Efficient Aerodynamic Optimization of Aircraft Wings

Pedro M. V. Rodrigues and André C. Marta(✉)

Center for Aerospace Science and Technology, IDMEC, Instituto Superior Técnico, Universidade de Lisboa, Av. Rovisco Pais 1, 1049-001 Lisbon, Portugal
{pedro.miguel.verissimo.rodrigues,andre.marta}@tecnico.ulisboa.pt

Abstract. Multidisciplinary design and optimization is a promising methodology for the efficient design of complex systems, in particular when it combines coupled analyses with gradient-based optimization techniques. In this case, it requires the derivatives evaluation of the functions of interest with respect to the design variables, which is the most demanding computational task in the process, so the goal of this work is to develop an efficient optimization framework to solve aerodynamic design problems using exact gradient information. To this end, the aerodynamic model based on the panel method is reformulated into five smaller modules, in which the respective sensitivity analysis blocks are constructed using exact gradient estimation methods: automatic differentiation, symbolic differentiation and the adjoint method. After the aerodynamic and corresponding sensitivity analysis tools are verified numerically, aerodynamic optimization problems are solved using the new tool with remarkable success since, when compared to the finite-differences method, the optimization time can be reduced by 90%.

Keywords: Gradient-based optimization · Sensitivity analysis
Automatic differentiation · Adjoint method

1 Introduction

Multidisciplinary Design and Optimization (MDO) is a promising tool to design complex systems, such those found in the aerospace field, combining optimization procedures with coupled multidisciplinary analysis [7]. Since hundred or even thousand design variables are usually required to faithfully parametrize the complete system, the use of gradient-based optimization techniques are imperative to efficient optimization due to faster convergence rates when comparing with heuristic and gradient free methods. However, the performance of these type of algorithms depends heavily on how efficiently the sensitivities/derivatives of the interest functions with respect to the design variables are calculated. The adjoint method is probably one of the most attractive to efficient sensitivity analysis since the derivatives may be calculated exactly and almost independently of the number of the system's inputs.

H. C. Rodrigues et al. (Eds.): *EngOpt 2018 Proceedings of the 6th International Conference on Engineering Optimization*, pp. 897–910, 2019.
https://doi.org/10.1007/978-3-319-97773-7_78

The objective of this work is to develop an efficient aerodynamic optimization tool to be incorporated in a wing aerostructural design framework [1]. The wing's structure was modeled using a 3D finite-element model, applied to the neutral axis and the aerodynamic loads were calculated using a panel method code [3]. Almeida took advantage of the framework's static aerostructural capabilities to minimize the wing mass, subject to lift and stress constraints, and concluded that an improved design was achieved. However, the optimization process was conducted slowly since the sensitivity analysis was performed inefficiently and inaccurately, using finite-differences (FD).

This work finds its motivation in solving the issue of inefficient aerodynamic optimization by providing an efficient sensitivity analysis framework to the panel method code implemented by Cardeira [3] using a gradient-based algorithm with an adjoint method for sensitivity analyses.

2 Optimization Methods

Following a deterministic approach, a typical engineering optimization problem may be expressed as nonlinear programming (NLP) as [2]

$$\begin{array}{lll} \text{minimize} & f(\mathbf{x}) & \\ \text{w.r.t.} & \mathbf{x} \in \mathbb{R}^n & \\ \text{subject to} & g_i(\mathbf{x}) \leq 0 & \text{for } i = 1, ..., m \\ & h_j(\mathbf{x}) = 0 & \text{for } j = 1, ..., \ell \\ & \mathbf{x}^L \leq \mathbf{x} \leq \mathbf{x}^U & \end{array} \tag{1}$$

where $\mathbf{x}$ is the design vector, or the independent variables, $\mathbf{x}^L$ and $\mathbf{x}^U$ are the lower and upper bounds, f is the objective function, h_j and g_i are the equality and inequality constraints, respectively.

Gradient-Based (GB) methods are usually preferable due to faster convergence rates and clear stopping criterion, but they require the gradient of both objective and constraint functions. The methodology to solve these problems consists in an iterative cycle comprised of two steps: find a descent/feasible direction based on gradient information; minimize the objective function in that direction (line search).

Among the several constrained GB methods, the Sequential Quadratic Programming [6] method will be used since it presents some advantages, including: the initial point may be unfeasible, only gradients of active constraints are needed, higher rate of convergence when comparing with similar methods and it is already implemented in MATLAB®.

