A discrete adjoint full potential formulation for fast aerostructural optimization in preliminary aircraft design

Adrien Crovato[a,*], Alex P. Prado[b], Pedro H. Cabral[b], Romain Boman[a], Vincent E. Terrapon[a], Grigorios Dimitriadis[a]

[a]*Department of Aerospace and Mechanical Engineering - University of Liège, 4000 Liège, Belgium*
[b]*Embraer S.A., São José dos Campos 12227-901, Brazil*

Abstract

Preliminary aircraft design is often carried out with the help of multi-disciplinary optimization processes which, in order to be effective and to take full advantage of composite materials, must model the aeroelastic behavior of the aircraft. Since many design variables are involved during the preliminary design stage, the optimization problem is usually solved using the adjoint method. Moreover, the level of fidelity of the fluid model and the associated simulation technique must also be selected with care, as they are the main contributors to the computational cost. The goal of the present work is twofold. Firstly, the discrete gradients of the full potential equation, a medium-fidelity model, are derived analytically so that they can be used in adjoint optimization problems. Moreover, the full potential solution and the computation of the gradients are implemented in an open-source and readily available finite element code. Secondly, aerodynamic shape and aerostructural optimization calculations are carried out on example wings to demonstrate the effectiveness and the computational efficiency of the proposed method. Overall, the results show that the newly implemented discrete adjoint nonlinear potential formulation is able to quickly optimize both the shape and the structural parameters of a typical wing. The next steps consist in improving the problem formulation and the model, by considering full aircraft configurations, along with more realistic design variables, constraints and objective function.

Keywords: Discrete adjoint, Transonic flows, Full potential, Aerostructural optimization, Preliminary aircraft design

1. Introduction

For the air transport sector to remain economically competitive and to keep its rather low environmental impact, aircraft fuel consumption must be reduced further. To this end, various strategies are being investigated, such as increasing aircraft engine efficiency, using new types of fuel, improving aircraft operations and air services, reducing structural weight, improving aerodynamic efficiency, etc. The combination of the last two approaches leads to the design of light, hence flexible, highly loaded wings, which requires suitable modeling of the aerostructural behavior of the aircraft. Aerodynamic shape optimization is therefore usually performed in conjunction with

*Corresponding author
Email addresses: a.crovato@uliege.be (Adrien Crovato), alex.prado@embraer.com.br (Alex P. Prado), pedro.cabral@embraer.com.br (Pedro H. Cabral), r.boman@uliege.be (Romain Boman), vincent.terrapon@uliege.be (Vincent E. Terrapon), gdimitriadis@uliege.be (Grigorios Dimitriadis)

aeroelastic tailoring, which was initially defined by Shirk and Hertz in 1986 as "*the embodiment of directional stiffness into an aircraft structural design to control aeroelastic deformation, static or dynamic, in such a fashion as to affect the aerodynamic and structural performance of that aircraft in a beneficial way*" [1]. Such an aerostructural optimization process enables the design of light wings with efficient shapes, exploiting the orthotropic mechanical properties of composite materials, that guarantee good aerodynamic performance and are able to withstand the aerodynamic loads and the aeroelastic instabilities.

Optimization methods fall into two categories: gradient-based or gradient-free. In the context of aircraft design, the former method has traditionally been favored over the latter, since the initial design is already quite close to the final design, hence not requiring a full exploration of the design space [2, 3]. Gradient-based optimization can be performed using finite differences or complex-step methods, or by resorting to the direct or adjoint formulations, as explained by Martins and Hwang [4, 5]. For the finite difference and the complex-step methods, the nonlinear aerostructural equations must be solved each time the total gradient with respect to a design variable is required. For the direct and adjoint formulations, the total gradients are computed by assembling the partial gradients and using the chain rule. If the direct formulation is used, a linear set of equations must be solved each time the total gradient with respect to a design variable is required. On the other hand, the adjoint formulation requires the solution of a linear set of equations once per response function evaluation. The optimization formulation of modern aircraft usually consists of a few functionals, like the fuel burn and the failure index, and thousands of design variables, such as the wing shape and the structural parameters. Consequently, the adjoint formulation is usually favored, as using the other methods would lead to a prohibitive computational cost [6, 7, 8, 9, 10, 11]. Examples of multidisciplinary optimizations for aircraft design carried out using the adjoint method can be found in several works [12, 13, 14, 15, 16, 17].

Aerostructural optimization is nowadays performed during preliminary aircraft design. Introducing such calculations in this early stage allows to directly consider complex fluid-structure interactions, and might prevent future failures resulting from otherwise unaccounted for aeroelastic behaviors, hence helping to make the design process more robust. In order to decrease the computational cost, low-fidelity linear modeling methodologies are usually favored [18]. The range of validity of these models is theoretically restricted to the subsonic regime. However, modern transport aircraft fly in the transonic regime, where nonlinear compressible effects become important and cannot be neglected. During a recent joint work with Embraer S.A. [1], the authors investigated the effect of various aerodynamic levels of fidelity on aeroelastic computations [19]. Based on the trade-off between accuracy and computational cost of the different models, the steady full potential equation was selected, and a finite element code was subsequently implemented to rapidly compute flow solutions [20, 21]. The objective of the present work is twofold. The flow solver is first enhanced by developing a discrete adjoint formulation allowing to quickly compute the gradients, and is integrated in a multi-disciplinary optimization framework. The methodology is then

[1] https://embraer.com/, accessed March 2022.

demonstrated on simple benchmark cases representative of analysis and optimization computations performed during preliminary aircraft design.

Several authors already developed adjoint formulations for the full potential equation. In 1983, Angrand [22] used the finite element method to discretize both the linear and nonlinear potential equations, and the adjoint method to formulate optimization problems. He used these developments to optimize the shapes of a nozzle and of an airfoil in incompressible and compressible flows. Later, Jameson and Reuther [23, 24, 25] formulated aerodynamic shape optimization problems using the adjoint method and a finite volume discretization of the full potential equation. The methodology was then implemented in the computer code `FLO36` and subsequently used to optimize the shape of airfoils parameterized using a conformal mapping technique, in order to reduce the wave drag produced through shocks in transonic flows. In 2000, Santos [26] used a similar formulation to carry out inverse design problems and wave drag minimization calculations using the parametrization of the NACA 4-digit airfoil series. More recently, Galbraith et al. [27] developed a finite element method based on a two-field formulation of the full potential equation, originally developed by Parrinello and Mantegazza [28, 29]. They then implemented the methodology in the `SANS` package [30] [2], which uses automatic differentiation to compute the partial gradients of the flow, and applied the adjoint method to recover the total gradients with respect to the angle of attack. Nevertheless, few full potential codes have been made widely available to date. Notable examples include the research code developed by Davari et al. [31] and Núñez et al. [32] in `Kratos Multiphysics` [33] [3], and the commercial package `TRANAIR` [34]. However, the former does not include sensitivity calculations, and the design and optimization features available in `TRANAIR` are based on the direct method, whereby the partial gradients are computed using finite differences [35, 36, 37, 38]. Moreover, to the best of the authors knowledge, the adjoint method has never been used together with the full potential equation to solve aerostructural optimization problems in preliminary aircraft design. The novelty of this work resides on two main points. Firstly, the analytical expressions of the discrete full potential gradients are derived and presented in the manuscript, so that they can be used in adjoint multi-disciplinary optimization problems. The full potential solution and the gradients computation are implemented in an open-source finite element code. The code is readily available and has a modular high-level interface so that it can be used within state-of-the-art optimization packages. Secondly, the effectiveness and the computational efficiency of the proposed full potential adjoint formulation are demonstrated by carrying out aerodynamic shape and aerostructural optimization computations on two benchmark wings.

The present manuscript is organized as follows. In Section 2, the aerostructural optimization problem is formulated using the adjoint method. Section 3 presents a discrete formulation of the full potential equation based on the finite element method. More specifically, the aerodynamic and mesh deformation residuals and functionals, as well as their partial derivatives, are derived analytically. The main aspects of the implementation are also highlighted. Two validation cases consisting of aerodynamic shape and aerostructural optimization calculations are

[2] https://darmofal.mit.edu/software, accessed March 2022.
[3] https://github.com/KratosMultiphysics/Kratos, accessed March 2022.

subsequently presented in Section 4. Finally, the last section summarizes and concludes the present manuscript, and suggests research directions for future work.

2. Formulation of the optimization problem

In the present work, the aerostructural optimization problem is formulated using the three-field formulation, initially developed by Farhat et al. [39] and later improved by Maute et al. [40]. This formulation is required when nonlinear aerodynamic models are used, since the fluid equations must be solved in the volume surrounding the body, which will deform according to the shape of the body. Furthermore, if the body undergoes large displacements, the contributions of the grid motion cannot be neglected. Figure 1 depicts typical structural and aerodynamic grids used to perform aerostructural analysis and optimization. The problem is formulated as follows,

$$\begin{aligned} \min_{\mathbf{p}} \; & F_{\text{obj}}(\mathbf{u}, \mathbf{v}, \mathbf{x}; \mathbf{p}) \\ \text{s.t. } & \mathbf{R}_{\text{u}} = 0 \\ & \mathbf{R}_{\text{v}} = 0 \\ & \mathbf{R}_{\text{x}} = 0 \\ & \mathbf{C}(\mathbf{u}, \mathbf{v}, \mathbf{x}; \mathbf{p}) = 0, \end{aligned} \tag{1}$$

where $\mathbf{u}$ and $\mathbf{v}$ denote the vector of structural and aerodynamic variables, $\mathbf{x}$ is the vector of volume mesh coordinates, $\mathbf{p}$ is the vector of design variables, $\mathbf{R}_{\text{u}}$ and $\mathbf{R}_{\text{v}}$ represent the structural and aerodynamic equations noted in residual form, $\mathbf{R}_{\text{x}}$ is the vector of residuals of the mesh morphing laws, and $F_{\text{obj}}(\mathbf{u}, \mathbf{v}, \mathbf{x}, \mathbf{p})$ is the functional to be minimized under the constraints $\mathbf{C}(\mathbf{u}, \mathbf{v}, \mathbf{x}, \mathbf{p})$. In order to minimize F_{obj}, the augmented Lagrangian $\mathcal{L}$ is first constructed as,

$$\mathcal{L} = F_{\text{obj}} + \lambda_{\text{u}}\mathbf{R}_{\text{u}} + \lambda_{\text{v}}\mathbf{R}_{\text{v}} + \lambda_{\text{x}}\mathbf{R}_{\text{x}}, \tag{2}$$

where λ_{u}, λ_{v} and λ_{x} are Lagrange multipliers, and then differentiated such that,

$$\delta\mathcal{L} = 0 \Rightarrow \begin{cases} \frac{\partial F_{\text{obj}}}{\partial \mathbf{u}} + \lambda_{\text{u}}\frac{\partial \mathbf{R}_{\text{u}}}{\partial \mathbf{u}} + \lambda_{\text{v}}\frac{\partial \mathbf{R}_{\text{v}}}{\partial \mathbf{u}} + \lambda_{\text{x}}\frac{\partial \mathbf{R}_{\text{x}}}{\partial \mathbf{u}} = 0 \\ \frac{\partial F_{\text{obj}}}{\partial \mathbf{v}} + \lambda_{\text{u}}\frac{\partial \mathbf{R}_{\text{u}}}{\partial \mathbf{v}} + \lambda_{\text{v}}\frac{\partial \mathbf{R}_{\text{v}}}{\partial \mathbf{v}} + \lambda_{\text{x}}\frac{\partial \mathbf{R}_{\text{x}}}{\partial \mathbf{v}} = 0 \\ \frac{\partial F_{\text{obj}}}{\partial \mathbf{x}} + \lambda_{\text{u}}\frac{\partial \mathbf{R}_{\text{u}}}{\partial \mathbf{x}} + \lambda_{\text{v}}\frac{\partial \mathbf{R}_{\text{v}}}{\partial \mathbf{x}} + \lambda_{\text{x}}\frac{\partial \mathbf{R}_{\text{x}}}{\partial \mathbf{x}} = 0 \\ \frac{\partial F_{\text{obj}}}{\partial \mathbf{p}} + \lambda_{\text{u}}\frac{\partial \mathbf{R}_{\text{u}}}{\partial \mathbf{p}} + \lambda_{\text{v}}\frac{\partial \mathbf{R}_{\text{v}}}{\partial \mathbf{p}} + \lambda_{\text{x}}\frac{\partial \mathbf{R}_{\text{x}}}{\partial \mathbf{p}} = 0 \\ \mathbf{R}_{\text{u}} = 0 \\ \mathbf{R}_{\text{v}} = 0 \\ \mathbf{R}_{\text{x}} = 0 \end{cases} . \tag{3}$$

The fourth equation represents the total gradient of the functional F_{obj} with respect to the design variables $\mathbf{p}$. It can be computed by first solving the set of the last three nonlinear equations for the variables $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{x}$, then solving the set of the first three linear equations for the Lagrange multipliers, and finally by injecting those variables into the expression of the total gradient. An optimizer can then be used to drive that gradient to zero, hence finding the optimal values of the design variables. Note that the constraints $\mathbf{C}$ are handled separately from the adjoint formulation.

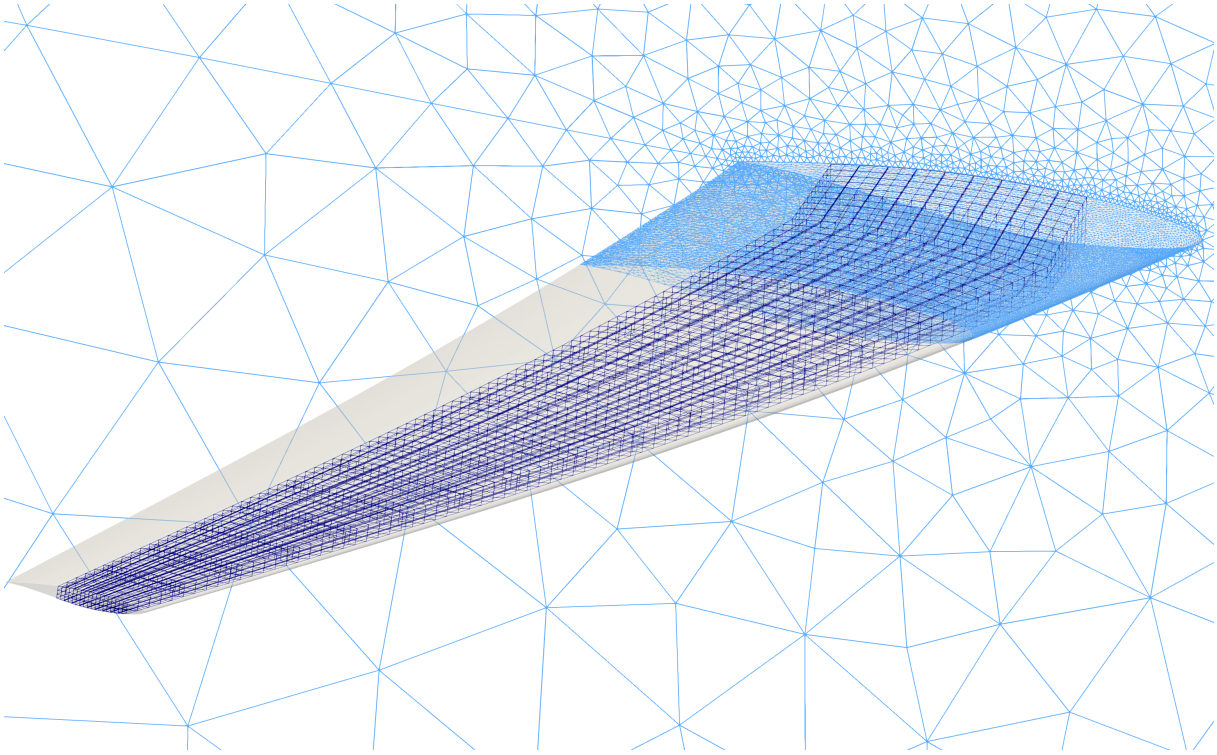


Figure 1: Typical structural and aerodynamic discretizations used to perform aerostructural analysis and optimization. The structural mesh, where $\mathbf{R}_\text{u}$ is solved, is depicted in dark blue. The aerodynamic grid, where $\mathbf{R}_\text{v}$ and $\mathbf{R}_\text{x}$ are solved, is depicted in light blue. Note that only a part of the aerodynamic mesh boundary is displayed.

