mathbf{F}}{\partial \mathbf{w}}\right|_{\mathbf{w}^*,\mathbf{z}^*} + \bar{\mathbf{f}}_\mathbf{f}^T \left.\frac{\partial \mathbf{F}_\mathbf{f}}{\partial \mathbf{w}}\right|_{\mathbf{w}^*,\mathbf{z}^*} = \bar{\mathbf{w}}_{n+1}^T$
$\frac{\partial J}{\partial \mathbf{z}} + \bar{\mathbf{w}}^T \left.\frac{\partial \mathbf{F}}{\partial \mathbf{z}}\right|_{\mathbf{w}^*,\mathbf{z}^*} + \bar{\mathbf{f}}_\mathbf{f}^T \left.\frac{\partial \mathbf{F}_\mathbf{f}}{\partial \mathbf{z}}\right|_{\mathbf{w}^*,\mathbf{z}^*} + \bar{\mathbf{z}}^T = 0$
$\frac{\partial J}{\partial \mathbf{u_{tot}}} + \bar{\mathbf{z}}^T \left.\frac{\partial \mathbf{M}}{\partial \mathbf{u_{tot}}}\right|_{\mathbf{u}_{\mathbf{tot}}^*} + \bar{\mathbf{u}}_{\mathbf{tot}}^T = 0$
$\frac{\partial J}{\partial \mathbf{u_f}} - \bar{\mathbf{u}}_\mathbf{f}^T - \bar{\mathbf{u}}_{\mathbf{tot}}^T = 0$
$\frac{\partial J}{\partial \mathbf{u}_{\mathbf{F}_\alpha}} - \bar{\mathbf{u}}_{\mathbf{tot}}^T = \frac{\mathrm{d}J}{\mathrm{d}\mathbf{u}_{\mathbf{F}_\alpha}}$
$\frac{\partial J}{\partial \mathbf{u_s}} + \bar{\mathbf{u}}_{\mathbf{s}n}^T \left.\frac{\partial \mathbf{S}}{\partial \mathbf{u_s}}\right|_{\mathbf{u}_\mathbf{s}^*,\mathbf{f}_\mathbf{s}^*} + \bar{\mathbf{u}}_\mathbf{f}^T \mathbf{H}_{MLS} = \bar{\mathbf{u}}_{\mathbf{s}n+1}^T$
$\frac{\partial J}{\partial \mathbf{f_s}} + \bar{\mathbf{u}}_\mathbf{s}^T \left.\frac{\partial \mathbf{S}}{\partial \mathbf{f_s}}\right|_{\mathbf{u}_\mathbf{s}^*,\mathbf{f}_\mathbf{s}^*} - \bar{\mathbf{f}}_\mathbf{s}^T = 0$
$\frac{\partial J}{\partial \mathbf{f_f}} - \bar{\mathbf{f}}_\mathbf{f}^T + \bar{\mathbf{f}}_\mathbf{s}^T \mathbf{H}_{MLS}^T = 0$

Fig. 2 Reverse computational path of the adjoint FSI solver.

2.3 Aerostructural wing shape optimization

To facilitate communication with the FSI orchestrator, the optimization tool is purely Python-based and wraps the modules used for OF, constraints, gradients and Jacobian evaluation. The algorithm selected for the gradient-based optimization is the Sequential Least Square Quadratic Programming (SLSQP),[38] a gradient based algorithm that uses a Broyden Fletcher Goldfarb Shanno (BFGS)-based second order approximation of the objective function.

With respect to the previous section and for optimization purposes, the number of DVs is reduced by parametrizing the wing jig-shape with an FFD technique.[61] Aerodynamic grid surface node positions are linked to the FFD Control Points (CPs) with a trivariate interpolation based on Bezier's basis functions:

$$\mathbf{x}_{\mathbf{F}_\alpha} = \sum_{i=0}^{l} \sum_{j=0}^{m} \sum_{k=0}^{n} N_i(\mu)\, N_j(\nu)\, N_k(\xi)\, \mathbf{x}_{ijk}^{CP} \tag{21}$$

In equation (21), $\mathbf{x}_{\mathbf{F}_\alpha}$ is the coordinate vector of the generic node of the aerodynamic mesh lying on the wing surface,

$\mathbf{x}_{ijk}^{CP}$ is the position of the CP identified by indexes i, j, k; N are Bernstein polynomials and μ, ν, ξ are parametric coordinates evaluated with a point-inversion procedure.[57] New DVs are then the FFD box CPs, and the gradient of the OF with respect to them is easily evaluated applying the chain rule:

$$\frac{\mathrm{d}J}{\mathrm{d}\mathbf{u}_{ijk}^{CP}} = \frac{\mathrm{d}J}{\mathrm{d}\mathbf{u}_{\mathbf{F}_\alpha}} \frac{\mathrm{d}\mathbf{u}_{\mathbf{F}_\alpha}}{\mathrm{d}\mathbf{u}_{ijk}^{CP}} \tag{22}$$

FFD deformation strategies are largely used in literature,[8,34,35,43] being independent of the grid topology and easy to use in automatic processes; they are well suited for cases in which the topology of the geometry is not expected to change (e.g., wings and fuselages).[40] Even though CPs do not have any direct engineering interpretation, strategical use of CP displacements can achieve consistent changes in the wing twist, chord and span[28] while ensuring the continuity of the surface.

An important reason to reduce the number of DVs is due to inherent limitations of the optimization algorithm: Kraft recommends only moderately large size problems for SLSQP,[38] although, for adjoint-based gradient evaluation methods, computational time is almost independent of the number of DVs.

As a reasonable approximation, the interface matrix of equations (10,11) and the FFD box parametric coordinates of equation (21) are evaluated for the initial jig-shape configuration and held constant throughout the whole optimization process, instead of being updated for each variation of the jig-shape.

Constraints So far no mention to the optimization constraints has been made, to focus on the aerostructural coupling problem and relative sensitivities. The lift coefficient C_L at which the drag is measured is prescribed and, hence, held constant throughout the optimization. In the proposed framework, fixed C_L constraint is imposed by gradually changing the angle of attack during the iterative process.[43] With the above procedure, the constraint is accommodated internally in the aerostructural solver and is not treated at the optimization level. This feature was originally present in the SU2 aerodynamic shape optimization tool,[2] and has been extended to the coupled aerostructural problem.

Geometrical constraints are imposed using SU2 module *GEO*. Providing the topology of the aerodynamic body and exploiting the FFD box parametrization, SU2_GEO can evaluate several kinds of constraints (e.g., wing curvature, volume and dihedral, airfoil chord, thickness, twist and LE radius) and their gradients with respect to the CP displacements, by means of FDs.

As pointed out by Lyu *et al.*[41] one of the weaknesses of single-point optimizations, without the use of an appropriate penalty, is the progressive thinning of wing leading edges. This can be avoided performing a more costly multi-point optimization, as sharp leading edges would perform poorly in off design point. In the proposed framework, instead, such issue is taken care of by manually setting to zero the OF sensitivities $\frac{\mathrm{d}J}{\mathrm{d}\mathbf{u}_{\mathbf{F}_\alpha}}$ relative to grid points close to sharp edge regions.

3 Aeroelastic test cases

3.1 Test case based on ONERA M6

The first test case is based on ONERA M6 wing geometry. A synthetic structure has been assembled based on the work of Bombardieri *et al.*,[6] in which wing box properties (i.e. wing box cross section and material Young Modulus) were selected for the aeroelastic model to exhibit flutter in transonic regime; in the current effort such properties have been fine-tuned to obtain sought levels of wingtip deflection in flying conditions. The wing box, whose elastic axis is located at 40% of the wing chord, is described by beam elements. Four rigid elements have been cross placed at several stations along the wing span to reproduce the position of the leading edge (LE), trailing edge (TE), upper and lower point positions of the current wing section (airfoil). This solution has been found successful for a correct application of the spline algorithm in order to transfer information between solid and fluid boundary meshes. The structural model is clamped in correspondence of the wing root. Although the ONERA M6 is a low aspect ratio wing for a Euler-Bernoulli beam model to deliver an accurate physic representation, such a test case has been selected for sensitivities verification and optimization trends assessment. Layout of the structural model is shown in Figure 3.

Concerning the aerodynamic part of the problem, for this test case, flow has been model with Euler equations. The CFD mesh[6,7] is shown in Figure 3. The volume contains 582,752 tetrahedral elements and wing surface is discretized with 36,454 triangular elements. The computational domain is a box shape extending approximately for 13 root chords downstream, for 12 root chords upstream and for 9 semi-spans laterally.

Solution of the CFD equation has been performed with a 3 level Multi-Grid scheme together with a 2^{nd} order in space Jameson-Schmidt-Turkel (JST) scheme for the convective flux. Main CFD options for this test case are given in Table 2.

For the purpose of sensitivities verification only, a coarser mesh, depicted in Figure 4, is considered, with 140,244 tetrahedral elements in the computational domain and 5,640 triangular elements on the wing surface. For all applications concerning this test case, considered flight condition is a steady flight at $M = 0.8395$ and sea level ($\rho = 1.2250\ kg/m^3$). For the splining process, interpolating polynomials of order 2 are selected together with weight Radial Basis Functions of

Table 2 Numerical options for Euler-based CFD problems.