3 Sensitivity Analysis

A common requirement among gradient-based optimization methods is the evaluation of the gradient of both objective and constraint functions. The performance of such methods greatly depends on how efficiently those gradients are calculated, and the following sections survey the sensitivity analysis methods available.

3.1 Symbolic Differentiation

Symbolic differentiation (SD) applies the rules of differentiation using computational software. It is restricted to explicit functions and, therefore, may become impracticable to be implemented in very large problems. Some tools are available include the `diff()` function from the *Symbolic Math Toolbox* in MATLAB®.

3.2 Finite-Differences Method

Finite-difference formulas approximate the derivative of a function using a quotient of a difference and they may be obtained through Taylor-series expansions. An example is the forward finite-difference (FFD) formula,

$$\frac{\partial F_j(\mathbf{x}_0)}{\partial x_i} = \frac{F_j(\mathbf{x}_0 + \mathbf{e}_i h) - F_j(\mathbf{x}_0)}{h} + \mathcal{O}(h)\,, \tag{2}$$

where $\mathbf{F} = [F_1, ..., F_m]^T$ are dependent functions and $\mathbf{x} = [x_1, ..., x_n]^T$ are independent variables. The truncation error is proportional to the step size h. If higher precision is desired, Taylor-series expansions may be combined.

These formulas are easy to implement as they can be used without detailed knowledge of the system. However, they suffer from errors of truncation and subtractive cancellation, and their cost depends linearly on the size of $\mathbf{x}$.

3.3 Complex-Step Derivative

The complex-step derivative (CSD) estimates the first derivative of a function using complex-variable calculus by considering Taylor-series expansion of $\mathbf{F}$ in the imaginary axis direction and taking the imaginary part. The resulting formulas do not suffer from the subtraction cancellation but complex algebra computationally expensive and is not supported by all programming languages.

3.4 Semi-analytical Methods

According to Peter and Dwight [5], semi-analytical methods such as the direct and the adjoint methods are the most efficient to sensitivity analysis. Consider again the vector valued function $\mathbf{F}$, which depends explicitly on $\mathbf{x}$ and implicitly on the state vector $\mathbf{y} = [y_1, ..., y_k]^T$, whose relation between the independent variables and the state vector is given by a system of residual equations

$$\mathbf{R}(\mathbf{x}, \mathbf{y}(\mathbf{x})) = 0 \tag{3}$$

According to the chain-rule for the total derivative of the function with respect to the independent variables, and since Eq. (3) must always be verified, results after some algebra

$$\frac{\mathrm{d}\mathbf{F}}{\mathrm{d}\mathbf{x}} = \frac{\partial \mathbf{F}}{\partial \mathbf{x}} + \frac{\partial \mathbf{F}}{\partial \mathbf{y}}\frac{\mathrm{d}\mathbf{y}}{\mathrm{d}\mathbf{x}} = \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \underbrace{-\frac{\partial \mathbf{F}}{\partial \mathbf{y}}\left[\frac{\partial \mathbf{R}}{\partial \mathbf{y}}\right]^{-1}}_{[\psi]^T}\left[\frac{\partial \mathbf{R}}{\partial \mathbf{x}}\right] \tag{4}$$

It is possible to assign the transpose of an adjoint matrix to the first two matrices of the second term, in the right hand side, as suggested by the under bracket. Consequently, $\frac{\mathrm{d}\mathbf{F}}{\mathrm{d}\mathbf{x}}$ may be calculated if the adjoint matrix is first solved for. This approach is the adjoint method and it is best suited if the number of outputs is smaller than the number of inputs ($m < n$).

3.5 Automatic Differentiation

A computer program with n inputs, l intermediate variables and m outputs can be decomposed into elementary functions such that each computer variable t_i depends only on the previous assigned variables: $t_i = T_i(t_1, ..., t_{i-1})$, where T_i is an elementary function [4]. The AD derivatives can be propagated in two ways: forward mode (FM) and reverse mode (RM). A sweep in FM corresponds to fix j and increment i from 1 to $n + l + m$, while a sweep in RM corresponds to fix i while j changes from $n + l + m$ to 1,

$$\text{FM: } \frac{dt_i}{dt_j} = \delta_{ij} + \sum_{k=j}^{i-1} \frac{\partial T_i}{\partial t_k}\frac{dt_k}{dt_j} \qquad \text{RM: } \frac{dt_i}{dt_j} = \delta_{ij} + \sum_{k=j+1}^{i} \frac{dt_i}{dt_k}\frac{\partial T_k}{\partial t_j} \tag{5}$$

The FM corresponds to obtain a column from the matrix $\frac{\mathrm{d}t_i}{\mathrm{d}t_j}$, while RM second corresponds to obtain a row. Therefore, RM is more efficient than FM if $m < n$, and the opposite is true if $m > n$.