3. Formulation of the aerodynamic model

3.1. Full potential equation

The steady full potential equation is derived form the Navier-Stokes equations by assuming that the fluid is inviscid and that the flow is steady, isentropic and irrotational, such that the velocity derives from a potential ϕ. Considering a domain Ω enclosed by a boundary surface Γ, split into a farfield boundary Γ_f, a body boundary Γ_b and a wake boundary Γ_w, as depicted in Figure 2, the full potential equation can be written in weak form as,

$$\int_\Omega \rho\nabla\phi\cdot\nabla\psi\, dV - \int_\Gamma \overline{\rho\nabla\phi}\cdot\hat{\mathbf{n}}\,\psi\, dS = 0, \qquad \forall\psi\in\Omega, \tag{4}$$

where ψ is a test function, $\hat{\mathbf{n}}$ is the unit vector normal to Γ pointing inwards, overlined variables denote quantities imposed through boundary conditions, and where the density ρ is given by the isentropic flow relationship,

$$\rho = \rho_\infty \left[1 + \frac{\gamma - 1}{2} M_\infty^2 \left(1 - |\nabla \phi|^2 \right) \right]^{\frac{1}{\gamma - 1}}. \tag{5}$$

In Equation 5, ρ_∞ is the freestream density, γ is the heat capacity ratio and M_∞ is the freestream Mach number. Note that the velocity $\nabla\phi$ is normalized by the freestream velocity. An important limitation of the nonlinear potential equation is the isentropicity assumption, which restricts its use to transonic flows with weak embedded shocks. A common upper limit for the local normal Mach number upstream of the shock is $M_n < 1.3$ [41].

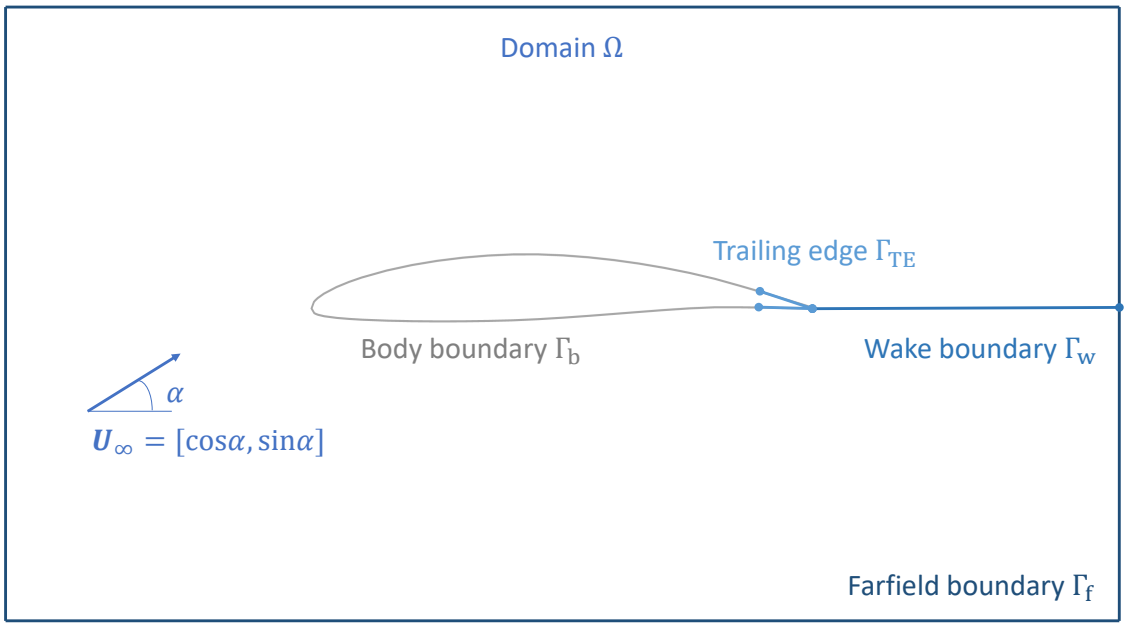


Figure 2: Typical domain used for a finite element computation, illustrated in two dimensions for simplicity.

Neumann boundary conditions are applied on the farfield boundary Γ_f, and on the body boundary Γ_b. Such boundary conditions impose a flux through the boundaries of the domain and are naturally recovered in the second term of the weak formulation of the full potential equation 4. Since the derivative of the potential is the velocity, the weak form of the Neumann boundary condition can be written as

$$\begin{aligned} \int_{\Gamma_f} \overline{\rho \nabla \phi} \cdot \hat{\mathbf{n}} \, \psi \, dS &= \int_{\Gamma_f} \rho_\infty \mathbf{U}_\infty \cdot \hat{\mathbf{n}} \, \psi \, dS, \qquad && \forall \psi \in \Gamma_f \\ \int_{\Gamma_b} \overline{\rho \nabla \phi} \cdot \hat{\mathbf{n}} \, \psi \, dS &= 0, && \forall \psi \in \Gamma_b \end{aligned} \tag{6}$$

where $\mathbf{U}_\infty$ is the freestream velocity vector given by,

$$\mathbf{U}_\infty = \begin{bmatrix} \cos\alpha\cos\beta \\ \sin\beta \\ \sin\alpha\cos\beta \end{bmatrix}, \tag{7}$$

and where α is the angle of attack and β is the angle of sideslip. Additionally, the Kutta condition needs to be enforced so that a physically meaningful solution is selected among all the possible solutions, and potential flows can produce aerodynamic loads acting on lifting bodies. This is accomplished by creating a wake sheet, denoted Γ_w, extending horizontally from the trailing edge of any lifting body to the farfield boundary located downstream of these bodies, as shown in Figure 2. The unknown potential value is discontinuous across the wake, and two boundary conditions are applied to restore the continuity in the flow variables. The first condition prescribes the equality of the mass flux across the wake,

$$\int_{\Gamma_w} [\![\rho\nabla\phi\cdot\hat{\mathbf{n}}]\!]\,\psi\,dS = 0, \qquad \forall\psi\in\Gamma_{w,l}, \tag{8}$$

and the second condition prescribes the equality of the pressure across the wake,

$$\int_{\Gamma_w} [\![|\nabla\phi|^2]\!]\,\Psi\,dS = 0, \qquad \forall\Psi\in\Gamma_{w,u}, \tag{9}$$

where Ψ is a test function stabilized using a Petrov-Galerkin formulation, the double square bracket indicates a jump between the quantities on the upper and lower sides of the wake, and the subscripts u and l refer to the upper and lower sides of the wake, respectively. Equations 8 and 9 are supplemented by a third condition that enforces the Kutta condition locally on the trailing edge Γ_{TE},

$$\int_{\Gamma_{TE,u}} \frac{1}{h^2}|\nabla\phi|^2\,\Psi\,dS - \int_{\Gamma_{TE,l}} \frac{1}{h^2}|\nabla\phi|^2\,\Psi\,dS = 0, \qquad \forall\Psi\in\Gamma_{TE,u}, \tag{10}$$

where h^2 denotes the trailing edge surface area covered by Γ_{TE} in Figure 2. More details about the Kutta condition can be found in Galbraith et al. [27] and in Crovato [21].

The physical behavior of the flow is reflected in the mathematical nature of the full potential equation: the equation is elliptic in subsonic regions of the flow, whereas it becomes hyperbolic in supersonic regions, as explained by Holst [42]. This change must be taken into account for the numerical method to be stable. Consequently, the physical density is upwinded in supersonic flow pockets, as proposed by Eberle [43] and Hafez et al. [44], and replaced by,

$$\tilde{\rho} = \rho - \mu(\rho - \rho_U), \tag{11}$$

where ρ_{U} is the density evaluated at an upwind point, and where the switching function is defined as

$$\mu = \mu_{\mathrm{C}} \max\left(0, 1 - \frac{M_{\mathrm{C}}^2}{M^2}\right). \tag{12}$$

The parameters μ_{C} and M_{C} are controlled by the numerical scheme. Additional details can be found in Crovato [21].

The total aerodynamic load vector coefficient is computed by integrating the pressure coefficient on the body surface,

$$\mathbf{C_F} = \frac{1}{S_{\mathrm{ref}}} \int_{\Gamma_{\mathrm{b}}} C_p \hat{\mathbf{n}}\, dS, \tag{13}$$

where S_{ref} is a reference area, and where the pressure coefficient is given by

$$C_p = \frac{2}{\gamma M_\infty^2} \left(\rho^\gamma - 1\right). \tag{14}$$

The total aerodynamic load is obtained by multiplying the aerodynamic load vector coefficient by the freestream dynamic pressure,

$$\mathbf{F} = \frac{1}{2} \rho_\infty |\mathbf{U}_\infty|^2 S_{\mathrm{ref}} \mathbf{C_F}. \tag{15}$$

The aerodynamic load coefficients are obtained by projecting $\mathbf{C_F}$ on the lift and drag directions, yielding

$$C_L = \mathbf{C_F} \cdot \mathbf{e_L}, \qquad C_D = \mathbf{C_F} \cdot \mathbf{e_D}, \tag{16}$$

where the directions are defined with respect to the angle of attack α, and the angle of sideslip β,

$$\mathbf{e_L} = \begin{bmatrix} -\sin\alpha \\ 0 \\ \cos\alpha \end{bmatrix}, \qquad \mathbf{e_D} = \begin{bmatrix} \cos\alpha\cos\beta \\ \sin\beta \\ \sin\alpha\cos\beta \end{bmatrix}. \tag{17}$$

3.1.1. Discretization

The domain Ω and its boundary Γ are discretized following a continuous Galerkin finite element methodology. An unstructured grid strategy is chosen in order to easily mesh three-dimensional complex shapes. The potential, test functions and coordinates are expressed using isoparametric linear tetrahedral and triangular finite elements in the volume and on the surfaces, respectively, as

$$\begin{aligned} \phi(\xi_k) &= N_i(\xi_k)\phi_i, \\ \psi(\xi_k) &= N_i(\xi_k)\psi_i, \\ x_k(\xi_k) &= N_i(\xi_k)x_{i,k} \end{aligned} \tag{18}$$

where N_i is the shape function of an element associated with its node i and interpolates the nodal values ϕ_i and ψ_i of the potential and the test functions, as well as the nodal coordinates $x_{i,k} = [x, y, z]_i$, on an element. The shape functions are expressed locally on each element as

$$N_i = N_i(\xi_k) = N_i\left([\xi, \eta, \zeta]\right), \tag{19}$$

where ξ_k is the vector of coordinates attached to the reference frame of an element and where k counts dimensions. The weak form of the full potential equation (4) must hold for any test function ψ. It can then be discretized and rewritten in residual form,

$$R_{\phi,i} = \sum_{\mathrm{e}} \int_{\Omega_{\mathrm{e}}} \tilde{\rho}_{\mathrm{e}} \nabla N_j \phi_j \cdot \nabla N_i \, dV_{\mathrm{e}} - \sum_{\mathrm{e}} \int_{\Gamma \mathrm{e}} \overline{\rho \nabla \phi}_{\mathrm{e}} \cdot \hat{\mathbf{n}}_{\mathrm{e}} N_i \, dS_{\mathrm{e}} = 0, \tag{20}$$

where $\sum_{\mathrm{e}}$ denotes a sum performed over all elements, Ω_{e} and Γ_{e} are the volume and the surface of an element, $\hat{\mathbf{n}}_{\mathrm{e}}$ is the unit normal vector to a surface element, and where $\tilde{\rho}_{\mathrm{e}}$ and $\overline{\rho \nabla \phi}_{\mathrm{e}}$ denote the upwinded density and the imposed flux evaluated on an element. The associated Neumann boundary conditions (6) become

$$\begin{aligned} \sum_{\mathrm{e}} \int_{\Gamma_{\mathrm{f_e}}} \overline{\rho \nabla \phi}_{\mathrm{e}} \cdot \hat{\mathbf{n}}_{\mathrm{e}} N_i \, dS_{\mathrm{e}} &= \sum_{\mathrm{e}} \int_{\Gamma_{\mathrm{f_e}}} \rho_\infty \mathbf{U}_\infty \cdot \hat{\mathbf{n}}_{\mathrm{e}} N_i \, dS_{\mathrm{e}}, \\ \sum_{\mathrm{e}} \int_{\Gamma_{\mathrm{b_e}}} \overline{\rho \nabla \phi}_{\mathrm{e}} \cdot \hat{\mathbf{n}}_{\mathrm{e}} N_i \, dS_{\mathrm{e}} &= 0. \end{aligned} \tag{21}$$

The equality of mass flux across the wake (8) is enforced on the lower wake nodes as

$$\sum_{\mathrm{e}} \int_{\Gamma_{\mathrm{w,l_e}}} \tilde{\rho}_{\mathrm{e}} \nabla N_j \phi_j \cdot \nabla N_i \, dS_{\mathrm{e}} - \sum_{\mathrm{e}} \int_{\Gamma_{\mathrm{w,u_e}}} \tilde{\rho}_{\mathrm{e}} \nabla N_j \phi_j \cdot \nabla N_i \, dS_{\mathrm{e}} = 0, \tag{22}$$

and the equality of the pressure (9) is enforced on the upper wake nodes as

$$\sum_{\mathrm{e}} \int_{\Gamma_{\mathrm{w_e}}} \left(\left[\nabla \phi_{\mathrm{e}} \cdot \nabla N_j \phi_j \right]_{\mathrm{w,u}} - \left[\nabla \phi_{\mathrm{e}} \cdot \nabla N_j \phi_j \right]_{\mathrm{w,l}} \right) \left(N_i + \frac{h_{\mathrm{e}}}{2} \hat{e}_{x,k} \, \partial_{x_k} N_i \right)_{\mathrm{w,u}} dS_{\mathrm{e}} = 0, \tag{23}$$

where h_{e} is the square root of an element surface area and where the stabilization direction is chosen to be $\hat{\mathbf{e}}_x = [1, 0, 0]$. The Kutta condition on the trailing edge (10) is enforced on the upper trailing edge nodes as

$$\begin{aligned} &\sum_{\mathrm{e}} \int_{\Gamma_{\mathrm{TE,u_e}}} \frac{1}{h_{\mathrm{e}}^2} \nabla \phi_{\mathrm{e}} \cdot \nabla N_j \phi_j \left(N_i + \frac{h}{2} \hat{e}_{x,k} \, \partial_{x_k} N_i \right) dS_{\mathrm{e}} \\ - &\sum_{\mathrm{e}} \int_{\Gamma_{\mathrm{TE,l_e}}} \frac{1}{h_{\mathrm{e}}^2} \nabla \phi_{\mathrm{e}} \cdot \nabla N_j \phi_j \left(N_i + \frac{h}{2} \hat{e}_{x,k} \, \partial_{x_k} N_i \right) dS_{\mathrm{e}} = 0. \end{aligned} \tag{24}$$

3.1.2. Partial gradients

The partial gradient of the potential residual at node i with respect to the potential variable at node j, also known as the flow Jacobian, is derived from Equation 20 by substituting Equations 5, 11 and 12, and is given by

$$\begin{aligned}\frac{\partial R_{\phi,i}}{\partial \phi_j} &= \sum_{\mathrm{e}} \int_{\Omega_{\mathrm{e}}} (1-\mu)\left[-M_\infty^2 \rho_{\mathrm{e}}^{2-\gamma} \partial_{x_k}\phi\, \partial_{x_k} N_j\, \partial_{x_k}\phi\, \partial_{x_k} N_i + \rho_{\mathrm{e}} \partial_{x_k} N_j\, \partial_{x_k} N_i\right] dV_{\mathrm{e}} \\ &+ \sum_{\mathrm{e}} \int_{\Omega_{\mathrm{e}}} \mu\left[-M_\infty^2 \rho_{\mathrm{U}}^{2-\gamma} \partial_{x_k}\phi_{\mathrm{U}}\, \partial_{x_k} N_{j_{\mathrm{U}}}\, \partial_{x_k}\phi\, \partial_{x_k} N_i + \rho_{\mathrm{U}} \partial_{x_k} N_j\, \partial_{x_k} N_i\right] dV_{\mathrm{e}} \\ &- \sum_{\mathrm{e}} \int_{\Omega_{\mathrm{e}}} (\rho_{\mathrm{e}} - \rho_{\mathrm{U}}) \left[\frac{2\mu_{\mathrm{C}} M_{\mathrm{C}}^2}{M_{\mathrm{e}}^3}\left(\frac{M_\infty}{\sqrt{\partial_{x_k}\phi^2\, \rho_{\mathrm{e}}^{\gamma-1}}} + \frac{\gamma-1}{2/M_\infty^3}\frac{\sqrt{\partial_{x_k}\phi^2}}{\sqrt{\rho_{\mathrm{e}}^{3(\gamma-1)}}}\right)\partial_{x_k}\phi\, \partial_{x_k} N_j\, \partial_{x_k}\phi\, \partial_{x_k} N_i\right] dV_{\mathrm{e}}.\end{aligned} \tag{25}$$

