Null space gradient flows for constrained optimization with applications to shape optimization

Florian Feppon, Grégoire Allaire, Charles Dapogny

NULL SPACE GRADIENT FLOWS FOR CONSTRAINED OPTIMIZATION WITH APPLICATIONS TO SHAPE OPTIMIZATION

F. FEPPON[12], G. ALLAIRE[1], C. DAPOGNY[3]

[1] *Centre de Mathématiques Appliquées, École Polytechnique, Palaiseau, France*
[2] *Safran Tech, Magny-les-Hameaux, France*
[3] *Univ. Grenoble Alpes, CNRS, Grenoble INP[1], LJK, 38000 Grenoble, France*

Abstract. The purpose of this article is to introduce a gradient-flow algorithm for solving equality and inequality constrained optimization problems, which is particularly suited for shape optimization applications. We rely on a variant of the Ordinary Differential Equation (ODE) approach proposed by Yamashita [65] for equality constrained problems: the search direction is a combination of a null space step and a range space step, aiming to decrease the value of the minimized objective function and the violation of the constraints, respectively. Our first contribution is to propose an extension of this ODE approach to optimization problems featuring both equality and inequality constraints. In the literature, a common practice consists in reducing inequality constraints to equality constraints by the introduction of additional slack variables. Here, we rather solve their local combinatorial character by computing the projection of the gradient of the objective function onto the cone of feasible directions. This is achieved by solving a dual quadratic programming subproblem whose size equals the number of active or violated constraints. The solution to this problem allows to identify the inequality constraints to which the optimization trajectory should remain tangent. Our second contribution is a formulation of our gradient flow in the context of—infinite-dimensional—Hilbert spaces, and of even more general optimization sets such as sets of shapes, as it occurs in shape optimization within the framework of Hadamard's boundary variation method. The cornerstone of this formulation is the classical operation of extension and regularization of shape derivatives. The numerical efficiency and ease of implementation of our algorithm are demonstrated on realistic shape optimization problems.

Keywords. nonlinear constrained optimization, gradient flows, shape and topology optimization, null space method.

AMS Subject classifications. 65K10, 49Q10, 34C35, 49B36, 65L05.

Contents

Corresponding author. Email: florian.feppon@polytechnique.edu.
[1]Institute of Engineering Univ. Grenoble Alpes

1. Introduction

The increasing popularity encountered by shape and topology optimization algorithms in industry calls for their use in more and more realistic physical contexts. In such applications, the optimized designs are often subjected to a large number of complex engineering constraints. To name a few of them, it is often required that the stress developed inside mechanical structures do not exceed a prescribed safety threshold [6, 28, 40]; it is also customary to impose constraints on the geometry of the optimized shape—e.g. on the thickness of its structural members, on its curvature radius, etc. [8, 55]—, or in keeping with manufacturability issues; see for instance [7, 4, 63], or [42] for an overview. This raises the need for advanced constrained optimization algorithms, adapted to the specificities of shape optimization.

Over the past decades, many iterative algorithms have been devised to solve for generic constrained optimization problems of the form:

$$\min_{x\in V} \quad J(x) \qquad \text{s.t.} \begin{cases} \boldsymbol{g}(x) = 0 \\ \boldsymbol{h}(x) \leq 0, \end{cases} \tag{1.1}$$

where V is typically a Hilbert space, $J : V \to \mathbb{R}$ is a differentiable objective function, $\boldsymbol{g} : V \to \mathbb{R}^p$ and $\boldsymbol{h} : V \to \mathbb{R}^q$ are differentiable functions accounting for p equality and q inequality constraints, respectively. 'Classical' gradient-based algorithms for the numerical resolution of (1.1) include, e.g., Penalty, Lagrangian, Interior Point and Trust Region Methods, Sequential Quadratic or Linear Programming (SQP or SLP) [17, 48, 66, 34, 61], the Method of Moving Asymptotes (MMA) [58], the Method of Feasible Directions (MFD) [68, 60].

As a matter of fact, advanced mathematical programming methods are not frequently described in the literature devoted to shape optimization based on Hadamard's method (see [35, 56] for an introduction). In most contributions, where, usually, only one constraint is considered, standard Penalty and Augmented Lagrangian Methods are used for the sake of implementation simplicity [9, 23]. Morin et. al. introduced a variant of SQP in [46] but the volume constraint featured in the optimization problem is treated with a Lagrange Multiplier method. When it comes to more complex applications, some authors have introduced adapted variants of Sequential Linear Programming [27] or of the Method of Feasible Direction [30]. However, a major difficulty related to the practical use of these algorithms in topology optimization lies in that the aforementioned techniques require fine tuning of the algorithm parameters in order to actually solve the minimization problem. These parameters are e.g. the penalty coefficients in the Augmented Lagrangian and Interior Point methods, the size of the trust region in SLP algorithms, the strategy for approximating the Hessian matrix in SQP, the bounds on the asymptotes in MMA and the Topkis parameters in MFD. The correct determination of these parameters is strongly case-dependent and often unintuitive: for instance, the penalty coefficients must be neither 'too large' nor 'too small' in Lagrangian methods, the SLP trust region size—which acts as a step length—cannot be chosen too small (otherwise the involved quadratic subproblems may not have a solution). In shape and topology optimization practice, a fair amount of trials and errors is often required in order to obtain satisfying minimizing sequences of shapes. Since every optimization step depends on the resolution of partial differential equations, such tunings are very tedious, time consuming for 2-d cases, and downright unaffordable for realistic 3-d applications.

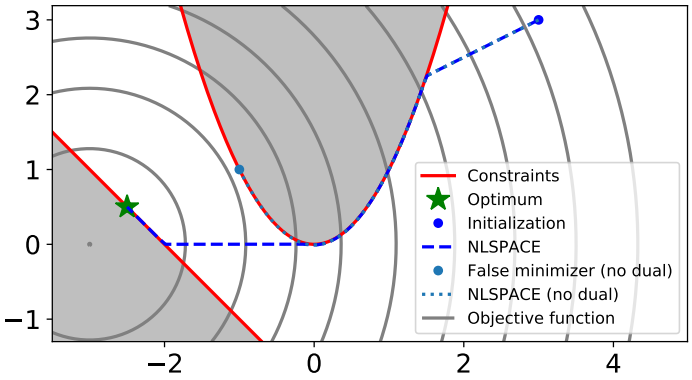


FIGURE 1. An example of optimization trajectory produced by our null space gradient flow (1.2) (labeled 'NLSPACE'). Trajectories travel tangentially to an optimal subset $\widehat{I}(x) \subset \widetilde{I}(x)$ of the active constraints $\widetilde{I}(x)$, which is determined by a dual problem (see Section 3). A less optimal trajectory (labeled 'NLSPACE (no dual)') is obtained if the set $\widehat{I}(x)$ is not identified, because it is unable to escape the tangent space to the constraints labeled by $\widetilde{I}(x)$.

The first main contribution of this article is to propose a novel algorithm for constrained optimization which is rather easy to implement and reliable in the sense that it allows to solve (1.1) without the need for tuning non physical parameters; it is therefore particularly well adapted to the specificities of shape and topology optimization applications. The essence of our method is a modification of the celebrated gradient flow which enables it to 'see the constraints': optimization trajectories $x(t)$ are obtained by solving an Ordinary Differential Equation (ODE):

$$\dot{x}(t) = -\alpha_J \boldsymbol{\xi}_J(x(t)) - \alpha_C \boldsymbol{\xi}_C(x(t)), \tag{1.2}$$

where the descent direction $\dot{x}$ is a combination of a so-called 'null space' direction $\boldsymbol{\xi}_J(x)$ and a 'range space' direction $\boldsymbol{\xi}_C(x)$, lying respectively in the null space of the constraint set and in its orthogonal complement (for this reason, we call the ODE (1.2) a 'null space' gradient flow). The null space direction $\boldsymbol{\xi}_J(x)$ is the projection of the gradient $\nabla J(x)$ onto the cone of feasible directions (see Figure 1). The range space direction $\boldsymbol{\xi}_C(x)$ is a Gauss-Newton direction, aimed to smoothly drive the optimization path toward the feasible domain. Finally, $\alpha_J, \alpha_C > 0$ are two (optional) parameters scaling the imposed decrease rates of the objective function and of the violation of the constraints; we shall see in particular that the latter quantity decreases along trajectories $x(t)$ at least as fast as $e^{-\alpha_C t}$. In numerical practice, (1.2) is solved by using a suitable discretization scheme; one which works well to solve a number of shape optimization examples or in $\mathbb{R}^d$ is proposed in Algorithm 1 below.

The cornerstone of our method is the computation of the null space direction $\boldsymbol{\xi}_J(x)$; it relies on the resolution of a dual program to identify an 'optimal' subset $\widehat{I}(x)$ of the set of active inequality constraints $\widetilde{I}(x) \subset \{1, \dots, q\}$ to which the optimization path must remain tangent. The remaining constraints $\widetilde{I}(x) \setminus \widehat{I}(x)$ become inactive, allowing the trajectory to naturally re-enter the feasible domain. More specifically, for a given subset of indices $I \subset \{1, \dots, q\}$, let us denote by $\boldsymbol{h}_I(x) := (h_i(x))_{i \in I}$ the collection of those inequality constraints indexed by I and by $\boldsymbol{C}_I(x)$ the vector:

$$\boldsymbol{C}_I(x) := \begin{bmatrix} \boldsymbol{g}(x) \\ \boldsymbol{h}_I(x) \end{bmatrix}. \tag{1.3}$$

Then, for inequality constrained problems, the directions $\boldsymbol{\xi}_J(x)$ and $\boldsymbol{\xi}_C(x)$ in (1.2) are defined as follows:

$$\boldsymbol{\xi}_J(x) = (\mathrm{I} - \mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widehat{I}(x)}(\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}\mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widehat{I}(x)})^{-1}\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)})\nabla J(x), \tag{1.4}$$

$$\boldsymbol{\xi}_C(x) = \mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widetilde{I}(x)}(\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}\mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widetilde{I}(x)})^{-1}\boldsymbol{C}_{\widetilde{I}(x)}(x), \tag{1.5}$$

where I is the identity matrix and $(\mathrm{D}\boldsymbol{C}(x))_{ij} = \partial_j C_i(x)$ denotes the Jacobian matrix of a vector function $\boldsymbol{C}(x) = (C_i(x))_i$ (the dependence with respect to x is omitted when the context is clear). The symbol

$\mathcal{T}$ denotes the transposition operator; it may differ from the usual transpose T if the optimization set V is infinite-dimensional (see below and Section 2). Formulas (1.4) and (1.5) involve two different subsets $\widehat{I}(x) \subset \widetilde{I}(x) \subset \{1, ..., q\}$ of indices of inequality constraints: the first one $\widetilde{I}(x)$ is the set of all saturated or violated constraints, defined by

$$\widetilde{I}(x) = \{i \in \{1, \dots, q\} \,|\, h_i(x) \geq 0\}. \tag{1.6}$$

The set $\widehat{I}(x) \subset \widetilde{I}(x)$ is characterized thanks to the introduction of the following 'dual' quadratic optimization subproblem:

$$(\boldsymbol{\lambda}^*(x), \boldsymbol{\mu}^*(x)) := \arg \min_{\substack{\boldsymbol{\lambda} \in \mathbb{R}^p \\ \boldsymbol{\mu} \in \mathbb{R}^{\widetilde{q}(x)}, \, \boldsymbol{\mu} \geq 0}} ||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}} \boldsymbol{\lambda} + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}} \boldsymbol{\mu}||_V. \tag{1.7}$$

The problem (1.7) amounts to computing the projection of $\nabla J(x)$ onto the cone of feasible directions. The set of indices $\widehat{I}(x)$ involved in the definition (1.4) of $\boldsymbol{\xi}_J(x)$ is inferred from the positive entries of the optimal Lagrange multiplier $\boldsymbol{\mu}^*(x)$:

$$\widehat{I}(x) := \{i \in \widetilde{I}(x) \,|\, \mu_i^*(x) > 0\}. \tag{1.8}$$

Proposition 4 shows that this definition of $\widehat{I}(x)$ is somehow 'optimal' since it ensures that $-\boldsymbol{\xi}_J(x)$ is the 'best' descent direction for $J(x)$ respecting locally equality and inequality constraints. This approach turns out to be very efficient when dealing with a large number of (possibly violated) inequality constraints (see Figure 1).

Our ODE (1.2) is a generalization of rather classical dynamical system approaches to nonlinear constrained optimization [44, 59, 65, 51], which seem to be little known within the topology optimization community. The flexibility of these methods lies in that, in principle, an admissible local minimizer to (1.1) is reached as the stationary point x^* of the continuous trajectory $x(t)$, for almost any (feasible or not) initialization $x(0)$, regardless of the value of the coefficients $\alpha_J, \alpha_C > 0$ (a formal proof for the ODE (1.2) is out of the scope of this work, but we refer to [39, 41, 50, 51] for related results when inequality constraints are not present). The descent properties of (1.2) are expected to be preserved at the discrete level provided it is discretized with a sufficiently small Euler step size Δt (see Remark 13 below). As a result, the success of the use of our method depends truly only on the selection of a sufficiently small step Δt, and to a less extent on the physically interpretable, dimensionless parameters α_J, α_C, whose tuning by the user is quite intuitive.

When the problem (1.1) features no constraint, (1.4) and (1.5) read simply $\boldsymbol{\xi}_J(x) = \nabla J(x)$ and $\boldsymbol{\xi}_C(x) = 0$, so that the ODE (1.2) reduces to the standard gradient flow

$$\dot{x}(t) = -\nabla J(x(t)). \tag{1.9}$$

When (1.1) features only equality constraints $\boldsymbol{g}(x) = 0$, but no inequality constraint (in that case $\boldsymbol{C}_{\widetilde{I}(x)}(x) = \boldsymbol{C}_{\widehat{I}(x)}(x) = \boldsymbol{g}(x)$), the same ODE (1.2) as ours was previously derived and studied in the early 1980s by Tanabe [59] (without the use of the Gauss-Newton direction $\boldsymbol{\xi}_C(x(t))$) and by Yamashita [65] (where a combination of both directions $\boldsymbol{\xi}_J(x(t))$ and $\boldsymbol{\xi}_C(x(t))$ is brought into play), in the finite-dimensional setting $V = \mathbb{R}^k$. In this particular case, the solution to our dual problem (1.7) has a closed-form expression and (1.2) reads with our notation:

$$\dot{x}(t) = -\alpha_J(\mathrm{I} - \mathrm{D}\boldsymbol{g}^{\mathcal{T}}(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\mathrm{D}\boldsymbol{g})\nabla J(x(t)) - \alpha_C \mathrm{D}\boldsymbol{g}^{\mathcal{T}}(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\boldsymbol{g}(x(t)). \tag{1.10}$$

In the general situation where (1.1) features both inequality and equality constraints, variants of the ODE (1.10) have been considered by various authors, with a different method from ours, however [51, 37, 38, 54]. The most common approach in the literature consists in introducing q slack variables $\{z_i\}_{1 \leq i \leq q} \in \mathbb{R}^q$ to transform the q inequalities $h_i(x) \leq 0$ for $1 \leq i \leq q$ into as many equality constraints $h_i(x) + z_i^2 = 0$, before solving the ODE (1.2) in the augmented space $(x, z) \in V \times \mathbb{R}^q$. This approach offers convergence guarantees [51] and could also be beneficial for shape optimization, however this is not the strategy we have retained. Indeed, our method does not need to resort to slack variables for handling inequality constraints, and it presents additional advantages described in Section 3.5.