4 Aerodynamic Model and Framework

4.1 Panel Method

The aerodynamic tool [3] is an implementation of the panel method, a numerical technique to solve inviscid potential flows around bodies of arbitrary shape by solving the Laplace equation, $\nabla^2\phi = 0$, subject to impermeable wall and farfield boundary conditions, $\nabla\phi.\mathbf{n} = 0$ and $\lim_{\mathbf{r}\to\infty}(\nabla\phi - \mathbf{V}_\infty) = 0$. The technique consists in distributing and finding the intensities of singularities along the body's surface to consequently determine the velocity field. The velocity potential ϕ in an arbitrary point $\mathbf{P}$ in the body's surface is a function of the singularities intensities and respective distance to point $\mathbf{P}$,

$$\frac{1}{4\pi}\int_{Body+Wake} \mu\mathbf{n}.\nabla\left(\frac{1}{r}\right)dS - \frac{1}{4\pi}\int_{Body} \sigma\left(\frac{1}{r}\right)dS = 0, \tag{6}$$

where σ and μ are the source and doublet intensities, and r is a distance from an arbitrary point in the body surface. σ is known since $\sigma = \mathbf{n}.\nabla\phi_\infty$

Since Eq. (6) holds for every point in the surface, the geometry can be discretized into panels, each containing a collocation point, resulting linear system of equations for the doublet intensities,

$$A_\mu \boldsymbol{\mu} = \mathbf{b}. \tag{7}$$

4.2 Aerodynamic Framework

The aerodynamic tool was reformulated into five modules, as schematically depicted in Fig. 3, as detailed next.

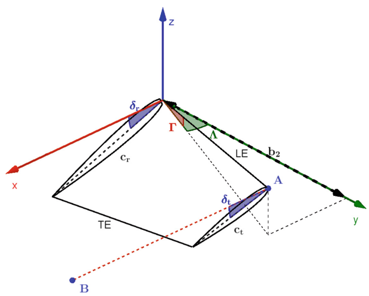

Fig. 1. Geometrical description of the aircraft half wing

Wing Parametrization. Translates the design variables $\mathbf{x_{DV}}$, into a discrete set of points representing the wing's geometry. According to Fig. 1, the exterior wing shape is defined by the leading edge, defined by the semi span length $b/2$, sweep angle Λ and dihedral angle Γ. The wing root and tip chords are c_r and c_t are related by the taper ratio $\lambda = c_t/c_r$ and their twist angles are δ_r and δ_t, respectively. Local twist and chord are assumed to vary linearly along the span. The wing is then discretized in the spanwise direction according to user specifications and an airfoil shape is assigned. The airfoil shapes are parametrized using control points ($\mathbf{WP_i}$) of four Bezier curves [8], In addition, the module also outputs the wing area S and the mean aerodynamic chord MAC.

Panels Definition. Creates the panels and calculates associated quantities. It outputs the panel's corner points **PP**, the collocation points **CP**, the panel's areas DS and the panel's basis vectors **LV**, as illustrated in Fig. 2.

Change of Basis. Given the **CP**, **PP** and **LV** vectors, the corner points **PP** are transformed to their own panel's frame of reference **PLP** since the influence coefficients may then be easily calculated.

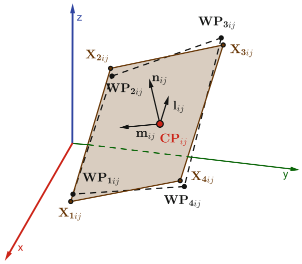

Fig. 2. Panel construction through a set of four non-coplanar adjacent points.

Aero Solver. Receives as inputs the vectors **LPP**, **CP**, **LV** and additionally, the free-stream airspeed V_∞ and angle-of-attack α. It then assembles and solves the linear system in Eq. (7) for the aerodynamic solution $\boldsymbol{\mu}$ and residuals $\mathbf{R}$. Since the wing could be assumed symmetric with respect to plane Oxz of Fig. 1, the method of images was used to diminish the computational cost of the implementation. The residuals may be written appropriately for the computational mesh as

$$R_{ij} = \sum_{n=1}^{N} \left[\sum_{m=1}^{M-1} \left(\mathcal{C}_{ijmn} \mu_{mn} + \mathcal{B}_{ijmn} \sigma_{mn} \right) + \mathcal{C}_{ijMn} \left(\mu_{(M-1)n} - \mu_{1n} \right) \right] = 0, \quad (8)$$

where $\mathcal{C}_{ijmn}$ and $\mathcal{B}_{ijmn}$ are the doublet and source influences of panel (m, n) on panel (i, j). Since the method of images was used, it can be proven that these quantities depend only on the corner points of panel (m, n), the (i, j) panel's collocation point and respective image, all written in (m, n) panel's frame of reference. The source intensities are easily known since $\sigma_{mn} = \mathbf{n}_{mn}.\mathbf{V}_\infty$.