Note that the last term of Equation 25 is zero when the switching function μ (12) is zero. Similar to the residuals $\mathbf{R}_\phi$, the Kutta condition is enforced by adding the contributions of the upper wake nodes to the lower wake rows, instead of the upper wake rows, in the Jacobian matrix,

$$\left.\frac{\partial R_{\phi,i}}{\partial \phi_j}\right|_{\mathrm{w,l}} \leftarrow \left.\frac{\partial R_{\phi,i}}{\partial \phi_j}\right|_{\mathrm{w,l}} + \left.\frac{\partial R_{\phi,i}}{\partial \phi_j}\right|_{\mathrm{w,u}}, \tag{26}$$

where $\leftarrow$ denotes an assignment. The following terms are then assembled on the upper wake rows,

$$\left.\frac{\partial R_{\phi,i}}{\partial \phi_j}\right|_{\mathrm{w,u}} = 2\sum_{\mathrm{e}} \int_{\Gamma_{\mathrm{w_e}}} \left(N_i + \frac{h}{2}\hat{e}_{x,k}\, \partial_{x_k} N_i\right)_{\mathrm{w,u}} \left(\left[\partial_{x_k}\phi\, \partial_{x_k} N_j\right]_{\mathrm{w,u}} - \left[\partial_{x_k}\phi\, \partial_{x_k} N_j\right]_{\mathrm{w,l}}\right) dS_{\mathrm{e}}. \tag{27}$$

Similarly, the lower and upper trailing edge contributions are assembled on the upper trailing edge rows,

$$\begin{aligned}\left.\frac{\partial R_{\phi,i}}{\partial \phi_j}\right|_{\mathrm{TE,u}} = 2\Bigg(&\sum_{\mathrm{e}} \int_{\Gamma_{\mathrm{TE,u_e}}} \frac{1}{h_{\mathrm{e}}^2}\left(N_i + \frac{h_{\mathrm{e}}}{2}\hat{e}_{x,k}\, \partial_{x_k} N_i\right)\partial_{x_k}\phi\, \partial_{x_k} N_j\, dS_{\mathrm{e}} \\ &- \sum_{\mathrm{e}} \int_{\Gamma_{\mathrm{TE,l_e}}} \frac{1}{h_{\mathrm{e}}^2}\left(N_i + \frac{h_{\mathrm{e}}}{2}\hat{e}_{x,k}\, \partial_{x_k} N_i\right)\partial_{x_k}\phi\, \partial_{x_k} N_j\, dS_{\mathrm{e}}\Bigg).\end{aligned} \tag{28}$$

The partial gradient of the potential residual at node i with respect to the mesh coordinates at node j is derived from Equation 20 and is given by

$$\begin{aligned}\frac{\partial R_{\phi,i}}{\partial x_j} &= \sum_{\mathrm{e}} \int_{\Omega_{\mathrm{e}}} (1-\mu)\left(-M_\infty^2 \rho_{\mathrm{e}}^{2-\gamma} \partial_{x_l}\phi \left(-J_{\mathrm{e},lk}^{-1}\, \partial_{x_j} J_{\mathrm{e},lk}\right)\partial_{x_k}\phi\right)\partial_{x_k}\phi\, \partial_{x_k} N_i\, dV_{\mathrm{e}} \\ &+ \sum_{\mathrm{e}} \int_{\Omega_{\mathrm{e}}} \mu\left(-M_\infty^2 \rho_{\mathrm{U}}^{2-\gamma} \partial_{x_l}\phi_{\mathrm{U}} \left(-J_{\mathrm{U},lk}^{-1}\, \partial_{x_j} J_{\mathrm{U},lk}\right)\partial_{x_k}\phi_{\mathrm{U}}\right)\partial_{x_k}\phi\, \partial_{x_k} N_i\, dV_{\mathrm{e}} \\ &+ \sum_{\mathrm{e}} \int_{\Omega_{\mathrm{e}}} \left[(1-\mu)\rho_{\mathrm{e}} + \mu\rho_{\mathrm{U}}\right]\left[\partial_{x_l}\phi\left(-J_{\mathrm{e},lk}^{-1}\, \partial_{x_j} J_{\mathrm{e},lk}\right)\partial_{x_k} N_i + \partial_{x_l} N_i\, J_{\mathrm{e},lk}^{-1}\, \partial_{x_j} J_{\mathrm{e},lk}\, \partial_{x_k}\phi\right] dV_{\mathrm{e}} \\ &- \sum_{\mathrm{e}} \int_{\Omega_{\mathrm{e}}} (\rho_{\mathrm{e}} - \rho_{\mathrm{U}}) \left[\frac{2\mu_{\mathrm{C}} M_{\mathrm{C}}^2}{M_{\mathrm{e}}^3}\left(\frac{M_\infty}{\sqrt{\partial_{x_k}\phi^2\, \rho_{\mathrm{e}}^{\gamma-1}}} + \frac{\gamma-1}{2/M_\infty^3}\frac{\sqrt{\partial_{x_k}\phi^2}}{\sqrt{\rho_{\mathrm{e}}^{3(\gamma-1)}}}\right)\partial_{x_l}\phi\left(-J_{\mathrm{e},lk}^{-1}\, \partial_{x_j} J_{\mathrm{e},lk}\right)\partial_{x_k}\phi\, \partial_{x_k}\phi\, \partial_{x_k} N_i\right] dV_{\mathrm{e}} \\ &+ \sum_{\mathrm{e}} \int_{\Omega_{\mathrm{e}}} \left[(1-\mu)\rho_{\mathrm{e}} + \mu\rho_{\mathrm{U}}\right]\partial_{x_k}\phi\, \partial_{x_k} N_i\, \partial_{x_j} dV_{\mathrm{e}}.\end{aligned} \tag{29}$$

The partial gradient of the Jacobian matrix of an element with respect to the mesh coordinates is defined and computed as,

$$\partial_{x_k} J_{\cdot,ij} = \frac{\partial}{\partial_{x_k}} \left(\partial_{\xi_j} N_l x_{l,i} \right), \tag{30}$$

where the subscript $\cdot$ refers to a variable evaluated on the current element e, or on the upstream element U. Since Gauss quadrature is used to compute the integrals in the finite element method, computing the partial gradient of an elementary volume with respect to the mesh coordinates amounts to computing the partial gradient of the Jacobian matrix determinant of an element as

$$\partial_{x_k} dV_\mathrm{e} = \partial_{x_k} \det(J_{\mathrm{e},ij}) = \det(J_{\mathrm{e},ij}) \operatorname{tr}(J_{\mathrm{e},ij}^{-1} \partial_{x_k} J_{\mathrm{e},ij}). \tag{31}$$

The contributions of the farfield boundary condition are not taken into account since the outer boundary is fixed. The partial gradients of the wake boundary conditions with respect to the mesh coordinates are assembled in a similar way as the flow residuals: the contributions of the upper wake rows are first added to the lower wake rows, and the upper wake rows are then computed as

$$\begin{aligned}
\left.\frac{\partial R_{\phi,i}}{\partial x_j}\right|_{\mathrm{w,u}} &= \sum_\mathrm{e} \int_{\Gamma_\mathrm{we}} \left(\partial_{x_j} \left(\frac{h_\mathrm{e}}{2} \hat{e}_{x,k}\, \partial_{x_k} N_i \right) \right)_{\mathrm{w,u}} \left(\left[\partial_{x_k}\phi\, \partial_{x_k}\phi \right]_{\mathrm{w,u}} - \left[\partial_{x_k}\phi\, \partial_{x_k}\phi \right]_{\mathrm{w,l}} \right) dS_\mathrm{e} \\
&+ 2 \sum_\mathrm{e} \int_{\Gamma_\mathrm{we}} \left(N_i + \frac{h_\mathrm{e}}{2} \hat{e}_{x,k}\, \tilde{U}_{\infty,k}\, \partial_{x_k} N_i \right)_{\mathrm{w,u}} \left[\partial_{x_l}\phi \left(-J_{\mathrm{e},lk}^{-1}\, \partial_{x_j} J_{\mathrm{e},lk} \right) \partial_{x_k}\phi \right]_{\mathrm{w,u}} dS_\mathrm{e} \\
&- 2 \sum_\mathrm{e} \int_{\Gamma_\mathrm{we}} \left(N_i + \frac{h_\mathrm{e}}{2} \hat{e}_{x,k}\, \partial_{x_k} N_i \right)_{\mathrm{w,u}} \left[\partial_{x_l}\phi \left(-J_{\mathrm{e},lk}^{-1}\, \partial_{x_j} J_{\mathrm{e},lk} \right) \partial_{x_k}\phi \right]_{\mathrm{w,l}} dS_\mathrm{e} \\
&+ \sum_\mathrm{e} \int_{\Gamma_\mathrm{we}} \left(N_i + \frac{h_\mathrm{e}}{2} \hat{e}_{x,k}\, \partial_{x_k} N_i \right)_{\mathrm{w,u}} \left(\left[\partial_{x_k}\phi\, \partial_{x_k}\phi \right]_{\mathrm{w,u}} - \left[\partial_{x_k}\phi\, \partial_{x_k}\phi \right]_{\mathrm{w,l}} \right) \partial_{x_j} dS_\mathrm{e}.
\end{aligned} \tag{32}$$

The partial gradient of the Kutta condition with respect to the mesh coordinates is computed as

$$\begin{aligned}
\left.\frac{\partial R_{\phi,i}}{\partial x_j}\right|_{\mathrm{TE},\cdot} &= \sum_\mathrm{e} \int_{\Gamma_{\mathrm{TE},\cdot\,\mathrm{e}}} \frac{-2}{h_\mathrm{e}^3} \partial_{x_j} h \left(N_i + \frac{h_\mathrm{e}}{2} \hat{e}_{x,k}\, \partial_{x_k} N_i \right) \partial_{x_k}\phi\, \partial_{x_k}\phi\, dS_\mathrm{e} \\
&+ \sum_\mathrm{e} \int_{\Gamma_{\mathrm{TE},\cdot\,\mathrm{e}}} \frac{1}{h_\mathrm{e}^2} \partial_{x_j} \left(\frac{h_\mathrm{e}}{2} \hat{e}_{x,k}\, \partial_{x_k} N_i \right) \partial_{x_k}\phi\, \partial_{x_k}\phi\, dS_\mathrm{e} \\
&+ 2 \sum_\mathrm{e} \int_{\Gamma_{\mathrm{TE},\cdot\,\mathrm{e}}} \frac{1}{h_\mathrm{e}^2} \left(N_i + \frac{h_\mathrm{e}}{2} \hat{e}_{x,k}\, \partial_{x_k} N_i \right) \partial_{x_l}\phi \left(-J_{\mathrm{e},lk}^{-1}\, \partial_{x_j} J_{\mathrm{e},lk} \right) \partial_{x_k}\phi\, dS_\mathrm{e} \\
&+ \sum_\mathrm{e} \int_{\Gamma_{\mathrm{TE},\cdot\,\mathrm{e}}} \frac{1}{h_\mathrm{e}^2} \left(N_i + \frac{h_\mathrm{e}}{2} \hat{e}_{x,k}\, \partial_{x_k} N_i \right) \partial_{x_k}\phi\, \partial_{x_k}\phi\, \partial_{x_j} dS_\mathrm{e}
\end{aligned} \tag{33}$$

where the subscript TE, $\cdot$ denotes a contribution from the lower or upper trailing edge. The lower trailing edge contributions are subtracted from those on the upper trailing edge and the result is then assembled on the upper trailing edge rows. The partial gradient of the stabilized shape function can further be developed as

$$\partial_{x_j} \left(\frac{h_\mathrm{e}}{2} \hat{e}_{x,k}\, \partial_{x_k} N_i \right) = \partial_{x_j} \frac{h_\mathrm{e}}{2} \hat{e}_{x,k}\, \partial_{x_k} N_i + \frac{h}{2} \hat{e}_{x,l}\, J_{\mathrm{e},lk}^{-1}\, \partial_{x_j} J_{\mathrm{e},lk} \partial_{x_k} N_i\,. \tag{34}$$

Computing the partial gradient of an elementary surface with respect to the mesh coordinates amounts to computing the partial gradient of the surface Jacobian matrix determinant of an element. For a two-dimensional surface in a three-dimensional space, this gradient is expressed as

$$\partial_{x_j} dS_\mathrm{e} = \partial_{x_j} \det(J_{\mathrm{S,e},i}) = \frac{J_{\mathrm{S,e},i}}{|J_{\mathrm{S,e},i}|} \left(\partial_{x_j} \partial_\xi N_k x_k \times \partial_\eta N_k x_k + \partial_\xi N_k x_k \times \partial_{x_j} \partial_\eta N_k x_k \right). \tag{35}$$

The angles of attack and of sideslip affect the potential residuals only through the farfield boundary condition. The partial gradient of the potential residual at node i with respect to the angles of attack and of sideslip are thus given by

$$\frac{\partial R_{\phi,i}}{\partial [\alpha, \beta]} = \sum_\mathrm{e} \int_{\Gamma_{\mathrm{f_e}}} \rho_\infty \frac{\partial \mathbf{U}_\infty}{\partial [\alpha, \beta]} \cdot \hat{\mathbf{n}}_\mathrm{e} \, N_i \, dS_\mathrm{e}, \tag{36}$$

where the gradient of the freestream velocity with respect to the angles of attack and of sideslip are

$$\frac{\partial \mathbf{U}_\infty}{\partial \alpha} = \begin{bmatrix} -\sin\alpha\cos\beta \\ 0 \\ \cos\alpha\cos\beta \end{bmatrix}, \qquad \frac{\partial \mathbf{U}_\infty}{\partial \beta} = \begin{bmatrix} -\cos\alpha\sin\beta \\ \cos\beta \\ -\sin\alpha\sin\beta \end{bmatrix}. \tag{37}$$

The partial gradient of the aerodynamic load at node i with respect to the potential variable at node j is given by

$$\begin{aligned} \frac{\partial F_i}{\partial \phi_j} &= \frac{1}{2} \rho_\infty u_\infty^2 \sum_\mathrm{e} \int_{\Gamma_{\mathrm{b,e}}} \partial_{\phi_j} C_{p_\mathrm{e}} \, \hat{n}_{\mathrm{e},i} \, dS_\mathrm{e} \\ &= -\rho_\infty u_\infty^2 \sum_\mathrm{e} \int_{\Gamma_{\mathrm{b,e}}} \rho_\mathrm{e}^\gamma \partial_{x_k} \phi \, \partial_{x_k} N_j \, \hat{n}_{\mathrm{e},i} \, dS_\mathrm{e}. \end{aligned} \tag{38}$$