Parameter	Value
Multi-Grid levels nr.	3
Convective flow num. method	JST
Pseudo-time num. method	Euler implicit
Linear solver	FGMRES
Linear solver precond.	LU_SGS

Table 3 Numerical options for the RANS-SA-based CFD problem.

Parameter	Value
Multi-Grid levels nr.	0
Convective flow num. method	JST
Pseudo-time num. method	Euler implicit
Linear solver	FGMRES
Linear solver precond.	ILU

degree of smoothness 1. The number of nearest structural neighbour points is 100.

3.2 Test cases based on the NASA CRM

The second aeroelastic test case is based on the NASA CRM[67] and will be referred to as Synthetic-CRM (SCRM). The SCRM structural beam model has been generated from the global FEM (gFEM) model “V15wingbox” available in the NASA Common Research Model website repository.[51] The gFEM has been converted to a beam-based FE model by means of a modal equivalence process[68] and is based on the CRM outer mold line at 1 g load factor. To exhibit the sought level of wingtip deflection in flying conditions and trigger geometric nonlinearities, the value of the synthetic Young Modulus has been fine-tuned. The same strategy as for the previous test case, i.e., adding four rigid elements at several wing sections along the span, guarantees an appropriate interfacing between the structural and fluid meshes. The model is simply supported at the section corresponding to the wing-fuselage intersection, and symmetry constraint is applied to the far inboard section, see Figure 5.

With respect to the wing shape, the original NASA CRM, featuring a wing + body layout, has been modified removing the fuselage and extending the wing root till the symmetry plane, maintaining the TE and LE sweep angles. The mesh used for the Euler case, built and validated in a previous effort[66] features 1,529,927 tetrahedral elements while the wing boundary consists of 71,998 triangular elements. As depicted in Figure 5, the computational domain has a bullet shape, extending approximately for 20 root chords downstream, for 21 root chords upstream and for 10 semi-spans laterally. Numerical solution of the CFD equation has been performed with same options as for the previous test case (see Table 2). For the splining process, interpolating polynomials of order 2 are selected together with weight Radial Basis Functions of degree of smoothness 2. The number of nearest structural neighbour points is 280.

For the RANS-SA simulation a different mesh is used which has been built and validated in a previous effort.[11] It consists of 1,549,052 hexahedral elements while the wing boundary features 10,669 elements. As depicted in Figure 6, the computational domain is a box, extending approximately for 16 root chords downstream, for 12 root chords upstream and for 4 semi-spans laterally. Main CFD options for this test case are given in Table 3. With this mesh, for the splining process, interpolating polynomials of order 2 are selected together with weight Radial Basis Functions of degree of smoothness 2. The number of nearest structural neighbour points is 230.

For all applications concerning this test case, considered flight condition is a steady flight at $M = 0.85$ and sea level ($\rho = 1.2250\, kg/m^3$).

4 Results

In this section results of the verification campaign of the coupled aerostructural sensitivities are shown first. Thereafter, wing shape optimizations carried out on the test cases are presented and results are discussed. In particular, when showing optimization results, several subcases are discussed to highlight, by means of physical considerations, the relevance of performing wing optimization including the aerostructural coupling. Since considered DVs are relative to the wing aerodynamic surface, such optimization will be referred to as AeroStructural Wing Shape Optimization (ASWSO). On the other hand, a less effective, though computationally less intensive, procedure would be performing the optimization of the wing surface considering a rigid configuration (i.e., without the inclusion of aerostructural coupling effects); such optimization will be referred to as Aerodynamic Wing Shape Optimization (AWSO).

4.1 Sensitivities verification

Extending the work of Bombardieri *et al.*[7] in which the calculation of selected AD-based cross sensitivities was verified for the proposed aeroelastic test case (i.e., the sensitivity of C_D and C_L with respect to the structure Young Modulus), in this effort further verification results are shown. Total derivatives of the drag coefficient with respect to the wing surface jig-shape parameters are considered. Sensitivities are evaluated for the ONERA M6 aeroelastic test case using the coarse mesh version (see Section 3) and Euler flow model.

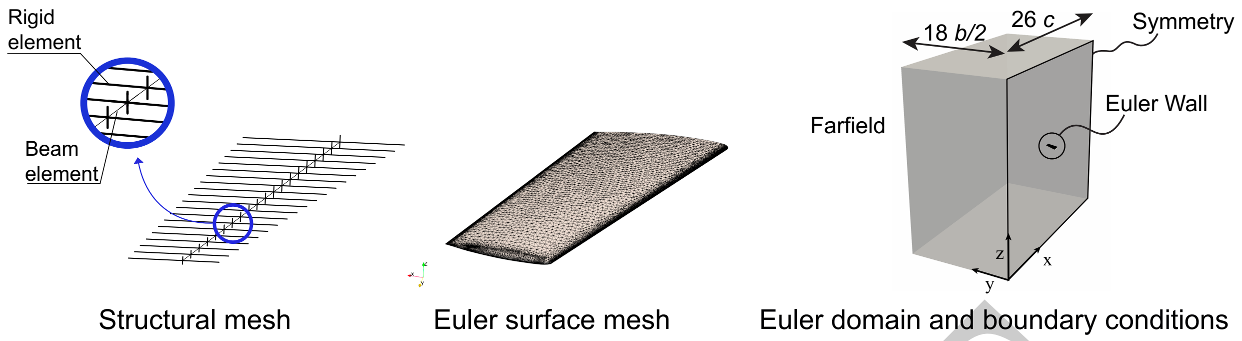


Fig. 3 Meshes for the Euler-based ONERA M6 test case. Fluid domain dimensions are given as function of the wing root chord c and semi-span $b/2$.

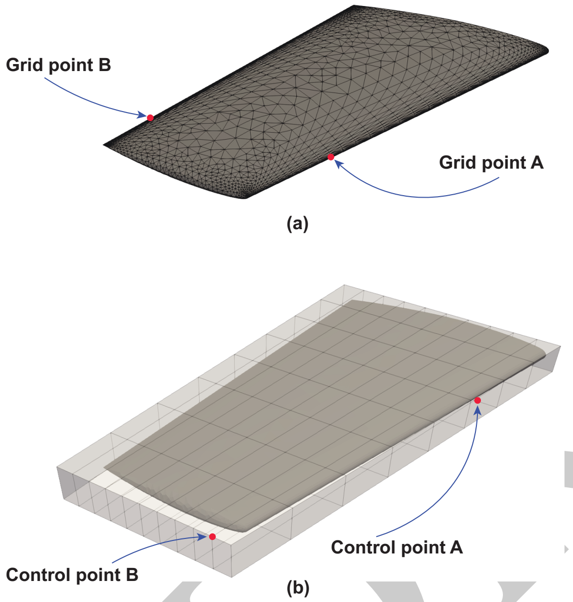


Fig. 4 ONERA M6 coarse mesh wing surface with the grid points (a) and control points (b) considered for sensitivities verification.

First, sensitivities of C_D with respect to the vertical displacement for two selected wing surface nodes of the jig-shape (variable $\mathbf{u}_{\mathbf{F}_\alpha}$ in Table 1) are verified. Such nodes are depicted in Figure 4(a) and are located outboard, in correspondence of the LE (grid point A) and of the TE (grid point B). The AD-based sensitivities provided by the framework are compared to the ones evaluated by a central FD scheme. For FD-based sensitivities a parametric study of the step size is conducted for the selected design variables in order to deliver converged values. Results of such campaign are shown in Table 4 for a fixed angle of attack (AoA = 3.06 deg) for two different synthetic Young Modulus E of the structure, whereas in Table 5 the same comparison is proposed for a fixed lift coefficient ($C_L = 0.22$) and one value of E.

For the calculation of the sensitivities with respect to the vertical FFD CP displacements, the chain rule is applied as shown in equation (22) in Section 2.3. The FFD-box uses Bezier basis functions of order 10, 8, 1 chord-wise, span-wise and along the thickness, respectively. The relative CPs chosen for verification are shown in Figure 4(b). CP A is located inboard, in correspondence of the compression side of the wing LE while CP B is located outboard, in correspondence of the suction side of the wing LE. Comparison between sensitivities predicted by the framework with AD and by central FD is provided in Table 6 for fixed AoA = 3.06 deg and one value of E.

Excellent agreement between sensitivities calculated by the two methods is found. It is, anyway, pointed out that, due to truncation errors, FDs are not reliable[33] in detecting errors below $O(10^{-2})$.

4.2 Euler-based optimization of the ONERA M6

This section discusses the results of the optimization of the ONERA M6 aeroelastic test case. Optimization constraints are shown in Table 7. DVs are vertical positions (z direction with respect to reference system in Figure 3) of CPs of the FFD-box shown in Figure 4(b), employing Bezier basis functions of order 10, 8, 1 chord-wise, span-wise and along the thickness, respectively. CPs on the symmetry plane are kept fixed as an effective way to prevent a change in shape of the relative airfoil. The synthetic Young Modulus has been tuned for the structure to exhibit wing-tip deflection of approximately 13% of the semi-span and trigger geometrical nonlinearities (see Figure 11).