Post-processing. Calculates the aerodynamic coefficients from the flow solution. It takes the angle-of-attack α, the vectors **PP**, **CP**, **LV**, DS, the aerodynamic solution $\boldsymbol{\mu}$, and MAC and S as inputs. Then computes the velocity on the panel (i, j) as

$$\mathbf{V}_{ij} = (V_{\infty l}, V_{\infty m}, V_{\infty n})_{ij} + (v_l, v_m, v_n)_{ij}, \quad (9)$$

where the first term is the free-stream and the second is the perturbation velocity, function of the aerodynamic solution μ_{ij}, both written in (i, j) panel's frame of reference. Next, the pressure coefficient is obtained as

$$C_{p_{ij}} = 1 - \frac{|\mathbf{V}_{ij}|^2}{|\mathbf{V}_\infty|^2}. \quad (10)$$

Knowing the pressure coefficients on each panel, the aerodynamic coefficients are calculated by numerical integration as

$$C_L = -\frac{2}{S} \sum_{i}^{M-1} \sum_{j}^{N} C_{p_{ij}} DS_{ij} \left(\mathbf{n}_{ij}.\mathbf{e}_L \right), \quad (11a)$$

$$C_D = -\frac{2}{S}\sum_{i}^{M-1}\sum_{j}^{N} C_{p_{ij}} DS_{ij}\,(\mathbf{n}_{ij}.\mathbf{e}_D), \tag{11b}$$

$$\mathbf{C_M} = -\frac{2}{S.l_0}\sum_{i}^{M-1}\sum_{j}^{N} C_{p_{ij}} DS_{ij}\,(\mathbf{CP}_{ij} \times \mathbf{n}_{ij}), \tag{11c}$$

where $\mathbf{e}_L$ and $\mathbf{e}_D$ are the unit vectors in the directions of the lift and drag, respectively, function of the angle-of-attack α. The variable l_0 is an appropriated reference length, which corresponds to MAC and b for the pitching and rolling moment coefficients, respectively.

5 Sensitivity Analysis Framework

The sensitivity analysis framework is depicted in Fig. 3. Each module presented before has its own sensitivity analysis, where the jacobians of the outputs with respect to the inputs are calculated. Then, the sensitivities of the functions of interest, f, with respect to the design variables, $\mathbf{x_{DV}}$, are calculated propagating the intermediate jacobians according to the chain-rule of differential calculus.

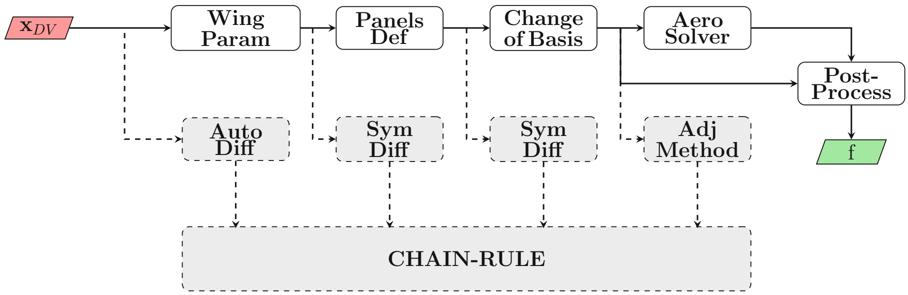

Fig. 3. Flowchart illustrating the sensitivity analysis framework

Mathematical Formulation. First, the vector of design variables must be defined. It is composed of three segments: one containing planform related parameters, another containing the control points that parametrize the airfoil shape in each cross section, and the angle-of-attack,

$$\mathbf{x_{DV}} = \left[\mathbf{x_{geo}}^T\ \mathbf{x_{airfoil}}^T\ \alpha\right]^T, \tag{12}$$

where each of the right hand side vectors are

$$\mathbf{x}_{geo} = \left[\Lambda\ \Gamma\ \delta_r\ \delta_t\ b\ c_r\ \lambda\right]^T \tag{13a}$$

and

$$\mathbf{x_{airfoil}}^T = \bigcup_j \left[A_x \ldots L_x\ A_y \ldots L_y\right]_j \quad \forall j \in \{1, \ldots, N+1\} \tag{13b}$$