The partial gradient of the aerodynamic load at node i with respect to the mesh coordinates at node j is given by

$$\begin{aligned} \frac{\partial F_i}{\partial x_j} &= \frac{1}{2} \rho_\infty u_\infty^2 \partial_{x_j} \sum_\mathrm{e} \int_{\Gamma_{\mathrm{b,e}}} C_{p_\mathrm{e}} \hat{n}_{\mathrm{e},i} \, dS_\mathrm{e} \\ &= \frac{1}{2} \rho_\infty u_\infty^2 \sum_\mathrm{e} \Bigg[\int_{\Gamma_{\mathrm{b,e}}} -2 \rho_\mathrm{e}^\gamma \partial_{x_l} \phi \left(-J_{\mathrm{e},lk}^{-1} \, \partial_{x_j} J_{\mathrm{e},lk} \right) \partial_{x_k} \phi \, \hat{n}_{\mathrm{e},i} \, dS_\mathrm{e} \\ &\qquad + \int_{\Gamma_{\mathrm{b,e}}} C_{p_\mathrm{e}} \partial_{x_j} \hat{n}_{\mathrm{e},i} \, dS_\mathrm{e} \\ &\qquad + \int_{\Gamma_{\mathrm{b,e}}} C_{p_\mathrm{e}} \hat{n}_{\mathrm{e},i} \, \partial_{x_j} dS_\mathrm{e} \Bigg], \end{aligned} \tag{39}$$

where the partial gradient of an elementary surface is computed as in Equation 35, and the partial gradient of the unit normal vector is given by

$$\frac{\partial \hat{n}_i}{\partial x_j} = (I_{ik} - \hat{n}_i \hat{n}_k) \frac{1}{|n_k|} \partial_{x_j} n_k. \tag{40}$$

The partial gradient of the normal vector to a two-dimensional triangular area with vertices $\mathbf{x}_0$, $\mathbf{x}_1$ and $\mathbf{x}_2$ embedded in a three-dimensional space is given by

$$\frac{\partial \mathbf{n}}{\partial x_j} = \partial_{x_j} (\mathbf{x}_1 - \mathbf{x}_0) \times (\mathbf{x}_2 - \mathbf{x}_0) + (\mathbf{x}_1 - \mathbf{x}_0) \times \partial_{x_j} (\mathbf{x}_2 - \mathbf{x}_0). \tag{41}$$

The partial gradients of the aerodynamic load coefficients can be readily obtained from Equations 38 and 39. Additionally, the partial gradients of the aerodynamic coefficients with respect to the angles of attack and of sideslip are given by,

$$\frac{C_L}{\partial[\alpha,\beta]} = \mathbf{C_F} \cdot \frac{\partial \mathbf{e_L}}{\partial[\alpha,\beta]}, \qquad \frac{C_D}{\partial[\alpha,\beta]} = \mathbf{C_F} \cdot \frac{\partial \mathbf{e_D}}{\partial[\alpha,\beta]}, \tag{42}$$

where the gradients of the directions are defined as,

$$\begin{aligned} \frac{\partial \mathbf{e_L}}{\partial \alpha} &= \begin{bmatrix} -\cos\alpha \\ 0 \\ -\sin\alpha \end{bmatrix}, & \frac{\partial \mathbf{e_D}}{\partial \alpha} &= \begin{bmatrix} -\sin\alpha\cos\beta \\ 0 \\ \cos\alpha\cos\beta \end{bmatrix}, \\ \frac{\partial \mathbf{e_L}}{\partial \beta} &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, & \frac{\partial \mathbf{e_D}}{\partial \beta} &= \begin{bmatrix} -\cos\alpha\sin\beta \\ \cos\beta \\ -\sin\alpha\sin\beta \end{bmatrix}. \end{aligned} \tag{43}$$

3.2. Mesh morphing laws

An efficient way to deform the grid for the kind of wing deflections considered in practical aeroelastic analysis and optimization, is to use linear elasticity theory, as suggested by various authors [45, 46, 47]. The grid is assumed to behave like an elastic body, rigid near the deforming boundaries, and flexible elsewhere. Moreover, the linear elasticity equations can be easily solved by the finite element method, and require little supplementary implementation work. For an elastic solid, the equilibrium between the internal and external forces can be written in weak form as

$$\int_\Omega \nabla\boldsymbol{\sigma} \cdot \nabla\psi \, dV - \int_\Gamma \overline{\nabla\boldsymbol{\sigma}} \cdot \hat{\mathbf{n}}\, \psi \, dS = \int_\Omega \mathbf{f}\, \psi \, dV, \qquad \forall \psi \in \Omega, \tag{44}$$

where the internal stress $\boldsymbol{\sigma}$ can be related to the displacements $\Delta\mathbf{x}$ using Hooke's constitutive law for linear isotropic solids,

$$\boldsymbol{\sigma} = \frac{E\nu}{2(1+\nu)(1-2\nu)} \mathrm{tr}\left(\nabla(\Delta\mathbf{x}) + \nabla(\Delta\mathbf{x})^{\mathrm{T}}\right)\mathbf{I} + \frac{E}{2(1+\nu)}\left(\nabla(\Delta\mathbf{x}) + \nabla(\Delta\mathbf{x})^{\mathrm{T}}\right). \tag{45}$$

The Young's modulus E and Poisson's ratio ν are constitutive parameters. In the present work, they are set to $1/V$, where V is the volume of an element, and 0, respectively, as suggested by Dwight [46]. As a result, the mesh behaves as a linear elastic solid, rigid close to the wing where the elements are small, and flexible in the farfield where the elements are large. Note that, in the context of mesh deformation, the external forces $\mathbf{f}$ are zero, and the deformation is driven by a Dirichlet boundary condition imposed on the moving boundaries. Also note that the wake is treated as an embedded surface and follows the deformation of the surrounding volume elements.

3.2.1. Discretization

After discretization, Equation 44 must hold for any test function ψ, and can therefore be rewritten as a set of equations,

$$\begin{aligned} R_{\mathrm{x},i} &= \sum_{\mathrm{e}} \int_{\Omega_\mathrm{e}} \left[\frac{E_\mathrm{e}\nu_\mathrm{e}}{2(1+\nu_\mathrm{e})(1-2\nu_\mathrm{e})} \partial_{x_k} N_l \Delta x_{l,k}\, \delta_{ij} + \frac{E_\mathrm{e}}{2(1+\nu_\mathrm{e})} \left(\partial_{x_j} N_l \Delta x_{l,i} + \partial_{x_i} N_l \Delta x_{l,j} \right) \right] \partial_{x_j} N_l \, dV_\mathrm{e} \\ &= 0. \end{aligned} \tag{46}$$

The Dirichlet boundary conditions on the deforming surfaces are enforced as

$$\overline{\Delta x_{i,j}}|_{\Gamma_\mathrm{b}} = \Delta x_{\mathrm{b}_{i,j}}. \tag{47}$$

On the wake, the periodic boundary conditions are discretized as follows. The upper wake volume element contributions are added to the lower wake equations, and the upper wake unknowns are prescribed to match the lower wake unknowns,

$$\begin{aligned} &\sum_{\mathrm{e}} \int_{\Omega_{\mathrm{w,le}}} \left[\frac{E_\mathrm{e}\nu_\mathrm{e}}{2(1+\nu_\mathrm{e})(1-2\nu_\mathrm{e})} \partial_{x_k} N_l \Delta x_{l,k}\, \delta_{ij} + \frac{E_\mathrm{e}}{2(1+\nu_\mathrm{e})} \left(\partial_{x_j} N_l \Delta x_{l,i} + \partial_{x_i} N_l \Delta x_{l,j} \right) \right] \partial_{x_j} N_l \, dV_\mathrm{e} \\ &+ \sum_{\mathrm{e}} \int_{\Omega_{\mathrm{w,ue}}} \left[\frac{E_\mathrm{e}\nu_\mathrm{e}}{2(1+\nu_\mathrm{e})(1-2\nu_\mathrm{e})} \partial_{x_k} N_l \Delta x_{l,k}\, \delta_{ij} + \frac{E_\mathrm{e}}{2(1+\nu_\mathrm{e})} \left(\partial_{x_j} N_l \Delta x_{l,i} + \partial_{x_i} N_l \Delta x_{l,j} \right) \right] \partial_{x_j} N_l \, dV_\mathrm{e} \\ &= 0, \\ &\Delta x_{i,j}|_{\Gamma_{\mathrm{w,u}}} - \Delta x_{i,j}|_{\Gamma_{\mathrm{w,l}}} = 0. \end{aligned} \tag{48}$$

3.2.2. Partial gradients

The mesh deformation residuals only depend linearly on the mesh coordinates. The partial gradients of the mesh deformation residual at node i with respect to the mesh coordinates at node j, also known as the mesh Jacobian, are therefore given by

$$\frac{\partial R_{\mathrm{x},i}}{\partial x_j} = \sum_{\mathrm{e}} \int_{\Omega_\mathrm{e}} \left[\frac{E_\mathrm{e}\nu_\mathrm{e}}{2(1+\nu_\mathrm{e})(1-2\nu_\mathrm{e})} \partial_{x_k} N_l\, \delta_{ij} + \frac{E_\mathrm{e}}{2(1+\nu_\mathrm{e})} \left(\partial_{x_j} N_l + \partial_{x_i} N_l \right) \right] \partial_{x_j} N_l \, dV_\mathrm{e}. \tag{49}$$

Note that, similar to the residuals $\mathbf{R}_\mathrm{x}$, periodic boundary conditions are prescribed on the wake by adding the upper wake volume element contributions to the lower wake equations, and by prescribing the upper wake unknowns to match the lower wake unknowns.

3.3. Implementation

The formulation presented in sections 3.1 and 3.2 has been implemented in the open-source code `DART` using the C++ language. The code is parallelized using the Intel `Threading Building Blocks` library [48] [4],

[4] https://www.intel.com/content/www/us/en/developer/tools/oneapi/onetbb.html, accessed March 2022.

and relies on the Eigen library [49] for linear algebra. Furthermore, optimized Basic Linear Algebra Subprograms (BLAS) libraries, such as the Intel Math Kernel Library [5], can be used as a back-end. In order to provide an easy and modular access to the code, while retaining its high computing efficiency, the C++ routines are exposed to python using SWIG [50] [6]. Furthermore, an Application Programming Interface (API) has been implemented using python. In practice, this API provides access to pre and post-processing features, as well as to solution and gradients computation procedures. It is currently compatible with two multidisciplinary frameworks: OpenMDAO [54] [7] and CUPyDO [51, 52] [8]. Since the API has been designed in a modular and flexible way, it could be easily extended to others.

The full potential solution is computed using a Newton method, combined with the line search procedure proposed by Bank and Rose [53]. Furthermore, the upwinding parameters in Equation 12 are varied during the computation. They are initially set to produce a strong stabilization over a wide portion of the flow. As the solution converges, the numerical dissipation is reduced and applied to a smaller portion of the flow. This strategy improves the robustness of the solver, while not impacting the accuracy of the final results. Additional details about this procedure can be found in Crovato [21]. The computation of all the partial gradients is interfaced in python such that they can be combined and included in any multidisciplinary adjoint formulation.

4. Results

One aerodynamic shape optimization and one aerostructural optimization calculation are performed in order to demonstrate the full potential and the mesh deformation formulations presented in Section 3. The computations are performed by means of the following tools, which are described in more details in Appendix A. The optimization process is formulated and solved using MPHYS and OpenMDAO [54], whereby scipy's Sequential Least Squares Programming (SLSQP) [55, 56] is used as the optimization driver. Gmsh [57] is used to create the aerodynamic grids, and pyGeo [58, 59] is used to parameterize and constrain the body's geometry. The full potential equation and the mesh morphing laws are solved using DART [21] and the structural model is solved using TACS [60, 61, 62, 63]. Finally, MELD [64] is used to interpolate and transfer data between the aerodynamic and structural grids.

4.1. Aerodynamic shape optimization - Onera M6 wing

The first calculation is carried out on the Onera M6 wing [65] [9] at a Mach number $M_\infty = 0.839$. The aerodynamic computation is performed in a domain discretized using tetrahedral elements. The objective of the optimization is to minimize the drag coefficient C_D. The geometry is embedded in a free-form deformation box discretized using 7 control points in the chordwise direction, 5 points in the spanwise direction and 2 points in the vertical direction, as shown in Figure 3. These points are restricted to move vertically to control the cross-sectional

[5] https://www.intel.com/content/www/us/en/developer/tools/oneapi/onemkl.html, accessed March 2022
[6] http://www.swig.org/, accessed March 2022.
[7] https://openmdao.org/, accessed March 2022.
[8] https://github.com/ulgltas/CUPyDO, accessed March 2022.
[9] https://www.grc.nasa.gov/www/wind/valid/m6wing/m6wing.html, accessed March 2022.

shape of the wing. Additionally, the points of the last 4 sections along the wing span can rotate rigidly in order to create a twist. Furthermore, the angle of attack of the freestream flow is allowed to change. The lift coefficient is constrained to $C_L = 0.30$. The internal volume of the wing is prevented from decreasing in order to emulate the presence of a structural wingbox. The leading edge radius and the trailing edge thickness are constrained to remain constant using 10 points distributed along the span, so that the leading edge cannot sharpen and the trailing edge cannot collapse. Additionally, the upper and lower control points at the wing leading and trailing edges are constrained to move in equal and opposite direction so that they cannot emulate a twist, which would be redundant with the twist design variables [13]. The optimization tolerance is set to a tenth of a drag count. The optimization problem is summarized in Table 1, and the N^2 diagram produced by `OpenMDAO` is provided in Appendix B.

Objective	Number of variables	Bounds	Scaling
drag coefficient	1	–	10^2
Design variables			
angle of attack	1	$[0; 5]^\circ$	10^{-1}
shape	70	$[-1; 1]$ cm	10^1
twist	4	$[-5; 5]^\circ$	10^{-1}
Nonlinear constraints			
lift coefficient	1	= 0.30	1
wing volume	1	[100; –] %	1
leading edge radius	10	[100; –] %	1
trailing edge thickness	10	[100; –] %	1
Linear constraints			
leading/trailing edge	10	= 0 m	1

Table 1: Summary of the Onera M6 optimization problem.

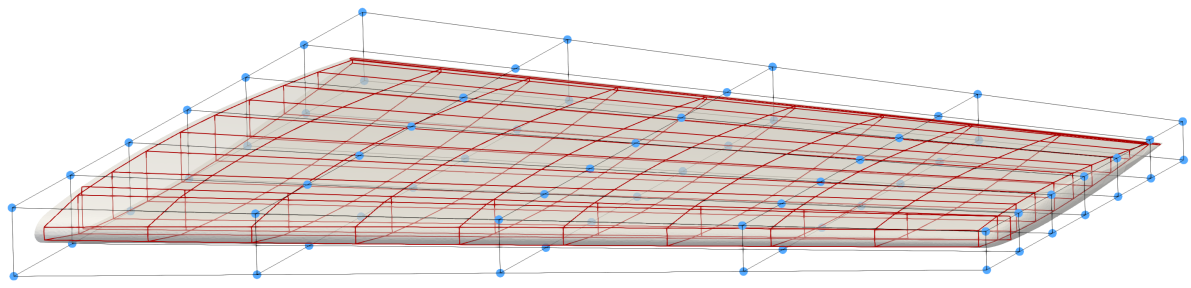


Figure 3: Optimization setup for the Onera M6 case. The points controlling the shape design variables are shown in light blue and the shape constraints are displayed in red.

4.1.1. Grid convergence analysis and gradient verification

Before running the optimization, a grid convergence analysis is first carried out. The results provided in 2 indicate that the medium mesh yields converged results.

Mesh	N. cells	C_L	C_D	CPU time
Coarse	196 080	0.2865	0.01062	0.5 min
Medium	477 857	0.2996	0.01126	1.5 min
Fine	826 182	0.3013	0.01129	3.5 min

Table 2: Cell count, aerodynamic coefficients and computational times obtained using three grids for the Onera M6 case.

