HAL
open science

Parametric optimization of pulsating jets in unsteady flow by Multiple-Gradient Descent Algorithm (MGDA)

Jean-Antoine Desideri, Régis Duvigneau

To cite this version:

Jean-Antoine Desideri, Régis Duvigneau. Parametric optimization of pulsating jets in unsteady flow by Multiple-Gradient Descent Algorithm (MGDA). J. Périaux; W. Fitzgibbon; B. Chetverushkin; O. Pironneau. Numerical Methods for Differential Equations, Optimization, and Technological Problems, Modeling, Simulation and Optimization for Science and Technology, 2017. hal-01414741

HAL Id: hal-01414741

https://inria.hal.science/hal-01414741

Submitted on 12 Dec 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Parametric optimization of pulsating jets in unsteady flow by Multiple-Gradient Descent Algorithm (MGDA)

J.-A. Désidéri and R. Duvigneau

Abstract Two numerical methodologies are combined to optimize six design characteristics of a system of pulsating jets acting on a laminar boundary layer governed by the compressible Navier-Stokes equations in a time-periodic regime. The flow is simulated by second-order in time and space finite-volumes, and the simulation provides the drag as a function of time. Simultaneously, the sensitivity equations, obtained by differentiating the governing equations w.r.t. the six parameters are also marched in time, and this provides the six-component parametric gradient of drag. When the periodic regime is reached numerically, one thus disposes of an objective-function, drag, to be minimized, and its parametric gradient, at all times of a period. Second, the parametric optimization is conducted as a multi-point problem by the Multiple-Gradient Descent Algorithm (MGDA) which permits to reduce the objective-function at all times simultaneously, and not simply in the sense of a weighted average.

Key words: Active-flow control, time-dependent Navier-Stokes equations, finite-volume schemes, sensitivity equations, multi-objective differentiable optimization, descent methods, robust design

1 Introduction: active flow control issues

Active flow control consists in exploiting natural flow instabilities by the use of actuators like oscillatory jets or vibrating membranes, and obtain some desirable ef-

J.-A. Désidéri
INRIA Acumes Team, 2004, Route des Lucioles, 06902 Sophia-Antipolis (France), e-mail: Jean-Antoine.Desideri@inria.fr

R. Duvigneau
INRIA Acumes Team, 2004, Route des Lucioles, 06902 Sophia-Antipolis (France), e-mail: Regis.Duvigneau@inria.fr

fects at a moderate energy expense. It has been a growing research area for the last decades, since this approach demonstrated its ability to improve aerodynamic performance [9], for a large range of applications. It is especially appealing in case of separated flows, for which natural instability phenomena can be efficiently exploited to manipulate flow characteristics using periodic flow excitation.

In this context, a major difficulty is related to the choice of actuation parameters, such as excitation frequency, amplitude, location, to obtain the expected flow response. In cases implying a single isolated actuator, it is relatively easy to carry out an experimental or numerical study to determine efficient control parameters. However, in the perspective of industrial applications involving hundreds of actuators, this task is far from being straightforward and the use of an automated optimization strategy is thus proposed, in the spirit of previous works [6, 7, 8].

The application of an optimization procedure to such problems is faced to the following difficulties: first, the choice of the optimization algorithm is conditioned by the huge computational time of the unsteady-flow simulation, and second, it is necessary to consider several objectives concurrently. Typically, the improvement of the single time-averaged performance is usually not satisfactory for realistic applications. Secondly, sensitivity analysis is tedious in the context of unsteady flows, due to the backward integration of the adjoint equation, which requires the storage, or partial storage / partial re-computation, of the unsteady solution.

The proposed work is based on two methodological ingredients to overcome the difficulties described above: the Sensitivity Equation Method (SEM) for unsteady flows on one side, which allows to compute the gradient of a cost-functional with respect to (w.r.t.) control parameters *at any time* using a *forward time-integration*, and the Multiple Gradient Descent Algorithm (MGDA) on the other side, which is an extension of the classical steepest-descent method to multiobjective problems and permits to compute a descent direction *common to a possibly-large set of cost-functions*.

2 Problem description: optimization of pulsating jets

We consider as model problem the two dimensional compressible flow over a flat plate equipped with three periodically oscillating jets (see Fig. 1). The Reynolds number based on the length h is $R = 10^3$ and the flow is laminar, while the Mach number is $M = 10^{-1}$. For the three jets, the crosswise velocity is imposed as:

$$v_k(x,t) = A_k \sin(2\pi N_k t + \varphi_k)\zeta(x) \quad k = 1,2,3, \tag{1}$$

where $\zeta(x)$ corresponds to a squared sine distribution. The jet frequencies are set to the fixed values $N_1 = N_\infty, N_2 = 2N_\infty$ and $N_3 = 1/2N_\infty$, with $N_\infty = u_\infty/h$. The jet amplitudes and phases are considered as control parameters $\mathbf{x} = \{A_1, A_2, A_3, \varphi_1, \varphi_2, \varphi_3\}$.

Initial values are chosen somewhat arbitrarily as $\mathbf{x}_0 = \{u_\infty, 3/2\,u_\infty, 2\,u_\infty, 0, \pi/4, 3\pi/4\}$.

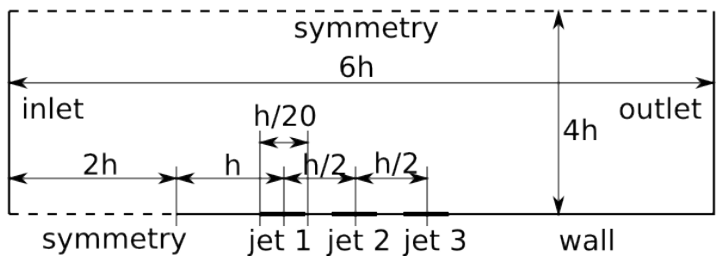

Fig. 1 Problem description.

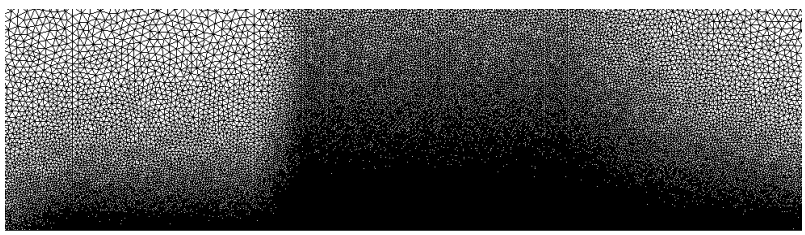

Fig. 2 Computational mesh.