Adjoint Method. Considering both the inputs of the *Aero Solver* and *Post Process* modules, it can be defined an intermediate information as

$$\mathbf{x_2} = \left[\mathbf{x_1}^T \; DS^T \; \mathbf{LPP}^T \; S \; \mathrm{MAC} \; \alpha \; V_\infty\right]^T , \tag{14}$$

where $\mathbf{x_1}$ is defined as

$$\mathbf{x_1} = \left[\mathbf{PP}^T \; \mathbf{CP}^T \; \mathbf{LV}^T\right]^T , \tag{15}$$

such that $\mathbf{R} = \mathbf{R}(\mathbf{x_2}, \boldsymbol{\mu})$ and $f = f(\mathbf{x_2}, \boldsymbol{\mu})$. Since the size of $\mathbf{x_2}$ is much larger than the size of f, the adjoint method is the best suited to calculate the sensitivity of f with respect to $\mathbf{x_2}$ as

$$\frac{\mathrm{d}f}{\mathrm{d}\mathbf{x_2}} = \frac{\partial f}{\partial \mathbf{x_2}} + \left[\boldsymbol{\psi}\right]^T \frac{\partial \mathbf{R}}{\partial \mathbf{x_2}} \tag{16a}$$

$$\left[\frac{\partial \mathbf{R}}{\partial \boldsymbol{\mu}}\right]^T \left[\boldsymbol{\psi}\right] = - \left[\frac{\partial f}{\partial \boldsymbol{\mu}}\right]^T \tag{16b}$$

Chain-Rule. The chain-rule is used to ultimately calculate the sensitivities of the interest functions with respect to the design variables as

$$\frac{\mathrm{d}f}{\mathrm{d}\mathbf{x_{DV}}} = \frac{\mathrm{d}f}{\mathrm{d}\mathbf{x_2}} \frac{\mathrm{d}\mathbf{x_2}}{\mathrm{d}\mathbf{x_{DV}}} \tag{17}$$

where

$$\frac{\mathrm{d}\mathbf{x_2}}{\mathrm{d}\mathbf{x_{DV}}} = \left[\left[\frac{\mathrm{d}\mathbf{x_1}}{\mathrm{d}\mathbf{x_{DV}}}\right]^T \left[\frac{\mathrm{d}DS}{\mathrm{d}\mathbf{x_{DV}}}\right]^T \left[\frac{\mathrm{d}\mathbf{LPP}}{\mathrm{d}\mathbf{x_{DV}}}\right]^T \left[\frac{\partial S}{\partial \mathbf{x_{DV}}}\right]^T \left[\frac{\partial \mathrm{MAC}}{\partial \mathbf{x_{DV}}}\right]^T \left[\frac{\partial \alpha}{\partial \mathbf{x_{DV}}}\right]^T [0]\right]^T \tag{18}$$

Note that some entries were already replaced by the respective partial derivatives where explicit dependence is observed. The remaining derivatives are assembled according to the variable dependencies presented for each module,

$$\frac{\mathrm{d}\mathbf{x_1}}{\mathrm{d}\mathbf{x_{DV}}} = \frac{\partial \mathbf{x_1}}{\partial \mathbf{WP}} \frac{\partial \mathbf{WP}}{\partial \mathbf{x_{DV}}} \tag{19}$$

$$\frac{\mathrm{d}DS}{\mathrm{d}\mathbf{x_{DV}}} = \frac{\partial DS}{\partial \mathbf{WP}} \frac{\partial \mathbf{WP}}{\partial \mathbf{x_{DV}}} \tag{20}$$

$$\frac{\mathrm{d}\mathbf{LPP}}{\mathrm{d}\mathbf{x_{DV}}} = \frac{\partial \mathbf{LPP}}{\partial \mathbf{x_1}} \frac{\partial \mathbf{x_1}}{\partial \mathbf{WP}} \frac{\partial \mathbf{WP}}{\partial \mathbf{x_{DV}}} \tag{21}$$

Sensitivities of Wing Parametrization. This module calculates three Jacobians: $\partial S/\partial \mathbf{x_{DV}}$, $\partial MAC/\partial \mathbf{x_{DV}}$ and $\partial \mathbf{WP}/\partial \mathbf{x_{DV}}$. The non-zero derivatives of the first two are calculated by hand since the expressions are simple. The last Jacobian is calculated with the aid of automatic differentiation using the forward mode since the number of outputs is larger than the number of inputs for the mesh sizes expected to be used in the optimization problems. The implementation was benchmarked with the complex-step derivative to compare both the results and performance. As it may be observed in Table 1, the AD implementation is faster, with savings up to 40.5%.