The analytic total gradients obtained using the adjoint method are also checked against the gradients computed using finite differences. The procedure is automated in `OpenMDAO`. The relative difference in the total gradients of the lift and drag coefficients, C_L and C_D, with respect to the angle of attack α and the twist angles θ_i on the original geometry are given by

$$\begin{aligned} \frac{\left|\frac{dC_L}{d\alpha}_{\text{AN}} - \frac{dC_L}{d\alpha}_{\text{FD}}\right|}{\frac{dC_L}{d\alpha}_{\text{AN}}} &= 2.4 \times 10^{-5}, \quad \frac{\left|\frac{dC_D}{d\alpha}_{\text{AN}} - \frac{dC_D}{d\alpha}_{\text{FD}}\right|}{\frac{dC_D}{d\alpha}_{\text{AN}}} = 3.0 \times 10^{-5}, \\ \frac{\left|\frac{dC_L}{d\theta_i}_{\text{AN}} - \frac{dC_L}{dz_i}_{\text{FD}}\right|}{\frac{dC_L}{d\theta_i}_{\text{AN}}} &= 2.5 \times 10^{-5}, \quad \frac{\left|\frac{dC_D}{d\theta_i}_{\text{AN}} - \frac{dC_D}{dz_i}_{\text{FD}}\right|}{\frac{dC_D}{d\theta_i}_{\text{AN}}} = 3.0 \times 10^{-5}, \end{aligned} \tag{50}$$

where subscripts AN and FD stand for analytic and finite differences, respectively, and where the step size for the finite differences has been set to 10^{-6}. Note that the gradients with respect to the motion of the control points in the direction normal to the chord has not been checked for each individual point. However, changing the twist amounts to changing the position of the control points. Therefore, computing the gradients with respect to the twist angle involves computing the gradients with respect to the displacements of the points, so that all gradients are implicitly checked.

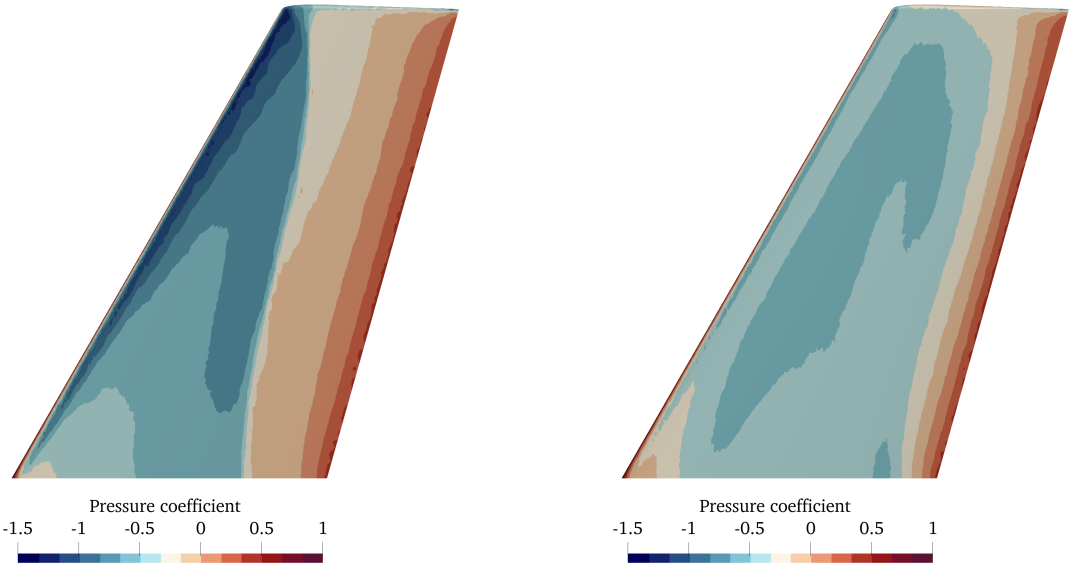
4.1.2. Optimization results

The integrated aerodynamic coefficients obtained on the baseline and optimized Onera M6 wing at $M_\infty = 0.839$ are given in Table 3. Additionally, the pressure coefficient contours on the suction side of the wing are shown in Figure 4, and the sectional pressure coefficients extracted at four stations along the semi-span are shown in Figure 5. Note that the second station corresponds to the mean aerodynamic chord. The initial solution, computed at an angle of attack of 3.06°, exhibits the well-known lambda shock pattern. Overall, the optimization process turned the original symmetric airfoil shape into a supercritical one, such that the strong shock is removed, and the flow is smoother. Moreover, the angle of attack has been slightly reduced and a negative twist angle has been added on the outboard sections of the wing. Consequently, the drag coefficient has been reduced from $C_{D,\text{baseline}} = 0.0113$ to $C_D = 0.0079$. The pressure distributions obtained using the present full potential formulation are also compared

to those previously obtained by Lyu et al. [70] using the Euler equations. The optimized solutions obtained from the full potential and Euler methodologies feature similar airfoil shapes and twist distributions. The Euler-based calculations produced a slightly sharper leading edge that completely eliminates the pressure peak, and leads to a drag reduction of 40%, compared to the 30% reduction obtained from the full potential optimization. This could be explained by the different solutions obtained on the baseline geometry: since the full potential equation predicts a smoother pressure peak initially, it is probably less penalized during the optimization process. Note that the differences could also be partly explained by the fact that different tools were used, especially the optimization driver.

Parameters	$\alpha(^\circ)$	C_L	C_D
Baseline	3.06	0.300	0.0113
Optimized	2.57	0.300	0.0079

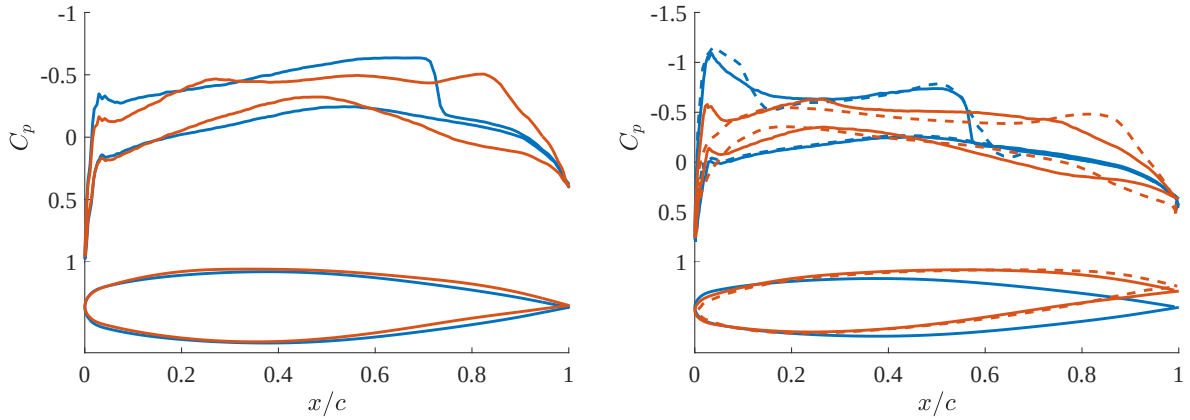
Table 3: Initial and final values of the angle of attack and aerodynamic coefficients of the Onera M6 wing at $M_\infty = 0.839$.



(a) Baseline solution.

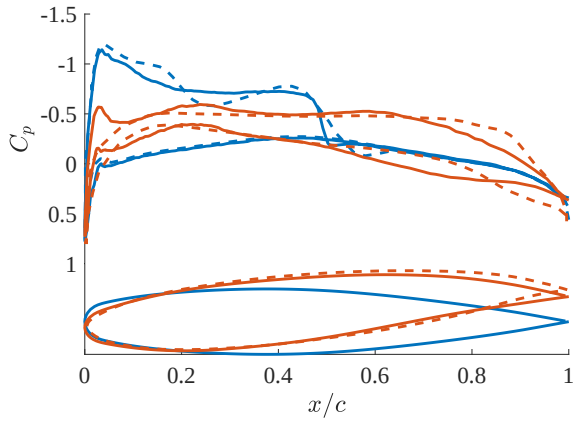
(b) Optimized solution.

Figure 4: Pressure coefficient contours on the suction side of the baseline and optimized Onera M6 wing at $M_\infty = 0.839$.

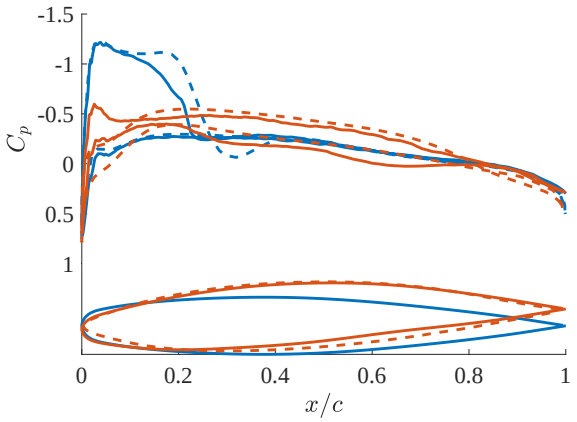


(a) $y/b = 0$.

(b) $y/b = 0.44$.



(c) $y/b = 0.65$.



(d) $y/b = 0.95$.

Figure 5: Sectional pressure coefficient and airfoil shape extracted at four stations along the span of the baseline and optimized Onera M6 wing at $M_\infty = 0.839$, obtained using the present approach and compared to Lyu et al. [70]. Blue: baseline design, red: optimized design. Solid: present approach, dashed: Lyu et al. [70].

The computation was run on a desktop workstation equipped with an AMD 3970X processor rated at 3.7 GHz. The mesh size, the number of gradient and functional evaluations, as well as the computational time are given in Table 4. Even though the Euler based optimization achieved better results, these were obtained at a much higher computational cost. While the full potential computation required only 11 gradient evaluations and 30 minutes, the Euler computation required 112 iterations for a total computational time of 348 core-hours [10].

N. cells	N. grad. eval.	N. func. eval.	N. threads	Wall-clock time
477 857	11	16	1	30 min

Table 4: Mesh size, number of gradient and functional evaluations, number of threads and computational time required for the Onera M6 case.

[10] The Euler computations were performed on a 2013 super-computer.

4.2. Aerostructural optimization - RAE benchmark wing

The second computation is carried out on the RAE benchmark wing at a Mach number $M_\infty = 0.82$ and an altitude of 35 000 ft. The wing is representative of a typical regional jet, and its geometrical and structural parameters are given in Table 5 [11]. The aerodynamic computation is performed in a domain discretized using tetrahedral elements. The structural model is made of an isotropic material representative of Aluminum 7000 series, and is discretized using second order shell elements based on the mixed interpolation of tensorial components formulation [60].

Parameters	Values
Aspect ratio	8.6
Taper ratio	0.3
Leading edge sweep	28.2°
Reference area	91 m^2
Airfoil	RAE 2822

(a) Geometric data.

Parameters	Values
Density	2 780 kg/m^3
Young modulus	73.1 GPa
Poisson ratio	0.33
Shear correction factor	5/6
Yield stress	324 MPa

(b) Structural data.

Table 5: Parameters of the RAE wing.

The objective of the optimization is to minimize the fuel burn W_{fuel}, which is computed using the Breguet formula,

$$W_{\text{fuel}} = \left(W_{\text{fixed}} + W_{\text{wing}}\right) \times \left(\exp\left(\frac{R c_t}{V_\infty} \frac{C_{D_0} + C_D}{C_L}\right) - 1\right), \tag{51}$$

where W_{fixed} is the fixed mass of the aircraft, W_{wing} is the mass of the structural wingbox, R is the mission range, c_t is the thrust specific fuel consumption, V_∞ is the true airspeed, and C_L is the lift coefficient. In order to account for the viscous drag, a fixed drag coefficient C_{D_0} is added to the drag coefficient C_D obtained using `DART`. The values of the different parameters have been derived from a typical mission profile of an Embraer regional aircraft and are provided in Table 6.

[11] The wing geometry is available at https://github.com/mdolab/MACH-Aero/tree/master/tutorial/aero/geometry/. The structural model has been built by the MDO Lab and is available at https://github.com/OpenMDAO/mphys/tree/main/examples/aerostructural/mach_tutorial_wing/vlm_meld_tacs/. Links accessed March 2022.

Parameters	Symbols	Values
Fixed weight	W_{fixed}	34 tons
Range	R	2 500 nm
Specific fuel consumption	c_t	0.65 h^{-1}
Airspeed	V_∞	473 KTAS
Dynamic pressure	q_∞	11 200 Pa
Fixed drag coefficient	C_{D_0}	0.022

Table 6: Mission parameters for the RAE optimization case.

Similar to the Onera M6 case, the wing skin and the wingbox are embedded in a box discretized using 7 control points in the chordwise direction, 5 points in the spanwise direction and 2 points in the vertical direction to control the cross-sectional shape of the wing, as shown in Figure 6. Additionally, the twist angle of the last 3 sections along the wing span and the angle of attack of the freestream flow are allowed to change. Furthermore, the thickness of the 18 ribs, leading and trailing edge spars, 16 stringers and 324 skin panel patches are also used as design variables. The lift coefficient and the aircraft weight are related through the load factor n, which is constrained to one, so that the aircraft is balanced at mid-cruise,

$$n = \frac{q_\infty S_{\text{ref}} C_L}{g\left(W_{\text{fixed}} + W_{\text{wing}} + \frac{W_{\text{fuel}}}{2}\right)} = 1, \tag{52}$$

where q_∞ is the dynamic pressure, S_{ref} is the reference area and g is the acceleration due to gravity. Note that the trim Equation 52 does not account for any lifting surfaces other than the wing. The structural failure index of the wing is constrained to remain below a given threshold. The local failure criterion is computed in each shell element by dividing the von Mises stress by the yield stress, and the global index is calculated by aggregating the criterion over all the elements using a Kreisselmeier-Steinhauser function [71, 62]. The global failure index is multiplied by a safety factor of 1.5. The ratio of the internal volume of the wingbox to its initial value is constrained to be no less than the ratio of the fuel weight to its initial value, so that the fuel required for the flight can be stored inside the fuel tanks. This volume constraint is enforced from the wing root to 75% of the span, where the fuel tanks would be located in an actual wing. Furthermore, the height of the trailing edge spar is constrained to 80% of its initial value using 10 points distributed along the span, such that sufficient space is reserved for the attachment of the control surface actuation devices. Similar to the Onera M6 case, the leading edge radius and the trailing edge thickness are constrained to remain constant using 10 points along the span, and the upper and lower control points at the wing leading and trailing edges are constrained to move in equal and opposite direction. Finally, structural smoothness is ensured by constraining the difference in the thicknesses between adjacent structural elements to remain below a given value. The optimization tolerance is set to 1 kg and the different physics are coupled using a Block Gauss-Seidel scheme [72], whereby relative tolerances on the residuals for the direct and adjoint solutions

are set to 10^{-6}. The optimization problem is summarized in Table 7, and the N^2 diagram is provided in Appendix B. Note that, as the present work is geared towards the demonstration of the full potential adjoint formulation and its interfacing in an optimization framework, the problem formulation is quite simple. Possible improvements will be listed in Section 5.

Objective	Number of variables	Bounds	Scaling
fuel burn	1	–	10^{-4}
Design variables			
angle of attack	1	$[0; 5]°$	1
shape	70	$[-10; 10]$ cm	10^{1}
twist	3	$[-5; 5]°$	1
ribs thickness	18	$[1; 20]$ mm	10^{2}
spar thickness	36	$[1; 20]$ mm	10^{2}
stringers thickness	288	$[1; 20]$ mm	10^{2}
skin thickness	324	$[1; 20]$ mm	10^{2}
Nonlinear constraints			
trim	1	$= 1$	1
failure	1	$= 1/1.5$	1.5
wing volume ratio	1	$[100; -]$ %	1
leading edge radius	10	$[100; -]$ %	1
trailing edge thickness	10	$[100; -]$ %	1
trailing edge spar height	10	$[80; -]$ %	1
Linear constraints			
leading/trailing edge	10	$= 0$ m	1
structural smoothness	1 256	$[-; 1]$ mm	10^{3}

Table 7: Summary of the RAE optimization problem.

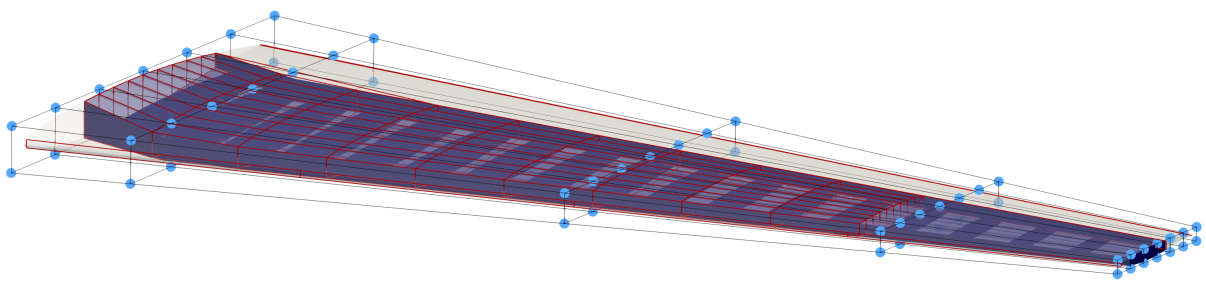


Figure 6: Optimization setup for the RAE wing case. The shape design variables are shown in light blue, the wingbox is displayed in dark blue and the shape constraints are depicted in red.