Our second main contribution is the exposure of our dynamical system strategy in a setting compatible with the inherently infinite-dimensional nature of shape optimization based on the method of Hadamard. In such a context, a clear distinction between the Fréchet derivative $\mathrm{D}J(x(t))$ (which is an element of the dual space V') and the gradient $\nabla J(x(t))$ (which is an element of V) is in order: the gradient $\nabla J(x(t))$ is

obtained by identifying $\mathrm{D}J(x(t))$ with an element of V thanks to the Riesz representation theorem. The same distinction is also needed between the differential of a vector valued function $\mathrm{D}\boldsymbol{C}(x(t))$ and its transpose $\mathrm{D}\boldsymbol{C}(x(t))^{\mathcal{T}}$; see Definition 1.

For shape optimization applications, the minimization set in (1.1) is the set of all open Lipschitz domains Ω enclosed in some 'hold-all' domain $D \subset \mathbb{R}^d$. This set is not a vector space, but it can be locally parameterized by the Sobolev space $W^{1,\infty}(D,\mathbb{R}^d)$. More precisely, for a given Lipschitz open set $\Omega \subset D$, one can restrict the minimization to the space

$$\mathcal{O} = \big\{(\mathrm{I}+\boldsymbol{\theta})(\Omega) \text{ with } \boldsymbol{\theta} \in W^{1,\infty}(D,\mathbb{R}^d)\big\} \tag{1.11}$$

of variations of Ω parametrized by vector fields $\boldsymbol{\theta}$. This space $\mathcal{O}$ is a Banach space (but not a Hilbert space), after being identified with $W^{1,\infty}(D,\mathbb{R}^d)$. The vector field $\boldsymbol{\theta}$ can be interpreted as a displacement field, and the minimized shape functional $\Omega \mapsto J(\Omega)$ (like the constraint functionals) is restricted to a functional $\boldsymbol{\theta} \mapsto J((\mathrm{I}+\boldsymbol{\theta})(\Omega))$ defined on $\mathcal{O}$, whose derivative can be computed in the sense of Fréchet [47, 56, 35]. The space $W^{1,\infty}(D,\mathbb{R}^d)$ can be interpreted as a 'tangent space' for the 'manifold' of all open Lipschitz domain $\Omega \subset D$. In practice, the identification of the aforementioned Fréchet derivative with a gradient is achieved by solving an extension and regularization problem, which has major consequences in numerical practice, see e.g. [21, 24]. This step is naturally and consistently included in our algorithm thanks to the suitable definition of the transposition operator $\mathcal{T}$. So far, this matter does not seem to have been clearly addressed in the literature concerned with constrained shape optimization: common approaches rather compute a descent direction *first*, before performing a regularization, see e.g. [27, 30].

Several contributions in the field of shape and topology optimization can be related to ours. In fact, our method is very close in spirit to the recent work of Barbarosie et. al. [16], where an iterative algorithm for equality constrained optimization is devised, which turns out to be a discretization of (1.2) with a variable scaling for the parameter α_C. When dealing with inequality constraints, the authors propose an active set strategy which is based—like ours— on the extraction of an appropriate subset of the active constraints (without convergence guarantee, however). This strategy relies on a different algorithm from ours, which generally yields a different (suboptimal) set than $\widehat{I}(x)$ whose mathematical properties are a little unclear; see Remark 4 below for more details. Finally, Yulin and Xiaoming also suggested in [67] to project the gradient of the objective function onto the cone of feasible directions; nevertheless, they remained elusive regarding how the projection problem is solved or how violated constraints are tackled.

An open-source implementation of our null space algorithm is made freely available online at

`https://gitlab.com/safrandrti/null-space-project-optimizer`.

The repository includes demonstrative pedagogical examples for the resolution of optimization problems in $\mathbb{R}^d$, but can also serve as a basis for the resolution of more complicated applications; it has been used as is for the realisation of the shape optimization test cases of this article and other of our related works [32, 31].

The present article is organized as follows. In Section 2, we review the definition and the properties of the gradient flow (1.2) for equality constrained optimization, in the case where the minimization set V is a Hilbert space. We detail then in Section 3 the necessary adaptations to account for inequality constraints and in particular the introduction of the dual subproblem allowing to determine the null space direction $\boldsymbol{\xi}_J(x)$. Under some technical assumptions, we prove in Proposition 5 the convergence of our 'null space' gradient flow (1.2) towards points satisfying the full Karush Kuhn Tucker (KKT) first-order necessary conditions for optimality. Several algorithmic implementation aspects of our method are discussed in Section 4. From Section 5 on, we then concentrate on shape optimization applications. After clarifying the necessary adaptations required to extend the discretization of (1.2) when the optimization set V is only locally a Banach space such as $W^{1,\infty}(D,\mathbb{R}^d)$, we explain how our algorithm can be integrated within the level set method for shape optimization [9, 62]. Numerical examples are eventually proposed in Section 6: we first solve an optimal design problem in thermoelasticity inspired from [64] for which the optimized solutions feature active and inactive inequality constraints—an example meant to emphasize the key role of our dual problem in the identification of the optimal subset of inequality constraints which need to be enforced at each stage of the optimization process. Then, we consider the shape optimization of a bridge structure subject to multiple loads, which involves up to ten constraints. Many more shape optimization applications built upon

this algorithm, including 3-d multiphysics test cases, are provided in the PhD dissertation [31]. The article ends with a technical appendix containing the proofs of two theoretical results stated in the main text.

2. Gradient flows for equality-constrained optimization in Hilbert spaces

Before turning to the general shape optimization setting in Section 5, we first consider in this section (and the next ones) the case where the optimization takes place in a Hilbert space V with inner product $\langle\cdot,\cdot\rangle_V$ and relative norm $||\cdot||_V = \langle\cdot,\cdot\rangle_V^{1/2}$. The first focus of our study is the minimization problem (1.1) in a simplified version where only equality constraints are present, namely:

$$\begin{aligned} &\min_{x\in V} \quad J(x) \\ &\text{s.t. } \boldsymbol{g}(x) = 0, \end{aligned} \tag{2.1}$$

where $J : V \to \mathbb{R}$ and $\boldsymbol{g} : V \to \mathbb{R}^p$ are Fréchet differentiable functions. Our purpose is to recall how the ODE approach (1.2) works in the present Hilbertian setting. Let us emphasize that, although this section is quite elementary and not new *per se*, it is not easily found as is in the literature. Since it is key in understanding our method for handling inequality constraints in Section 3, the present context is thoroughly detailed for the reader's convenience.

2.1. Notation and first-order optimality conditions

We start by setting notation about differentiability and gradients in Hilbert spaces that we use throughout this article. As we have mentioned indeed, a clear distinction between gradients and Fréchet derivatives proves crucial in our shape optimization applications in Section 5.

Definition 1.

(1) A vector-valued function $\boldsymbol{g} : V \to \mathbb{R}^p$ is *differentiable* at a point $x \in V$ if there exists a continuous linear mapping $\mathrm{D}\boldsymbol{g}(x) : V \to \mathbb{R}^p$ such that

$$\boldsymbol{g}(x+h) = \boldsymbol{g}(x) + \mathrm{D}\boldsymbol{g}(x)h + o(h) \text{ with } \frac{o(h)}{||h||_V} \xrightarrow{h\to 0} 0. \tag{2.2}$$

$\mathrm{D}\boldsymbol{g}(x)$ is called the Fréchet derivative of $\boldsymbol{g}$ at x.

(2) If $\boldsymbol{g} : V \to \mathbb{R}^p$ is differentiable, for any $\boldsymbol{\mu} \in \mathbb{R}^p$, the Riesz representation theorem [18] ensures the existence of a unique function $\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\mu} \in V$ satisfying

$$\forall \boldsymbol{\mu} \in \mathbb{R}^p, \forall \boldsymbol{\xi} \in V, \ \langle \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\mu}, \boldsymbol{\xi}\rangle_V = \boldsymbol{\mu}^T \mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi}, \tag{2.3}$$

where the superscript ^{T} stands for the usual transpose of a vector in the Euclidean space $\mathbb{R}^p$. The linear operator $\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}} : \mathbb{R}^p \to V$ thus defined is called the *transpose* of $\mathrm{D}\boldsymbol{g}(x)$.

(3) If $J : V \to \mathbb{R}$ is a scalar function which is differentiable at $x \in V$, the Riesz representation theorem ensures the existence of a unique vector $\nabla J(x) \in V$ satisfying

$$\forall \boldsymbol{\xi} \in V, \langle \nabla J(x), \boldsymbol{\xi}\rangle_V = \mathrm{D}J(x)\boldsymbol{\xi}. \tag{2.4}$$

This vector $\nabla J(x)$ is called the *gradient* of J at x.

Throughout the following, we shall sometimes omit the explicit mention to x in the notation for differentials or gradients when the considered point $x \in V$ is clear, so as to keep expressions as light as possible.

Remark 1.

(1) If V is the (finite-dimensional) Euclidean space $\mathbb{R}^k$, and $\langle\cdot,\cdot\rangle_V$ is the usual inner product, the Fréchet derivative (resp. its transpose) of a vector function $\boldsymbol{g} : \mathbb{R}^k \to \mathbb{R}^p$ are respectively given by the Jacobian matrix $(\mathrm{D}\boldsymbol{g})_{ij} = \partial_j g_i$ (resp. $(\mathrm{D}\boldsymbol{g}^{\mathcal{T}})_{ij} = (\mathrm{D}\boldsymbol{g}^T)_{ij} = \partial_i g_j$). In the literature, the differential matrix $\mathrm{D}\boldsymbol{g}$ is often denoted with the nabla notation $\nabla\boldsymbol{g}$. For the sake of clarity, we reserve the ∇ symbol for the gradient of scalar functions $J : V \to \mathbb{R}$. The calligraphic transpose notation $\mathcal{T}$ appearing in the objects $\mathrm{D}J(x)^{\mathcal{T}}$ or $\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}$ encodes at the same time the operator transposition (reversing the input and range spaces) and the Riesz identifications. In particular, it holds that $\nabla J(x) = \mathrm{D}J(x)^{\mathcal{T}}1$.

(2) Still in the case where $V = \mathbb{R}^k$ is finite-dimensional, but the inner product $\langle \cdot, \cdot \rangle_V$ is given by a symmetric positive definite matrix A (that is, $\langle \boldsymbol{\xi}, \boldsymbol{\xi} \rangle_V = \boldsymbol{\xi}^T A \boldsymbol{\xi}$), the calligraphic transpose of a $p \times k$ matrix $M : \mathbb{R}^k \to \mathbb{R}^p$ with respect to $\langle \cdot, \cdot \rangle_V$ now reads $M^{\mathcal{T}} = A^{-1} M^T$. In shape optimization applications, the inner product $\langle \cdot, \cdot \rangle_V$ often stands for the bilinear form associated to an elliptic operator, and the calligraphic transpose $\mathcal{T}$ encompasses the extension and regularization steps of the shape derivative, see Section 5.2 below.

(3) If V is replaced by the tangent space to a Riemannian manifold, the bilinear form $\langle \cdot, \cdot \rangle_V$ can be interpreted as a metric and $\nabla J(x)$, as given by (2.4), is the covariant gradient with respect to this metric.

(4) When V is a general Hilbert space, for a vector-valued function $\boldsymbol{g} : V \to \mathbb{R}^p$ with components $\boldsymbol{g}(x) = (g_i(x))_{1 \leq i \leq p}$, $\mathrm{D}\boldsymbol{g} : V \to \mathbb{R}^p$ is the 'row' matrix whose entries are the p linear forms $\mathrm{D}g_i(x) : V \to \mathbb{R}$. The transpose $\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}$ is the 'column' matrix gathering the p vectors $(\nabla g_i(x))_{1 \leq i \leq p}$ obtained by solving the p identification problems, similar to (2.4), $\langle \nabla g_i(x), \boldsymbol{\xi} \rangle_V = \mathrm{D}g_i(x)\boldsymbol{\xi}$ for any $\boldsymbol{\xi} \in V$. More precisely, $\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}} \boldsymbol{\mu} = \sum_{i=1}^{p} \mu_i \nabla g_i(x)$ for any $\boldsymbol{\mu} \in \mathbb{R}^d$. In particular, the $p \times p$ matrix $\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}}$ has entries $(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})_{ij} = \langle \nabla g_i, \nabla g_j \rangle_V = Dg_i(x)(\nabla g_j(x))$.

In the present section, the equality constraints are said to satisfy the "Linear Independence Constraint Qualification" (LICQ) condition at a point $x \in V$ if

$$\mathrm{rank}(\mathrm{D}\boldsymbol{g}(x)) = p, \text{ or equivalently } \mathrm{D}\boldsymbol{g}(x)\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}} \text{ is an invertible } p \times p \text{ matrix.} \tag{2.5}$$

Note that (2.5) makes sense even at points $x \in V$ where $\boldsymbol{g}(x) \neq 0$, an essential fact in the sequel: as we have mentioned in the introduction, realistic shape optimization problems are often initialized with an unfeasible design. Under the above notation, let us recall the classical first-order necessary optimality conditions (KKT) for the equality-constrained problem (2.1) at some point $x^* \in V$ where the constraints are satisfied and qualified: there exists $\boldsymbol{\lambda}(x^*) \in \mathbb{R}^p$ such that,

$$\begin{cases} \nabla J(x^*) + \mathrm{D}\boldsymbol{g}(x^*)^{\mathcal{T}} \boldsymbol{\lambda}(x^*) = 0, \\ \boldsymbol{g}(x^*) = 0; \end{cases} \tag{2.6}$$

see for instance [17, 48].

2.2. Definitions and properties of the null space and range space steps $\boldsymbol{\xi}_J$ and $\boldsymbol{\xi}_C$

We are now in position to define the null space and range space steps $\boldsymbol{\xi}_J(x)$ and $\boldsymbol{\xi}_C(x)$ featured in the ODE (1.2) for equality constrained problems in the present Hilbert space setting.

Definition 2. Consider the optimization problem (2.1). For any point $x \in V$ satisfying the LICQ condition (2.5), we define the *null space* and *range space* directions $\boldsymbol{\xi}_J(x)$ and $\boldsymbol{\xi}_C(x)$ by, respectively:

$$\boldsymbol{\xi}_J(x) := (\mathrm{I} - \mathrm{D}\boldsymbol{g}^{\mathcal{T}}(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\mathrm{D}\boldsymbol{g})\nabla J(x), \tag{2.7}$$

$$\boldsymbol{\xi}_C(x) := \mathrm{D}\boldsymbol{g}^{\mathcal{T}}(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\boldsymbol{g}(x). \tag{2.8}$$

In the finite-dimensional case where $V = \mathbb{R}^k$, it is well-known that the null space step $\boldsymbol{\xi}_J(x)$ in (2.7) is the orthogonal projection of the gradient $\nabla J(x)$ onto the null space of the constraints

$$\mathrm{Ker}(\mathrm{D}\boldsymbol{g}(x)) = \{\boldsymbol{\xi} \in V \,|\, \mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi} = 0\},$$

which is also the tangent space at x to the manifold $\{y \in V \,|\, \boldsymbol{g}(y) = \boldsymbol{g}(x)\}$. This is still true when V is a Hilbert space, as we recall in the next lemma.