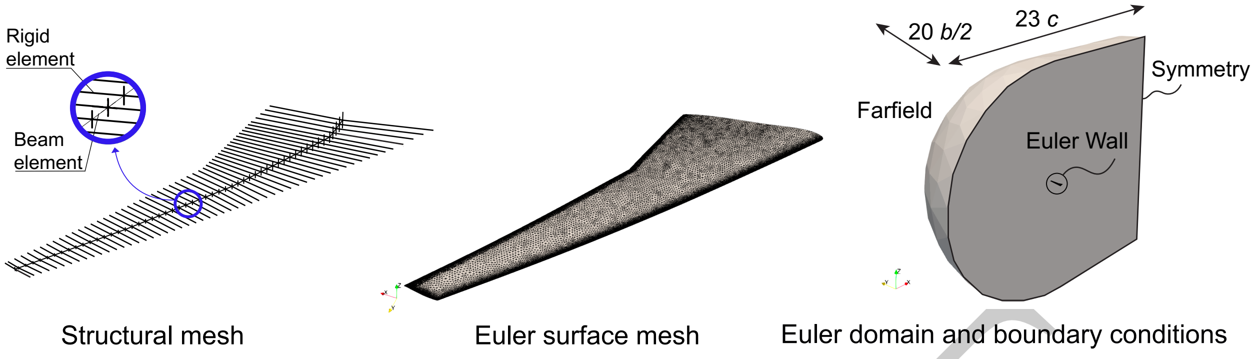


Fig. 5 Meshes for the Euler-based SCRM test case. Fluid domain dimensions are given as function of the wing root chord c and semi-span $b/2$.

Table 4 Aerostructural sensitivities of C_D with respect to vertical jig shape boundary displacements $\mathbf{u}_{\mathbf{F}_{\alpha_z}}$ calculated using FD and AD for two values of the synthetic E, $AoA = 3.06$ and $M_\infty = 0.8395$.

	E = 40 [GPa]		E = 20 [GPa]	
Grid pt. A	Sens.	Relative error to FD	Sens.	Relative error to FD
FD	- 0.0043**72394124**	–	0.00669**1539871**	–
AD	- 0.0043**66897122**	0.1257%	0.00669**8868402**	0.1094%
Grid pt. B	Sens.	Relative error to FD	Sens.	Relative error to FD
FD	0.0112**88968085**	–	0.001710**656774**	–
AD	0.0112**65754131**	0.2056%	0.001710**362610**	1.719e-04%

Table 5 Aerostructural sensitivities of C_D with respect to vertical jig-shape node displacements $\mathbf{u}_{\mathbf{F}_{\alpha_z}}$ calculated using FD and AD. $C_L = 0.22$, $M_\infty = 0.8395$ and $E = 40$ GPa.

Grid pt. A	Sens.	Relative error to FD
FD	0.00805**59423**	–
AD	0.00805**89472**	0.0373%
Grid pt. B	Sens.	Relative error to FD
FD	0.001627**56262**	–
AD	0.001627**95035**	0.0238%

Table 6 Aerostructural sensitivities of C_D with respect to vertical CP displacements $\mathbf{u}^{\mathbf{CP}}_{\mathbf{F}_{\alpha_z}}$ calculated using FD and AD. $AoA = 3.06$, $M_\infty = 0.8395$ and $E = 40$ GPa.

CP A	Sens.	Relative error to FD
FD	0.0043**083629927**	–
AD	0.0043**296403878**	0.4914%
CP B	Sens.	Relative error to FD
FD	0.01203**79174801**	–
AD	0.01203**49705805**	0.0244%

Table 7 Set of constraints and total number of DVs used for the optimization of the ONERA M6 aeroelastic test case.

Aerodynamic constraints		
C_L	=	0.286
Geometric constraints		
t/c (sec. at 16.4% span)	$\geq$	9.64%
t/c (sec. at 32.8% span)	$\geq$	9.60 %
t/c (sec. at 49.2% span)	$\geq$	9.58%
t/c (sec. at 65.7% span)	$\geq$	9.49%
Number of DVs	=	198

4.2.1 Aerodynamic wing shape optimization

First, an AWSO is run for the test case. Result of this optimization, in terms of C_D vs optimization iterations is shown in Figure 7. For the optimal shape a C_D reduction of 35.38% with respect to the baseline configuration is achieved. Figure 8 shows the C_p distribution for the top and front view of the baseline (left) and optimal (right) designs. It can be noted how the optimal design doesn't feature the characteristic lambda shock of the original design.

4.2.2 Aerostructural wing shape optimization

An ASWSO is then run. C_D evolution is shown for this case in Figure 9. For the optimal wing a reduction of C_D of 35.39%

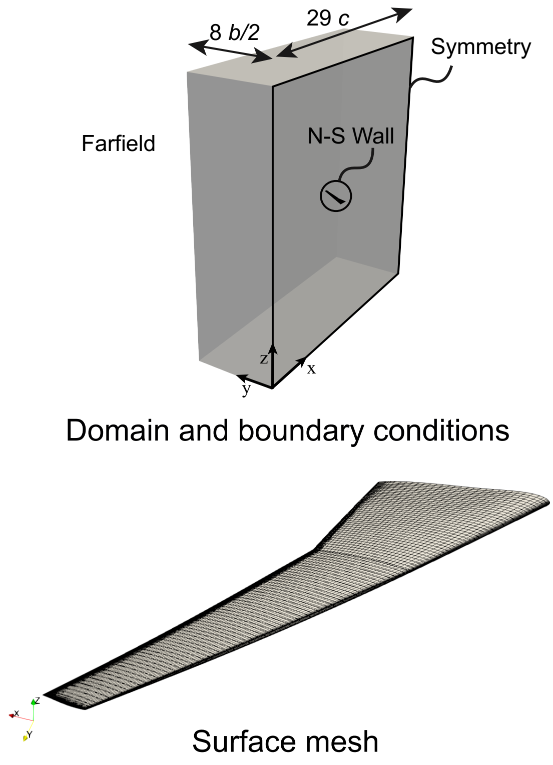


Fig. 6 Aerodynamic mesh for the RANS-SA-based SCRM test case. Fluid domain dimensions are given as function of the wing root chord c and semi-span $b/2$.

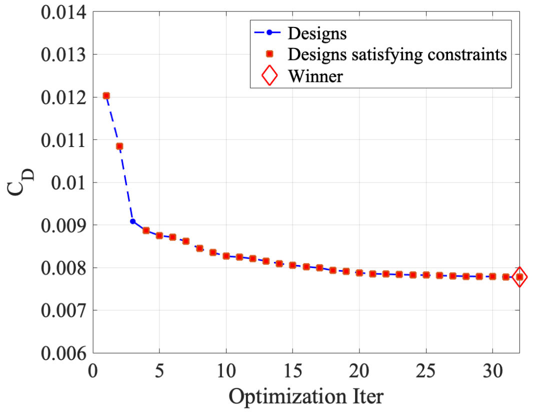


Fig. 7 C_D reduction for the Euler-based AWSO of the ONERA M6 wing.

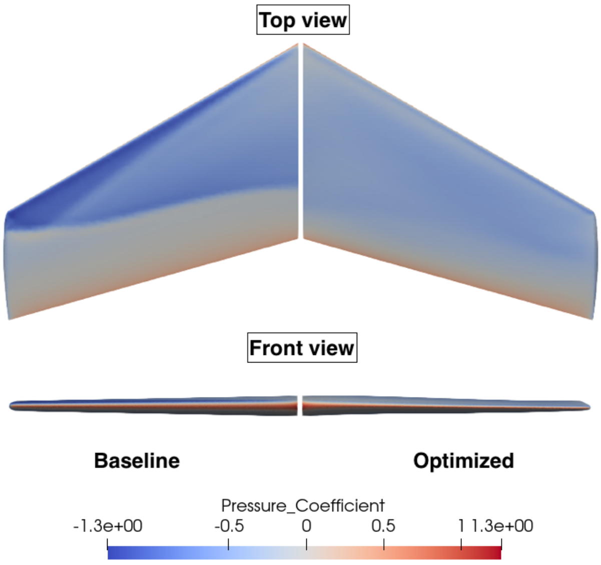


Fig. 8 Euler-based AWSO of the ONERA M6 wing: C_p distribution on the baseline and the optimized designs.

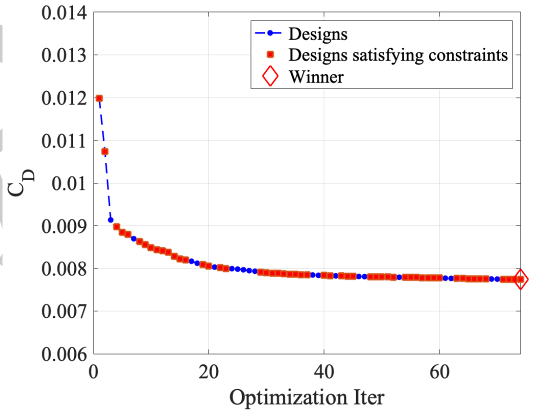


Fig. 9 C_D reduction for the Euler-based ASWSO of the ONERA M6 aeroelastic test case.

is obtained with respect to the baseline configuration in its relative *flying shape* (i.e., the wing in its deformed shape at aeroelastic equilibrium).

Figure 10 shows the C_p distribution for the baseline (left) and optimal (right) designs. From the top view it is apparent how the original design at aeroelastic equilibrium features a similar shock wave pattern as the one observed in its undeflected condition. Aerostructural optimal wing achieves a reduction of C_D by alleviating the shock wave in its flying shape. The front view allows to appreciate the maximum wing-tip deflection for both designs.