4.2.1. Grid convergence analysis and gradient verification

Before running the optimization, a grid convergence analysis is carried out at an angle of attack of 2.55°. The three computations converged in nine coupling iterations. The results provided in Table 8 indicate that the medium mesh yields sufficiently converged results. Note that only the inviscid drag contribution is shown.

Mesh	N. cells	C_L	C_D	Δ_{tip}	CPU time
Coarse	330 978	0.3830	0.00802	1.18 m	5.4 min
Medium	655 569	0.4069	0.00931	1.22 m	12.8 min
Fine	1 118 925	0.4080	0.00996	1.24 m	21.4 min

Table 8: Aerodynamic coefficients, tip displacement and computational times obtained using three grids for the RAE case.

The analytic total gradients computed using the adjoint method are also checked against the gradients computed using finite differences in `OpenMDAO`. The relative difference in the total gradients of the fuel burn W_{fuel} with respect to the angle of attack α and the thickness of the structural elements $t_{\star,i}$ are given by

$$\begin{aligned}
\frac{\left|\frac{dW_{\text{fuel}}}{d\alpha}_{\text{AN}} - \frac{dW_{\text{fuel}}}{d\alpha}_{\text{FD}}\right|}{\frac{dW_{\text{fuel}}}{d\alpha}_{\text{AN}}} &= 1.2 \times 10^{-4}, \\
\frac{\left|\frac{dW_{\text{fuel}}}{dt_{\star,i}}_{\text{AN}} - \frac{dW_{\text{fuel}}}{dt_{\star,i}}_{\text{FD}}\right|}{\frac{dW_{\text{fuel}}}{dt_{\star,i}}_{\text{AN}}} &= \begin{bmatrix} 5.4 \times 10^{-4} \\ 1.1 \times 10^{-4} \\ 3.7 \times 10^{-4} \\ 7.3 \times 10^{-4} \\ 7.1 \times 10^{-4} \\ 1.1 \times 10^{-4} \\ 1.7 \times 10^{-4} \end{bmatrix},
\end{aligned} \tag{53}$$

where the step size for the finite differences has been set to 10^{-6}. Note that the structural elements are listed in the following order: ribs, leading and trailing edge spars, upper and lower skins, and upper and lower stringers.

4.2.2. Optimization results

Figure 7 shows the variation of the load factor, the failure criterion, the lift-to-drag ratio, the wingbox mass and the fuel burn as a function of the number of functional evaluations for the RAE case. Initially, the wing is stiffened in order to lower the failure criterion. Once it has reached the prescribed value, the thickness of the structural elements is redistributed, and the wing mass decreases. Overall the fuel burn smoothly decreases from 10.90 to 10.09 tons over 42 iterations. The reduction in fuel burn is solely due to the increase of the lift-to-drag ratio, varying from 12.8 to 13.7, since the wingbox mass slightly increases from 1.27 to 1.34 tons.

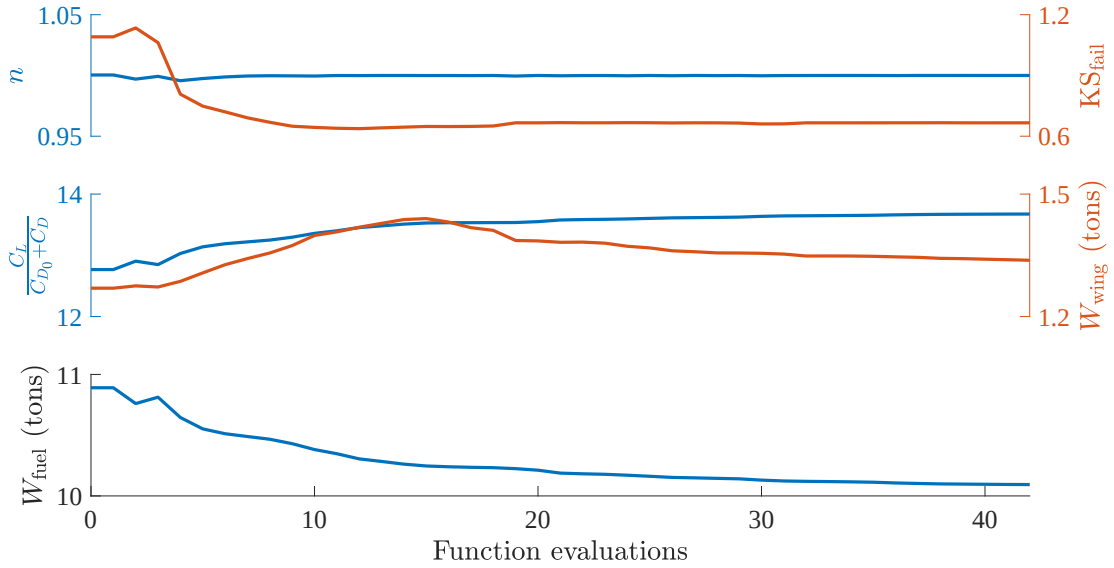
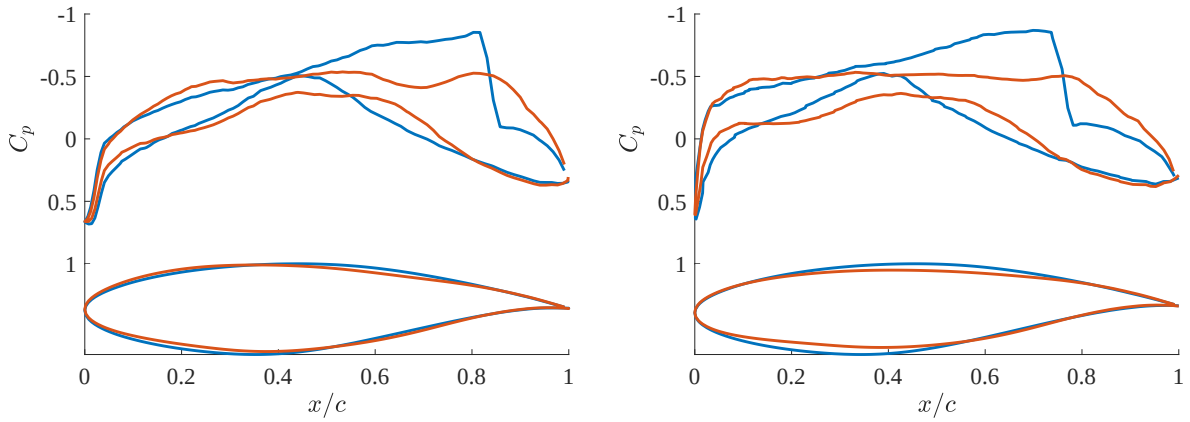


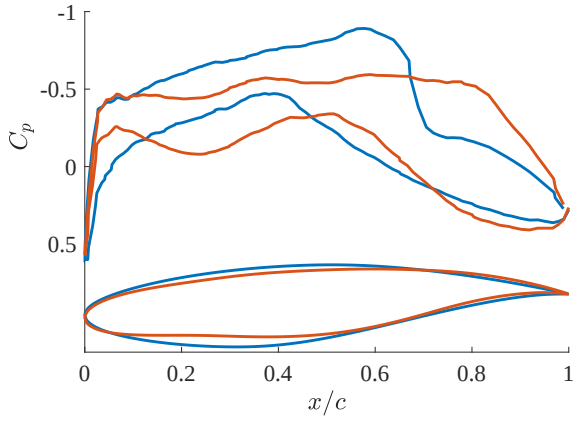
Figure 7: Variation of the load factor, the failure criterion, the lift-to-drag ratio, the wingbox mass and the fuel burn as a function of the number of function evaluations for the RAE benchmark case.

The sectional pressure coefficient and the airfoil shape extracted at six stations along the span of the baseline and optimized RAE wing are depicted in Figure 8. Note that Fig. 8b corresponds to the location of the wingbox kink and that Fig. 8c corresponds to the location of the mean aerodynamic chord. Overall, the optimization process has slightly modified the baseline RAE 2822 supercritical airfoil in order to suppress the strong shock on the suction side of the wing, except near the trailing edge of the station located at 80% of the semi-span, where a weak shock is still present. Consequently, the drag coefficient has been reduced from $C_{D,\text{baseline}} = 0.0307$ to $C_D = 0.0284$. Although the drag has decreased and the lift-to-drag ratio has increased, it is worth noting that the optimizer produced bumps on the pressure side of the wing. While these bumps are barely noticeable on the inboard section of the wing, they become more pronounced outboard. Furthermore, the presence of a bump near the mid-chord reduces lift production locally and shifts the loads towards the trailing edge, especially outboard. These bumps are probably due to the thickness constraint imposed at the rear spar, and to the inviscid nature of the calculations. If the fluid were viscous, a boundary layer would be created, and any bumps would produce more drag, which would drive the optimizer away from such a solution. Viscous effects can be taken into account by complementing the full potential equation by the integral boundary layer equations through a viscous-inviscid interaction method [66]. Such a coupling is currently being developed and implemented in `DART` [67, 68]. Alternatively, imposing the thickness to change uniformly along the chord might also help alleviate this issue, probably at the cost of obtaining a less optimal solution. It should be mentioned that similar bumps were already noticed in the context of aerostructural optimization based on the Euler equations [69].

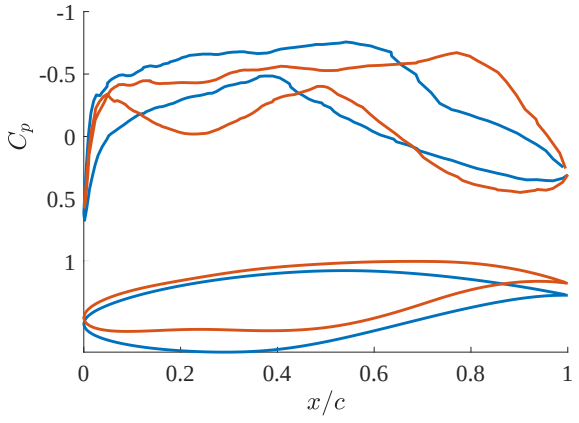


(a) $y/b = 0.$

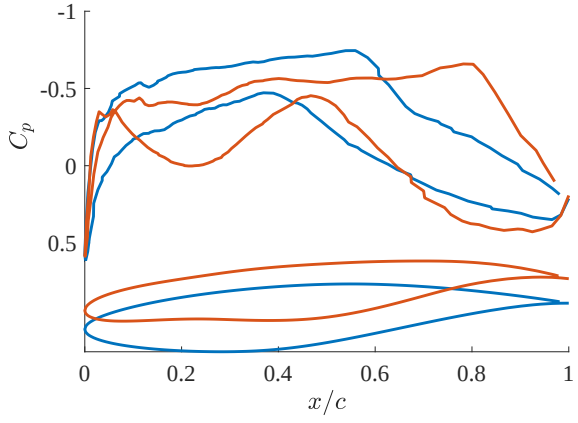
(b) $y/b = 0.10.$



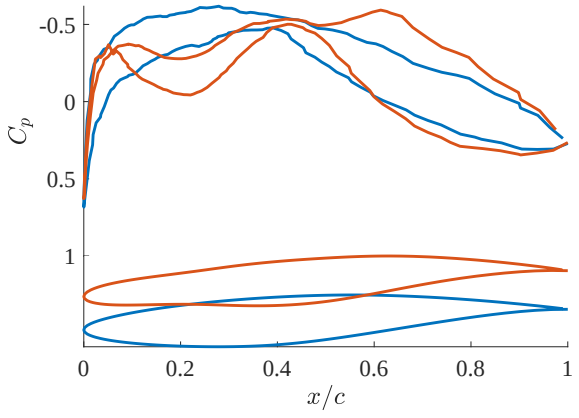
(c) $y/b = 0.41.$



(d) $y/b = 0.65.$



(e) $y/b = 0.8.$

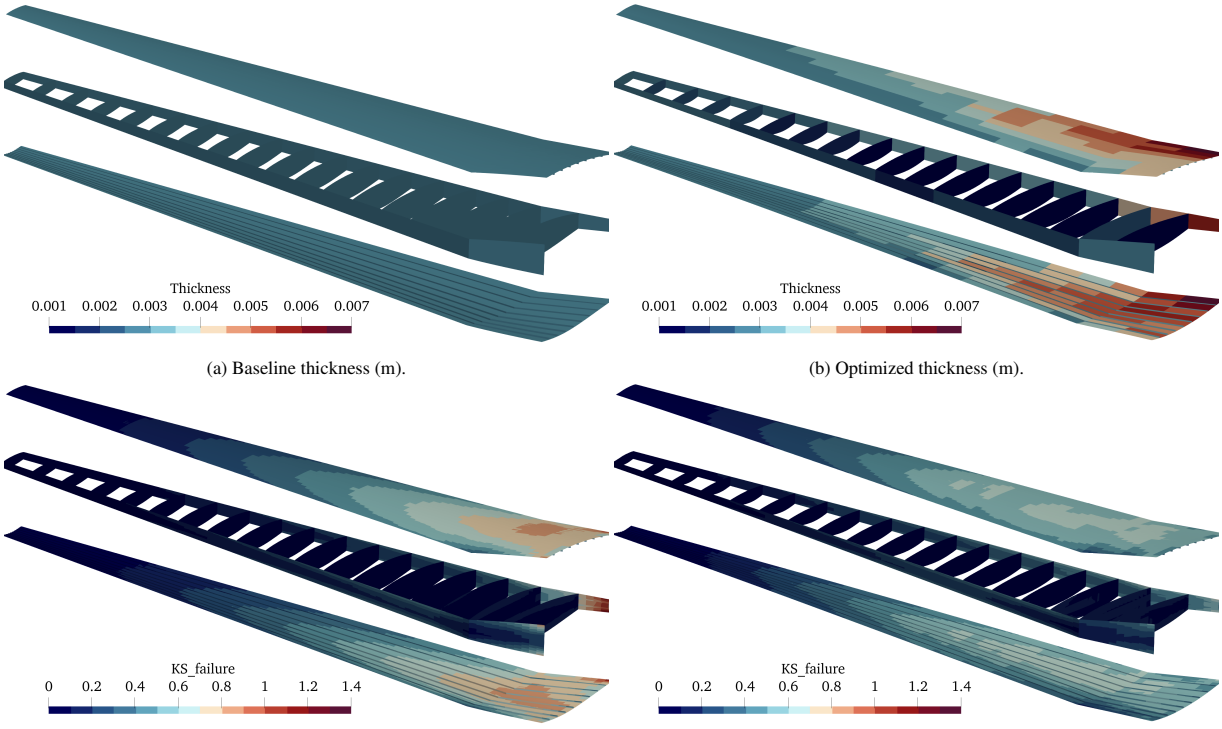


(f) $y/b = 0.95.$

Figure 8: Sectional pressure coefficient and airfoil shape extracted at six stations along the span of the RAE wing at $M_\infty = 0.82$. Blue: baseline design, red: optimized design.

Figure 9 illustrates the thickness of the patches and the failure criterion on the elements of the baseline and optimized structural wingbox for the RAE case. Overall, the thickness of the ribs has decreased uniformly to about 1 mm and that of the stringers is nearly left unchanged. Moreover, the thickness of the spars and the skin panels has increased inboard, where the failure criterion was initially violated, while the thickness decreased outboard, where the stresses are the lowest. Finally, the wingbox volume has decreased on the outboard region, where it is

not constrained.



(a) Baseline thickness (m).

(b) Optimized thickness (m).

(c) Baseline failure criterion.

(d) Optimized failure criterion.

Figure 9: Thickness and failure criterion of the baseline and optimized structural wingbox for the RAE case.