Lemma 1. *Let $x \in V$ be a point where the LICQ condition (2.5) is satisfied. Then:*

(1) The space V has the following orthogonal decomposition:

$$V = \mathrm{Ker}(\mathrm{D}\boldsymbol{g}(x)) \oplus \mathrm{Ran}(\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}),$$

where the range is defined as $\mathrm{Ran}(\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}) := \{\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda} \,|\, \boldsymbol{\lambda} \in \mathbb{R}^p\}$. Moreover, the operator $\Pi_{g(x)} : V \to V$ defined by $\Pi_{g(x)} = \mathrm{I} - \mathrm{D}\boldsymbol{g}^{\mathcal{T}}(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\mathrm{D}\boldsymbol{g}(x)$ is the orthogonal projection onto $\mathrm{Ker}(\mathrm{D}\boldsymbol{g}(x))$.

(2) *When* $\Pi_{g(x)}(\nabla J(x)) \neq 0$, $\boldsymbol{\xi}_J(x) = \Pi_{g(x)}(\nabla J(x))$ *is the best feasible descent direction for* J *from* x *(up to a normalization) in the sense that*

$$-\frac{\boldsymbol{\xi}_J(x)}{||\boldsymbol{\xi}_J(x)||_V} = \arg\min_{\boldsymbol{\xi}\in V} \quad \mathrm{D}J(x)\boldsymbol{\xi} \quad \text{s.t.} \begin{cases} \mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi} = 0 \\ \langle \boldsymbol{\xi}, \boldsymbol{\xi}\rangle_V \leq 1. \end{cases} \tag{2.9}$$

(3) *The null space direction* $\boldsymbol{\xi}_J(x) = \Pi_{g(x)}(\nabla J(x))$ *is the closest least-squares approximation of* $\nabla J(x)$ *within the space* $\mathrm{Ker}(\mathrm{D}\boldsymbol{g}(x))$*. It alternatively reads*

$$\boldsymbol{\xi}_J(x) = \nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda}^*(x), \tag{2.10}$$

where the Lagrange multiplier $\boldsymbol{\lambda}^*(x) := -(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\mathrm{D}\boldsymbol{g}\nabla J(x) \in \mathbb{R}^p$ *is the unique solution to the dual problem of* (2.9)*, which reads:*

$$\boldsymbol{\lambda}^*(x) = \arg\min_{\boldsymbol{\lambda}\in\mathbb{R}^p} ||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda}||_V. \tag{2.11}$$

Proof.

(1) Any $\boldsymbol{\xi} \in V$ may be decomposed as $\boldsymbol{\xi} = \Pi_{g(x)}(\boldsymbol{\xi}) + (\mathrm{I} - \Pi_{g(x)})(\boldsymbol{\xi})$, where it is straightforward to verify that $\Pi_{g(x)}(\boldsymbol{\xi}) \in \mathrm{Ker}(\mathrm{D}\boldsymbol{g}(x))$, and $(\mathrm{I}-\Pi_{g(x)})(\boldsymbol{\xi}) \in \mathrm{Ran}(\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}})$. In addition, $\mathrm{Ker}(\mathrm{D}\boldsymbol{g}(x))$ and $\mathrm{Ran}(\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}})$ are orthogonal for the inner product a since from (2.3), one has,

$$\forall \boldsymbol{\zeta} \in \mathrm{Ker}(\mathrm{D}\boldsymbol{g}(x)),\ \forall \boldsymbol{\lambda} \in \mathbb{R}^p,\ \ \langle \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda}, \boldsymbol{\zeta}\rangle_V = \boldsymbol{\lambda}^T\mathrm{D}\boldsymbol{g}(x)\boldsymbol{\zeta} = 0.$$

(2) It follows from the first point that for any $\boldsymbol{\xi} \in \mathrm{Ker}(\mathrm{D}\boldsymbol{g}(x))$ such that $||\boldsymbol{\xi}||_V \leq 1$,

$$\mathrm{D}J(x)\boldsymbol{\xi} = \langle \nabla J(x), \boldsymbol{\xi}\rangle_V = \langle \Pi_{g(x)}(\nabla J(x)), \boldsymbol{\xi}\rangle_V \geq -||\Pi_{g(x)}(\nabla J(x))||_V,$$

whence we easily infer that $\boldsymbol{\xi} := -\Pi_{g(x)}(\nabla J(x))/||\Pi_{g(x)}(\nabla J(x))||_V$ is the global minimizer of (2.9).

(3) The Pythagore identity yields, for any $\boldsymbol{\xi} \in \mathrm{Ker}(\mathrm{D}\boldsymbol{g}(x))$,

$$||\nabla J(x) - \boldsymbol{\xi}||_V^2 = ||(\mathrm{I} - \Pi_{g(x)})\nabla J(x)||_V^2 + ||\Pi_{g(x)}\nabla J(x) - \boldsymbol{\xi}||_V^2 \geq ||\nabla J(x) - \Pi_{g(x)}\nabla J(x)||_V^2.$$

Hence the orthogonal projection $\Pi_{g(x)}(\nabla J(x))$ is the best approximation of $\nabla J(x)$ within $\mathrm{Ker}(\mathrm{D}\boldsymbol{g}(x))$. On the other hand, recalling from the first point that $\mathrm{Ran}(\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}})$ is the orthogonal complement of $\mathrm{Ker}(\mathrm{D}\boldsymbol{g}(x))$, we obtain also, for any $\boldsymbol{\lambda} \in \mathbb{R}^p$,

$$||\Pi_{g(x)}(\nabla J(x))||_V = ||\nabla J(x) - (\mathrm{I} - \Pi_{g(x)})(\nabla J(x))||_V \leq ||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda}||_V,$$

whence the expression (2.10) and the minimization property (2.11) follow. Note that the uniqueness of the solution $\boldsymbol{\lambda}^*(x)$ to (2.11) results from the LICQ condition (2.5).
Finally, the optimization problem (2.9) can be rewritten as

$$\min_{\substack{\boldsymbol{\xi}\in V \\ \langle \boldsymbol{\xi},\boldsymbol{\xi}\rangle_V \leq 1}} \ \max_{\boldsymbol{\lambda}\in\mathbb{R}^p} \quad \mathrm{D}J(x)\boldsymbol{\xi} + \boldsymbol{\lambda}^T\mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi}.$$

Hence the (formal) dual problem of (2.9) reads:

$$\max_{\boldsymbol{\lambda}\in\mathbb{R}^p} \ \min_{\substack{\boldsymbol{\xi}\in V \\ \langle \boldsymbol{\xi},\boldsymbol{\xi}\rangle_V \leq 1}} \quad \mathrm{D}J(x)\boldsymbol{\xi} + \boldsymbol{\lambda}^T\mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi}.$$

According to the definitions (2.3) and (2.4) of the gradient and of the Hilbertian transpose, the latter problem rewrites:

$$\max_{\boldsymbol{\lambda}\in\mathbb{R}^p} \ \min_{\substack{\boldsymbol{\xi}\in V \\ \langle \boldsymbol{\xi},\boldsymbol{\xi}\rangle_V \leq 1}} \ \langle \nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda}, \boldsymbol{\xi}\rangle_V = -\max_{\boldsymbol{\lambda}\in\mathbb{R}^p} ||\nabla J + \mathrm{D}\boldsymbol{g}^{\mathcal{T}}\boldsymbol{\lambda}||_V, \tag{2.12}$$

where for given $\boldsymbol{\lambda} \in \mathbb{R}^p$, the value

$$\boldsymbol{\xi}^* := \frac{\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda}}{||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda}||_V}$$

is that achieving the minimum in the inner minimization problem at the left-hand side of (2.12). This shows that (2.11) is the dual problem of (2.9).

□

The next lemma is also fairly classical in the literature, at least in the finite-dimensional case. It characterizes the range space step $\boldsymbol{\xi}_C(x)$, defined by (2.8), as the unique Gauss-Newton direction for the cancellation of the constraint function $\boldsymbol{g}(x)$ which is orthogonal to the (linearized) set of constraints:

Lemma 2. *Let $x \in V$ be a point where the LICQ condition (2.5) is satisfied. Then:*

(1) The range space step $\boldsymbol{\xi}_C(x) = \mathrm{D}\boldsymbol{g}^{\mathcal{T}}(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\boldsymbol{g}(x)$ is orthogonal to $\mathrm{Ker}(\mathrm{D}\boldsymbol{g}(x))$:

$$\forall \boldsymbol{\xi} \in \mathrm{Ker}(\mathrm{D}\boldsymbol{g}(x)),\ \ \langle \boldsymbol{\xi}_C(x), \boldsymbol{\xi} \rangle_V = 0.$$

(2) $-\boldsymbol{\xi}_C(x)$ is a descent direction for the violation of the constraints:

$$\mathrm{D}\boldsymbol{g}(x)(-\boldsymbol{\xi}_C(x)) = -\boldsymbol{g}(x). \tag{2.13}$$

(3) The set of solutions to the Gauss-Newton program

$$\min_{\boldsymbol{\xi} \in V} ||\boldsymbol{g}(x) + \mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi}||^2 \tag{2.14}$$

is the affine subspace $\{-\boldsymbol{\xi}_C(x) + \boldsymbol{\zeta} \mid \boldsymbol{\zeta} \in \mathrm{Ker}(\mathrm{D}\boldsymbol{g}(x))\}$ of V.

Proof. Point (1) is an immediate consequence of point (1) in Lemma 1. Point (2) is obvious from the definition (2.8) of $\boldsymbol{\xi}_C(x)$. Note that (2.13) means that $-\boldsymbol{\xi}_C(x)$ is a descent direction for the violation of the constraints in the sense that it ensures that any coordinate $g_i(x)$, $i = 1, ..., p$, decreases along $-\boldsymbol{\xi}_C(x)$ if $g_i(x) \geq 0$ and increases if $g_i(x) \leq 0$.

To prove point (3), since (2.14) is a convex optimization problem, a necessary and sufficient condition for $\boldsymbol{\xi} \in V$ to be one solution is given by the usual first-order condition:

$$\forall \boldsymbol{\zeta} \in V,\ \ (\boldsymbol{g}(x) + \mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi})^T(\mathrm{D}\boldsymbol{g}(x)\boldsymbol{\zeta}) = \langle \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}(\boldsymbol{g}(x) + \mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi}), \boldsymbol{\zeta} \rangle_V = 0,$$

which rewrites:

$$\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi} = -\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{g}(x).$$

Since the matrix $(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})$ is invertible, this is in turn equivalent to $\mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi} = -\boldsymbol{g}(x)$. Finally, we know from point (2) that $-\boldsymbol{\xi}_C(x)$ is one particular solution to this last equation. Thus, point (3) follows from the fact that any two solutions of this problem differ up to an element $\boldsymbol{\zeta} \in V$ such that $\mathrm{D}\boldsymbol{g}(x)\boldsymbol{\zeta} = 0$. □

2.3. Decrease properties of the equality constrained gradient flow

The main features of the definitions of $\boldsymbol{\xi}_J(x)$ and $\boldsymbol{\xi}_C(x)$ are the facts that $\boldsymbol{\xi}_J(x)$ is orthogonal to the set of constraints, i.e. $\mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi}_J(x) = 0$, and that $-\boldsymbol{\xi}_C(x)$ decreases the violation of the constraints while being orthogonal to $\boldsymbol{\xi}_J(x)$. These ensure that the entries of the constraint functional $\boldsymbol{g}(x(t))$ tend to 0 along the trajectories of the ODE (1.2), independently of the value of $\boldsymbol{\xi}_J(x)$. Then, as soon as the violation of the constraints becomes sufficiently small, the objective function $J(x(t))$ decreases without compromising the asymptotic vanishing of $\boldsymbol{g}(x(t))$. We review these properties in the next proposition, which was also observed in [65] in the finite-dimensional context; see Appendix A for the proof.

Proposition 1. *Assume that the trajectories $x(t)$ of the flow*

$$\begin{cases} \dot{x} = -\alpha_J(\mathrm{I} - \mathrm{D}\boldsymbol{g}^{\mathcal{T}}(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\mathrm{D}\boldsymbol{g}(x))\nabla J(x) - \alpha_C \mathrm{D}\boldsymbol{g}^{\mathcal{T}}(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\boldsymbol{g}(x) \\ x(0) = x_0 \end{cases} \tag{2.15}$$

exist on some time interval $[0, T]$ for $T > 0$, and that the LICQ condition (2.5) holds at any point $x(t)$, $t \in [0, T]$. Then the following properties hold true:

(1) The violation of the constraints vanishes at exponential rate:

$$\forall t \in [0, T],\ \ \boldsymbol{g}(x(t)) = e^{-\alpha_C t}\boldsymbol{g}(x_0). \tag{2.16}$$

(2) The value $J(x(t))$ of the objective decreases 'as soon as the violation (2.16) of the constraints is sufficiently small' in the following sense: assume that $\mathrm{rank}(\mathrm{D}\boldsymbol{g}) = p$ on $K = \{x \in V \mid ||\boldsymbol{g}(x)||_\infty \leq ||\boldsymbol{g}(x_0)||_\infty\}$ and that

$$\sup_{x \in K} ||\nabla J(x)||_V |\sigma_p^{-1}(x)| < +\infty, \tag{2.17}$$

where $\sigma_p(x)$ is the smallest singular value of $\mathrm{D}\boldsymbol{g}(x)$. Then there exists a constant $C > 0$ such that

$$\forall t \in [0,T],\ ||\Pi_{g(x)}(\nabla J(x(t)))||_V^2 > Ce^{-\alpha_C t} \Rightarrow \frac{\mathrm{d}}{\mathrm{d}t}J(x(t)) < 0. \tag{2.18}$$

(3) Any stationary point x^* of (2.15) satisfies the first-order KKT conditions (2.6) of the optimization program (2.1), that is:

$$\begin{cases} \boldsymbol{g}(x^*) = 0 \\ \exists \boldsymbol{\lambda}^* \in \mathbb{R}^p,\ \nabla J(x^*) + \mathrm{D}\boldsymbol{g}^{\mathcal{T}}(x^*)\boldsymbol{\lambda}^* = \Pi_{g(x^*)}(\nabla J(x^*)) = 0. \end{cases} \tag{2.19}$$

3. Extension to equality and inequality constraints

We now proceed to extend the dynamical system (1.2) or (2.15) so as to handle inequality constraints as well. We return to the full optimization problem (1.1), still posed in a Hilbert space V with inner product $\langle \cdot, \cdot \rangle_V$, and where the objective $J : V \to \mathbb{R}$, equality constraints $\boldsymbol{g} : V \to \mathbb{R}^p$ and inequality constraints $\boldsymbol{h} : V \to \mathbb{R}^q$ are differentiable functions.