4.2.3 AWSO and ASWSO comparison

Relevance of pursuing an aerostructural optimization can be inferred from Table 8, which compares the C_D for the flying shapes of the original design, the AWSO optimal design, and the ASWSO optimal design. It can be seen how the AWSO optimum in operation shows a value of the C_D which is far from the “real” optimum evaluated by means of an ASWSO. This discrepancy can be explained by the fact that AWSO

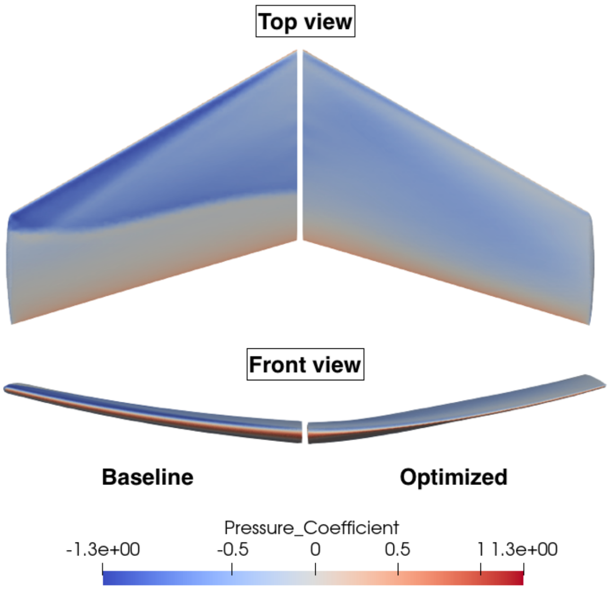


Fig. 10 Euler-based ASWSO of the ONERA M6 wing: C_p distribution on the baseline and the optimized designs at aeroelastic equilibrium.

Table 8 Comparison of C_D between ASWSO, AWSO optima and the original configuration in flying shape, for the Euler-based ONERA M6 test case.

Configuration	$\mathbf{C_D}$	Diff. %
ASWSO optimum	0.00775	–
AWSO optimum	0.00824	6.32%
Original	0.01199	35.39%

optimizes the wing around its rigid configuration which is, de-facto, an off-design point with respect to the flying shape in which the wing operates and which is naturally considered by an ASWSO. The more the wing is flexible, the more the rigid shape differs from the shape at aeroelastic equilibrium. For this same reason, if large wing deflections are expected, geometrical nonlinearities should be considered in aerostructural optimization.

Figure 11 shows a detail of the tip deflection for the three configurations. It is interesting to notice how the wing optimized considering the aerostructural coupling displays a larger tip deflection than the other wings.

For the flying shapes of the AWSO and ASWSO optima, Figure 12 shows, for selected sections along the wing span, the airfoil shapes (with their relative position in space) and the C_p distribution. Airfoils for both optimized configurations are different than the symmetric airfoils characteristic of the original ONERA M6 aerodynamic surface. It can be noticed, close to the wing tip (sections at 70% and 90% span), the higher deflection of the ASWSO optimal wing.

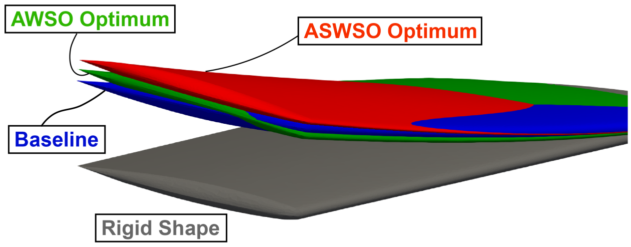


Fig. 11 Comparison between ASWSO, AWSO optima and the original configuration flying shapes for the Euler-based ONERA M6 test case: detail of the wing tip.

Table 9 Set of constraints and total number of DVs used for the optimization of the Euler-based SCRM aeroelastic test case.

Aerodynamic constraints		
C_L	=	0.5
Geometric constraints		
t/c (sec. at 0.34% span)	$\geq$	15.6%
t/c (sec. at 16.32% span)	$\geq$	12.5%
t/c (sec. at 27.01% span)	$\geq$	11.2%
t/c (sec. at 38.49% span)	$\geq$	10.4%
t/c (sec. at 49.76% span)	$\geq$	10.0%
t/c (sec. at 60.74% span)	$\geq$	9.8%
t/c (sec. at 71.89% span)	$\geq$	9.6%
t/c (sec. at 83.07% span)	$\geq$	9.5%
t/c (sec. at 94.14% span)	$\geq$	9.5%
Number of DVs	=	418

4.3 Euler-based optimization of the SCRM

This section discusses the results of the Euler-based optimization performed on the SCRM aeroelastic test case. Optimization costraints are shown in Table 9. An FFD box is built based on Bezier functions of order 10, 18, 1 chord-wise, span-wise and along the thickness, respectively, as depicted in Figure 13. DVs are the vertical positions of all FFD-box CPs. CPs on the symmetry plane are kept fixed as an effective way to prevent changes in the shape of the relative airfoil. The synthetic Young Modulus has been tuned for the structure to exhibit wing-tip deflection of approximately the 6% of the semi-span (see Figure 18).

4.3.1 Aerodynamic wing shape optimization

Results of the AWSO process in terms of drag coefficient evolution are shown in Figure 14, from which a C_D reduction of 9.13% can be inferred. Figure 15 shows the C_p distribution for the top and front view of the baseline (left) and optimal (right) designs. On the optimal design, the shock redistribution and alleviation along the wing span are apparent.

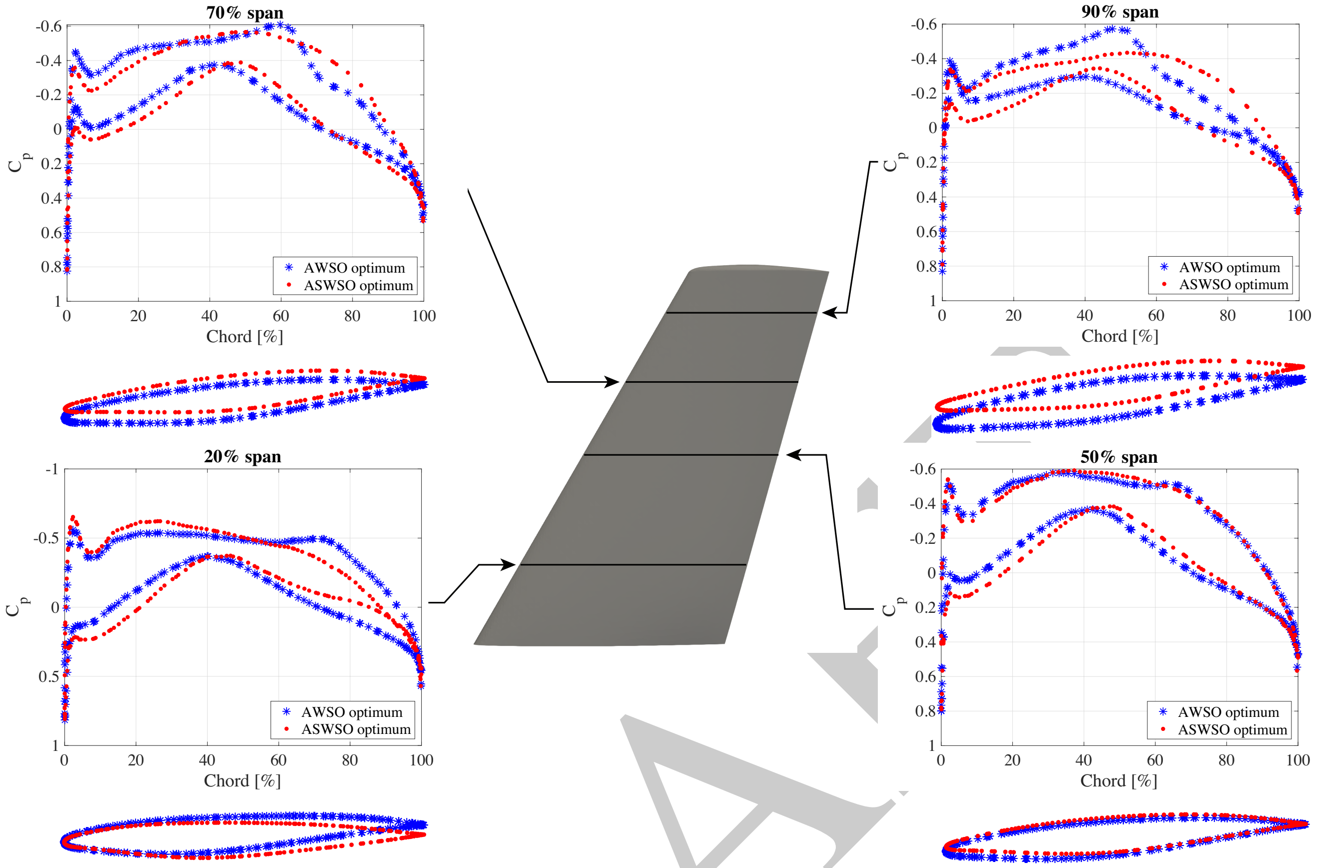


Fig. 12 Flying shape comparison between AWSO and ASWSO optima for the Euler-based ONERA M6 test case. For selected stations along the wing span both C_p distributions and airfoil shapes (with relative position in space) are shown.