The computation was run on a desktop workstation equipped with an AMD 3970X processor rated at 3.7 GHz. The mesh size, the number of gradient and functional evaluations, and computational time are given in Table 9. Note that `DART` uses shared-memory parallelization while `TACS` and `MELD` use distributed-memory parallelization. To avoid any conflict between the two strategies, `TACS` and `MELD` have been run on one process. As such, the computational time could be reduced by enabling both shared and distributed memory parallelization.

N. aero. cells	N. struct. cells	N. grad. eval.	N. func. eval.	N. threads	Wall-clock time
655 569	7 632	40	42	1	13 h

Table 9: Aerodynamic and structural mesh size, number of gradient and functional evaluations, number of threads and computational time required for the RAE benchmark case.

5. Conclusion

In the present work, a discrete adjoint full potential formulation has first been developed to quickly compute the flow solution and its gradients, so that they can be provided to an optimization process. More specifically, analytical expressions for the full potential and the mesh morphing law residuals and gradients have been derived and presented in this manuscript. The formulation was then implemented in an open-source and readily available finite element code, `DART`, and integrated in an `OpenMDAO`-based framework, able to perform aerostructural analysis and optimization suited for preliminary aircraft design. The methodology was subsequently demonstrated

on two cases: an aerodynamic shape optimization of the Onera M6 wing, and an aerostructural optimization of a typical regional jet wing. Overall, the results indicate that the gradients provided by the full potential formulation are able to quickly optimize the shape and the structural thickness of wings. More specifically, the optimizer turns the wing shape into supercritical airfoils adapted to transonic conditions, so that the strength of shockwaves and the associated wave drag are reduced. Moreover, a washout is created along the wing span, such that the tip vortices and the induced drag are reduced. Additionally, for aerostructural cases, the thickness of the structural wing box increases where the stresses are high, and decreases where they are low. This redistribution produces a light wing that respects the failure criterion of the material. The two calculations, performed on a desktop workstation, only took 30 minutes and 13 core-hours. In the future, the integration of the code in the optimization framework will be improved so that both shared and distributed memory parallelization can be used at the same time, hence further reducing the computational cost. Although the aerostructural case presented in this work is representative of an actual process, it remains quite simple. The next steps consist in improving the model, its parametrization, as well as the formulation of the constraints and the objective function, following various ideas already present in the literature. Particularly, a more detailed structural model, made of composite materials including more design variables and non-aerodynamic loads, will be considered. Additionally, viscous effects will be accounted for by resorting to viscous-inviscid interaction, whereby the full potential model will be complemented by the boundary layer equations. Moreover, the planform shape will also be parametrized, so that the taper ratio, the sweep angle and the dihedral angle of the wing are allowed to change. Consequently, the reference wing area will also have to be adequately constrained. The internal volume of the wingbox will also be constrained by computing the actual volume of the fuel tanks, rather than by imposing a ratio. Furthermore, the failure criterion will be constrained by maneuvers, such as a 2.5 g pull-up, rather than by cruise conditions. Buckling constraints will also be considered. The formulation of the objective function will also include the fuel burn in various flight conditions, and will not be limited to the nominal cruise. Ultimately, the process will be applied on a full aircraft configuration, and will also allow to recover the jig shape. The results will then be compared to higher-fidelity computations, such as Reynolds-Averaged Navier-Stokes calculations.

6. Acknowledgments

The authors gratefully thank Anil Yildirim (University of Michigan), Kevin Jacobson (NASA Langley Research Center) and Justin Gray (NASA Glenn Research Center) for granting an early access to the software `MPHYS`. The authors would also like to thank Luc Papeleux (University of Liège) for his precious help. The authors would finally like to acknowledge the aerospace company Embraer S.A. for partly funding the present work, the MDO Lab members for providing the structural model of the RAE benchmark, and the Consortium des Équipements de Calculs Intensifs (CÉCI), funded by the Fonds de la Recherche Scientifique de Belgique under Grant No. 2.5020.11, for providing the computational resources that helped to develop and validate `DART`.

References

[1] M. H. Shirk, T. J. Hertz, Aeroelastic Tailoring - Theory, Practise and Promise, Journal of Aircraft 23 (1) (1986) 6–18.

[2] G. L. Mura, Mesh Sensitivity Investigation in the Discrete Adjoint Framework, Ph.D. thesis, University of Sheffield (March 2017).

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

[4] J. R. R. A. Martins, J. T. Hwang, Review and Unification of Methods for Computing Derivatives of Multidisciplinary Computational Models, AIAA Journal 51 (11) (2013) 2582–2599.

[5] J. T. Hwang, J. R. R. A. Martins, A computational architecture for coupling heterogeneous numerical models and computing coupled derivatives, ACM Transactions on Mathematical Software 44 (4) (2018) Article 37.

[6] G. R. Shubin, P. D. Frank., A comparison of the implicit gradient approach and the variational approach to aerodynamic design optimization, in: Proceedings of the Third International Conference on Inverse Design Concepts and Optimization in Engineering Sciences, 1991.

[7] J. Brezillon, N. Gauger, 2d and 3d aerodynamic shape optimisation using the adjoint approach, Aerospace Science and Technology 8 (8) (2004) 715–727. doi:https://doi.org/10.1016/j.ast.2004.07.006.

[8] J. Martins, J. Alonso, J. Reuther, A Coupled-Adjoint Sensitivity Analysis Method for High-Fidelity Aero-Structural Design, Optimization and Engineering 6 (1) (2005) 33–62. doi:https://doi.org/10.1023/B:OPTE.0000048536.47956.62.

[9] R. P. Liem, J. R. Martins, G. K. Kenway, Expected drag minimization for aerodynamic design optimization based on aircraft operational data, Aerospace Science and Technology 63 (2017) 344–362. doi:https://doi.org/10.1016/j.ast.2017.01.006.

[10] R. Sanchez, T. Albring, R. Palacios, N. R. Gauger, T. D. Economon, J. J. Alonso, Coupled adjoint-based sensitivities in large-displacement fluid-structure interaction using algorithmic differentiation, International Journal for Numerical Methods in Engineering 113 (7) (2018) 1081–1107. doi:https://doi.org/10.1002/nme.5700.

[11] G. K. Kenway, C. A. Mader, P. He, J. R. Martins, Effective adjoint approaches for computational fluid dynamics, Progress in Aerospace Sciences 110 (2019) 100542. doi:https://doi.org/10.1016/j.paerosci.2019.05.002.

[12] D. A. Burdette, J. R. Martins, Design of a transonic wing with an adaptive morphing trailing edge via aerostructural optimization, Aerospace Science and Technology 81 (2018) 192–203. doi:https://doi.org/10.1016/j.ast.2018.08.004.

[13] T. R. Brooks, G. K. W. Kenway, J. R. R. A. Martins, Benchmark Aerostructural Models for the Study of Transonic Aircraft Wings, AIAA Journal 56 (7) (2018) 2840–2855.

[14] T. Brooks, J. R. R. A. Martins, G. J. Kennedy, High-fidelity Aerostructural Optimization of Tow-steered Composite Wings, Journal of Fluids and Structures 88 (2019) 122–147.

[15] N. P. Bons, J. R. R. A. Martins, Aerostructural Design Exploration of a Wing in Transonic Flow, Aerospace 7 (8) (2020) 118.

[16] J. E. Hoogervorst, A. Elham, Wing aerostructural optimization using the Individual Discipline Feasible Architecture, Aerospace Science and Technology 65 (2017) 90–99. doi:https://doi.org/10.1016/j.ast.2017.02.012.

[17] M. Li, J. Bai, L. Li, X. Meng, Q. Liu, B. Chen, A gradient-based aero-stealth optimization design method for flying wing aircraft, Aerospace Science and Technology 92 (2019) 156–169. doi:https://doi.org/10.1016/j.ast.2019.05.067.

[18] J. R. Wright, J. E. Cooper, Static Aeroelasticity and Flutter, John Wiley and Sons, 2015, Ch. 22, pp. 475–480.

[19] A. Crovato, H. S. Almeida, G. Vio, G. H. Silva, A. P. Prado, C. Breviglieri, H. Güner, P. H. Cabral, R. Boman, V. E. Terrapon, G. Dimitriadis, Effect of Levels of Fidelity on Steady Aerodynamic and Static Aeroelastic Computations, Aerospace 7 (4) (2020) 42. doi:https://doi.org/10.3390/aerospace7040042.

[20] A. Crovato, R. Boman, G. Güner, Hüseyin abd Dimitriadis, V. Terrapon, H. Almeida, A. Prado, C. Breviglieri, P. Cabral, G. Silva, A Full Potential Static Aeroelastic Solver for Preliminary Aircraft Design, in: Proceedings of the 18th International Forum on Aeroelasticity and Structural Dynamics, 2019.

[21] A. Crovato, Steady Transonic Aerodynamic and Aeroelastic Modeling for Preliminary Aircraft Design, Ph.D. thesis, University of Liège (October 2020).
URL http://hdl.handle.net/2268/251906

[22] F. Angrand, Optimum design for potential flows, International journal for numerical methods in fluid dynamics 3 (1983) 265–282.

[23] A. Jameson, Aerodynamic design via control theory, Journal of scientific computing 3 (3) (1988) 233–260.

[24] A. Jameson, Automatic Design of Transonic Airfoils to Reduce the Shock Induced Pressure Drag, in: 31st Israel Annual Conference on Aviation and Aeronautics, 1990, pp. 5–17.

[25] J. Reuter, A. Jameson, Control theory based airfoil design for potential flow and a finite volume discretization, in: Proceedings of the 32nd Aerospace Sciences Meeting and Exhibit, 1994. doi:10.2514/6.1994-499.

[26] L. Carlos De Castro Santos, An Adjoint Formulation for the Non-Linear Potential Flow Equation, Applied Mathematics and Computations 108 (1) (2000) 11–21. doi:10.1016/S0096-3003(98)10137-6.

[27] M. C. Galbraith, S. R. Allmaras, R. Haimes, Full Potential Revisited: A Medium Fidelty Aerodynamic Analysis Tool, in: Proceedings of the 55th AIAA Aerospace Sciences Meeting, 2017. doi:10.2514/6.2017-0290.

[28] A. Parrinello, P. Mantegazza, Independent two-fields solution for full-potential unsteady transonic flows, AIAA Journal 48 (7).

[29] A. Parrinello, P. Mantegazza, Improvements and extensions to a full-potential formulation based on independent fields, AIAA Journal 50 (3).

[30] M. C. Galbraith, S. Allmaras, D. L. Darmofal, A Verification Driven Process for Rapid Development of CFD Software, in: Proceedings of the 53rd AIAA Aerospace Sciences Meeting, 2015. doi:10.2514/6.2015-0818.

[31] M. Davari, R. Rossi, P. Dadvand, I. Lopez, R. Wuchner, A cut finite element method for the solution of the full potential equation with an embedded wake, Computational Mechanics 63 (5) (2018) 821–833.

[32] M. Núñez, I. López, J. Baiges, R. Rossi, An embedded approach for the solution of the full potential equation with finite elements, Computer Methods in Applied Mechanics and Engineering 388 (2022) 114244.

[33] P. Dadvand, R. Rossi, E. Onate, An object-oriented environment for developing finite element codes for multi-disciplinary applications, Archives of Computational Methods in Engineering 17 (3) (2010) 253–297.

[34] F. Johnson, S. S.S., M. Bieterman, R. Melvin, D. Young, J. Bussoletti, C. Hilmes, Tranair: A Full-Potential, Solution-Adaptative, Rectangular Grid-Code for Predicting Subsonic, Transonic, and Supersonic Flows About Arbitrary Configurations, Tech. rep., NASA (1992).

[35] W. Huffman, R. Melvin, D. Young, F. Johnson, J. Bussoletti, Practical Design and Optimization in Computational Fluid Dynamics, AIAA paper.

[36] D. Young, W. Huffman, R. Melvin, M. Bieterman, C. Hilmes, F. Johnson, Inexactness and Global Convergence in Design Optimization, AIAA paper.

[37] R. Melvin, W. Huffman, D. Young, F. Johnson, C. Hilmes, M. Bieterman, Recent Progress in Aerodynamic Design and Optimization, Internatonal Journal for Numerical Methods in Fluids 30 (1999) 205–216.

[38] D. Young, W. Huffman, R. Melvin, C. Hilmes, F. Johnson, Nonlinear Elimination in Aerodynamic analysis and Design Optimization, in: S. Verlag (Ed.), Large-Scale PDE-constrained Optimization, Lecture Notes in Computational Science and Engineering, Biegler, L., 2004, pp. 17–44.

[39] C. Farhat, M. Lesoinne, N. Maman, Mixed explicit/implicit time integration of coupled aeroelastic problems: Three-field formulation, geometric conservation and distributed solution, International Journal for Numerical Methods in Fluids 21 (10) (1995) 807–835. doi:https://doi.org/10.1002/fld.1650211004.

[40] K. Maute, M. Nikbay, C. Farhat, 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) (2003) 911–933. doi:https://doi.org/10.1002/nme.599.

[41] J. L. Steger, B. S. Baldwin, Shock waves and drag in the numerical calculation of isentropic transonic flows, Tech. rep., NASA (1972).

[42] T. L. Holst, Transonic flow computations using nonlinear potential methods, Progress in Aerospace Sciences 36 (1) (2000) 1–61. doi:https://doi.org/10.1016/S0376-0421(99)00010-X.

[43] A. Eberle, A finite volume method for calculating transonic potential flow around wings from the pressure minimum integral, NASA (1978).
URL https://ntrs.nasa.gov/citations/19780019138

[44] M. Hafez, J. South, E. Murman, Artificial compressibility methods for numerical solutions of transonic full potential equation, AIAA Journal 17 (8) (1979) 838–844. doi:10.2514/3.61235.

[45] E. J. Nielsen, M. A. Park, Using an adjoint approach to eliminate mesh sensitivities in computational design, AIAA Journal 44 (5) (2006) 948–953. doi:10.2514/1.16052.

[46] R. P. Dwight, Robust Mesh Deformation using the Linear Elasticity Equations, Journal of Computational Fluid Dynamics 12 (2009) 401–406.

[47] M. Widhalm, J. Brezillon, C. Ilic, T. Leicht, Investigation on adjoint based gradient computations for realistic 3d aero-optimization, in: Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, 2010. doi:10.2514/6.2010-9129.

[48] A. D. Robison, Intel® Threading Building Blocks (TBB), Springer US, 2011, pp. 955–964. doi:10.1007/978-0-387-09766-4_51.

[49] G. Guennebaud, B. Jacob, et al., Eigen v3, http://eigen.tuxfamily.org (2010).

[50] D. M. Beazley, 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, Monterey, California, USA, 1996, p. 15.

[51] D. Thomas, M. Cerquaglia, R. Boman, T. Economon, J. Alonso, G. Dimitriadis, V. Terrapon, CUPyDO - An integrated Python environment for coupled fluid-structure simulations, Advances in Engineering Software 128 (2019) 69–85. doi:https://doi.org/10.1016/j.advengsoft.2018.05.007.

[52] M. Cerquaglia, D. Thomas, R. Boman, V. Terrapon, J.-P. Ponthot, A fully partitioned lagrangian framework for fsi problems characterized by free surfaces, large solid deformations and displacements, and strong added-mass effects, Computer Methods in Applied Mechanics and Engineering 348 (2019) 409–442. doi:https://doi.org/10.1016/j.cma.2019.01.021.

[53] R. E. Bank, D. J. Rose, Global approximate Newton methods, Numerische Mathematik 37 (2) (1981) 279–295. doi:https://doi.org/10.1007/BF01398257.

[54] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, B. A. Naylor, OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization, Structural and Multidisciplinary Optimization 59 (4) (2019) 1075–1104.

[55] D. Kraft, A software package for sequential quadratic programming, Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht, Wiss. Berichtswesen d. DFVLR, 1988.

[56] J. Nocedal, J. W. Stephen, Sequential Quadratic Programming, Springer New York, New York, NY, 2006, pp. 529–562.

[57] C. Geuzaine, J.-F. Remacle, Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities, International Journal for Numerical Methods in Engineering 79 (2009) 1309–1331.

[58] G. K. W. Kenway, G. J. Kennedy, J. R. R. A. Martins, A CAD-Free Approach to High-Fidelity Aerostructural Optimization, in: Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, Fort Worth, TX, 2010, aIAA 2010-9231.