Inspired by the methodology developed in Section 2, we still propose to solve the equality and inequality constrained problem (1.1) by means of a dynamical system of the form:

$$\begin{cases} \dot{x}(t) = -\alpha_J \boldsymbol{\xi}_J(x(t)) - \alpha_C \boldsymbol{\xi}_C(x(t)) \\ x(0) = x_0, \end{cases} \tag{3.1}$$

whose forward Euler discretized version reads:

$$x_{n+1} = x_n - \Delta t(\alpha_J \boldsymbol{\xi}_J(x_n) + \alpha_C \boldsymbol{\xi}_C(x_n)). \tag{3.2}$$

After introducing notation conventions related to inequality constraints in Section 3.1, the range space step $\boldsymbol{\xi}_C(x)$ is defined in Section 3.2 from a formula analogous to (1.5). The definition of the null space step $\boldsymbol{\xi}_J(x)$ is examined in details in Section 3.3; it involves a procedure for identifying a relevant subset $\widehat{I}(x) \subset \widetilde{I}(x)$ of the saturated or violated constraints, which relies on the dual problem (1.7). Finally, the properties of the resulting flow (3.1) are outlined in Section 3.4.

3.1. Notation and preliminaries

Let us recall the definition (1.6) for the set $\widetilde{I}(x)$ of saturated or violated inequality constraints at $x \in V$. We denote by $\widetilde{q}(x) := \mathrm{Card}(\widetilde{I}(x))$ the number of such constraints. For a given subset $I \subset \{1, ..., q\}$, the vector $\boldsymbol{h}_I(x) = (h_i(x))_{i \in I}$ collects the inequality constraints indexed by I. The vector $\boldsymbol{C}_I(x)$, defined by (1.3), collects all equality constraints $\boldsymbol{g}(x)$ and those selected inequality constraints $\boldsymbol{h}_I(x)$.

In the present context, the constraints are said to satisfy the "Linear Independence Constraint Qualification" (LICQ) condition at $x \in V$ if the linearized saturated or violated constraints are independent, namely if

$$\mathrm{rank}(\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}(x)) = p + \widetilde{q}(x). \tag{3.3}$$

If the point x satisfies the constraints, (3.3) is one usual LICQ condition (of course, there are other possible qualification conditions, see [17, 48]), but (3.3) also applies to points x where constraints are not satisfied. Define $\Pi_{\boldsymbol{C}_I} : V \to V$, the orthogonal projection operator onto $\mathrm{Ker}(\mathrm{D}\boldsymbol{C}_I(x))$, by

$$\Pi_{\boldsymbol{C}_I} = I - \mathrm{D}\boldsymbol{C}_I(x)^{\mathcal{T}}(\mathrm{D}\boldsymbol{C}_I(x)\mathrm{D}\boldsymbol{C}_I(x)^{\mathcal{T}})^{-1}\mathrm{D}\boldsymbol{C}_I(x), \tag{3.4}$$

and let $(\boldsymbol{\lambda}_I(x), \boldsymbol{\mu}_I(x)) \in \mathbb{R}^p \times \mathbb{R}_+^{\mathrm{Card}(I)}$ be the corresponding Lagrange multipliers:

$$\begin{bmatrix} \boldsymbol{\lambda}_I(x) \\ \boldsymbol{\mu}_I(x) \end{bmatrix} := -(\mathrm{D}\boldsymbol{C}_I\mathrm{D}\boldsymbol{C}_I^{\mathcal{T}})^{-1}\mathrm{D}\boldsymbol{C}_I(x)\nabla J(x). \tag{3.5}$$

Last but not least, let us recall that in the present context of the equality and inequality constrained problem (1.1), the necessary first-order optimality conditions (the KKT conditions) at a given point $x^* \in V$

satisfying the LICQ condition (3.3), read as follows: there exist $\boldsymbol{\lambda}(x^*) \in \mathbb{R}^p$ and $\boldsymbol{\mu}(x^*) \in \mathbb{R}^q_+$ such that

$$\begin{cases} \nabla J(x^*) + \mathrm{D}\boldsymbol{g}(x^*)^{\mathcal{T}}\boldsymbol{\lambda}(x^*) + \mathrm{D}\boldsymbol{h}(x^*)^{\mathcal{T}}\boldsymbol{\mu}(x^*) = 0, \\ \boldsymbol{g}(x^*) = 0, \;\; \boldsymbol{h}(x^*) \leq 0, \\ \forall i = 1, ..., q, \;\; \mu_i h_i(x^*) = 0; \end{cases} \tag{3.6}$$

see again [17, 48].

3.2. Definition of the range step direction

Definition 3 (range space step)**.** For the optimization problem (1.1), the range step $\boldsymbol{\xi}_C(x)$ is defined at any $x \in V$ satisfying the LICQ condition (3.3) by

$$\boldsymbol{\xi}_C(x) := \mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widetilde{I}(x)}(\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}\mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widetilde{I}(x)})^{-1}\boldsymbol{C}_{\widetilde{I}(x)}(x), \tag{3.7}$$

where $\widetilde{I}(x)$ is the set of all saturated or violated constraints, defined by (1.6).

The purpose of the range space step $\boldsymbol{\xi}_C(x)$ is to decrease the violation of the constraints as we shall see in Proposition 5 below. The exact counterpart of Lemma 2 holds in this context, in particular:

(1) $\boldsymbol{\xi}_C(x)$ is orthogonal to $\mathrm{Ker}(\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)})$.

(2) $-\boldsymbol{\xi}_C(x)$ is a Gauss-Newton direction for the violation of the constraints:

$$\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}(-\boldsymbol{\xi}_C(x)) = -\boldsymbol{C}_{\widetilde{I}(x)}(x).$$

3.3. Definition and properties of the null space step

The definition of the null space direction $\boldsymbol{\xi}_J(x)$ is slightly more involved in the present context, as it is not obtained by simply replacing $\mathrm{D}\boldsymbol{g}(x)$ by $\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}$ in (2.7). It requires the introduction of a subset $\widehat{I}(x) \subset \widetilde{I}(x)$ of saturated or violated inequality constraints, as we now discuss.

Like in the equality-constrained case discussed in Section 2, the rationale behind the definition of the null space step $\boldsymbol{\xi}_J(x)$ is that it is sought, up to a change of sign, as a best normalized descent direction diminishing violated or saturated inequality constraints: $-\boldsymbol{\xi}_J(x)$ is set positively proportional to the solution $\boldsymbol{\xi}^*(x)$ of the following minimization problem (compare with Lemma 1):

$$\min_{\boldsymbol{\xi} \in V} \quad \mathrm{D}J(x)\boldsymbol{\xi}$$
$$\text{s.t.} \quad \begin{cases} \mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi} = 0 \\ \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)\boldsymbol{\xi} \leq 0 \\ ||\boldsymbol{\xi}||_V \leq 1. \end{cases} \tag{3.8}$$

The problem (3.8) could be solved directly with standard quadratic programming algorithms. However, it is convenient to characterize explicitly the minimizer $\boldsymbol{\xi}^*(x)$ of (3.8) by examining the dual problem. This will allow us to obtain in Definition 4 an explicit formula for the null space direction $\boldsymbol{\xi}_J(x)$, under the form (1.4).

We now introduce the dual optimization problem to (3.8), which is analogous to that (2.11) in the previous section.

Proposition 2. *Let $x \in V$ satisfy the LICQ condition (3.3). There exists a unique couple of multipliers $\boldsymbol{\lambda}^*(x) \in \mathbb{R}^p$ and $\boldsymbol{\mu}^*(x) \in \mathbb{R}^{\widetilde{q}(x)}_+$ solving the following quadratic optimization problem which is the dual of (3.8):*

$$(\boldsymbol{\lambda}^*(x), \boldsymbol{\mu}^*(x)) := \arg \min_{\substack{\boldsymbol{\lambda} \in \mathbb{R}^p \\ \boldsymbol{\mu} \in \mathbb{R}^{\widetilde{q}(x)}, \, \boldsymbol{\mu} \geq 0}} ||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda} + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}}\boldsymbol{\mu}||_V. \tag{3.9}$$

Proof. At first, (3.8) is equivalent to the following min-max formulation:

$$\min_{\substack{\boldsymbol{\xi} \in V \\ ||\boldsymbol{\xi}||_V \leq 1}} \max_{\substack{\boldsymbol{\lambda} \in \mathbb{R}^p \\ \boldsymbol{\mu} \in \mathbb{R}^{\widetilde{q}(x)}_+}} \left(\mathrm{D}J(x)\boldsymbol{\xi} + \boldsymbol{\lambda}^T \mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi} + \boldsymbol{\mu}^T \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)\boldsymbol{\xi} \right).$$

Exchanging formally the min and the max and solving explicitly the inner minimization problem with respect to $\boldsymbol{\xi}$ (see the proof of Lemma 1) yields that (3.9) is the dual problem of (3.8) up to a change of signs (the duality gap between (3.9) and (3.8) will be shown to vanish in Proposition 3). The program (3.9) brings into play the closed convex set $\mathbb{R}^p \times \mathbb{R}^{\widetilde{q}(x)}_+$ and the least-squares functional

$$(\boldsymbol{\lambda}, \boldsymbol{\mu}) \mapsto \left\| \nabla J(x) + \mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}(x)^{\mathcal{T}} \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\mu} \end{bmatrix} \right\|_V .$$

The latter is strictly convex over $\mathbb{R}^p \times \mathbb{R}^{\widetilde{q}(x)}_+$ by virtue of (3.3). Hence, (3.9) has a unique solution. □

The optimization problem (3.9) belongs to the class of non negative least-squares problems; it can be solved efficiently with a number of dedicated solvers, such as `cvxopt` [12] or `IPOPT` [61]. One nice feature of (3.9) lies in that its dimension is the number $p + \widetilde{q}(x)$ of saturated or violated constraints (and not the total number $p + q$ of constraints), which can be small for many practical cases, as e.g. in our shape optimization applications of Section 5. It is also possible to exploit the *sparsity* of the constraints if $p + \widetilde{q}(x)$ is large, see Remark 5 below.

The next proposition relates the optimal values and the solutions $\boldsymbol{\xi}^*(x)$ and $(\boldsymbol{\lambda}^*(x), \boldsymbol{\mu}^*(x))$ of the primal and dual problems (3.8) and (3.9). Roughly speaking, it claims that the optimal feasible descent direction $\boldsymbol{\xi}^*(x)$ of (3.11) is the projection of the gradient $\nabla J(x)$ onto the cone of feasible directions. The proof follows classical arguments of duality theory in linear programming and it is detailed for the convenience of the reader.

Proposition 3. *Let $x \in V$ satisfy the LICQ condition (3.3) and denote by*

$$m^*(x) := ||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}} \boldsymbol{\lambda}^*(x) + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}} \boldsymbol{\mu}^*(x)||_V$$

the value of the dual problem (3.9), where $(\boldsymbol{\lambda}^*(x), \boldsymbol{\mu}^*(x))$ is the unique solution to the latter. Then the value of the primal problem (3.8) is $p^*(x) = -m^*(x)$ and the following alternative holds:

(1) $m^*(x) = 0$: the first line of the KKT conditions (3.6) for the minimization problem (1.1) holds with (necessarily unique) Lagrange multipliers $(\boldsymbol{\lambda}^*(x), \boldsymbol{\mu}^*(x)) \in \mathbb{R}^p \times \mathbb{R}^{\widetilde{q}(x)}_+$:

$$\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}} \boldsymbol{\lambda}^*(x) + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}} \boldsymbol{\mu}^*(x) = 0. \tag{3.10}$$

One particular minimizer of (3.8) is $\boldsymbol{\xi}^*(x) = 0$.

(2) $m^*(x) > 0$: (3.10) does not hold and (3.8) has a unique minimizer $\boldsymbol{\xi}^*(x)$ given by

$$\boldsymbol{\xi}^*(x) = -\frac{\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}} \boldsymbol{\lambda}^*(x) + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}} \boldsymbol{\mu}^*(x)}{||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}} \boldsymbol{\lambda}^*(x) + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}} \boldsymbol{\mu}^*(x)||_V}. \tag{3.11}$$

Proof. Let $\boldsymbol{\xi} \in V$ be a feasible direction for the problem (3.8), i.e. $\mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi} = 0$, $\mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)\boldsymbol{\xi} \leq 0$ and $||\boldsymbol{\xi}||_V \leq 1$. Then for any $(\boldsymbol{\lambda}, \boldsymbol{\mu}) \in \mathbb{R}^p \times \mathbb{R}^{\widetilde{q}(x)}_+$, it holds

$$\begin{aligned} \mathrm{D}J(x)\boldsymbol{\xi} \geq & \quad \mathrm{D}J(x)\boldsymbol{\xi} + \boldsymbol{\lambda}^T \mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi} + \boldsymbol{\mu}^T \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)\boldsymbol{\xi} \\ = & \quad \langle \nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}} \boldsymbol{\lambda} + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}} \boldsymbol{\mu}, \boldsymbol{\xi} \rangle_V \\ \geq & \quad -||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}} \boldsymbol{\lambda} + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}} \boldsymbol{\mu}||_V \end{aligned} \tag{3.12}$$

Since (3.12) holds for any feasible direction $\boldsymbol{\xi}$ for (3.8), and for any $(\boldsymbol{\lambda}, \boldsymbol{\mu}) \in \mathbb{R}^p \times \mathbb{R}^{\widetilde{q}(x)}_+$, it follows:

$$\min_{\substack{\boldsymbol{\xi} \in V \\ \boldsymbol{\xi} \text{ feasible for (3.8)}}} \mathrm{D}J(x)\boldsymbol{\xi} \geq - \min_{\substack{\boldsymbol{\lambda} \in \mathbb{R}^p \\ \boldsymbol{\mu} \in \mathbb{R}^{\widetilde{q}(x)}, \boldsymbol{\mu} \geq 0}} ||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}} \boldsymbol{\lambda} + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}} \boldsymbol{\mu}||_V. \tag{3.13}$$

Therefore, we have proven that $p^*(x) \geq -m^*(x)$. We now examine the alternative $m^*(x) = 0$ or $m^*(x) > 0$:

(1) If $m^*(x) = 0$, then (3.13) implies $p^*(x) \geq 0$. Therefore, the value of (3.8) is $p^*(x) = -m^*(x) = 0$, attained in particular at $\boldsymbol{\xi}^* = 0$. Furthermore, the KKT condition (3.10) is satisfied by definition of $m^*(x) = 0$.