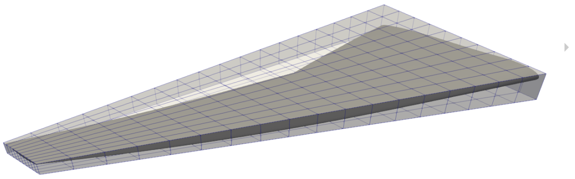


Fig. 13 FFD box used for the Euler-based optimization of the SCRM aeroelastic model.

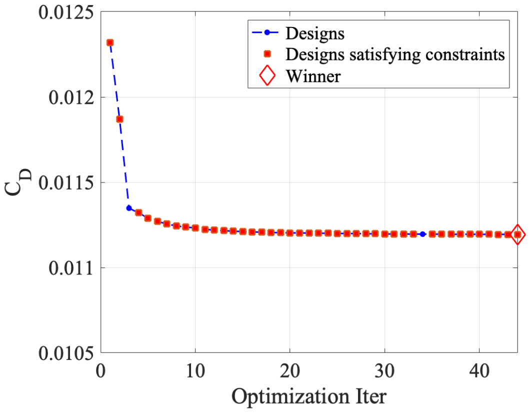


Fig. 14 C_D reduction for the Euler-based AWSO of the SCRM wing.

4.3.2 Aerostructural wing shape optimization

Figure 16 depicts the drag coefficient evolution versus optimization iterations for an ASWSO on the aeroelastic test case. A C_D reduction of 3.84% with respect to the baseline configuration is obtained, which, even if representing a significant improvement, is smaller than the one observed for the AWSO case. It is also underlined how optimization parameters needed more tuning if compared to the AWSO case, witnessing an increased complexity of the problem due to the aerostructural coupling. It can be speculated that for such test case, opening the design space, in particular adding DVs relative to the structural domain (e.g., wing-box element sizes), may be needed for larger efficiency improvements. Figure 17 depicts the C_p distribution on baseline (left) and optimal (right) designs. It can be noted, from the top view

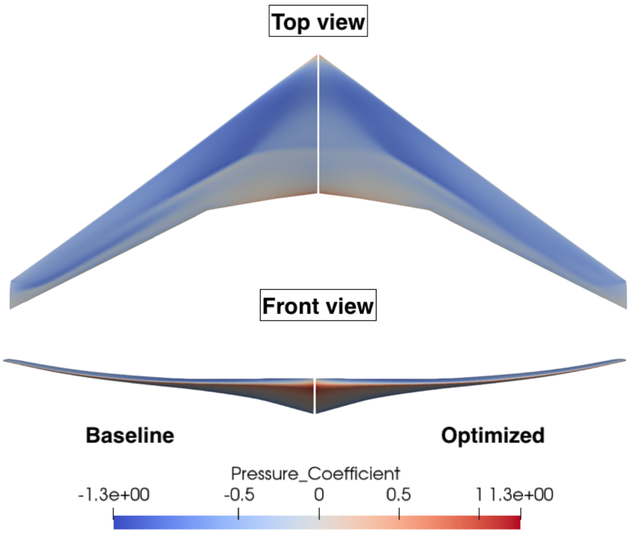


Fig. 15 Euler-based AWSO of the SCRM wing: C_p distribution on the baseline and the optimized designs.

of the optimized configuration, the shock redistribution and alleviation close to the wing tip. The front view allows to appreciate the maximum wing-tip deflection for both designs.

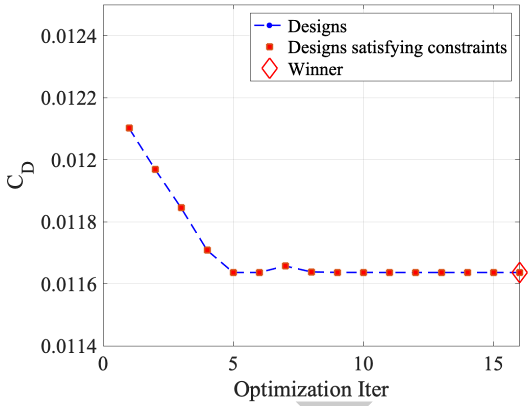


Fig. 16 C_D reduction for the Euler-based ASWSO of the SCRM wing.

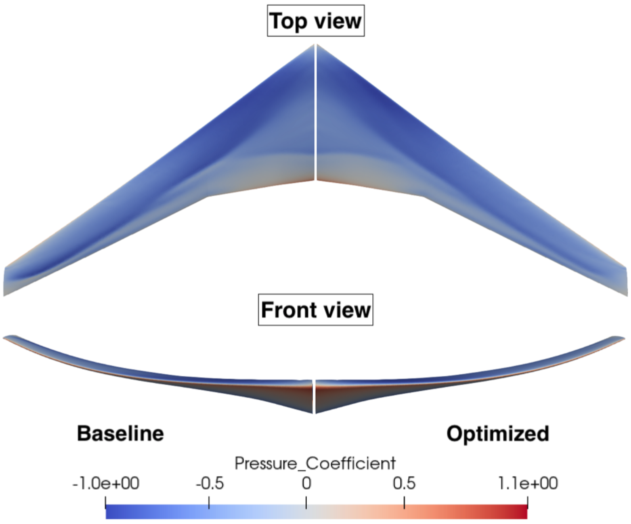


Fig. 17 Euler-based ASWSO of the SCRM wing: C_p distribution on the baseline and the optimized designs at aeroelastic equilibrium.

Table 10 Comparison of C_D between ASWSO, AWSO optima and the original configuration in flying shape, for the Euler-based SCRM test case.

Configuration	C_D	Diff. %
ASWSO optimum	0.01163	–
AWSO optimum	0.01243	6.87%
Original	0.01210	4.04%

4.3.3 AWSO and ASWSO comparison

Results of the optimization campaign are summarized in Table 10 where the C_D for the flying shapes of the original design, the AWSO and the ASWSO optimal designs are shown, while Figure 18 depicts a detail of the tip deflection for the three configurations. As already observed in the previous test case, the AWSO optimum in its flying shape has a considerably higher C_D than the one of the ASWSO. However, this test case shows a relevant peculiarity: performances of the AWSO optimal wing are considerably poorer than the ones of the original (unoptimized) design at aeroelastic equilibrium. Hence, the computational costs of performing an aerodynamic shape optimization without considering the flexibility of the structure might not payback in more efficient wings when operating in their actual flying shape.

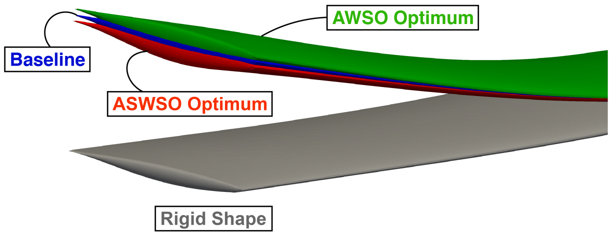


Fig. 18 Comparison between ASWSO, AWSO optima and the original configuration flying shapes for the Euler-based SCRM test case: detail of wing tip.

For the flying shapes of the AWSO and ASWSO optima, Figure 19 shows, for selected sections along the wing span, the C_p distribution and the airfoil shapes (with their relative position in space). C_p distribution highlights how shock-related gradients are generally much weaker for the ASWSO optimum with respect to its AWSO counterpart, which has been, de-facto, optimized about a different operating condition. It can also be noted how ASWSO optimum has a smaller tip deflection compared to the AWSO one (as also shown in Figure 18), showing hence an opposite trend with respect to the one seen for the ONERA M6 test case.

4.3.4 Optimization with uncoupled sensitivities

Section 4.3.3 underlines how, for certain conditions, the computational effort of an AWSO doesn't necessarily payback with more efficient wings with respect to the baseline. On the other hand, ASWSO is computationally more demanding. An intermediate approach that might combine the advantages of AWSO and ASWSO is to consider the aerostructural coupling in the primal problem and neglect it for sensitivities evaluation. Such uncoupled sensitivities are the purely aerodynamic ones, calculated around the wing flying shape, and can be evaluated using the current framework with a single iteration of the adjoint solver ($N_{ADJ} = 1$ in Algorithm 2).

For the current application, Figure 20 depicts the comparison between the ASWSO and the intermediate optimization approach in terms of drag coefficient evolution versus optimization iterations, for a fixed computational time. The ASWSO reaches the minimum drag coefficient after 6 iterations, and the relative computational time is taken as reference. In the case of the intermediate approach, less accurate gradients evaluation results in a lower convergence rate of the objective function, providing, for the same run time (equivalent to approximately 8 iterations), a drag coefficient 0.43% higher than the ASWSO optimum. It is important to stress out that such intermediate approach is less robust and can compromise convergence of the optimization process due to the inconsistency between the primal and the dual problems.