The grid employed in this study counts 111 161 nodes (see Fig. 2). The initial solution corresponds to uniform flow variables based on inlet conditions. The time step is set to $\Delta t = 1/(400N_\infty)$. The unsteady flow tends rapidly to a periodic regime, whose period corresponds to the lowest actuation frequency $1/2N_\infty$ and is thus described by 800 time-steps. As transient effects have vanished, one period is defined as observation interval (see Fig. 3). Instantaneous velocity fields are shown in Figs. 4-5 as illustration.

We consider a set of objective-functions $\{f_j(\mathbf{x})\}_{j=1,\dots,m}$, defined as the values of the drag $\mathscr{J}$, estimated at discrete times $\{t_j\}_{j=1,\dots,m}$, chosen in the observation interval:

$$f_j(\mathbf{x}) = \mathscr{J}(\mathbf{W}_j) \quad \text{with } \mathbf{W}_j = \mathbf{W}(\mathbf{x}, t_j) \quad \forall j \in \{1, \cdots, m\}, \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^n$ represents the vector of the $n = 6$ control parameters and $\mathbf{W}$ the flow variables. The objective of this work is to *reduce simultaneously* these m cost-functions. The following sections describe how the gradient $\nabla_{\mathbf{x}} f_j$ of f_j w.r.t. $\mathbf{x}$ is evaluated, and the optimization algorithm proposed to conduct the simultaneous optimization of these functions.

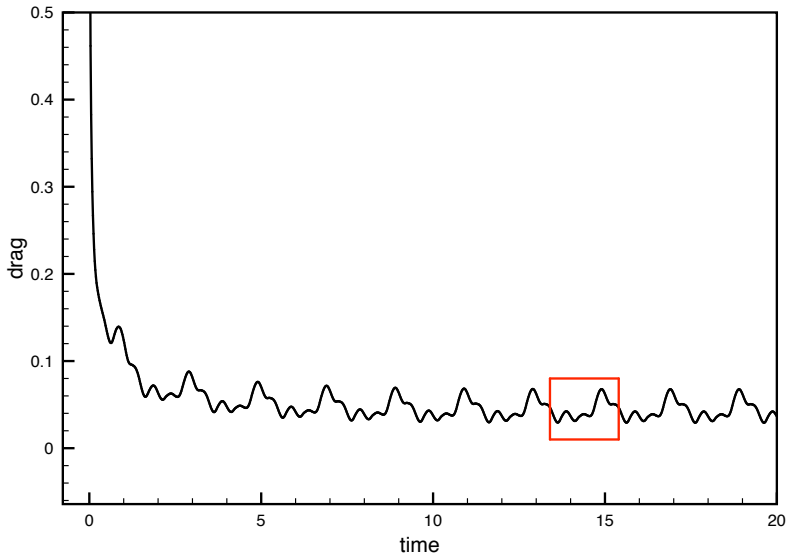

Fig. 3 History of drag for initial parameters and observation area.

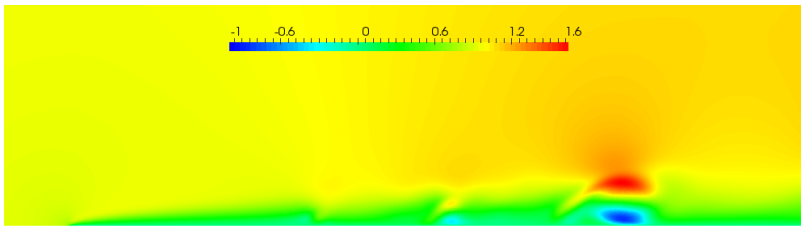

Fig. 4 Snapshot of the streamwise velocity field.

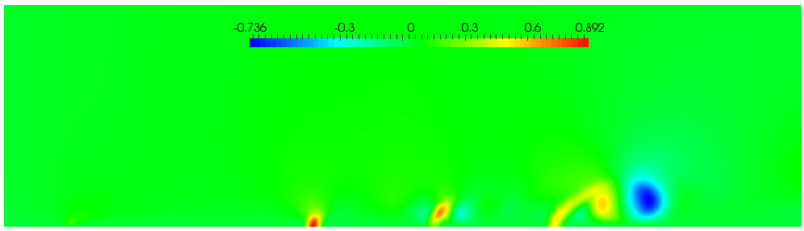

Fig. 5 Snapshot of the crosswise velocity field.

3 Sensitivity analysis for unsteady flow

3.1 Method

The governing flow equations are written in conservative form as follows:

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla \cdot \mathscr{F} = \nabla \cdot \mathscr{G}, \tag{3}$$

where $\mathbf{W} = (\rho, \rho u, \rho v, \rho e)$ is the vector of conservative mean-flow variables, ρ is density, u and v are the velocity components, and e the total energy per unit mass; $\mathscr{F} = (\mathbf{F}_x(\mathbf{W}), \mathbf{F}_y(\mathbf{W}))$ and $\mathscr{G} = (\mathbf{G}_x(\mathbf{W}), \mathbf{G}_y(\mathbf{W}))$ are the vectors of convective and diffusive fluxes respectively. Here ∇ stands for the gradient w.r.t. the spatial Cartesian coordinates x and y, and $(\nabla.)$ for the divergence operator. The pressure p is obtained from the perfect-gas state equation:

$$p = \rho(\gamma - 1)(e - \frac{u^2 + v^2}{2}) = \rho(\gamma - 1)e_i \tag{4}$$

where $\gamma = \frac{7}{5}$ is the ratio of the specific heats for diatomic gas, and e_i the internal energy.

The inviscid fluxes are given by:

$$\mathbf{F}_x(\mathbf{W}) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u(e + \frac{p}{\rho}) \end{pmatrix} \quad \mathbf{F}_y(\mathbf{W}) = \begin{pmatrix} \rho v \\ \rho v u \\ \rho v^2 + p \\ \rho v(e + \frac{p}{\rho}) \end{pmatrix}. \tag{5}$$

The viscous fluxes are written as:

$$\mathbf{G}_x(\mathbf{W}) = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{yx} \\ u\tau_{xx} + v\tau_{yx} - q_x \end{pmatrix} \quad \mathbf{G}_y(\mathbf{W}) = \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} - q_y \end{pmatrix}, \tag{6}$$

where $\bar{\bar{\tau}}$ is the symmetric viscous stress tensor and $\mathbf{q}$ the heat flux.