(2) Assume now $m^*(x) > 0$. The KKT condition for the convex problem (3.8) states that for any local optimum $\boldsymbol{\xi}'$, there exists $(\boldsymbol{\lambda}', \boldsymbol{\mu}') \in \mathbb{R}^p \times \mathbb{R}_+^{\widetilde{q}(x)}$ and $\alpha \geq 0$ such that,

$$\forall \boldsymbol{\xi} \in V,\ (\mathrm{D}J(x) + \boldsymbol{\lambda}'^T \mathrm{D}\boldsymbol{g}(x) + \boldsymbol{\mu}'^T \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x))\boldsymbol{\xi} = -\alpha \langle \boldsymbol{\xi}', \boldsymbol{\xi} \rangle_V. \tag{3.14}$$

Using Riesz identifications of the gradient and the differentials, we obtain

$$\alpha \boldsymbol{\xi}' = -(\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^\mathcal{T} \boldsymbol{\lambda}' + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^\mathcal{T} \boldsymbol{\mu}'),$$

and since $m^*(x) > 0$, it is necessary that $\alpha \neq 0$. The complementarity condition $\alpha(\langle \boldsymbol{\xi}', \boldsymbol{\xi}' \rangle_V - 1) = 0$ yields then $||\boldsymbol{\xi}'||_V = 1$, which readily implies:

$$\boldsymbol{\xi}' = -\frac{\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^\mathcal{T} \boldsymbol{\lambda}' + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^\mathcal{T} \boldsymbol{\mu}'}{||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^\mathcal{T} \boldsymbol{\lambda}' + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^\mathcal{T} \boldsymbol{\mu}'||_V}.$$

Then the complementarity condition for (3.8) implies $\boldsymbol{\mu}'^T \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)\boldsymbol{\xi}' = 0$. Therefore it holds that

$$\begin{aligned} \mathrm{D}J(x)\boldsymbol{\xi}' &= \mathrm{D}J(x)\boldsymbol{\xi}' + \boldsymbol{\lambda}'^T \mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi}' + \boldsymbol{\mu}'^T \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)\boldsymbol{\xi}' \\ &= \langle \nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^\mathcal{T} \boldsymbol{\lambda}' + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^\mathcal{T} \boldsymbol{\mu}', \boldsymbol{\xi}' \rangle_V \\ &= -||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^\mathcal{T} \boldsymbol{\lambda}' + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^\mathcal{T} \boldsymbol{\mu}'||_V. \end{aligned} \tag{3.15}$$

The previous equation together with the inequality (3.12) with $\boldsymbol{\xi} = \boldsymbol{\xi}'$ then implies that $(\boldsymbol{\lambda}', \boldsymbol{\mu}')$ achieves the minimum of (3.9). By uniqueness (see Proposition 2), this implies $\boldsymbol{\lambda}' = \boldsymbol{\lambda}^*(x)$ and $\boldsymbol{\mu}' = \boldsymbol{\mu}^*(x)$, and so $\boldsymbol{\xi}' = \boldsymbol{\xi}^*(x)$, as defined by (3.11). Furthermore, $p^*(x) = \mathrm{D}J(x)\boldsymbol{\xi}^*(x) = \mathrm{D}J(x)\boldsymbol{\xi}' = -m^*(x)$.

□

Finally, the next proposition characterizes explicitly the optimal descent direction $\boldsymbol{\xi}^*(x)$ from the signs of the multiplier $\boldsymbol{\mu}^*(x)$, and highlights in which sense the problem (3.8) is combinatorial. Recalling the definitions (3.4) and (3.5) of the projection operator Π_{C_I} and the multipliers $(\boldsymbol{\lambda}_I(x), \boldsymbol{\mu}_I(x))$, we have:

Proposition 4. *In the situation of point (2) in Proposition 3, let $\boldsymbol{\xi}^*(x)$ and $(\boldsymbol{\lambda}^*(x), \boldsymbol{\mu}^*(x))$ be the unique minimizers of the primal and dual problems (3.8) and (3.9). Define the subset $\widehat{I}(x) \subset \widetilde{I}(x)$ by*

$$\widehat{I}(x) := \{i \in \widetilde{I}(x) \,|\, \mu_i^*(x) > 0\}. \tag{3.16}$$

(1) $(\boldsymbol{\lambda}^*(x), \boldsymbol{\mu}^*(x))$ and $\boldsymbol{\xi}^*(x)$ are explicitly given in terms of $\widehat{I}(x)$ by:

$$\begin{bmatrix} \boldsymbol{\lambda}^*(x) \\ \widehat{\boldsymbol{\mu}}^*(x) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\lambda}_{\widehat{I}(x)}(x) \\ \boldsymbol{\mu}_{\widehat{I}(x)}(x) \end{bmatrix} = -(\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)} \mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^\mathcal{T})^{-1} \mathrm{D}\boldsymbol{C}_{\widehat{I}(x)} \nabla J(x), \tag{3.17}$$

$$\boldsymbol{\xi}^*(x) = -\frac{\Pi_{\boldsymbol{C}_{\widehat{I}(x)}}(\nabla J(x))}{||\Pi_{\boldsymbol{C}_{\widehat{I}(x)}}(\nabla J(x))||_V}, \tag{3.18}$$

where $\widehat{\boldsymbol{\mu}}^*(x) := (\mu_i^*(x))_{i \in \widehat{I}(x)}$ is the vector collecting all positive components of $\boldsymbol{\mu}^*(x)$.

(2) $\widehat{I}(x)$ is equivalently the unique solution to each of the following discrete optimization problems:

$$\begin{aligned} \widehat{I}(x) = \arg \max_{I \subset \widetilde{I}(x)} \quad & ||\Pi_{\boldsymbol{C}_I}(\nabla J(x))||_V \\ s.t. \quad & \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x) \Pi_{C_I}(\nabla J(x)) \geq 0, \end{aligned} \tag{3.19}$$

$$\begin{aligned} \widehat{I}(x) = \arg \min_{I \subset \widetilde{I}(x)} \quad & ||\Pi_{C_I}(\nabla J(x))||_V \\ s.t. \quad & \boldsymbol{\mu}_I(x) \geq 0. \end{aligned} \tag{3.20}$$

In particular, $\widehat{I}(x)$ is the unique subset $I \subset \widetilde{I}(x)$ satisfying simultaneously both feasibility conditions

$$\mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x) \Pi_{C_I}(\nabla J(x)) \geq 0 \text{ and } \boldsymbol{\mu}_I(x) \geq 0.$$

Proof.

(1) The complementarity condition for the primal and dual problems (3.8) and (3.9) reads

$$\forall i \in \widetilde{I}(x), \quad \mu_i^*(x)\mathrm{D}h_i(x)\boldsymbol{\xi}^*(x) = 0. \tag{3.21}$$

Therefore, $\mathrm{D}h_i(x)\boldsymbol{\xi}^*(x) = 0$ for all indices $i \in \widehat{I}(x)$, which implies that $\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}(x)\boldsymbol{\xi}^*(x) = 0$. Then, after left multiplication of (3.11) by $(\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^{\mathcal{T}})^{-1}\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}$, we obtain (3.17), whence (3.18) follows.

(2) For any subset $I \subset \widetilde{I}(x)$ satisfying $\mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)\Pi_{C_I}(\nabla J(x)) \geq 0$, the direction

$$\boldsymbol{\xi} = -\frac{\Pi_{\boldsymbol{C}_I}(\nabla J(x))}{||\Pi_{\boldsymbol{C}_I}(\nabla J(x))||_V}$$

is feasible for the primal problem (3.8), and we obtain by definition of $\boldsymbol{\xi}^*(x)$ that

$$-||\Pi_{\boldsymbol{C}_{\widehat{I}(x)}}(\nabla J(x))||_V = \mathrm{D}J(x)\boldsymbol{\xi}^*(x) \leq \mathrm{D}J(x)\boldsymbol{\xi} = -||\Pi_{\boldsymbol{C}_I}(\nabla J(x))||_V, \tag{3.22}$$

whence the maximization property (3.19).

For $I \subset \widetilde{I}(x)$ satisfying $\boldsymbol{\mu}_I(x) \geq 0$, we obtain feasible multipliers $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ for the dual problem (3.9) by taking the components of $\boldsymbol{\mu}$ to coincide with those of $\boldsymbol{\mu}_I$ on the indices of I and assigning them the value 0 in the complementary subset $\widetilde{I}(x) \setminus I$. Then the optimality of $(\boldsymbol{\lambda}^*(x), \boldsymbol{\mu}^*(x))$ for (3.9) reads:

$$\begin{aligned} ||\Pi_{\boldsymbol{C}_{\widehat{I}(x)}}(\nabla J(x))||_V &= ||\nabla J + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda}^*(x) + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}}\boldsymbol{\mu}^*(x)||_V \\ &\leq ||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda} + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}^{\mathcal{T}}\boldsymbol{\mu}||_V = ||\Pi_{\boldsymbol{C}_I}(\nabla J(x))||_V, \end{aligned} \tag{3.23}$$

whence the minimization property (3.20).

□

Remark 2. In view of (3.16), the optimal multiplier $\boldsymbol{\mu}^*(x)$ can be interpreted as an indicator specifying which constraints of $\widetilde{I}(x)$ are 'not aligned' with the gradient $\nabla J(x)$ and should be kept in the subset $\widehat{I}(x)$. The best descent direction (in the sense of (3.8)) is obtained by projecting the gradient $\nabla J(x)$ onto the tangent space of the constraint subset $\widehat{I}(x)$ rather than onto the full set of violated or saturated constraints $\widetilde{I}(x)$. Indeed, the descent direction $\boldsymbol{\xi} = -\Pi_{C_{\widetilde{I}(x)}}\nabla J(x)$ that would be obtained by projecting $\nabla J(x)$ on the whole set $\widetilde{I}(x)$ would only keep them constant at first order, i.e. $\mathrm{D}h_i(x)\boldsymbol{\xi} = 0$, (see Remark 6 for more details), whereas considering $\boldsymbol{\xi} = -\Pi_{C_{\widehat{I}(x)}}\nabla J(x)$ instead allows the inequality constraints associated to $i \in \widetilde{I}(x) \setminus \widehat{I}(x)$ to decrease, which is much more efficient.

Remark 3. Note that actually, the use of a dual problem such as (3.9) in order to obtain information about which constraints should remain active is quite classical in active set methods, see e.g. [19, 36, 48].

In principle, the subset $\widehat{I}(x)$ could be found by solving the discrete problems (3.19) or (3.20). However, we expect that in practice, it is more efficient to rely on iterative solvers based on gradient descent strategies for the dual problem (3.9), e.g. a cone programming solver or a non negative least-squares algorithm such as that in [19]. This is what we do in the sequel.

Having introduced the subset $\widehat{I}(x)$ (defined in (3.16)), we are now in position to define the null space direction $\boldsymbol{\xi}_J(x)$ in the present context: $-\boldsymbol{\xi}_J(x)$ is set to be a positive multiple of the optimal descent direction $\boldsymbol{\xi}^*(x)$ supplied by (3.18).

Definition 4. For any point $x \in V$ satisfying the LICQ condition (3.3), the null space direction $\boldsymbol{\xi}_J(x)$ at x for the optimization problem (1.1) is defined by:

$$\boldsymbol{\xi}_J(x) := \Pi_{\boldsymbol{C}_{\widehat{I}(x)}}(\nabla J(x)) = (I - \mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}(x)^{\mathcal{T}}(\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^{\mathcal{T}})^{-1}\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)})\nabla J(x), \tag{3.24}$$

where $\widehat{I}(x)$ is the set defined by (3.16).

Observe that all violated and saturated constraints are taken into account in the Gauss-Newton direction $\boldsymbol{\xi}_C(x)$ defined by (3.7), while only those constraints in $\widehat{I}(x)$, not aligned with the gradient $\nabla J(x)$, occur in the definition of $\boldsymbol{\xi}_J(x)$.

Remark 4. With our notation conventions, the discrete optimization scheme proposed by Barbarosie et. al. [15, 16] reads

$$\begin{cases} x_{n+1} = x_n - \Delta t \nabla J(x_n) - \mathrm{D}\boldsymbol{C}_{I(x_n)}^{\mathcal{T}} \boldsymbol{\nu}_n \\ \quad \boldsymbol{\nu}_n = -\Delta t (\mathrm{D}\boldsymbol{C}_{I(x_n)} \mathrm{D}\boldsymbol{C}_{I(x_n)}^{\mathcal{T}})^{-1} \mathrm{D}\boldsymbol{C}_{I(x_n)} \nabla J(x_n) + (\mathrm{D}\boldsymbol{C}_{I(x_n)} \mathrm{D}\boldsymbol{C}_{I(x_n)}^{\mathcal{T}})^{-1} \boldsymbol{C}_{I(x_n)}, \end{cases} \tag{3.25}$$

where the set $I(x_n)$ is obtained by removing indices from $\widetilde{I}(x_n)$ one by one, starting from the index i_0 associated with the most negative multiplier $\nu_{n,i_0} < 0$, until all of them become non negative. Therefore, the set $I(x_n)$ used in the latter strategy and that $\widehat{I}(x_n)$ featured in ours (given by (3.16)) do not coincide in general; one could think of configurations where the procedure of [16] would fail to find the optimal set $\widehat{I}(x_n)$ (for example if $i_0 \in \widehat{I}(x_n)$) and would project the gradient on a less optimal subset of constraints. We note that no convergence result is given by the authors of [15, 16] about their procedure.

Remark 5. Let us discuss two extreme cases related to the involved computational effort in the numerical implementation of (3.24). Upon discretization, we may assume that $V = \mathbb{R}^k$ is a finite-dimensional space.

(1) If the total number $p + \widetilde{q}$ of saturated or violated constraints is small compared to the dimension k of V, it is best, for numerical efficiency, to assemble the small square matrix $(\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)} \mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}^{\mathcal{T}})$ and to invert it by a direct method.

(2) If $V = \mathbb{R}^k$ is equipped with an inner product encoded by a matrix A, and if $p + \widetilde{q}$ is of the order of k or larger, the computation of the inverse of $(\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)} \mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^{\mathcal{T}})$ can be expensive. However, it is still tractable if both $\mathrm{D}\boldsymbol{C}$ and A are sparse matrices (i.e. matrices with many 0 entries). For instance, this occurs in the case of bound constraints on the optimization variable $x = (x_1, ..., x_k)$, i.e. constraints of the form $\alpha_i \leq x_i \leq \beta_i$, $i = 1, ..., k$.

Recalling from Remark 1 that in this setting, $\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^{\mathcal{T}} = A^{-1} \mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^{T}$, the calculation of $\boldsymbol{\xi}_J(x)$ can be carried out thanks to the following procedure: at first, the vector

$$X := A^{-1} \mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^{T} (\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)} A^{-1} \mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^{T})^{-1} \mathrm{D}\boldsymbol{C}_{\widehat{I}(x)} \nabla J(x)$$

is computed as the solution to the sparse linear system

$$\begin{bmatrix} A & -\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^{T} \\ \mathrm{D}\boldsymbol{C}_{\widehat{I}(x)} & 0 \end{bmatrix} \begin{bmatrix} X \\ Z \end{bmatrix} = \begin{bmatrix} 0 \\ \mathrm{D}\boldsymbol{C}_{\widehat{I}(x)} \nabla J(x) \end{bmatrix},$$

where $Z \in \mathbb{R}^{p + \mathrm{Card}(\widehat{I}(x))}$ is an extra slack variable. Then, the desired null space direction is obtained as $\boldsymbol{\xi}_J(x) = \nabla J(x) - X$. A similar strategy, exploiting the sparsity of A and $\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}$, can be used to compute the range space direction $\boldsymbol{\xi}_C(x)$ of (3.7), or to solve the dual quadratic subproblem (3.9).