[59] T. W. Sederberg, S. R. Parry, Free-form deformation of solid geometric models, SIGGRAPH Computer Graphics 20 (4) (1986) 151–160. doi:10.1145/15886.15903.

[60] G. J. Kennedy, A parallel finite-element framework for large-scale gradient-based design optimization of high-performance structures, Finite Elements in Analysis and Design 87 (2014) 56–73.

[61] G. J. Kennedy, Strategies for adaptive optimization with aggregation constraints using interior-point methods, Computers and Structures 153 (2015) 217–229.

[62] G. J. Kennedy, J. E. Hicken, Improved constraint-aggregation methods, Computer Methods in Applied Mechanics and Engineering 289 (2015) 332–354.

[63] K. Boopathy, G. J. Kennedy, Parallel Finite Element Framework for Rotorcraft Multibody Dynamics and Discrete Adjoint Sensitivities, AIAA Journal 57 (8) (2019) 3159–3172.

[64] J. F. Kiviaho, G. J. Kennedy, Efficient and Robust Load and Displacement Transfer Scheme Using Weighted Least Squares, AIAA Journal 57 (5) (2019) 2237–2243.

[65] V. Schmitt, F. Charpin, Pressure distributions on the ONERA-M6-wing at transonic Mach numbers, Experimental data base for computer program assessment 4.

[66] M. Drela, M. Giles, Viscous-Inviscid Analysis of Transonic and Low Reynolds Number Airfoils, AIAA Journal 25 (10) (1987) 1347–1355.

[67] A. Bilocq, Implementation of a viscous-inviscid interaction scheme in a finite element full potential solver, Master's thesis, University of Liège (June 2020).
URL https://orbi.uliege.be/handle/2268/252195

[68] P. Dechamps, Improvement of the viscous-inviscid interaction method implemented in DARTFlo, Master's thesis, University of Liège (January 2022).

[69] G. K. Kenway, J. R. R. A. Martins, Multi-point High-fidelity Aerostructural optimization of a Transport Aircraft Configuration, Journal of Aircraft 51 (1) (2014) 144–160.

[70] Z. Lyu, G. K. Kenway, C. Paige, J. R. R. A. Martins, Automatic Differentiation Adjoint of the Reynolds-Averaged Navier-Stokes Equations with a Turbulence Model, in: 21st AIAA Computational Fluid Dynamics Conference, 2013. doi:10.2514/6.2013-2581.

[71] G. Kreisselmeier, R. Steinhauser, Systematische Auslegung von Reglern durch Optimierung eines vektoriellen Gütekriteriums, Automatisierungstechnik 27 (1-12) (1979) 76–79.

[72] C. Wood, A. J. Gil, O. Hassan, J. Bonet, Partitioned Block-Gauss–Seidel coupling for dynamic fluid–structure interaction, Computers and Structures 88 (23) (2010) 1367–1382.

[73] A. Lambe, J. R. R. A. Martins, Extensions to the Design Structure Matrix for the Description of Multidisciplinary Design, Analysis, and Optimization Processes, Structural and Multidisciplinary Optimization.

Appendix A. Numerical tools

The overall procedure used to solve the optimization problem 1 is illustrated in Figure A.10 using the eXtended Design Structure Matrix (XDSM) tool [73] [12], and the different software used in the present work are briefly described in the next paragraphs.

The optimization process is solved using `MPHYS` [13], a modular multiphysics simulation package built on top of the `OpenMDAO` framework [54] [14], that particularizes it to analysis and optimization calculations usually performed during aircraft design, such as aerodynamic analysis, aerostructural optimization, etc. More specifically, `MPHYS` automatizes the connection of the different software components used in the optimization process and required by `OpenMDAO`. The code is open-source and is developed jointly by NASA [15] and the MDO Lab of the University of Michigan [16].

The aerodynamic meshes are created using `Gmsh` [57] [17], a three-dimensional unstructured mesh generation software. The code is actively developed at the University of Liège.

The aerodynamic surfaces are parametrized using `pyGeo` [58] [18], a set of python tools providing geometry manipulation features. More specifically, `pyGeo` allows to parametrize and constrain a geometry, represented by a surface grid, using the free form deformation technique [59]. It then allows to deform that geometry according to some design parameters, and to provide the sensitivities of the surface deformation with respect to these parameters. The code is open-source and developed by the MDO Lab.

The aerodynamic full potential equation is solved using `DART` [21] [19], a finite element code designed to quickly compute inviscid transonic flows. `DART` allows to calculate the aerodynamic loads and to provide the sensitivities of flow functionals using analytic gradients. The volume mesh deformation procedure in also embedded in `DART`. The code is open-source, written in C++ and interfaced in python, and developed at the University of Liège.

The structural equations are solved using `TACS` [60, 61, 62, 63] [20], a finite element code dedicated to the analysis of composite structures. `TACS` allows to compute the deflection of the structural model when it is subjected to external loads, and to provide the sensitivities of structural functionals using analytic gradients. The code is written in C++ and interfaced in python. It is developed both by the SMDO group at Georgia Tech [21] and the MDO Lab.

Since the structure and the aerodynamic equations are solved on different grids, the relevant data need to be interpolated from one mesh to another. `MELD` [64] [22] implements such a transfer scheme, and is used to pass the

[12] https://github.com/mdolab/pyXDSM, accessed March 2022.
[13] https://github.com/OpenMDAO/mphys, accessed March 2022.
[14] https://openmdao.org/, accessed March 2022.
[15] https://www.nasa.gov/, accessed March 2022.
[16] https://mdolab.engin.umich.edu/, accessed March 2022.
[17] http://gmsh.info, accessed March 2022.
[18] https://github.com/mdolab/pygeo, accessed March 2022.
[19] https://gitlab.uliege.be/am-dept/dartflo, accessed March 2022.
[20] https://github.com/smdogroup/tacs, accessed March 2022.
[21] http://gkennedy.gatech.edu/, accessed March 2022.
[22] https://github.com/smdogroup/funtofem, accessed March 2022.

aerodynamic loads to the structural mesh and the structural displacements to the aerodynamic mesh. The code is written in C++ and interfaced in python, and is developed by the SMDO group.

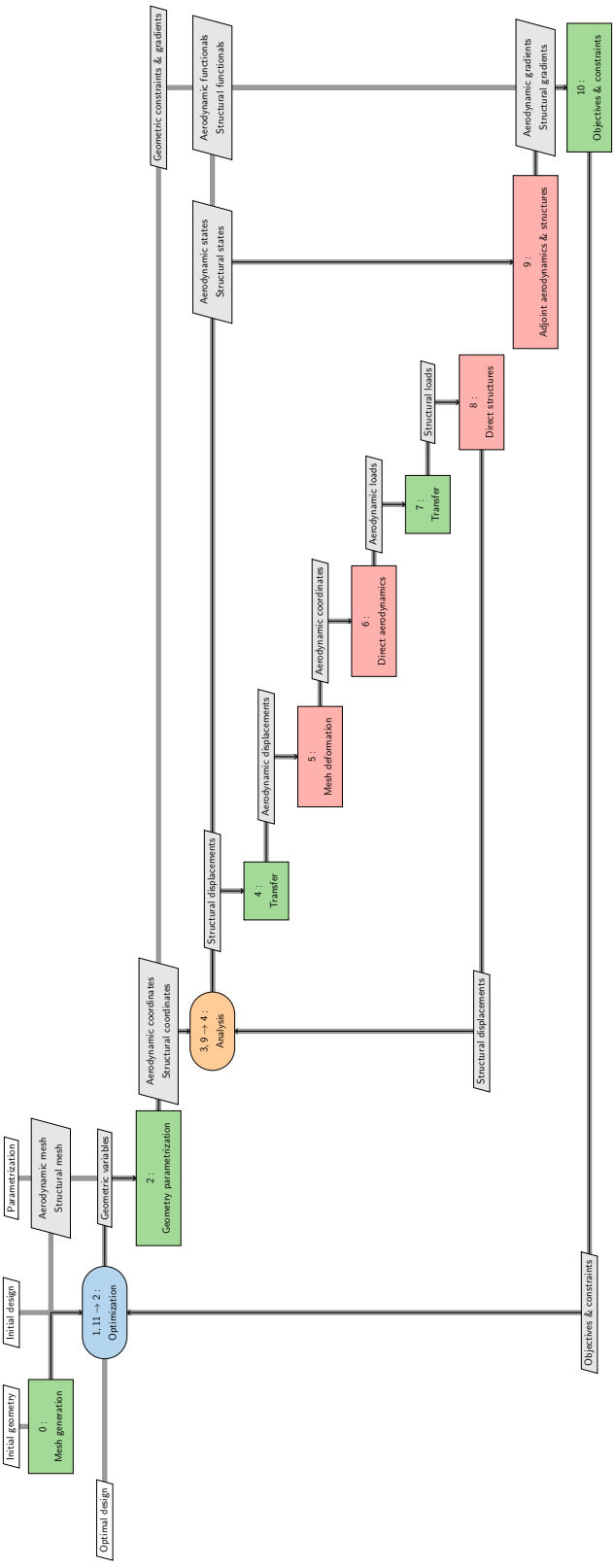


Figure A.10: eXtended Design Structure Matrix of a generic aerostrucutral optimization process. The processes illustrated in green are explicit while those depicted in red are implicit.

Appendix B. N^2 diagrams

The N^2 diagram representing the optimization process of the Onera M6 case is depicted in Figure B.11.

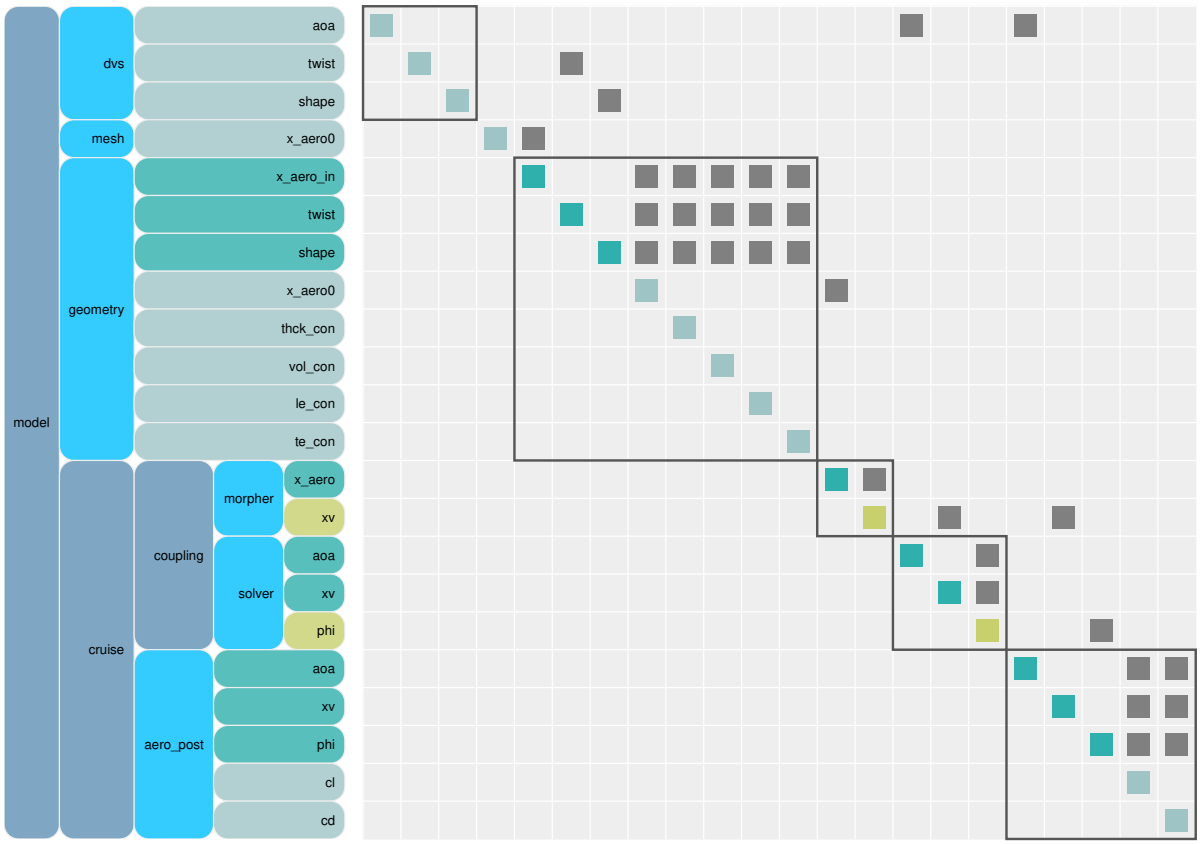


Figure B.11: N^2 diagram for the shape optimization of the Onera M6 wing. The input variables are displayed in green, the output variables are in grey, and the implicit (state) variables are in yellow.

The N^2 diagram representing the optimization process of the RAE case is depicted in Figure B.12. Note that the variables have been collapsed due to their large number.

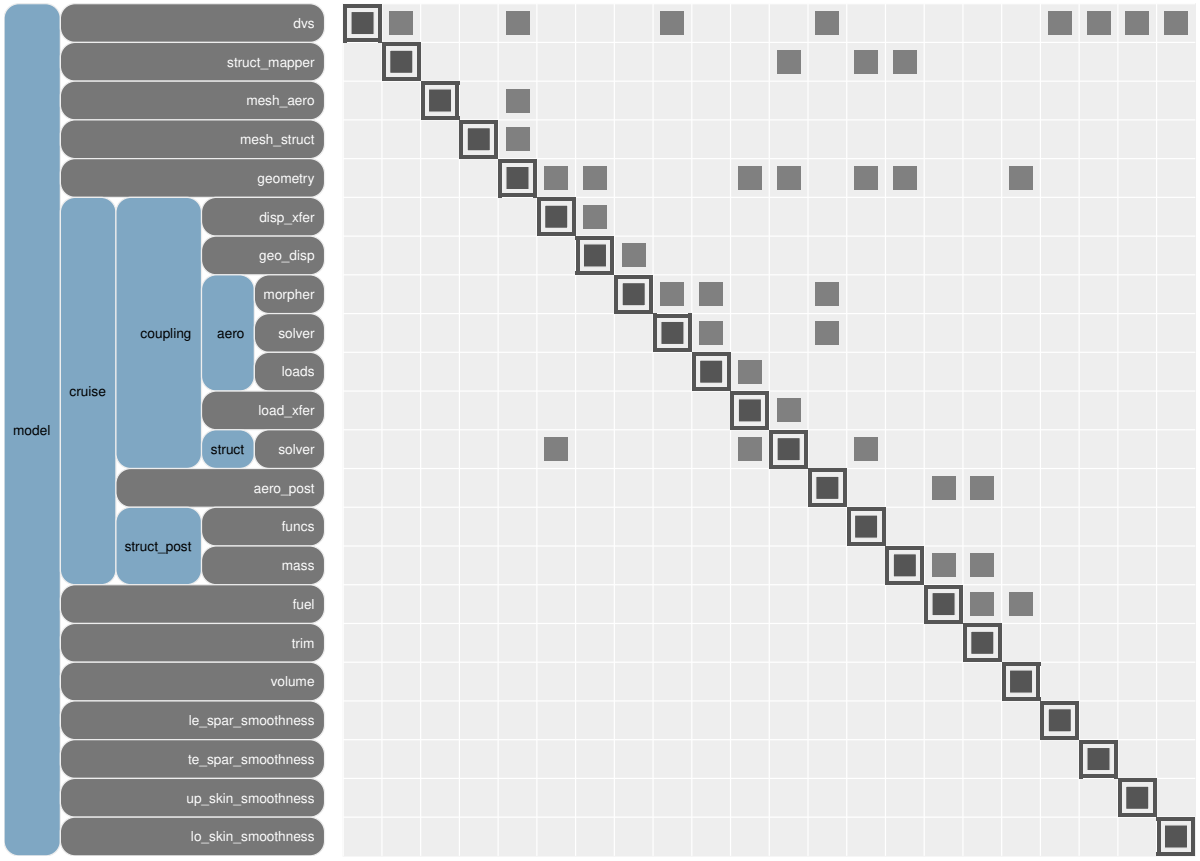


Figure B.12: N^2 diagram for the shape optimization of the RAE wing.