We can now introduce the sensitivity field $\mathbf{W}'$, which is defined as the derivative of the flow solution $\mathbf{W}$ w.r.t. a given control parameter a, component of $\mathbf{x}$:

$$\mathbf{W}' = \frac{\partial \mathbf{W}}{\partial a}. \tag{7}$$

The equations governing the sensitivity field can be obtained by differentiating (3) w.r.t. a:

$$\frac{\partial}{\partial a}\left(\frac{\partial \mathbf{W}}{\partial t}\right) + \frac{\partial}{\partial a}(\nabla \cdot \mathscr{F}) = \frac{\partial}{\partial a}(\nabla \cdot \mathscr{G}). \tag{8}$$

By switching the derivatives w.r.t. a and those w.r.t time or space coordinates, one obtains:

$$\frac{\partial}{\partial t}\left(\frac{\partial \mathbf{W}}{\partial a}\right)+\nabla\cdot\left(\frac{\partial \mathscr{F}}{\partial a}\right)=\nabla\cdot\left(\frac{\partial \mathscr{G}}{\partial a}\right), \tag{9}$$

or:

$$\frac{\partial \mathbf{W}'}{\partial t}+\nabla\cdot\mathscr{F}'=\nabla\cdot\mathscr{G}', \tag{10}$$

which is formally similar to (3), by introducing the sensitivity of the convective flux $\mathscr{F}'=(\mathbf{F}'_x(\mathbf{W},\mathbf{W}'),\mathbf{F}'_y(\mathbf{W},\mathbf{W}'))$ and the sensitivity of the diffusive flux $\mathscr{G}'=(\mathbf{G}'_x(\mathbf{W},\mathbf{W}'),\mathbf{G}'_y(\mathbf{W},\mathbf{W}'))$. The sensitivity of the convective fluxes can be expressed as:

$$\mathbf{F}'_x(\mathbf{W},\mathbf{W}')=\begin{pmatrix}(\rho u)'\\ (\rho u)'u+(\rho u)u'+p'\\ (\rho u)'v+(\rho u)v'\\ (\rho u)'(e+\frac{p}{\rho})+(\rho u)(e'+(\frac{p}{\rho})')\end{pmatrix} \tag{11}$$

$$\mathbf{F}'_y(\mathbf{W},\mathbf{W}')=\begin{pmatrix}(\rho v)'\\ (\rho v)'u+(\rho v)u'\\ (\rho v)'v+(\rho v)v'+p'\\ (\rho v)'(e+\frac{p}{\rho})+(\rho v)(e'+(\frac{p}{\rho})')\end{pmatrix}. \tag{12}$$

The sensitivity of the diffusive fluxes reads:

$$\mathbf{G}'_x(\mathbf{W},\mathbf{W}')=\begin{pmatrix}0\\ \tau'_{xx}\\ \tau'_{yx}\\ u'\tau_{xx}+v'\tau_{yx}+u\tau'_{xx}+v\tau'_{yx}-q'_x\end{pmatrix} \tag{13}$$

$$\mathbf{G}'_y(\mathbf{W},\mathbf{W}')=\begin{pmatrix}0\\ \tau'_{xy}\\ \tau'_{yy}\\ u'\tau_{xy}+v'\tau_{yy}+u\tau'_{xy}+v\tau'_{yy}-q'_y\end{pmatrix}, \tag{14}$$

where $\bar{\bar{\tau}}'$ is the sensitivity of the viscous stress tensor and $\mathbf{q}'$ the sensitivity of the heat flux. The boundary conditions for the sensitivity equations are obtained by differentiating the boundary conditions applied to the flow.

Since the flow and sensitivity equations are formally similar, both are solved using the same finite-volume approach [5], based on a second-order vertex-centered discretization scheme. Temporal integration relies on a second-order implicit backward method, with a dual time-stepping technique. Note that the implicit part of the scheme is the same for the flow and sensitivity equations, since both involve the same Jacobian matrix.

The sensitivity equation depends on the parameter a, component of $\mathbf{x}$ of interest. Therefore, six sensitivity equations have to be solved, possibly in parallel, to estimate the components of the gradient for all m cost-functions:

$$\nabla_a f_j = \partial_{\mathbf{W}} \mathscr{J}(\mathbf{W}_j) \cdot \mathbf{W}'(t_j) \quad \forall j \in \{1, \cdots, m\}. \tag{15}$$

We emphasize that if m is large, the solution of six sensitivity equations is far more cost-efficient than solving m adjoint equations backward in time. Additionally, the memory-storage requirement remains moderate.

3.2 Verification

To verify the implementation of the sensitivity equations, we compute a neighboring solution according to a first-order extrapolation $\mathbf{W}(a) + \mathbf{W}'\delta a$ and compare it with the solution $\mathbf{W}(a + \delta a)$. This exercise is conducted in a simplified case including a single jet, a being the jet amplitude A_1. Figs. 6-7 provide illustrations for the flow fields at selected times and Fig. 8 for the resulting drag history, for a perturbation of the jet amplitude $\delta A_1 = A_1/4$. A similar exercise has been achieved for the phase, to fully verify the gradient estimation.

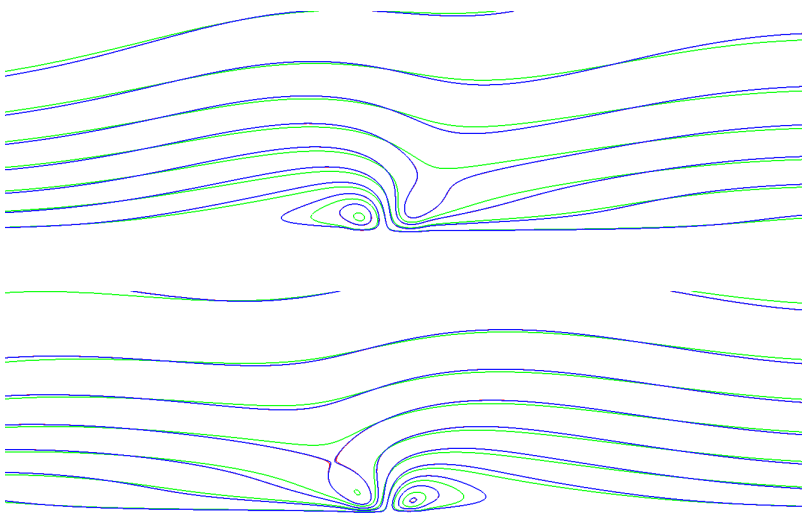

Fig. 6 Linear extrapolation of streamwise velocity field u w.r.t. jet amplitude A_1 for blowing (top) and suction (bottom) phases: reference state $u(A_1)$ in green, extrapolated state $u(A_1) + u'\delta A_1$ in red, non-linear perturbed state $u(A_1 + \delta A_1)$ in blue, for $\delta A_1 = A_1/4$.