Table 1. Computational cost of the *Wing Parametrization* sensitivity analysis module.

No. panels	50	200	450	800	1250	1800
CSD time [s]	0.639	3.011	9.332	19.815	38.970	64.553
AD time [s]	0.380	1.151	4.191	10.400	27.219	55.999
Savings [%]	40.5	61.8	55.1	47.5	30.2	13.3

Sensitivities of Panels Definition. The sensitivity analysis of *Panels Definition* corresponds to the calculation of four jacobians: $\partial\mathbf{PP}/\partial\mathbf{WP}$, $\partial\mathbf{CP}/\partial\mathbf{WP}$, $\partial\mathbf{LV}/\partial\mathbf{WP}$ and $\partial DS/\partial\mathbf{WP}$. These matrices were all obtained with the aid of symbolic differentiation. Although the procedure was quite lengthy, this approach allowed to obtain huge computational savings since only different from zero partial derivatives were calculated, algebraic simplifications were made and MATLAB® vectorization techniques were applied. According to Table 2, the implementation can be about 1000 times faster, comparing with the CSD and AD.

Since this approach is very susceptible to errors, the implementation was benchmarked with AD and the complex-step derivative. Figure 4 shows the absolute difference for each entry of $\frac{\partial\mathbf{CP}}{\partial\mathbf{WP}}$ when benchmarked with the CSD and AD. As observed, the difference is bounded and really small. Similar results were obtained for the remaining Jacobians thus verified the module.

Table 2. Computational cost of the *Panels Definition* sensitivity analysis module.

No. panels	200	450	800	1250	1800
CSD time [s]	20.05006	109.4079	289.9539	761.1905	2821.0137
AD time [s]	22.93737	148.1591	599.1266	2296.589	8623.2376
SD time [s]	0.117399	0.294621	0.554629	1.005761	1.822698
Savings CSD [%]	99.4	99.7	99.8	99.9	99.9
Savings AD [%]	99.5	99.8	99.9	100	100

Sensitivities of Change of Basis. The Jacobians $\partial\mathbf{LPP}/\partial\mathbf{PP}$, $\partial\mathbf{LPP}/\partial\mathbf{CP}$ and $\partial\mathbf{LPP}/\partial\mathbf{LV}$ and $\partial\mathbf{LPP}/\partial\mathbf{x_1}$ were obtained by hand differentiation since the expressions were simple. No benchmark is provided here, although the module's verification was indeed performed.

Sensitivities of Aero Solver. Given the inputs of *Aero Solver* module, the respective sensitivity analysis corresponds to the calculation of six Jacobians that are used to construct $\partial\mathbf{R}/\partial\mathbf{x_2}$ and $\partial\mathbf{R}/\partial\boldsymbol{\mu}$, that are used in the adjoint method in Eqs. (16a) and (16b). The sensitivity analysis corresponds to differentiate

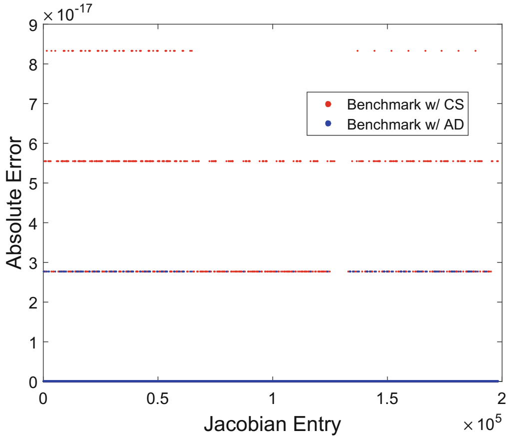

Fig. 4. Absolute difference for all entries of $\partial\mathbf{CP}/\partial\mathbf{WP}$

Eq. (8) with respect to the inputs of Aero Solver, that was performed by hand with the aid of symbolic differentiation. The reason to follow this methodology was the same as for the sensitivity analysis of *Panels Definition*. The benefits in computational terms are described in Table 3, where savings of about 99.5% were observed, when comparing the followed approach with the forward mode of AD.

Table 3. Computational cost of the *Aero Solver* sensitivity analysis module.