Remark 6. As we have already mentioned, the Lagrange multiplier $\boldsymbol{\mu}^*(x)$ given by (3.17) may be understood as an indicator of which inequality constraints are aligned with the gradient of J at x. This insight is especially intuitive in the particular situation where the gradients of the constraint functions are orthogonal to one another, i.e.:

$$\begin{aligned} &\langle \nabla g_i(x), \nabla g_j(x) \rangle_V = 0, \text{ for } i, j = 1, ..., p, \ i \neq j, \\ &\langle \nabla h_i(x), \nabla h_j(x) \rangle_V = 0, \text{ for } i, j = 1, ..., q, \ i \neq j, \\ &\langle \nabla g_i(x), \nabla h_j(x) \rangle_V = 0, \text{ for } i = 1, ..., p, \ j = 1, ..., q. \end{aligned}$$

Indeed, in this case, it easily follows from the Pythagore theorem that for any $(\boldsymbol{\lambda}, \boldsymbol{\mu}) \in \mathbb{R}^p \times \mathbb{R}_+^{\widetilde{q}(x)}$,

$$||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}} \boldsymbol{\lambda} + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}} \boldsymbol{\mu}||_V^2 = \left|\left| \nabla J(x) + \sum_{i=1}^{p} \lambda_i \nabla g_i(x) + \sum_{j \in \widetilde{I}(x)} \mu_j \nabla h_j(x) \right|\right|_V^2$$

$$= ||\nabla J(x)||_V^2 + \sum_{i=1}^{p} \left(\lambda_i^2 ||\nabla g_i(x)||_V^2 + 2\lambda_i \langle \nabla J(x), \nabla g_i(x) \rangle_V \right) + \sum_{j \in \widetilde{I}(x)} \left(\mu_j^2 ||\nabla h_j(x)||_V^2 + 2\mu_j \langle \nabla J(x), \nabla h_j(x) \rangle_V \right).$$

Therefore the minimization problem (3.9) is separable with respect to the components of $(\boldsymbol{\lambda}, \boldsymbol{\mu}) \in \mathbb{R}^p \times \mathbb{R}_+^{\widetilde{q}(x)}$: $(\lambda_i^*(x))_{1 \leq i \leq p}$ and $(\mu_i^*(x))_{i \in \widetilde{I}(x)}$ are the respective solutions to the minimization problems:

$$\forall i \in 1 \dots p, \;\; \lambda_i^*(x) = \arg\min_{t \in \mathbb{R}} \left(t^2 ||\nabla g_i(x)||_V^2 + 2t \langle \nabla J(x), \nabla g_i(x) \rangle_V \right),$$

$$\forall i \in \widetilde{I}(x), \;\; \mu_i^*(x) = \arg\min_{\substack{t \in \mathbb{R} \\ t \geq 0}} \left(t^2 ||\nabla h_i(x)||_V^2 + 2t \langle \nabla J(x), \nabla h_i(x) \rangle_V \right),$$

which yields eventually:

$$\lambda_i^*(x) = -\frac{\langle \nabla J(x), \nabla g_i(x) \rangle_V}{||\nabla g_i(x)||_V^2}, \qquad \mu_i^*(x) = \begin{cases} 0 & \text{if } \langle \nabla J(x), \nabla h_i(x) \rangle_V \geq 0, \\ -\frac{\langle \nabla J(x), \nabla h_i(x) \rangle_V}{||\nabla h_i(x)||_V^2} & \text{otherwise.} \end{cases}$$

Hence, $\mu_i^*(x)$ is positive if and only if following the descent direction $-\nabla J(x)$ leads to an increase (i.e. violation) of the i^{th} inequality constraint.

In the general case where all the constraint gradients are not mutually orthogonal, the interpretation of $\boldsymbol{\mu}^*(x)$ is similar, up to the additional complication that (3.9) accounts for the possible alignments between different constraint gradients. In the following, with a slight abuse of language, we shall nevertheless refer to the indices $i \in \widetilde{I}(x) \setminus \widehat{I}(x)$ as those associated to constraints which are 'aligned' with $\nabla J(x)$, in the sense that $-\mathrm{D}h_i(x)\boldsymbol{\xi}_J(x) \leq 0$, i.e. the violation $h_i(x)$ decreases along $-\boldsymbol{\xi}_J(x)$ (or, at least, stays constant).

3.4. Decrease properties of the trajectories of the null space ODE

The final result of this section is the counterpart of Proposition 1 in the case of the equality and inequality constrained optimization problem (1.1); see Appendix A for the proof and further remarks.

Proposition 5. *Assume that the trajectories $x(t)$ of the flow*

$$\begin{cases} \dot{x}(t) = -\alpha_J \boldsymbol{\xi}_J(x(t)) - \alpha_C \boldsymbol{\xi}_C(x(t)) \\ x(0) = x_0, \end{cases} \tag{3.26}$$

with $\boldsymbol{\xi}_J$ and $\boldsymbol{\xi}_C$ given by (3.7) and (3.24) exist on some interval $[0, T]$ for $T > 0$ and are such that:

(a) the set $\widetilde{I}(x(t))$ defined in (1.6) is constant over $[0, T]$:

$$\forall t \in [0, T], \; \widetilde{I}(x(t)) = \widetilde{I}(x_0);$$

(b) the constraints remain qualified along the flow $x(t)$, in the sense of the LICQ condition (3.3).

Then the following properties hold true:

(1) The violation of the constraints decreases exponentially:

$$\forall t \in [0, T], \; \boldsymbol{g}(x(t)) = e^{-\alpha_C t} \boldsymbol{g}(x_0) \text{ and } \boldsymbol{h}_{\widetilde{I}(x_0)}(x(t)) \leq e^{-\alpha_C t} \boldsymbol{h}_{\widetilde{I}(x_0)}(x_0). \tag{3.27}$$

(2) $J(x(t))$ decreases 'as soon as the violation (3.27) of the constraints is sufficiently small' in the following sense. Assume that $\mathrm{rank}(\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x_0)}(x))$ is maximal for all x in $K = \{x \in V \,|\, ||\boldsymbol{C}_{\widetilde{I}(x_0)}(x)||_\infty \leq ||\boldsymbol{C}_{\widetilde{I}(x_0)}(x_0)||_\infty\}$ and that:

$$\sup_{x \in K} ||\nabla J(x)||_V |\sigma_p^{-1}(x)| < +\infty. \tag{3.28}$$

where $\sigma_p(x)$ is the smallest singular value of $\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}(x)$. Then there exists a constant $C > 0$ such that

$$\forall t \in [0, T], \; ||\Pi_{C_{\widetilde{I}(x(t))}}(\nabla J(x(t)))||_V^2 > Ce^{-\alpha_C t} \Rightarrow \frac{\mathrm{d}}{\mathrm{d}t} J(x(t)) < 0. \tag{3.29}$$

(3) Any stationary point x^* of the flow (3.26) satisfies the KKT optimality conditions (3.6) which equivalently rewrite:

$$\begin{cases} \nabla J(x^*) + \mathrm{D}\boldsymbol{g}(x^*)^\mathcal{T} \boldsymbol{\lambda}^*(x^*) + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x^*)}(x^*)^\mathcal{T} \boldsymbol{\mu}^*(x^*) = 0, \\ \boldsymbol{g}(x^*) = 0 \text{ and } \boldsymbol{h}_{\widetilde{I}(x^*)}(x^*) = 0 \Leftrightarrow \boldsymbol{C}_{\widetilde{I}(x^*)}(x^*) = 0, \end{cases} \tag{3.30}$$

where $(\boldsymbol{\lambda}^*(x^*), \boldsymbol{\mu}^*(x^*)) \in \mathbb{R}^p \times \mathbb{R}_+^{\widetilde{q}(x^*)}$ are defined in (3.9) or (3.17).

3.5. **Comparison with the method of slack variables for inequality constraints**

One popular method in the literature to address inequality constraints in problems such as (1.1), which is significantly different from ours, is to introduce slack variables so as to turn them into equality constraints of an augmented problem. In this section, we briefly review this idea which was investigated by [51] in the context of dynamical system approaches to constrained optimization, and we compare it with our method based on the dual problem (3.9).

The method of slack variables consists in replacing (1.1) with the following equivalent equality-constrained problem, involving as many extra variables $(z_1, \dots, z_q) \in \mathbb{R}^q$ as there are inequality constraints in (1.1):

$$\begin{aligned} &\min_{\substack{x \in V \\ z \in \mathbb{R}^q}} \quad J(x) \\ &\text{s.t.} \quad \boldsymbol{C}(x,z) = 0, \end{aligned} \tag{3.31}$$

where the augmented vector of constraints $\boldsymbol{C}(x,z)$ reads:

$$\boldsymbol{C}(x,z) := \begin{bmatrix} \boldsymbol{g}(x) \\ h_1(x) + \frac{1}{2} z_1^2 \\ \vdots \\ h_q(x) + \frac{1}{2} z_q^2 \end{bmatrix} \in \mathbb{R}^{p+q}.$$

Problem (3.31) is an equality constrained optimization problem of the form (2.1), set over the Hilbert space $\widetilde{V} = V \times \mathbb{R}^q$ with inner product $\langle (x,z), (x',z') \rangle_{\widetilde{V}} = \langle x, x' \rangle_V + z^T z'$. It can be solved thanks to the proposed algorithm in Section 2; the associated gradient flow for (3.31) reads:

$$\begin{cases} \begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} = -\alpha_J (\mathrm{I} - \mathrm{D}\boldsymbol{C}^{\mathcal{T}} (\mathrm{D}\boldsymbol{C} \mathrm{D}\boldsymbol{C}^{\mathcal{T}})^{-1} \mathrm{D}\boldsymbol{C}) \begin{bmatrix} \nabla J(x(t)) \\ 0 \end{bmatrix} - \alpha_C \mathrm{D}\boldsymbol{C}^{\mathcal{T}} (\mathrm{D}\boldsymbol{C} \mathrm{D}\boldsymbol{C}^{\mathcal{T}})^{-1} \boldsymbol{C}(x(t), z(t)), \\ x(0) = x_0 \text{ and } z(0) = z_0, \end{cases} \tag{3.32}$$

where $x_0 \in V$ is the considered initial point in the resolution of (1.1), and the variable z is initialized with a value $z_0 \in \mathbb{R}^q$ in such a way that the inequality constraints of (1.1) which are inactive for x_0 (i.e. $h_i(x_0) < 0$) are associated with satisfied equality constraints $C_{p+i}(x_0, z_0) = 0$ in (3.31), namely $z_{0,i} = \sqrt{2|h_i(x_0)|}$. In the finite-dimensional setting $V = \mathbb{R}^k$ and when J, $\boldsymbol{g}$ and $\boldsymbol{h}$ are $\mathcal{C}^2$ functions, Schropp and Singer proved in [51] that:

(i) Stationary points of the extended flow (3.32) are exactly critical points of (1.1), that is points x^* satisfying (3.6) but whose multiplier $\boldsymbol{\mu}(x^*) \in \mathbb{R}^q$ may have negative entries.

(ii) Among all possible critical points of (1.1), only KKT points (fulfilling all three conditions (3.6) with $\boldsymbol{\mu}(x^*) \in \mathbb{R}^q_+$) are asymptotically stable equilibria. As a consequence, the solution vector $x(t)$ to (3.32) converges in practice to a KKT point for problem (1.1).

The main differences between this slack variable approach and our new approach in Section 3.2 for dealing with equality and inequality constrained problems can be summarized as follows.

(1) Any point x^{crit} satisfying the constraints $(\boldsymbol{C}_{\widetilde{I}(x^{\mathrm{crit}})}(x^{\mathrm{crit}}) = 0)$ and $\Pi_{\boldsymbol{C}_{\widetilde{I}(x^{\mathrm{crit}})}}(\nabla J(x^{\mathrm{crit}})) = 0$ is a stationary point of the slack variable dynamical system (3.32), although it might violate the full KKT condition (because (3.5) may yield negative values of the multiplier $\boldsymbol{\mu}_{\widetilde{I}(x^{\mathrm{crit}})}(x^{\mathrm{crit}})$). BY contrast, only true, feasible KKT points are stationary points of our flow (3.26), see Proposition 5.

(2) The computation of the directions $\boldsymbol{\xi}_J(x)$ and $\boldsymbol{\xi}_C(x)$ involved in our flow (3.26) requires to invert a matrix of size at most $(p + \widetilde{q}(x))$-by-$(p + \widetilde{q}(x))$ where $\widetilde{q}(x)$ is the number of active or violated constraints at x. The process of equalizing inequality constraints as in [51, 54] rather requires to invert the full $(p+q)$-by-$(p+q)$ matrix $\mathrm{D}\boldsymbol{C}(x,z)\mathrm{D}\boldsymbol{C}(x,z)^{\mathcal{T}}$. Our method is therefore more efficient if $\widetilde{q}(x) \ll q$, that is, if a lot of inequality constraints are inactive.

(3) At feasible points, our ODE (3.26) follows the best locally admissible descent direction (with respect to the norm of V), which is not the case for the extended ODE (3.32). Therefore, from a common feasible point x, the ODE (3.26) always decreases the objective function with a 'steeper slope'.

All in all, our numerical observations (not reported in the present article, but extensively discussed in [31]) tend to illustrate that both flows (3.26) and (3.32) may have equivalent performances for solving the non linear optimization problem (1.1), this performance being measured in term of the total length covered by the optimization path to reach the optimum. However, the two ODEs (3.26) and (3.32) yield optimization paths of essentially different natures. Our null space flow (3.26) ignores inactive constraints and those aligned with the gradient of the objective function. As a result, it produces non smooth paths that are more likely to reach quickly the saturation of the constraint. The extended flow (3.32) yields smoother trajectories that more likely stay away from the constraints, at the cost of inverting at every step the full matrix $\mathrm{D}\boldsymbol{C}(x,z)\mathrm{D}\boldsymbol{C}(x,z)^{\mathcal{T}}$ whose size equals the total number of constraints (active and inactive).

4. Numerical discretization and time-stepping schemes for the null space ODE

This short section describes practical implementation details for the discretization of the ODE (1.2) by an explicit Euler scheme. Two important issues are discussed in Sections 4.1 and 4.2, respectively. First, we propose small adaptations in the computation of $\boldsymbol{\xi}_J(x)$ and $\boldsymbol{\xi}_C(x)$ in order to account for the discontinuous changes of the right-hand side $-(\alpha_J\boldsymbol{\xi}_J + \alpha_C\boldsymbol{\xi}_C)$. Then, a merit function is proposed for adapting the time step Δt. The complete implementation of the algorithm is summarized in Section 4.3 below.

4.1. Accounting for discontinuities near the inequality constraint barriers

A potential issue when implementing the above Algorithm 1 comes from the fact that the vector fields $\boldsymbol{\xi}_J$ and $\boldsymbol{\xi}_C$ given by (3.7) and (3.24) suffer from the same discontinuities as the discrete index mapping $x \mapsto \widetilde{I}(x)$. As a result, abrupt oscillations of the discrete optimization path (x_n) may occur near the boundary of the feasible set: if $h_i(x_n) = 0$ and $i \in \widehat{I}(x_n)$ for some index $i \in \{1,\dots,q\}$, then in the definition (3.24) of $\boldsymbol{\xi}_J(x_n)$, the gradient $\nabla J(x_n)$ is projected tangentially to the constraint h_i, but it is not projected after any slight deviation (e.g. due to the discretization) making this constraint inactive ($h_i(x_{n+1}) < 0$). This kind of issue is very classical in the discretization of ODEs with discontinuous vector fields and can be tackled by various methods, see e.g. [25] for a review.