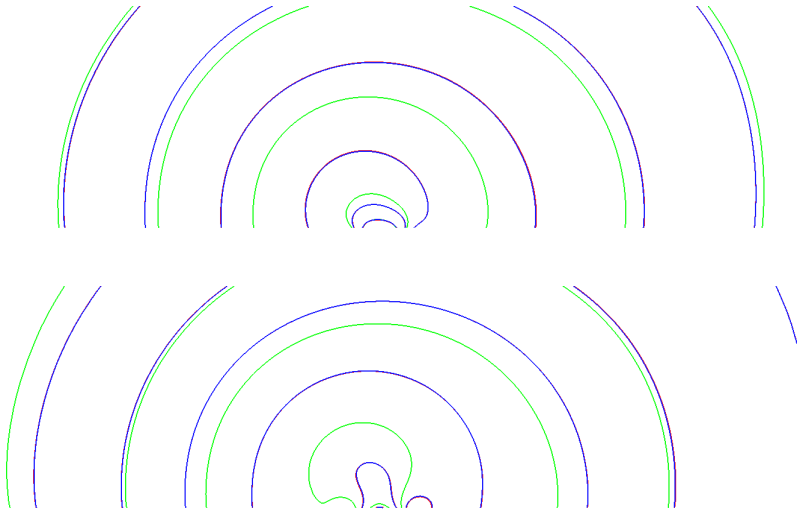

Fig. 7 Linear extrapolation of pressure field p w.r.t. jet amplitude A_1 for blowing (top) and suction (bottom) phases: reference state $p(A_1)$ in green, extrapolated state $p(A_1)+p'\delta A_1$ in red, non-linear perturbed state $p(A_1+\delta A_1)$ in blue, for $\delta A_1 = A_1/4$.

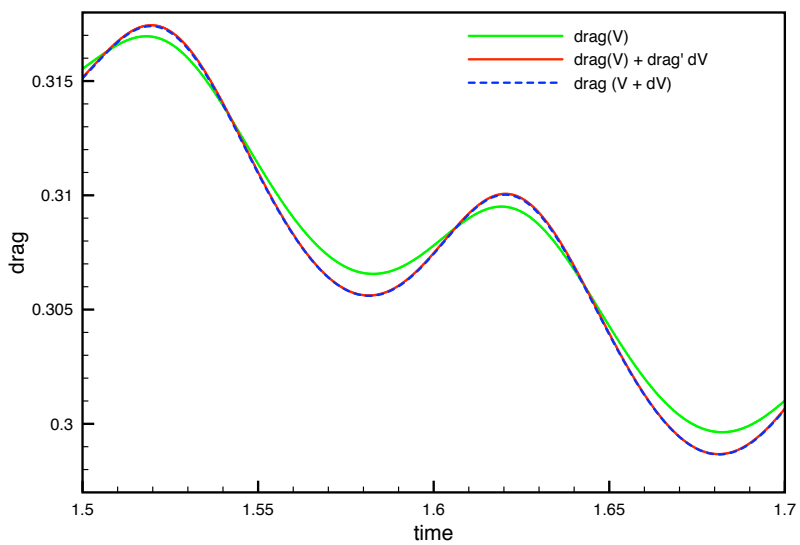

Fig. 8 Linear extrapolation of the drag w.r.t. jet amplitude A_1: reference drag $\mathscr{J}(A_1)$ in green, extrapolated drag $\mathscr{J}(A_1)+\mathscr{J}'\delta A_1$ in red, non-linear perturbed drag $\mathscr{J}(A_1+\delta A_1)$ in blue, for $\delta A_1 = A_1/4$.

4 Multiobjective descent algorithm MGDA

Equipped with procedures for calculating the objective-functions and their gradients, we now turn to the issue of constructing the multi-objective optimization method.

The Multiple-Gradient Descent Algorithm (MGDA) was originally introduced in [1] and [2] to solve general multi-objective optimization problems involving differentiable cost-functions. Variants were proposed in [3], but more recently the algorithm was slightly revised in [4] to apply to cases where the number m of objective-functions exceeds the dimension n of the working design space. We recall here the basic definition of the revised version and provide some details about the application to the present parametric optimization.

4.1 Multi-Objective problem statement

Let m and n be two arbitrary integers, and consider the multi-objective optimization problem consisting in minimizing m differentiable objective-functions $\{f_j(\mathbf{x})\}$ in some open admissible domain $\Omega_a \subseteq \mathbb{R}^n$ ($j = 1,\ldots,m$; $f_j \in C^1(\Omega_a)$). Given a starting point $\mathbf{x}_0 \in \Omega_a$ and a vector $\mathbf{d} \in \mathbb{R}^n$, one forms the directional derivatives

$$f'_j = [\nabla_{\mathbf{x}} f_j(\mathbf{x}_0)]^t \, \mathbf{d} \tag{16}$$

where $\nabla_{\mathbf{x}}$ is the symbol for the gradient w.r.t. $\mathbf{x}$ and the superscript ^{t} stands for transposition. One seeks for a vector $\mathbf{d}$ such that

$$f'_j > 0 \quad (\forall j)\,. \tag{17}$$

If such a vector $\mathbf{d}$ exists, the direction of vector $(-\mathbf{d})$ is said to be a local descent direction common to all objective-functions. Then evidently, infinitely-many other such directions also exist, and our algorithm permits to identify at least one.