4.3.5 Speed, scalability and memory usage

In this section basic benchmarks are given for the discussed ASWSO. For this particular application, 10 primal FSI iterations ($N_{FSI} = 10$ in Algorithm 1) deliver an OF converged up to 5 significant digits while 9 adjoint FSI iterations ($N_{ADJ} = 9$ in Algorithm 2) deliver sensitivities converged up to 4 significant digits, which is a common value for many optimization applications.[27] For the primal and the relative adjoint problems, CFD density residual tolerance is set to 10^{-10}. Table 11 shows the speed and scalability of the primal and adjoint computations. Benchmarks have been performed on Majorana, a one node machine equipped with two Intel Xeon E5-2697 v4 and 128 GB of memory. It is underlined how, for the current implementation, SU2 performs flow/mesh deformation problems in parallel while pyBeam works in single core, both in the primal and the adjoint problems execution. It can be appreciated how, the adjoint run takes always less than the relative primal one. Considering the number of DVs for this application (see Table 10), advantages in terms of sensitivities evaluation time are evident if comparing AD-based adjoint with other strategies such as FD. In terms of memory usage, the primal problem has a memory peak of 9 GB, which coincides with the flow solution stage, whereas the adjoint solution takes 57.9 GB of memory during the solution of the mesh deforming/flow adjoint problems. The large memory peak is due to the computational path recording operated by CoDiPack which, for the current implementation, occurs simultaneously for the mesh deformation and the flow adjoint problems solution.

Table 11 Wall-clock run time for the primal and adjoint computations with increasing number of CPU treads. CFD mesh has 1,529,927 volume elements.

Treads	Primal run time [s]	Adjoint run time [s]	Adjoint/Primal
8	9480	3499	0.37
16	6077	2369	0.39
32	4247	1912	0.45
64	4766	3062	0.64

4.4 RANS-SA-based optimization of the SCRM

One last optimization is performed for the SCRM aeroelastic test case considering a RANS-SA-based flow model for CFD: this counts as the highest-fidelity optimization performed within this effort. A Reynolds number of 40 millions is used in standard air conditions; Sutherland viscosity model is employed.

With respect to the previous test case of Section 4.3, some changes have been performed to reduce the overall computational effort of the optimization: the synthetic Young Modulus is 20% larger and FFD box Bezier functions are of order 4, 9, 1 chord-wise, span-wise and along the thickness respectively. DVs are the vertical positions of all FFD-box CPs, for a total number of 100. CPs on the symmetry plane are kept fixed as an effective way to prevent change in the shape of the relative airfoil.

Aerodynamic and geometric constraints are the same as for the test case in Section 4.3. To solve the fluid primal and relative adjoint problem, the non-frozen-turbulent approach is used.[3,41]

Figure 21 depicts the drag coefficient evolution versus optimization iterations for an ASWSO on test case. A C_D reduction of 17.97% with respect to the baseline configuration

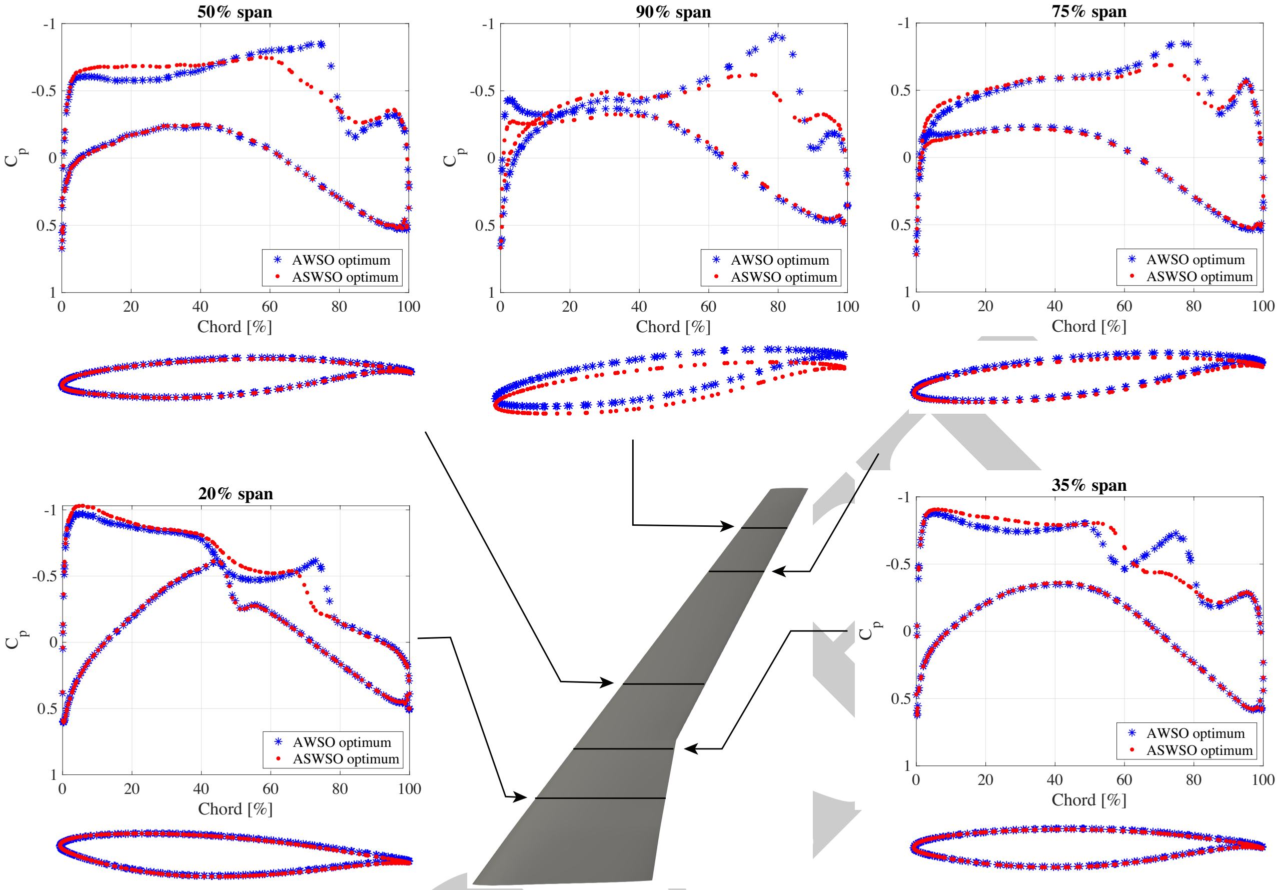


Fig. 19 Comparison between AWSO and ASWSO optima in flight condition for the Euler-based SCRM M6 test case. C_p distributions and airfoil shapes on selected stations.

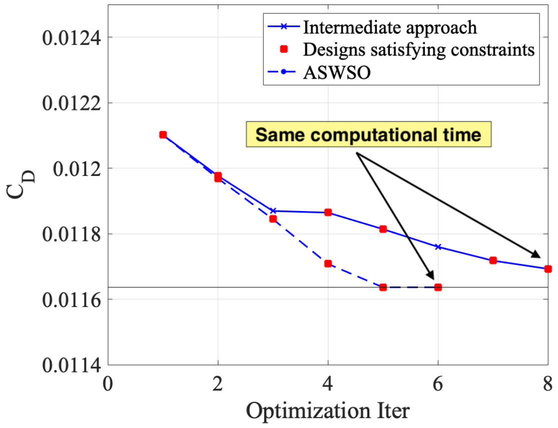


Fig. 20 Comparison between the C_D reduction of the ASWSO and the intermediate optimization approach for the Euler-based SCRM test case.

(in its relative flying shape) is obtained. Figure 22 depicts the C_p distribution on baseline (left) and optimal (right) designs.

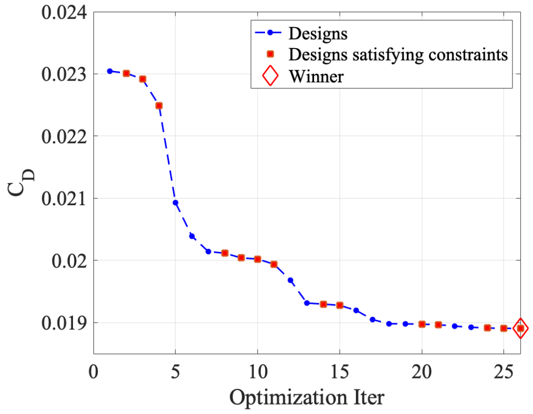


Fig. 21 C_D reduction for RANS-SA-based ASWSO of the SCRM wing.

It can be noted the shock alleviation in correspondence of the wing kink and its redistribution close to the wing tip. Moreover, the flying shape of the optimized configuration has a higher wing-tip deflection with respect to the baseline one, showing an opposite trend with respect to the one observed in the Euler case (see Figure 18).

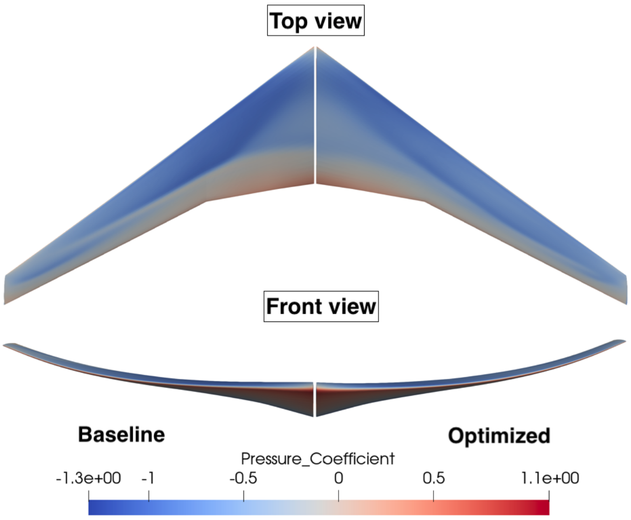


Fig. 22 RANS-SA-based ASWSO of the SCRM wing: C_p distribution on the baseline and the optimized designs at aeroelastic equilibrium.