In this section, we suggest a simple alternative: constraints are felt from a short distance by replacing the set $\widetilde{I}(x_n)$ in (1.6) with the set $\widetilde{I}_\epsilon(x_n)$ of inequality constraints violated "up to ϵ_i":

$$\widetilde{I}_\epsilon(x_n) = \{i \in \{1,\dots,q\} \,|\, h_i(x_n) \geq -\epsilon_i\}. \tag{4.1}$$

The tolerances $\epsilon_i > 0$ can be estimated in an automatic fashion, independently of an arbitrary rescaling of the constraints, thanks to an posteriori bound which we now detail. Let $\mathtt{h}$ be a user-defined parameter, representing the distance from a point x to the boundary of the feasible set at which we desire that the constraints should be 'felt'. This characteristic length $\mathtt{h}$ should be defined in accordance with the typical distance $||x_{n+1} - x_n||_V$ between two successive iterates of the algorithm. For our shape optimization applications in Section 5, $\mathtt{h}$ is typically of the order of the size of the mesh discretizing the shape, see Section 5.3.2 below.

Assume now that the current point x_n satisfies the constraint h_i up to the uncertainty $\mathtt{h}$ on its location: by this we mean that there exists some unknown point x_n^* such that $||x_n^* - x_n|| \leq \mathtt{h}$, $h_i(x_n) > 0$ and $h_i(x_n^*) = 0$. Then the error $\mathtt{h}$ for the location of x_n propagates to the constraint values $h_i(x_n)$ according to the following inequality:

$$h_i(x_n) = |h_i(x_n) - h_i(x_n^*)| \simeq |\mathrm{D}h_i(x_n)(x_n^* - x_n)| \leq ||\nabla h_i(x_n)||_V \mathtt{h}. \tag{4.2}$$

It is therefore natural to set

$$\epsilon_i := ||\nabla h_i(x_n)||_V \mathtt{h} \tag{4.3}$$

for the value of ϵ_i in (4.1).

Note that more generally, the a posteriori bound (4.2) allows to assert whether a constraint $C_i(x_n)$ can be considered as satisfied or not with respect to the numerical discretization.

The dual problem (3.9) is then solved with $\widetilde{I}_\epsilon(x_n)$ instead of $\widetilde{I}(x_n)$ in order to obtain a new subset of indices $\widehat{I}_\epsilon(x_n)$ which indicates which constraints are likely to be not aligned with the gradient $\nabla J(x_n)$ when

approaching the barrier $\Big\{h_i = 0 \,|\, i \in \{1,\dots,q\}\backslash\widetilde{I}(x_n)\Big\}$. The null space and range space directions $\boldsymbol{\xi}_J(x_n)$ and $\boldsymbol{\xi}_C(x_n)$ in Definitions 3 and 4 are finally replaced with $\boldsymbol{\xi}_{J,\epsilon}(x_n)$ and $\boldsymbol{\xi}_{C,\epsilon}(x_n)$ computed as follows:

$$\boldsymbol{\xi}_{J,\epsilon}(x_n) := (\mathrm{I} - \mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widehat{I}_\epsilon(x_n)}(\mathrm{D}\boldsymbol{C}_{\widehat{I}_\epsilon(x_n)}\mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widehat{I}_\epsilon(x_n)})^{-1}\mathrm{D}\boldsymbol{C}_{\widehat{I}_\epsilon(x_n)})\nabla J(x_n), \tag{4.4}$$

$$\boldsymbol{\xi}_{C,\epsilon}(x_n) := \mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{I^*_\epsilon(x_n)}(\mathrm{D}\boldsymbol{C}_{I^*_\epsilon(x_n)}\mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{I^*_\epsilon(x_n)})^{-1}\boldsymbol{C}_{I^*_\epsilon(x_n)}(x_n), \tag{4.5}$$

where $I^*_\epsilon(x_n) = \widetilde{I}(x_n) \cup \widehat{I}_\epsilon(x_n)$ is the set of constraints that are either violated, saturated or not aligned with the gradient $\nabla J(x_n)$ at $\boldsymbol{h} = -(\epsilon_1, ..., \epsilon_q)^T$. The use of $\widehat{I}_\epsilon(x_n)$ in the definition of $\boldsymbol{\xi}_{J,\epsilon}(x_n)$ ensures that the gradient $\nabla J(x_n)$ is being projected tangentially to the constraint in a small neighborhood of the boundary of the feasible set. The use of $I^*_\epsilon(x_n)$ in (4.5) (instead of $\widetilde{I}(x_n)$) helps ensuring that the optimization trajectory eventually stabilizes on the constraint barrier $\{h_i = 0 \,|\, i \in \widehat{I}_\epsilon(x_n)\backslash\widetilde{I}(x_n)\}$ (and not on the ϵ_i-level set). As a result, no abrupt discontinuity occurs for $\boldsymbol{\xi}_{J,\epsilon}$ and $\boldsymbol{\xi}_{C,\epsilon}$ when crossing the boundary of the feasible domain as long as the optimization path stays in this neighborhood. Including inequality constraints indexed by $i \in \widehat{I}_\epsilon(x_n)$ in the Gauss-Newton direction $\boldsymbol{\xi}_{C,\epsilon}(x_n)$ even if they are satisfied (i.e. if $-\epsilon_i \leq h_i(x_n) \leq 0$) further allows to stabilize the values of these constraints closer to zero.

4.2. **Time step adaptation based on a merit function.**

The ODE (1.2) is discretized by an explicit scheme of the form:

$$x_{n+1} = x_n - \Delta t_n\big(\alpha_J\boldsymbol{\xi}_J(x_n) + \alpha_C\boldsymbol{\xi}_C(x_n)\big), \tag{4.6}$$

with a variable time step $\Delta t_n > 0$. The practical implementation of such a strategy is often guided by a merit function, i.e. an indicator allowing to detect when a step has been too large, a situation where a choice has to be made regarding whether the step should be reduced or accepted. For our null space algorithm, a merit function which resembles very much that of the Augmented Lagrangian Method is readily available, however with a specific choice of multipliers:

Lemma 3. *For a given $x_n \in V$, let $\mathtt{merit}_{x_n} : V \to \mathbb{R}$ be the function defined by*

$$\mathtt{merit}_{x_n}(x) := \alpha_J\left(J(x) + \boldsymbol{\Lambda}(x_n)^T\boldsymbol{C}_{\widetilde{I}(x_n)}(x)\right) + \frac{\alpha_C}{2}\boldsymbol{C}_{\widetilde{I}(x_n)}(x)^T\boldsymbol{S}(x_n)\boldsymbol{C}_{\widetilde{I}(x_n)}(x) \tag{4.7}$$

where $\boldsymbol{\Lambda}(x_n) = \begin{bmatrix}\boldsymbol{\lambda}^*(x_n)^T & \boldsymbol{\mu}^*(x_n)^T\end{bmatrix}^T$ is the vector of multipliers defined as the solution to the dual problem (3.9) (see (3.17)) and $\boldsymbol{S}(x_n) = (\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x_n)}(x_n)\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x_n)}(x_n)^{\mathcal{T}})^{-1}$ is symmetric positive definite. Then (4.6) is a gradient step for the decrease of the function $\mathtt{merit}_{x_n}$, namely:

$$\nabla\mathtt{merit}_{x_n}(x_n) = \alpha_J\boldsymbol{\xi}_J(x_n) + \alpha_C\boldsymbol{\xi}_C(x_n).$$

Proof. It is a straightforward computation of the gradient of (4.7). □

One possible implementation of an optimization strategy of the form (4.6) based on this merit function is summarized in Algorithm 1, which requires the introduction of a few extra parameters:

- `time_step`: choose a fixed time step $\Delta t > 0$.
- `maxtrials`: the optimization time step is decreased up to `maxtrials` times until the value of the merit function has decreased. If the merit function has not decreased after all `maxtrials` steps, the smallest step is accepted.
- `tolLag`: a small threshold for the values of the Lagrange multipliers μ^*_i under which these are considered to be 0 (in our examples, we took `tolLag=1e-8`). This value should be set in accordance with the machine precision and that of the quadratic programming solver for the dual problem (3.9).

Let us emphasize that these parameters have a quite intuitive and physical meaning, so that the task of assigning their values does not involve fine tuning in practice.

Importantly, the rescaling induced by the inverse of the correlation matrix $(\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x_n)}\mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widetilde{I}(x_n)})^{-1}$ normalizes all the constraints; in particular, the whole Algorithm 1 as outlined below is invariant under multiplication of the constraints by arbitrary positive constants (up to the machine precision for the step 3) and no preliminary rescaling of the constraints is required from the user.

4.3. **Overall algorithm pseudo code**

The resulting algorithmic implementation of our null space gradient flow taking into account both adaptations of Section 4.1 and (4.2) is summarized in Algorithm 1 below.

Algorithm 1 Discretization of the null space gradient flow (3.26).

for $n = 1 \dots \texttt{maxiter}$ **do**

1. Compute the gradients $\nabla J(x_n)$, $\nabla g_i(x_n)$ and $\nabla h_j(x_n)$ for $1 \leq i \leq p$, $1 \leq j \leq q$ by solving the identification problems (2.3) and (2.4).

2. For all inequality constraints $1 \leq i \leq q$, compute the tolerance

$$\epsilon_i := ||\nabla h_i(x_n)||_V \, \texttt{h}.$$

3. Determine the set $\widetilde{I}(x_n)$ of active or violated constraints and the set $\widetilde{I}_\epsilon(x_n)$ of constraints violated "up to ϵ_i":

$$\widetilde{I}(x_n) = \{i \in \{1, \dots, q\} \,|\, h_i(x_n) \geq 0\}$$
$$\widetilde{I}_\epsilon(x_n) = \{i \in \{1, \dots, q\} \,|\, h_i(x_n) \geq -\epsilon_i\}.$$

4. Denoting by $\widetilde{q}_\epsilon := \mathrm{Card}(\widetilde{I}_\epsilon)$, solve the dual problem

$$(\boldsymbol{\lambda}^*_\epsilon(x_n), \boldsymbol{\mu}^*_\epsilon(x_n)) := \arg \min_{\substack{\boldsymbol{\lambda} \in \mathbb{R}^p \\ \boldsymbol{\mu} \in \mathbb{R}^{\widetilde{q}_\epsilon(x)}, \, \boldsymbol{\mu} \geq 0}} ||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^\mathcal{T} \boldsymbol{\lambda} + \mathrm{D}\boldsymbol{h}_{\widetilde{I}_\epsilon(x)}(x)^\mathcal{T} \boldsymbol{\mu}||_V$$

to obtain the optimal Lagrange multiplier $\boldsymbol{\mu}^*(x_n)$. Infer the subset $\widehat{I}_\epsilon(x_n) \subset \widetilde{I}_\epsilon(x_n)$ indicating which constraints must remain active (Proposition 4) :

$$\widehat{I}_\epsilon(x_n) = \{i \in \widetilde{I}_\epsilon(x_n) \,|\, \mu^*_{\epsilon,i}(x_n) > \texttt{tolLag}\}. \tag{4.8}$$

5. Let $I^*_\epsilon(x_n) := \widetilde{I}(x_n) \cup \widehat{I}_\epsilon(x_n)$. Extract the vectors $\boldsymbol{C}_{\widehat{I}_\epsilon(x_n)}(x_n)$ and $\boldsymbol{C}_{I^*_\epsilon(x_n)}(x_n)$ (defined by (1.3)) and compute

$$\begin{aligned} \boldsymbol{\xi}_J(x_n) &= (\mathrm{I} - \mathrm{D}\boldsymbol{C}^\mathcal{T}_{\widehat{I}_\epsilon(x_n)}(\mathrm{D}\boldsymbol{C}_{\widehat{I}_\epsilon(x_n)}\mathrm{D}\boldsymbol{C}^\mathcal{T}_{\widehat{I}_\epsilon(x_n)})^{-1}\mathrm{D}\boldsymbol{C}_{\widehat{I}_\epsilon(x_n)})\nabla J(x_n), \\ \boldsymbol{\xi}_C(x_n) &= \mathrm{D}\boldsymbol{C}^\mathcal{T}_{I^*_\epsilon(x_n)}(\mathrm{D}\boldsymbol{C}_{I^*_\epsilon(x_n)}\mathrm{D}\boldsymbol{C}^\mathcal{T}_{I^*_\epsilon(x_n)})^{-1}\boldsymbol{C}_{I^*_\epsilon(x_n)}. \end{aligned} \tag{4.9}$$

for $k = 1 \dots \texttt{maxtrials}$ **do**

Compute the step

$$x_{n+1} = x_n - \frac{\Delta t}{2^{k-1}}(\alpha_J \boldsymbol{\xi}_J(x_n) + \alpha_C \boldsymbol{\xi}_C(x_n)).$$

if $\texttt{merit}_{x_n}(x_{n+1}) < \texttt{merit}_{x_n}(x_n)$ **then**

break

end if

end for

end for

5. Application to shape optimization

With the previous material at hand, we are now in position to present our optimization strategy dedicated to shape and topology optimization problems. In such applications, the optimization takes place within a set of shapes in $\mathbb{R}^d$ ($d = 2$ or 3):

$$\mathcal{X} = \{\Omega \subset D \,|\, \Omega \text{ open and Lipschitz}\}, \tag{5.1}$$

where $D \subset \mathbb{R}^d$ is an enclosing 'hold-all' domain. Since $\mathcal{X}$ is not even a vector space, the present context does not fall into the optimization framework described in Sections 2 and 3. However, $\mathcal{X}$ can be locally replaced by a subset which is naturally identified to the (infinite-dimensional) Banach space $W^{1,\infty}(D, \mathbb{R}^d)$ (see below). This is achieved in the framework of Hadamard's method of boundary variations, as explained in Section 5.1. Introducing a suitable Hilbert space $V \subset W^{1,\infty}(D, \mathbb{R}^d)$, the set $\mathcal{X}$ can be endowed with a manifold structure, where V plays the role of a tangent space, as we outline in Section 5.2. These facts make

it possible to extend our dynamical system (3.1) to this context, up to small adaptations, as described below. Section 5.3 explains several implementations details of Algorithm 1 that are specific to shape optimization. In particular, we highlight how the classical extension and regularization procedures of shape derivatives are naturally included in our method when using the definition (2.3) of the Hilbertian transposition $\mathcal{T}$.

5.1. Hadamard's framework for gradient-based shape optimization

The essence of Hadamard's method (see for instance [3, 35, 47, 57]) is to replace the complicated set of shapes $\mathcal{X}$ in (5.1), by the set $\mathcal{O}$ defined in (1.11). which has the simpler structure of a Banach space. The induced parametrization of shapes by vector fields $\boldsymbol{\theta} \in W^{1,\infty}(D,\mathbb{R}^d)$ gives rise to the following definition of shape derivative.

Definition 5. A function $F : \mathcal{X} \to \mathbb{R}$; $\Omega \mapsto F(\Omega)$ is *shape differentiable* at $\Omega \in \mathcal{X}$ if the underlying mapping $\boldsymbol{\theta} \mapsto F((\mathrm{I}+\boldsymbol{\theta})\Omega)$, from $W^{1,\infty}(D,\mathbb{R}^d)$ into $\mathbb{R}$, is Fréchet differentiable at $\boldsymbol{\theta} = 0$. The corresponding derivative, denoted by $\mathrm{D}F(\Omega) : W^{1,\infty}(D,\mathbb{R}^d) \to \mathbb{R}$, is called the *shape derivative* of F at Ω and the following expansion holds in the vicinity of $\boldsymbol{\theta} = 0$:

$$F((\mathrm{I}+\boldsymbol{\theta})\Omega) = F(\Omega) + \mathrm{D}F(\Omega)(\boldsymbol{\theta}) + o(\boldsymbol{\theta}), \text{ where } \frac{|o(\boldsymbol{\theta})|}{||\boldsymbol{\theta}||_{W^{1,\infty}(D,\mathbb{R}^d)}} \xrightarrow{\boldsymbol{\theta}\to 0} 0. \tag{5.2}$$

When dealing with shape optimization problems of the form (1.1), we consider objective and constraint functions $J : \mathcal{O} \to \mathbb{R}$, $\boldsymbol{g} : \mathcal{O} \to \mathbb{R}^p$ and $\boldsymbol{h} : \mathcal{O} \to \mathbb{R}^q$ which are shape differentiable in the sense of Definition 5.