4.2 Convex hull, two lemmas and basic MGDA

We recall the following :

Definition 1. The convex hull of a family of m vectors $\{\mathbf{u}_j\}$ ($j = 1,\ldots,m$; $\mathbf{u}_j \in \mathbb{R}^n$), is the set of all their convex combinations:

$$\overline{\mathrm{U}} = \left\{ \mathbf{u} \in \mathbb{R}^n \text{ such that } \mathbf{u} = \sum_{j=1}^{m} \alpha_j \mathbf{u}_j;\ \alpha_j \in \mathbb{R}+;\ \sum_{j=1}^{m} \alpha_j = 1 \right\}. \tag{18}$$

Then, we have :

Lemma 1. *Given an $n \times n$ real-symmetric positive-definite matrix $\mathbf{A}_n$, the associated scalar product*

$$\big(\mathbf{u},\mathbf{v}\big) = \mathbf{u}^t \mathbf{A}_n \mathbf{v} \quad (\mathbf{u},\mathbf{v} \in \mathbb{R}^n), \tag{19}$$

and Euclidean norm

$$\|\mathbf{u}\| = \sqrt{\mathbf{u}^t \mathbf{A}_n \mathbf{u}}, \tag{20}$$

the convex hull $\overline{\mathrm{U}}$ admits a unique element ω of minimum norm.

Proof. - Existence : $\overline{\mathrm{U}}$ is closed and $\|.\|$ is a continuous function.
- Uniqueness : suppose that ω_1 and ω_2 are two realizations of the minimum $\mu = \arg\min_{\mathbf{u}\in\overline{\mathrm{U}}} \|\mathbf{u}\|$ so that $\mu = \|\omega_1\| = \|\omega_2\|$ and let

$$\omega_s = \tfrac{1}{2}\left(\omega_2 + \omega_1\right), \quad \omega_d = \tfrac{1}{2}\left(\omega_2 - \omega_1\right),$$

so that:

$$\big(\omega_s, \omega_d\big) = \tfrac{1}{4}\big(\omega_2 + \omega_1, \omega_2 - \omega_1\big) = \tfrac{1}{4}\left(\|\omega_2\|^2 - \|\omega_1\|^2\right) = 0\,.$$

Hence $\omega_s \perp \omega_d$, and since $\omega_s \in \overline{\mathrm{U}}$, $\|\omega_s\| \geq \mu$, and :

$$\mu^2 = \|\omega_2\|^2 = \|\omega_s + \omega_d\|^2 = \|\omega_s\|^2 + \|\omega_d\|^2 \geq \mu^2 + \|\omega_d\|^2 \Longrightarrow \omega_d = 0\,. \qquad \square$$

Lemma 2. *The minimum-norm element ω defined in Lemma 1 satisfies :*

$$\forall u \in \overline{\mathrm{U}}, \quad \big(\mathbf{u},\omega\big) \geq \|\omega\|^2\,. \tag{21}$$

Proof. Let $\mathbf{u} \in \overline{\mathrm{U}}$, arbitrary. Let $\delta = \mathbf{u} - \omega$; by convexity of $\overline{\mathrm{U}}$:

$$\forall \varepsilon \in [0,1],\ (1-\varepsilon)\omega + \varepsilon\mathbf{u} = \omega + \varepsilon\delta \in \overline{\mathrm{U}}\,,$$

and by definition of ω, $\|\omega + \varepsilon\delta\| \geq \|\omega\|$, that is :

$$\big(\omega + \varepsilon\delta, \omega + \varepsilon\delta\big) - \big(\omega,\omega\big) = 2\varepsilon\big(\omega,\delta\big) + \varepsilon^2\|\delta\|^2 \geq 0\,,$$

and this requires that the coefficient of ε be non-negative. $\square$

Then consider the case where

$$\mathbf{u}_j = \nabla_{\mathbf{x}} f_j(\mathbf{x}_0) \quad (\forall j). \tag{22}$$

If the vector ω defined in Lemma 1 is nonzero, the vector

$$\mathbf{d} = \mathbf{A}_n \omega \tag{23}$$

is also nonzero, and is a solution to the problem stated in (16)-(17) since by virtue of Lemma 2:

$$\left(\mathbf{u}_j, \omega\right) = \mathbf{u}_j^t \mathbf{A}_n \omega = \mathbf{u}_j^t \mathbf{d} \geq \|\omega\|^2 > 0. \tag{24}$$

The situation in which $\omega = 0$, or equivalently,

$$\exists \alpha = \{\alpha_j\} \in \mathbb{R}+^m \text{ such that } \sum_{j=1}^{m} \alpha_j \nabla f_j(\mathbf{x}_0) = 0 \text{ and } \sum_{j=1}^{m} \alpha_j = 1, \tag{25}$$

is said to be one of "Pareto-stationarity". The relationship between Pareto-optimality and Pareto-stationarity was made precise by the following [3]-[4]

Theorem 1. *If the objective-functions are differentiable and convex in some open ball $\mathscr{B} \subseteq \Omega_a$ about $\mathbf{x}_0$, and if $\mathbf{x}_0$ is Pareto-optimal, then the Pareto-stationarity condition is satisfied at $\mathbf{x}_0$.*

Hence, the Pareto-stationarity condition generalizes to the multi-objective context, the classical stationarity condition expressing that an unconstrained differentiable function is extremal.

We now return to the non-trivial case of a point $\mathbf{x}_0$ that is *not Pareto-stationary* and we suppose that the vectors ω and $\mathbf{d}$ ($\omega \neq 0$; $\mathbf{d} \neq 0$) have been identified (see next subsection). Then we define MGDA as the iteration which transforms $\mathbf{x}_0$ in

$$\mathbf{x}_1 = \mathbf{x}_0 - \rho \mathbf{d} \tag{26}$$

where $\rho > 0$ is some appropriate step-size. Thus MGDA is an extension to the multi-objective context of the classical steepest-descent method, in which the direction of search is taken to be the vector $\mathbf{d}$ defined above. At convergence, the limiting point is Pareto-stationary.

We now examine how can the vector $\mathbf{d}$ be computed in practice.

4.3 QP formulation and hierarchical Gram-Schmidt orthogonalization

By letting

$$\omega = \sum_{j=1}^{m} \alpha_j \mathbf{u}_j = \mathbf{U}\alpha \tag{27}$$

where $\mathbf{u}_j = \nabla f_j(\mathbf{x}_0)$, $\mathbf{U}$ is the $n \times m$ matrix whose jth column contains the n components of vector $\mathbf{u}_j$, the identification of vector ω can be made by solving the following Quadratic-Programming (QP) problem for the unknown vector of coefficients $\alpha = \{\alpha_j\}$:

$$\omega = \arg \min_{\alpha \in \mathbb{R}^m} \tfrac{1}{2} \alpha^t \mathbf{H} \alpha \tag{28}$$

subject to:

$$\alpha_j \geq 0 \ (\forall j), \quad \sum_{j=1}^{m} \alpha_j = 1, \tag{29}$$

where $\mathbf{H} = \mathbf{U}^t \mathbf{A}_n \mathbf{U}$. Note that if vector ω is unique, vector α may not be.