No. panels	50	200	450
AD time [s]	749.964	8653.022	40582.198
SD time [s]	3.796	40.797	222.411
Savings [%]	99.5	99.5	99.5

Since obtaining the derivatives using this approach is very susceptible to programming errors, a benchmark with AD differentiation was performed. Figure 5 presents the absolute difference for all entries of $\partial\mathbf{R}/\partial\mathbf{LPP}$, taking the respective values calculated using AD as reference. As observed, the differences are bounded and small. A similar analysis was performed for all the produced Jacobians and the results were similar, proving that the implementation was correct.

Sensitivities of Post Process. The sensitivity analysis of *Post Process* corresponds to calculate the Jacobians $\partial f/\partial \mathbf{x_2}$ and $\partial f/\partial \boldsymbol{\mu}$. These matrices, required to the adjoint method in Eqs. (16a) and (16b), were calculated using the reverse mode of automatic differentiation since the number of inputs is much higher

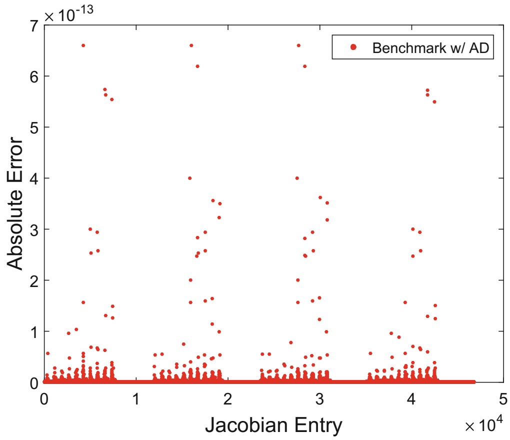

Fig. 5. Absolute difference for all entries of $\partial\mathbf{R}/\partial\mathbf{LPP}$

than the number of outputs. Furthermore, the number of outputs is, at most, equal to five, corresponding to all the aerodynamic coefficients the program may compute.

Benchmark with Finite Differences. After the sensitivity analysis framework had been constructed, it was benchmarked with the finite-differences method to guarantee that the program was free of programming errors and also to measure its performance. Figure 6 shows the absolute difference of each entry of the aerodynamic coefficients sensitivities with respect to the design variables, when compared to the finite-differences method with a step size of $h = 10^{-7}$. As observed, the difference is at most of $\mathcal{O}(10^{-6})$, thus verifying the framework's results. On the other hand, Table 4 shows the time spent by the sensitivity analysis framework and the implementation using FD, taking the time spent by the aerodynamic model as reference, for an increasing number of design variables. As observed, using the sensitivity analysis framework translates into increased time savings for increasing number of design variables. This result allows to conclude that the new tool is much more efficient than the implementation of FD for accurate wing description.

6 Aerodynamic Optimization

An illustrative optimization problem of a full wing was solved to demonstrate the benefits of gradient-based optimization using efficient sensitivity analysis. The problem was solved using both the new sensitivity analysis framework (FW) and the finite-difference (FD) method, to compare the required computational time, the number of iterations and the number of function evaluations.

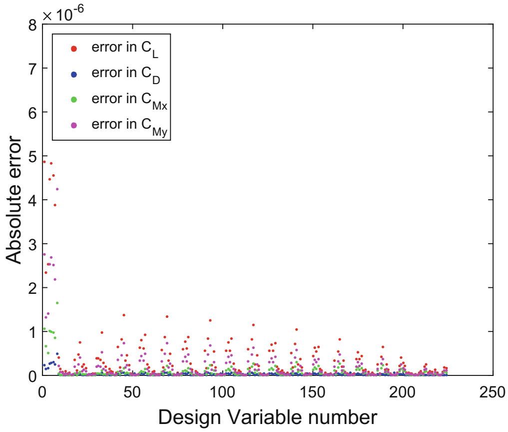

Fig. 6. Absolute difference of the aerodynamic coefficients sensitivities w.r.t. $\mathbf{x_{DV}}$, benchmarked with FD

Table 4. Runtime comparison for increasing number of design variables

No. panels	32	64	128	192	240
No. DV	80	128	224	320	392
t_{model} [s]	0.083	0.239	0.876	1.946	3.037
t_{sens}/t_{model} [–]	41.65	34.56	27.77	25.07	24.01
t_{FD}/t_{model} [–]	61.47	120.07	224.78	316.24	390.19
Savings [%]	32.2	71.1	87.6	92.1	93.8

The lift and pitching moment constrained wing optimization problem was expressed as

$$\begin{aligned} &\text{minimize} && C_D \\ &\text{w.r.t.} && \mathbf{x_{DV}} \\ &\text{subject to} && C_L = 0.3\,, \quad S = S_0\,, \quad C_M = C_{M_0} \\ &&& \mathbf{x}^L \le \mathbf{x_{DV}} \le \mathbf{x}^U \end{aligned} \tag{22}$$

where $\mathbf{x_{DV}}$ includes all design parameters, as defined in Eq. (12).