5 Conclusions and future works

In this work a method for high-fidelity gradient-based aerostructural optimization of wings is presented, assisted by algorithmic differentiation and including aerodynamic and structural nonlinearities. The proposed approach is modular: each one of the single discipline solvers is interfaced at high level through a Python wrapper to solve the static aeroelastic equilibrium (primal problem). Moreover, each solver has implemented its own adjoint capability, employing algorithmic differentiation and levering on a fixed-point formulation, hence, allowing the evaluation of discrete-consistent coupled aerostructural sensitivities to be used in gradient-based optimization.

For the fluid problem the solver is the CFD module of SU2, whereas for the structural problem a nonlinear beam FE solver, embedding AD library CodiPack at native level, has been ad-hoc developed to demonstrate the approach. An interface module provides loads/displacements transfer between the two solvers.

Capability of the method is demonstrated performing aerostructural wing shape optimization on aeroelastic test cases of potential industrial interest, based on the ONERA M6 and CRM wings. Geometrical nonlinearities are always taken into account, and different levels of fidelity are employed at aerodynamic level (Euler and RANS-SA flow models).

Results of numerical optimization campaign show interesting trends, all highlighting the relevance of considering aerostructural coupling. Wings optimized neglecting such coupling, i.e., optimized not considering the deflection of the wing, perform relatively worse when considered in their actual flying shape configuration. For one test case it is even found that optimization carried out neglecting the aerostructural coupling leads to wings with lower performances with respect to the ones of the initial nonoptimized baseline, when both are considered in their relative flying shapes. Such result strongly warns against the practice of performing aerodynamic shape optimization without considering flexibility of the structure. An intermediate approach is also explored, in which the aerostructural coupling is considered at the primal solver level only. Comparison with the fully-coupled approach is discussed.

An optimization performed considering RANS-SA turbulence model is then shown. The current AD implementation easily allowed the adoption of non-frozen turbulence assumption both in the primal and the adjoint execution and optimization of the wing shape delivered a noticeable drag coefficient reduction.

Basic benchmarks of the implementation, in terms of speed, scalability and memory usage are also shown.

The optimization framework is released as open-source within the SU2 multiphysics suite in order to provide easy access to an aerostructural optimization tool (and relative primal aerostructural solver) to a potentially large audience. Thanks to the modular approach, users can easily experiment with different discipline modules.

Concerning future works, the framework can be expanded, with little effort, to introduce structural objective functions and design variables, compatibly with the discipline solver capabilities. Moreover, revision of the AD operational scheme of the current framework may allow to achieve a reduced RAM footprint during the code registration process performed by CoDiPack, for computationally intensive cases featuring large meshes and higher-fidelity flow models (RANS or higher).

Replication of results

The employed framework is currently on GitHub in the branch *feature_pyBeam_ShapeDesignV2* of SU2 repository and will soon be available in the official release of the suite.

PyBeam organization on GitHub provides the complete set of test cases discussed above in the repository *SAMO_testcases*.

Declaration

Conflicts of interest/Competing interests The authors declare that they have no conflict of interest.

Acknowledgments Part of the simulations were executed on the high performance cluster "Elwetritsch" at TU Kaiserslautern, which is part of the Alliance for High Performance Computing in Rhineland-Palatinate (AHRP). The authors would like to thank Dr. Beckett Y. Zhou and Guillermo Suàrez of the Chair for Scientific Computing of TU Kaiserslautern for their assistance.

References

1. Albring, T., Sagebaum, M., Gauger, N.R.: Development of a consistent discrete adjoint solver in an evolving aerodynamic design framework. 16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (2015)
2. Albring, T., Sagebaum, M., Gauger, N.R.: Efficient aerodynamic design using the discrete adjoint method in su2. In: 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (2016)
3. Barcelos, M., Maute, K.: Aeroelastic design optimization for laminar and turbulent flows. Computer Methods in Applied Mechanics and Engineering **197**(19-20), 1813–1832 (2008)
4. Beazley, D.M.: Swig: An easy to use tool for integrating scripting languages with c and c++. In: Proceedings of the 4th Conference on USENIX Tcl/Tk Workshop, 1996 - Volume 4, TCLTK'96, pp. 15–15. USENIX Association, Berkeley, CA, USA (1996)
5. Belytschko, T., Liu, W., Moran, B.: Nonlinear finite elements for continua and structures. Wiley (2000)
6. Bombardieri, R., Cavallaro, R., Luis Sáez de Teresa, J., Karpel, M.: Nonlinear aeroelasticity: a cfd-based adaptive methodology for flutter prediction. AIAA 2019-1866. AIAA Scitech 2019 Forum, San Diego, California (2019)
7. Bombardieri, R., Sanchez, R., Cavallaro, R., Gauger, N.R.: Towards an open-source framework for aero-structural design and optimization within the su2 suite. In: A. Gaspar-Cunha, J. Periaux, K. Giannakoglou, N. Gauger, D. Quagliarella, D. Greiner (eds.) Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences, chap. 19. Springer (2021)
8. Brezillon, J., Ronzheimer, A., Haar, D., Abu-Zurayk, M., Lummer, M., Krüger, W., Natterer, F.J.: Development and application of multi-disciplinary optimization capabilities based on high-fidelity methods. In: 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 2012 (2012)
9. Brooks, T.R., Kenway, G.K.W., Martins, J.R.R.A.: Benchmark aerostructural models for the study of transonic aircraft wings. AIAA Journal **56**(7), 2840–2855 (2018)
10. Cambier, L., Gazaix, M.: Elsa: An efficient object-oriented solution to cfd complexity. In: 40th AIAA Aerospace Sciences Meeting and Exhibit (2002)
11. Castro Lòpez, E.: Aerodynamic shape optimization of wings in transonic regime using discrete adjoint method. B.Sc. thesis uc3m (2020)
12. Cavallaro, R., Bombardieri, R., Demasi, L., Iannelli, A.: Prandtlplane Joined Wing: Body freedom flutter, limit cycle oscillation and freeplay studies. Journal of Fluids and Structures **59**, 57–84 (2015)
13. Cavallaro, R., Iannelli, A., Demasi, L., Razón, A.M.: Phenomenology of nonlinear aeroelastic responses of highly deformable Joined Wings. Advances in Aircraft and Spacecraft Science **2**(2), 125–168 (2015)
14. Cramer, E.J., Dennis Jr., J.E., Frank, P.D., Lewis, R.M., Shubin, G.R.: Problem formulation for multidisciplinary optimization. SIAM Journal on Optimization **4**(4), 754–776 (1994)
15. Degroote, J., Bathe, K.J., Vierendeels, J.: Performance of a new partitioned procedure versus a monolithic procedure in fluid–structure interaction. Computers & Structures **87**(11–12), 793 – 801 (2009). Fifth MIT Conference on Computational Fluid and Solid Mechanics
16. Dener, A., Hicken, J.E., Kenway, G.K., Martins, J.R.R.A.: Enabling modular aerostructural optimization: Individual discipline feasible without the jacobians
17. Donea, J., Giuliani, S., Halleux, J.: An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions. Computer Methods in Applied Mechanics and Engineering **33**(1), 689 – 723 (1982)
18. Donea, J., Huerta, A., Ponthot, J.P., Rodríguez-Ferran, A.: Arbitrary Lagrangian–Eulerian Methods. John Wiley & Sons, Ltd (2004)
19. Dwight, R.: Robust mesh deformation using the linear elasticity equations. pp. 401–406. Springer Berlin, Ghent (2009)
20. Economon, T.D., Palacios, F., Copeland, S.R., Lukaczyk, T.W., Alonso, J.J.: Su2: An open-source suite for multiphysics simulation and design. AIAA Journal **Vol. 54**(No. 3), pp. 828–846 (2016)
21. Farhat, C., Geuzaine, P., Brown, G.: Application of a three-field nonlinear fluid–structure formulation to the prediction of the aeroelastic parameters of an f-16 fighter. Computers & Fluids **32** (2003)
22. Ghazlane, I., Carrier, G., Dumont, A., antoine Desideri, J.: Aerostructural Adjoint Method for Flexible Wing Optimization
23. Gori, G., Vimercati, D., Guardone, A.: Non-ideal compressible-fluid effects in oblique shock waves. Journal of Physics: Conference Series **821**(1), 012003 (2017)
24. Grossman, B., Gurdal, Z., Haftka, R.: Integrated aerodynamic/structural design of a sailplane wing. In: Aircraft Systems, Design and Technology Meeting. American Institute of Aeronautics and Astronautics (1986)
25. Haftka, R.T.: Optimization of flexible wing structures subject to strength and induced drag constraints. AIAA Journal **15**, 1101–1106 (1977)
26. He, P., Mader, C.A., Martins, J.R.R.A., Maki, K.: An Object-oriented Framework for Rapid Discrete Adjoint Development using OpenFOAM
27. He, P., Mader, C.A., Martins, J.R.R.A., Maki, K.J.: Dafoam: An open-source adjoint framework for multidisciplinary design optimization with openfoam. AIAA Journal **58**(3), 1304–1319 (2020)
28. Hoogervorst, J.E., Elham, A.: Wing aerostructural optimization using the individual discipline feasible architecture. Aerospace Science and Technology **65**, 90–99 (2017)
29. Irons, B.M., Tuck, R.C.: A version of the aitken accelerator for computer iteration. International Journal for Numerical Methods in Engineering **1**(3), 275–277 (1969)
30. Kennedy, G., Martins, J.: Parallel Solution Methods for Aerostructural Analysis and Design Optimization
31. Kennedy, G.J., Kenway, G.K., Martins, J.R.R.A.: A comparison of metallic, composite and nanocomposite optimal transonic transport wings. Tech. rep., NASA (2014). CR-2014-218185
32. Kenway, G., Kennedy, G., Martins, J.: Aerostructural optimization of the common research model configuration. In: 15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (2014)
33. Kenway, G., Kennedy, G., Martins, J.: Scalable parallel approach for high-fidelity steady-state aeroelastic analysis and adjoint derivative computations. AIAA Journal **52**(5), 935–951 (2014)