If the family of gradients is linearly-independent, which requires in particular that $m \leq n$, it is possible to choose the scalar product, through the definition of matrix $\mathbf{A}_n$, in such a way that these gradients form an orthogonal basis of their span. Then vector ω is explicitly determined by the orthogonal projection of 0 onto the convex hull $\overline{\mathrm{U}}$:

$$\alpha_j = \frac{1}{\left\|\mathbf{u}_j\right\|^2 \sum_{k=1}^{m} \frac{1}{\|\mathbf{u}_k\|^2}} \tag{30}$$

In the inverse case where $m > n$ (and even $m \gg n$), focus of interest presently, let $\mathsf{r} \leq n < m$ be the rank of the family of gradients. Using first the standard Euclidean scalar product ($\mathbf{A}_n = \mathbf{I}_n$), the Gram-Schmidt orthogonalization process stops in r steps and produces an orthogonal basis (in the usual sense) $\{\mathbf{v}_j\}$ ($j = 1, \ldots, \mathsf{r}$), that span a subspace $\mathscr{G}$ of dimension r. The orthogonal vectors, $\{\mathbf{v}_j\}$, are calculated from a subfamily of the original vectors, $\{\mathbf{u}_j\}$, $j \in J$, where J is a subfamily of r indices from 1 to m. In this process, a hierarchical principle was introduced in [4] to select these indices in such a way that the cone bounded by the reduced family $\{\mathbf{u}_j\}$ ($j \in J$) be as large as possible to contain the directions, in the most favorable situation, of all the other gradients, unused in the Gram-Schmidt process by redundancy. When this occurs, the subfamily $\{\mathbf{u}_j\}$ ($j \in J$) not only is a basis of the subspace $\mathscr{G}$, but also, its convex hull contains all the directions of interest. In the more general case where the directions of some gradients among the unused vectors $\{\mathbf{u}_j\}$ ($j \notin J$) are not in the cone, we resort to the QP-formulation, but expressed after changing the basis to become $\{\mathbf{u}_j\}$ ($j \in J$) and by choosing a new scalar product to make this subfamily orthogonal. The technical steps are the following [4]:

- Once the $n \times m$ matrix $\mathbf{U}$ is formed with the components of the given gradients $\{\mathbf{u}_j\}$ ($\mathbf{u}_j \in \mathbb{R}^n$, $j = 1, \ldots, m$), these gradients, or column-vectors are made (physically) dimensionless, by component-wise normalization to form the initial matrix $\mathbf{G}$:
$$\forall i \in 1, \ldots, n: \ s_i = \max_j \left|\mathbf{u}_{i,j}\right|, \quad \mathbf{S} = \mathbf{Diag}\big(s_i\big), \quad \mathbf{G} = \mathbf{S}^{-1}\mathbf{U}. \tag{31}$$
This normalization is essential to make the subsequent calculations of scalar products physically meaningful and computationally well-balanced.
- Throughout the Gram-Schmidt process, columns of the $\mathbf{G}$ matrix are permuted by the hierarchical selection of basis vectors. Upon exit, the actually used gradients are placed in the first r columns. The corresponding $n \times \mathsf{r}$ leftmost block of the final matrix $\mathbf{G}$ is then denoted $\underline{\mathbf{G}}$. Note that in the present version of the Gram-Schmidt process, the computed orthogonal vectors $\{\mathbf{v}_j\}$ are not normalized to unity, but in a special way for which r directional derivatives are equal [4]. Let these vectors be stored in matrix $\mathbf{V}$, and define the following diagonal matrix:

$$\Delta = \mathbf{Diag}\left(\mathbf{v}_j^t \mathbf{v}_j\right). \tag{32}$$

Then:

$$\mathbf{A}_n = \mathbf{W}^t\mathbf{W} + (\mathbf{I} - \Pi)^2, \quad \mathbf{W} = \left(\underline{\mathbf{G}}^t\underline{\mathbf{G}}\right)^{-1}\underline{\mathbf{G}}^t\mathbf{V}\Delta^{-1}\mathbf{V}^t, \tag{33}$$

where $\Pi = \mathbf{V}\Delta^{-1}\mathbf{V}^t$ is the projection matrix onto subspace $\mathscr{G}$ [4].

In this way, the QP-formulation is well-conditioned and easily solved by a library procedure. We have used the procedure qpsolve from the Scilab library which is equivalent to the quadprog procedure from the MATLAB library. As a result of this, exactly r directional derivatives $\{f_j'\}$ are equal, and by experience, the remaining ones differ only slightly.

5 Results

For each control parameter, component of $\mathbf{x} = \{A_1, A_2, A_3, \varphi_1, \varphi_2, \varphi_3\}$, the SEM is applied to obtain sensitivity fields. We provide as illustration (see Figs. 9-10) instantaneous sensitivity fields of the velocity w.r.t. the amplitude for the first jet. The derivatives of the drag w.r.t. the control parameters are then computed at all time-steps, yielding 800 values of cost-function and gradient for the whole observation period, as illustrated in Fig. 11.

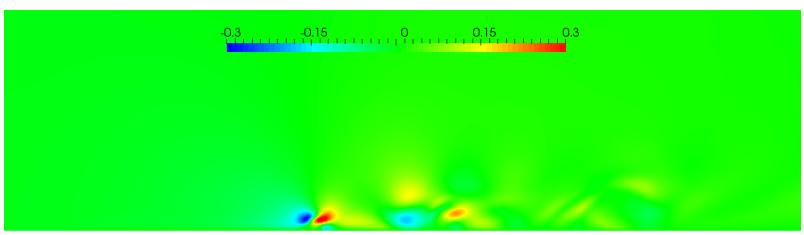

Fig. 9 Sensitivity of streamwise velocity w.r.t. first jet amplitude.