Both the baseline and optimized wing configurations are depicted in Fig. 7.

The output functions and part of the design vector baseline and optimized values are presented in Table 5, for both approaches of sensitivity analysis. The drag was reduced in 72% again due to increased aspect ratio, lift reduction and lift redistribution. The performance of both approaches was also measured and presented in Table 6. As observed, the number of function evaluations using the new framework is much less when compared to finite-differences. As a consequence, the optimization time is considerably much shorter, about 9 times faster than the implementation using finite-differences, clearly proving its efficiency for accurate wing optimization using many design variables.

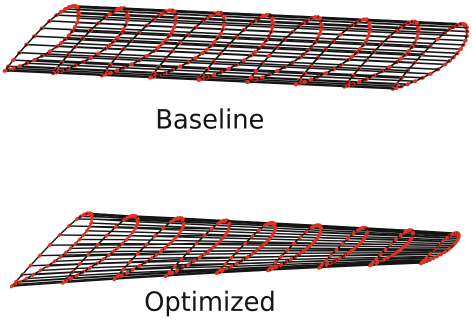

Fig. 7. Baseline and optimized wing configurations

Table 5. Baseline, optimized design vector and output values in the second optimization problem

Design variables	Baseline	Optimized FW	Optimized FD
$\alpha\,[°]$	4	1.000000	1.000000
$\Lambda\,[°]$	0	6.011407	6.007157
$\Gamma\,[°]$	0	5.000000	5.000000
$\delta_r\,[°]$	0	2.230745	2.227752
$\delta_t\,[°]$	0	−0.124150	−0.124778
$b\,[\mathrm{m}]$	6	8.000000	8.000000
$c_r\,[\mathrm{m}]$	1	1.000000	1.000000
λ	1	0.500000	0.500000
Outputs	Baseline	Optimized FW	Optimized FD
C_D	0.013429	0.003724	0.003725
C_L	0.313735	0.300000	0.300000
C_{M_x}	0.071235	0.065081	0.065076
C_{M_y}	0.220788	0.220788	0.220788
S	6.000000	6.000000	6.000000
AR	6.000000	10.666667	10.666667

Table 6. Second optimization case performance benchmark between different sensitivity analysis methods

Gradient calculation method	Time [s]	Iterations	Function evaluations
Sensitivity framework	7607.1	44	75
Forward finite differences	68726.9	44	10155

7 Conclusions

An efficient aerodynamic optimization tool was developed. To accomplish that, a sensitivity analysis framework was constructed based on exact gradient calculation using analytical methods such as automatic differentiation, symbolic differentiation and the adjoint method. Special concern was employed to obtain high computational efficiency, which was translated in combining good programming practices in MATLAB® with code simplifications, whenever possible. An aerodynamic wing optimization problem was solved to illustrate the performance of the new sensitivity analysis framework and it was concluded that savings up to 90% in the computational cost can be achieved.

References

1. Almeida, J.: Structural dynamics for aeroelastic analysis. Master's thesis, Instituto Superior Técnico (2015)
2. Belegundu, A.D., Chandrupatla, T.R.: Optimization Concepts and Applications in Engineering. Cambridge University Press, Cambridge (2011)
3. Cardeira, A.: Aeroelastic analysis of aircraft wings. Master's thesis, Instituto Superior Técnico (2014)
4. Martins, J.R.R.A., Hwang, J.: Review and unification of methods for computing derivatives of multidisciplinary computational models. AIAA J. **51**(11), 2582–2599 (2013). https://doi.org/10.2514/1.J052184
5. Peter, J., Dwight, R.: Numerical sensitivity analysis for aerodynamic optimization: a survey of approaches. Comput. Fluids **39**(3), 373–391 (2010)
6. Schittkowski, K., Yuan, Y.: Sequential quadratic programming methods. Wiley Encyclopedia of Operations Research and Management Science (2011)
7. Sobieski, J., Haftka, R.T.: Multidisciplinary aerospace design optimization: survey of recent developments. Struct. Optim. **14**(1), 1–23 (1997)
8. Venkataraman, P.: A new procedure for airfoil definition. In: AIAA Paper, pp. 95–1875 (1995)