34. Kenway, G.K., Kennedy, G., Martins, J.: High aspect ratio wing design: Optimal aerostructural tradeoffs for the next generation of materials. In: AIAA SciTech Forum 2014, 52nd Aerospace Sciences Meeting (2014)
35. Kenway, G.K.W., Martins, J.R.R.A.: Multipoint high-fidelity aerostructural optimization of a transport aircraft configuration. Journal of Aircraft **51**(1), 144–160 (2014)
36. Kiviaho, J.F., Jacobson, K., Smith, M.J., Kennedy, G.: A robust and flexible coupling framework for aeroelastic analysis and optimization. 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (2017)
37. Korivi, V.M., Taylor, A.C., Newman, P.A., Hou, G., Jones, H.E.: An incremental strategy for calculating consistent discrete cfd sensitivity derivatives (1992)
38. Kraft, D.: A Software Package for Sequential Quadratic Programming. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht. Wiss. Berichtswesen d. DFVLR (1988)
39. Levy, R., Spillers, W.: Analysis of geometrically nonlinear structures. v. 1. Kluwer Academic Publishers, Dordrecht, Netherlands (2003)
40. Liem, R.P., Martins, J.R.R.A., Kenway, G.K.W.: Expected drag minimization for aerodynamic design optimization based on aircraft operational data. Aerospace Science and Technology **63**, 344–362 (2017)
41. Lyu, Z., Kenway, G.K.W., Martins, J.R.R.A.: Aerodynamic shape optimization investigations of the common research model wing benchmark. AIAA Journal **53**, 968–985 (2015)
42. Martins, J., Alonso, J., Reuther, J.: Aero-structural wing design optimization using high-fidelity sensitivity analysis. In: Proceedings - CEAS Conference on Multidisciplinary Aircraft Design Optimization. Cologne, Germany, (2001)
43. Martins, J.R.R.A., Alonso, J.J., Reuther, J.J.: High-fidelity aerostructural design optimization of a supersonic business jet. Journal of Aircraft **41**, 523–530 (2004)
44. Martins, J.R.R.A., Hwang, J.T.: Review and unification of methods for computing derivatives of multidisciplinary computational models. AIAA Journal **51**(11), 2582–2599 (2013)
45. Maute, K., Nikbay, M., Farhat, C.: Coupled analytical sensitivity analysis and optimization of three-dimensional nonlinear aeroelastic systems. AIAA Journal **39**(11), 2051–2061 (2001)
46. Maute, K., Nikbay, M., Farhat, C.: Sensitivity analysis and design optimization of three-dimensional non-linear aeroelastic systems by the adjoint method. International Journal for Numerical Methods in Engineering **56**(6), 911–933 (2003)
47. Mishra, A., Mani, K., Mavriplis, D., Sitaraman, J.: Time dependent adjoint-based optimization for coupled fluid–structure problems. Journal of Computational Physics **292** (2015)
48. Mishra, A., Mavriplis, D., Sitaraman, J.: Time-dependent aeroelastic adjoint-based aerodynamic shape optimization of helicopter rotors in forward flight. AIAA Journal (2016)
49. Molina, E.S., Spode, C., Da Silva, R.G.A., Manosalvas-Kjono, D.E., Nimmagadda, S., Economon, T.D., Alonso, J.J., Righi, M.: Hybrid rans/les calculations in su2. In: 23rd AIAA Computational Fluid Dynamics Conference, 2017 (2017)
50. Mount M., D., Arya, S.: Ann: A library for approximate nearest neighbor searching (2010). URL http://www.cs.umd.edu/ mount/ANN/
51. NASA: Nasa common research model (2008). URL https://commonresearchmodel.larc.nasa.gov
52. Palacios, F., Alonso, J., Duraisamy, K., Colonno, M., Hicken, J., Aranake, A., Campos, A., Copeland, S., Economon, T., Lonkar, A., Lukaczyk, T., Taylor, T.: Stanford University Unstructured (SU^2): An open-source integrated computational environment for multiphysics simulation and design. In: 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition. American Institute of Aeronautics and Astronautics (2013)
53. Palacios, F., Economon, T.D., Aranake, A., Copeland, S.R., Lonkar, A.K., Lukaczyk, T.W., Manosalvas, D.E., Naik, K.R., Padron, S., Tracey, B., Variyar, A., Alonso, J.J.: Stanford University Unstructured (SU2): Analysis and Design Technology for Turbulent Flows. In: 52nd Aerospace Sciences Meeting. American Institute of Aeronautics and Astronautics (2014)
54. Palacios, F., Economon, T.D., Wendorff, A.D., Alonso, J.J.: Large-scale aircraft design using su2. In: 53rd AIAA Aerospace Sciences Meeting (2015)
55. Peter, J.E., Dwight, R.P.: Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches. Computers & Fluids **39**(3), 373 – 391 (2010)
56. Pini, M., Vitale, S., Colonna, P., Gori, G., Guardone, A., Economon, T., Alonso, J., Palacios, F.: Su2: the open-source software for non-ideal compressible flows. Journal of Physics: Conference Series **821**(1), 012013 (2017)
57. Pustina, L., Cavallaro, R., Bernardini, G.: Nerone: An open-source based tool for aerodynamic transonic optimization of nonplanar wings. Aerotec. Missili Spaz. **98**, 85–104 (2019)
58. Quaranta, G., Masarati, P., Mantegazza, P.: A conservative mesh-free approach for fluid structure problems in coupled problems. In: International Conference for Coupled Problems in Science and Engineering, Santorini, Greece, pp. 24–27 (2005)
59. Romanelli Giulio; Castellani, M., Mantegazza, P., Ricci, S.: Coupled csd/cfd non-linear aeroelastic trim of free-flying flexible aircraft. In: 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference20th AIAA/ASME/AHS Adaptive Structures Conference14th AIAA - Honolulu, Hawaii. (2012)
60. Sagebaum, M., Albring, T., Gauger, N.R.: High-performance derivative computations using codipack. arXiv preprint arXiv:1709.07229 (2017)
61. Samareh, J.: Aerodynamic shape optimization based on free-form deformation. AIAA 2004-4630. 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (2014)
62. Sanchez, R.: Coupled adjoint-based sensitivities in large-displacement fluid-structure interaction using algorithmic differentiation. Ph.D. thesis, Imperial College London (2018)
63. Sanchez, R., Albring, T., Palacios, R., Gauger, N.R., Economon, T.D., Alonso, J.J.: Coupled adjoint-based sensitivities in large-displacement fluid-structure interaction using algorithmic differentiation. International Journal for Numerical Methods in Engineering **113**(7), 1081–1107 (2018)
64. Sanchez, R., Kline, H., Thomas, D., Variyar, A., Righi, M., Economon, T., Alonso, J., Palacios, R., Dimitriadis, G., Terrapon, V.: Assessment of the Fluid-Structure Interaction Capabilities for Aeronautical Applications of the Open-Source Solver SU2. In: VII European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2016). Crete Island, Greece, 5-10 June (2016)
65. Spalart, P., Allmaras, S.: A one-equation turbulence model for aerodynamic flows
66. Luis Sàez de Teresa, J.: Towards correction of aerodynamic coefficients for aeroelastic calculations with open-source su2 software. M.Sc. thesis uc3m (2019)
67. Vassberg, J., Dehaan, M., Rivers, M., Wahls, R.: Development of a common research model for applied cfd validation studies. In: American Institute of Aeronautics and Astronautics 26th AIAA Applied Aerodynamics Conference - Honolulu, Hawaii (2008)
68. Vela Peña, J.: Aeroelastic calculations on an equivalent beam-based nasa crm. B.Sc. thesis uc3m (2019)
69. Wang, L., Diskin, B., Biedron, R.T., Nielsen, E.J., Bauchau, O.A.: High-fidelity multidisciplinary sensitivity analysis and design optimization for rotorcraft applications. AIAA Journal (2019)
70. van der Weide, E., Kalitzin, G., Schluter, J., Alonso, J.: Unsteady Turbomachinery Computations Using Massively Parallel Platforms

71. White, F.M.: Viscous Fluid Flow. McGraw–Hill, New York (1974)
72. Wilcox, D.: Turbulence Modeling for CFD. DCW Industries, Inc. (1998)
73. Zhou, B.Y., Albring, T., Gauger, N.R., Ilario, C., Economon, T., Alonso, J.J.: Reduction of airframe noise components using a discrete adjoint approach. AIAA 2017-3658 (2017)