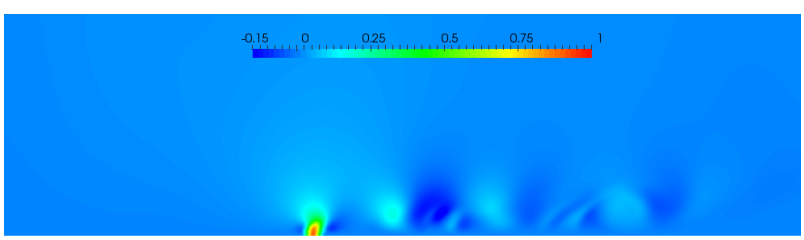

Fig. 10 Sensitivity of crosswise velocity w.r.t. first jet amplitude.

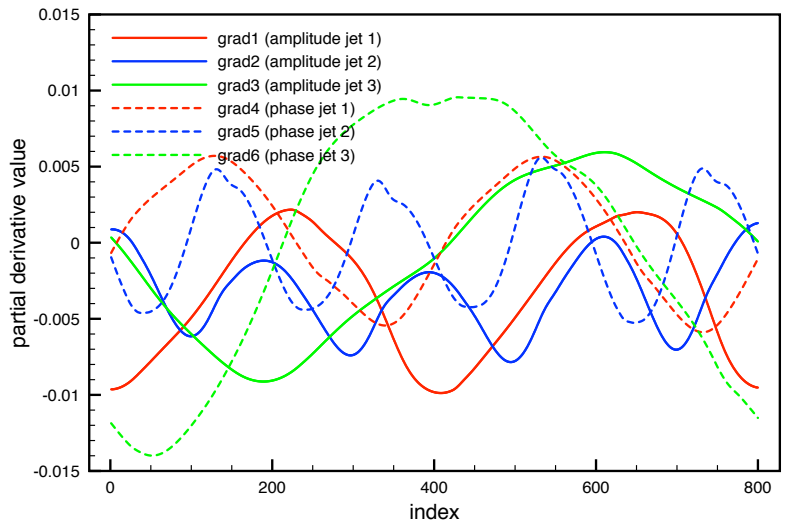

Fig. 11 Gradient components for the 800 cost-functionals.

We aim now at determining the vector of parameters $\mathbf{x} = \{A_1, A_2, A_3, \varphi_1, \varphi_2, \varphi_3\}$ that reduces simultaneously the 800 cost-functions associated with the observation period. To reduce somewhat the computational complexity without altering greatly the transient behavior, the MGDA approach is applied to only $m = 20$ homogenized gradients, obtained by averaging the gradients by time-intervals of 40 time-steps. As a result, one disposes of a common descent direction associated with vector $\mathbf{d}$ satisfying (17).

Once the vector $\mathbf{d}$ is determined, a practical step-size ρ must be estimated. For this, we first note that a natural scale for the variations of a time-dependent objective-function is given by its standard deviation, $\bar{\sigma}$, a more significant value than its average which can be 0. Then if $\delta\mathbf{x} = -\bar{\rho}\mathbf{d}$, the variation of the objective-function average can be estimated as $-\bar{\rho}\bar{g}.\mathbf{d}$ where $\bar{g}$ is the average gradient. Thus, a meaningful reference step-size can be defined by the condition:

$$\bar{\rho}\bar{g}.\mathbf{d} = \bar{\sigma} \tag{34}$$

where $\bar{\sigma} = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\left(f_j - \bar{f}\right)^2}$, $\bar{f} = \frac{1}{m}\sum_{j=1}^{m} f_j$, and $\bar{g} = \frac{1}{m}\sum_{j=1}^{m}\nabla f_j(\mathbf{x}_0)$. This gives:

$$\bar{\rho} = \frac{\bar{\sigma}}{\bar{g}.\mathbf{d}}. \tag{35}$$

In practice, in the present experiments, we have used the step-size $\rho = \frac{1}{10}\bar{\rho}$ to update the control parameters by the descent method, and this resulted in a successful iteration, stable and effective.

The history of the drag in the observation period is represented in Fig. 12, from the baseline flow to full convergence of the optimization approach. As expected, *each update of the design vector has resulted in a diminished drag over the entire observation period*. As it can be noticed, some points in time are more critical than others. In contrast, when one applies, more classically, the steepest-descent method to the time-averaged cost-function $\overline{\mathscr{J}}$, by setting the search direction to the average gradient, an increase of the drag can be observed at some times, as illustrated in Fig. 13. Finally, also note that actuation permits a significant drag reduction w.r.t. the case without suction/blowing for which the value of drag is indicated on the figure by a dotted horizontal line.

Finally, a second exercise is conducted: the drag values computed over the last 40% of the observation period only are considered as optimization criteria in the MGDA approach (in this interval, the drag is especially high for the baseline flow). The history of the drag in the observation period is represented in Fig. 14, for the first 16 iterations of the optimization algorithm. As expected, a more significant decrease is achieved during the last 40% of the observation period, whereas the drag is free to vary in the first 60%, and in fact, increases.

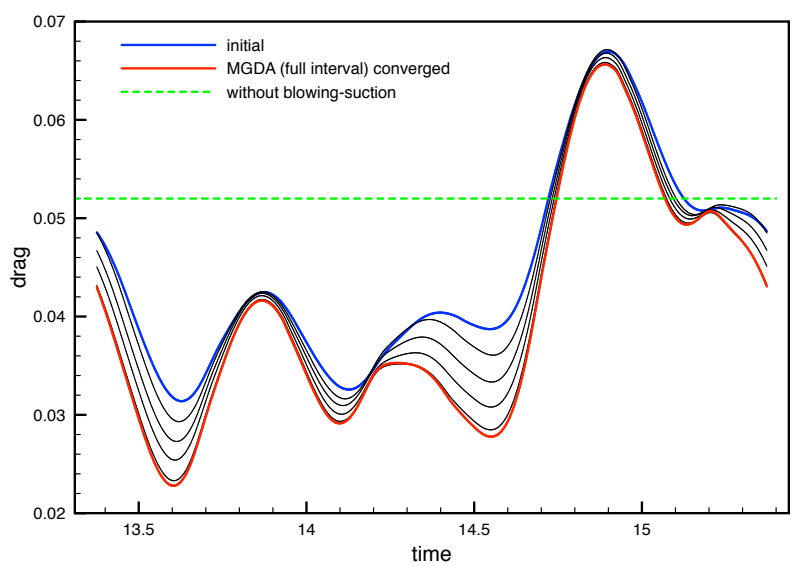

Fig. 12 Evolution of the drag history w.r.t. optimization iterations: case of MGDA approach (blue: initial, red: final, black: intermediate iterations).

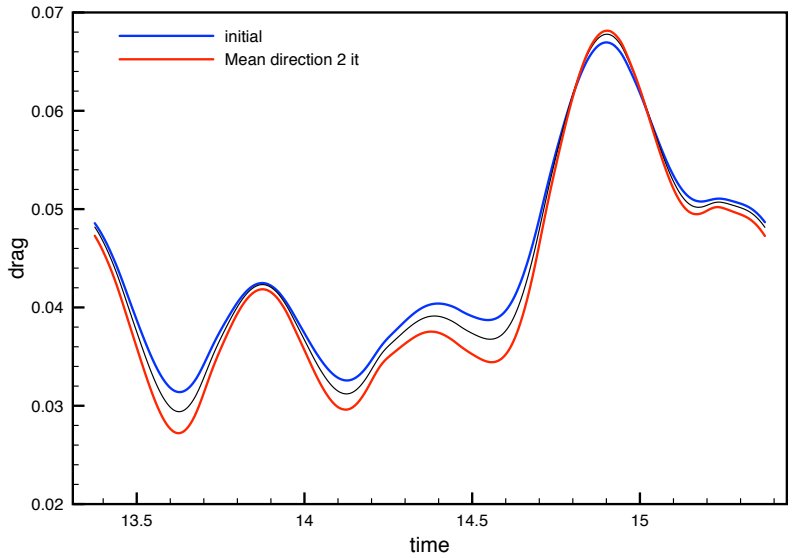

Fig. 13 Evolution of the drag history w.r.t. optimization iterations: case of a mean direction descent (blue: initial, red: after 2 iterations, black: intermediate iteration).

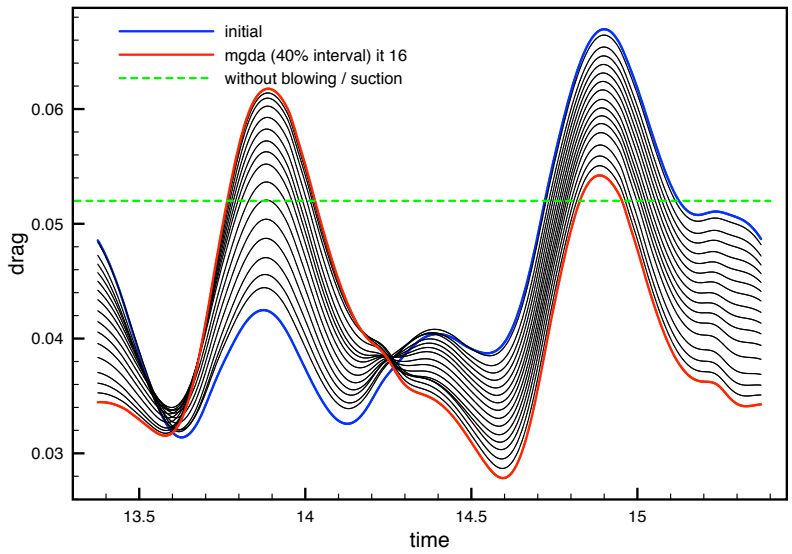

Fig. 14 Evolution of the drag history w.r.t. optimization iterations: case of MGDA approach based on the last 40% of the observation period (blue: initial, red: after 7 iterations, black: intermediate iterations).

6 Conclusion

In this work, we solved for demonstration an exercise of active-flow control in which drag over a flat plate has been reduced by three pulsating jets acting on the boundary layer. The flow, governed by the time-dependent compressible Navier-Stokes equations, has been simulated numerically by second-order in time and space finite-volumes, yielding, when the periodic regime is achieved, drag as a function of time. The simultaneous solution of the sensitivity equations has provided additionally the six-component gradient of drag w.r.t. the design characteristics of the jets. The accuracy of these gradients has been verified by comparison with finite-differences via fine-mesh computations.

By the Multiple-Gradient Descent Algorithm (MGDA), a direction of search has been identified permitting to reduce drag at all times of the period, or a selected segment of it. The process was repeated iteratively, and at all intermediate steps of the optimization process as well as at convergence, the drag was effectively reduced over the entire observation time-interval.

Hence, we dispose of a numerical optimization tool whose efficacy is demonstrated uniformly, that is, over a possibly-large range of operational conditions, here different discretization times. This contrasts with more classical approaches in which a single functional, usually defined as a somewhat arbitrary weighted average, is minimized at the risk of a degradation of certain elements composing the average.

This method is currently being extended to solve more general robust design problems.

References

1. J.-A. Désidéri. Multiple-gradient descent algorithm (mgda). Research Report 6953, INRIA, 2009 (revised version November 5, 2012). http://hal.inria.fr/inria-00389811/fr/.
2. J.-A. Désidéri. Multiple-gradient descent algorithm (MGDA) for multiobjective optimization. *Comptes Rendus de l'Académie des Sciences Paris*, 350:313–318, 2012.
3. J.-A. Désidéri. *Numerical Methods for Differential Equations, Optimization, and Technological Problems*, volume 34 of *Modeling, Simulation and Optimization for Science and Technology, Fitzgibbon, W.; Kuznetsov, Y.A.; Neittaanmäki, P.; Pironneau, O. Eds.*, chapter Multiple-Gradient Descent Algorithm (*MGDA*) for Pareto-Front Identification. Springer-Verlag, 2014. J. Périaux and R. Glowinski Jubilees.
4. J.-A. Désidéri. Révision de l'algorithme de descente à gradients multiples (MGDA) par orthogonalisation hiérarchique. Research Report 8710, INRIA, April 2015. https://hal.inria.fr/hal-01139994.
5. R. Duvigneau. A sensitivity equation method for unsteady compressible flows: Implementation and verification. Technical report, INRIA Research Report No 8739, June 2015.
6. R. Duvigneau, A. Hay, and M. Visonneau. Optimal location of a synthetic jet on an airfoil for stall control. *Journal of Fluid Engineering*, 129(7):825–833, July 2007.
7. R. Duvigneau, J. Labroquère, and E. Guilmineau. Comparison of turbulence closures for optimized active control. *Computers & Fluids*, (124):67–77, January 2016.
8. R. Duvigneau and M. Visonneau. Optimization of a synthetic jet actuator for aerodynamic stall control. *Computers and Fluids*, 35:624–638, July 2006.
9. M. Gad El-Hack, A. Pollard, and J.-P. Bonnet. *Flow control: fundamentals and practices*. Springer-Verlag, 1998.