Since $W^{1,\infty}(D,\mathbb{R}^d)$ is not a Hilbert space, the shape derivative $\mathrm{D}J(\Omega)$ of J at Ω (and those of $\boldsymbol{g}$ and $\boldsymbol{h}$) cannot be readily identified with a gradient vector $\boldsymbol{\xi} \in W^{1,\infty}(D,\mathbb{R}^d)$. To circumvent this drawback, we introduce a Hilbert space of vector fields $V \subset W^{1,\infty}(D,\mathbb{R}^d)$, with inner product $\langle\cdot,\cdot\rangle_V$, and where the inclusion is continuous. This ensures that $\mathrm{D}J(\Omega)$, $\mathrm{D}\boldsymbol{g}(\Omega)$ and $\mathrm{D}\boldsymbol{h}(\Omega)$ are also continuous linear operators on V, hence the definitions of the gradient $\nabla J(\Omega) \in V$ and of the transposed operators $\mathrm{D}\boldsymbol{g}^{\mathcal{T}}(\Omega) : \mathbb{R}^p \to V$, $\mathrm{D}\boldsymbol{h}^{\mathcal{T}}(\Omega) : \mathbb{R}^q \to V$ with respect to the inner product $\langle\cdot,\cdot\rangle_V$ make sense; see Definition 1. For instance, the gradient $\nabla J(\Omega) \in V$ is obtained by solving the so-called identification problem:

$$\forall \boldsymbol{\theta} \in V,\ \ \langle \nabla J(\Omega), \boldsymbol{\theta}\rangle_V = \mathrm{D}J(\Omega)(\boldsymbol{\theta}). \tag{5.3}$$

A typical choice as for the Hilbert space $V \subset W^{1,\infty}(D,\mathbb{R}^d)$ is the Sobolev space $V = H^m(D,\mathbb{R}^d)$ with $m > 1 + d/2$, equipped with its standard inner product (the inclusion $H^m(D,\mathbb{R}^d) \subset W^{1,\infty}(D,\mathbb{R}^d)$ being a consequence of the Sobolev embedding theorem, see [18]). In this case, the identification problem (5.3) boils down to a linear elliptic problem of order $2m$.

Let us recall that, under mild regularity assumptions on the objective function $J(\Omega)$, the shape derivative of $J(\Omega)$ can be written in the form of a boundary integral involving only the normal component of the deformation $\boldsymbol{\theta}$ (this is the so-called *Hadamard structure theorem* [35, 47, 56]). In practice, in all the considered applications hereafter, there exists $v_J(\Omega) \in L^1(\partial\Omega)$ such that:

$$\forall \boldsymbol{\theta} \in W^{1,\infty}(D,\mathbb{R}^d),\ \ \mathrm{D}J(\Omega)\boldsymbol{\theta} = \int_{\partial\Omega} v_J(\Omega)\ \boldsymbol{\theta}\cdot\boldsymbol{n}\mathrm{d}s. \tag{5.4}$$

A common strategy in the literature (see for instance [10, 14, 21, 32, 24, 45]) consists in taking simply $H^1(D,\mathbb{R}^d)$ as for the Hilbert space V, equipped with the inner product

$$\forall \boldsymbol{\theta}, \boldsymbol{\theta}' \in V,\ \langle \boldsymbol{\theta}, \boldsymbol{\theta}'\rangle_V = \int_D (\gamma^2 \nabla\boldsymbol{\theta} : \nabla\boldsymbol{\theta}' + \boldsymbol{\theta}\cdot\boldsymbol{\theta}')\mathrm{d}x, \tag{5.5}$$

where $\gamma > 0$ is a user-defined parameter which can physically be interpreted as a length-scale for the regularity of deformations $\boldsymbol{\theta}$ (typically, $\gamma = 3\,\texttt{hmin}$ where `hmin` is the minimum edge length of the mesh discretization). Note that this choice of V is an abuse of the above framework since $H^1(D,\mathbb{R}^d)$ is not a subspace of $W^{1,\infty}(D,\mathbb{R}^d)$. However, under the very mild assumption $v_J(\Omega) \in L^2(\partial\Omega)$, (which is for instance satisfied in the situations considered in Section 6), the identification problem (5.3) is still well-posed because (5.4) defines a continuous linear form on $H^1(D,\mathbb{R}^d)$. In such a situation, the identification (5.3) to (5.5) is interpreted as an extension and regularization of the normal velocity $v_J(\Omega)$ to the whole domain D. This practice and its consistency with respect to optimization are very classical issues in shape optimization, see [10, 14, 21, 24, 45]. In particular, variants can be considered for tuning more finely the smoothness of such

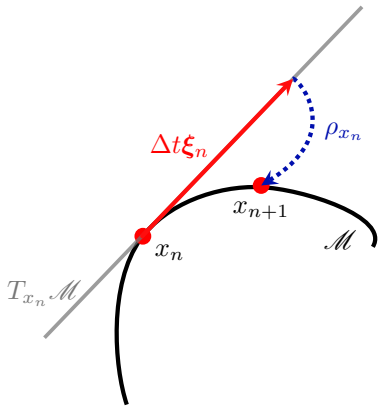


FIGURE 2. Optimization on a manifold $\mathscr{M}$: a retraction map ρ_{x_n} is used to project a tangential motion $\Delta t\boldsymbol{\xi}_n \in T_{x_n}\mathscr{M}$ from $x_n \in \mathscr{M}$ back onto the optimization domain $\mathscr{M}$.

extensions, or to prescribe non optimizable boundaries by imposing a zero Dirichlet boundary condition in (5.4). Eventually, the choice $V = H^1(D, \mathbb{R}^d)$ is quite convenient because this space is easily discretized with $\mathbb{P}_1$ finite elements. Since this leads to very good results in practice, we shall rely on this strategy in the following.

In light of the previous discussion, the proposed dynamical system (3.1) for tackling shape optimization problems of the form (1.1) is extended and discretized as follows.

(1) The null space and range space directions $\boldsymbol{\xi}_J(\Omega)$ and $\boldsymbol{\xi}_C(\Omega)$ are computed as elements of $V = H^1(D, \mathbb{R}^d)$ thanks to the formulas (3.7) and (3.24). This requires the computation of the gradient $\nabla J(\Omega)$ and of the transposes $\mathrm{D}\boldsymbol{g}^\mathcal{T}(\Omega)$, $\mathrm{D}\boldsymbol{h}^\mathcal{T}(\Omega)$ via the resolution of identification problems such as (5.3). In particular, steps 1 to 4 of Algorithm 1 including the resolution of the dual problem (3.9) are achieved from the knowledge of the Fréchet derivatives and of their transposes.

(2) The update (3.2) of the design from one iteration to the next is performed by

$$\Omega_{n+1} := (I - \Delta t(\alpha_J \boldsymbol{\xi}_J(\Omega_n) + \alpha_C \boldsymbol{\xi}_C(\Omega_n)))\Omega_n, \tag{5.6}$$

and the step 5 of Algorithm 1 is adapted accordingly.

The numerical procedure to account for the deformation from Ω_n to Ω_{n+1} is presented in Section 5.3.

5.2. Manifold structures for shape optimization

Another interpretation of the Hadamard framework of Section 5.1 can be made in terms of manifold structures, see e.g. [13, 52]. This allows to extend the material of Sections 2 and 3 to shape optimization purposes by using 'classical' optimization strategies on smooth embedded manifolds $\mathscr{M} \subset \mathbb{R}^k$. In such a context, a descent direction at a point $x_n \in \mathscr{M}$ for some objective functional is typically sought as an element $\boldsymbol{\xi}_n \in T_{x_n}\mathscr{M}$ of the tangent space $T_{x_n}\mathscr{M}$ to $\mathscr{M}$ at x_n; see e.g. [29, 1]. Then one relies on a *retraction*, i.e., a mapping $\rho_{x_n} : T_{x_n}\mathscr{M} \to \mathscr{M}$, satisfying the following two consistency conditions:

$$\begin{cases} \rho_{x_n}(0) = x_n \\ \forall \boldsymbol{\xi} \in T_{x_n}\mathscr{M}, \ \dfrac{\mathrm{d}}{\mathrm{d}t}\Big|_{t=0} \rho_{x_n}(t\boldsymbol{\xi}) = \boldsymbol{\xi}. \end{cases}$$

The mapping ρ_{x_n} then allows to convert $\boldsymbol{\xi}_n$ into a practical update on $\mathscr{M}$:

$$x_{n+1} := \rho_{x_n}(\Delta t\boldsymbol{\xi}_n), \tag{5.7}$$

where $\Delta t > 0$ is the descent step; see [2] and Figure 2. Since the new point x_{n+1} belongs to $\mathscr{M}$, this procedure can be repeated iteratively.

In the context of Section 5.1, the set of shapes $\mathcal{X}$ plays the role of the manifold $\mathscr{M}$ and, by Hadamard's method, in view of (1.11), the tangent space to $\mathcal{X}$ at Ω can be identified to $W^{1,\infty}(D, \mathbb{R}^d)$. The corresponding

retraction is, for any $\boldsymbol{\theta} \in W^{1,\infty}(D, \mathbb{R}^d)$,

$$\rho_\Omega(\boldsymbol{\theta}) := (\mathrm{I} + \boldsymbol{\theta})(\Omega). \tag{5.8}$$

Formally, the set $W^{1,\infty}(D, \mathbb{R}^d)$ may be interpreted as the tangent space to $\mathcal{X}$ at Ω and the mapping ρ_Ω, which is defined by (1.11) on a neighborhood of $\boldsymbol{\theta} = 0$ in $W^{1,\infty}(D, \mathbb{R}^d)$, plays the role of a retraction. Finally, the bilinear form $\langle \cdot, \cdot \rangle_V$ introduced in (5.3), can be interpreted as a metric on the 'manifold of shapes' $\mathcal{X}$, see e.g. [52, 53] about this idea.

5.3. Implementation of the constrained gradient flow for level set based shape optimization

The employed level set framework for numerical shape and topology optimization is recalled in Section 5.3.1. Further technical details about the practical implementation of Algorithm 1 are then presented in Section 5.3.2.

5.3.1. *Numerical shape optimization using the level set method and a mesh evolution strategy*

Our numerical representation of shapes and their deformations relies on the level set method, pioneered in [49], then introduced in the shape optimization context in [9, 62]. A given shape Ω inside the fixed 'hold-all' domain D is represented by means of a scalar, level set function $\phi : D \to \mathbb{R}$ such that:

$$\begin{cases} \phi(x) < 0 & \text{if } x \in \Omega, \\ \phi(x) = 0 & \text{if } x \in \partial\Omega, \\ \phi(x) > 0 & \text{if } x \in D \setminus \overline{\Omega}. \end{cases} \tag{5.9}$$

The motion of a domain $\Omega(t)$ in D evolving over a period of time $(0, T)$, starting from a known shape $\Omega(0) = \Omega$, according to a velocity field $\boldsymbol{\theta} : D \to \mathbb{R}^d$ translates in terms of an associated level set function $\phi(t, x)$ (i.e. (5.9) holds at every time $t \in (0, T)$) by the following advection equation:

$$\begin{cases} \dfrac{\partial \phi}{\partial t}(t, x) + \boldsymbol{\theta}(x) \cdot \nabla \phi(t, x) = 0, & t \in (0, t), \ \ x \in D, \\ \phi(0, x) = \phi_0(x), & x \in D, \end{cases} \tag{5.10}$$

where ϕ_0 is one level set function for Ω.

In the implementation of the discretized optimization flow (1.2), passing from the current iteration, indexed by n to the next one $n + 1$ implies the motion of the corresponding shape Ω_n along the descent direction $\boldsymbol{\theta}_n \in H^1(D, \mathbb{R}^d)$ given by

$$\boldsymbol{\theta}_n := -(\alpha_{J,n} \boldsymbol{\xi}_J(\Omega_n) + \alpha_{C,n} \boldsymbol{\xi}_C(\Omega_n)), \tag{5.11}$$

for a small time step Δt_n. Here, the coefficients α_J and α_C of the update procedure (5.6) may vary from one iteration to the next, as reflected by the $_n$ subscript (this slight modification of Algorithm 1 is detailed in Section 5.3.2 below). The level set function ϕ_n, corresponding to the current shape Ω_n, is updated by solving equation (5.10) on the current mesh $\mathcal{T}_n$ of D with $\boldsymbol{\theta} = \boldsymbol{\theta}_n$ and $\phi_0 = \phi_n$. After a time step Δt the new shape Ω_{n+1} is defined as $\{x \in D \,|\, \phi(\Delta t_n, x) < 0\}$.

In the implementation, we use the mesh evolution technique of our previous works [5, 32]. In a few words, at every iteration n, the current shape Ω_n is explicitly discretized as a submesh of a triangulated mesh $\mathcal{T}_n$ of D as a whole (see e.g. Figure 10 below). After solving equation (5.10) thanks to an adapted solver (in practice we use that of our previous work [20]), $\mathcal{T}_n$ is remeshed adaptively into a new mesh $\mathcal{T}_{n+1}$ featuring a discretization of Ω_{n+1} as a submesh, by using the open-source library `Mmg` [22].

Remark 7. In our method, the velocity $\boldsymbol{\theta} \in H^1(D, \mathbb{R}^d)$ is a vector field, in contrast with more classical level set methods [9, 62] that rather rely on a non linear Hamilton-Jacobi equation, which contrary to (5.10) involves only the normal component of $\boldsymbol{\theta}$. In the latter case, it is enough to regularize only the normal component $\boldsymbol{\theta} \cdot \boldsymbol{n}$ (a scalar field) of the shape derivative; see [31] for more details.

5.3.2. *Adaptive normalizations for the null space and range space directions*

A few comments are in order regarding the appropriate scaling of the null and range space steps with respect to the size of the mesh discretization in the practice of Algorithm 1, and the choice of variable coefficients $\alpha_{J,n}$ and $\alpha_{C,n}$ in the descent direction $\boldsymbol{\theta}_n$ given by (5.11).

For stability reasons, the vertices of the current mesh $\mathcal{T}_n$ accounting for Ω_n should move by a distance which equals at most a few mesh elements in order to produce the subsequent shape Ω_{n+1}. Hence, the minimum edge length `hmin` of the computational mesh is a natural candidate for the limiting step size value `h` in the discussion in Section 4.1. In our practical implementation, we set $\Delta t = 1$ and a descent direction $\boldsymbol{\theta}_n \in H^1(D, \mathbb{R}^d)$ is computed by estimating

$$\boldsymbol{\theta}_n := -(\alpha_{J,n}\boldsymbol{\xi}_J(\Omega_n) + \alpha_{C,n}\boldsymbol{\xi}_C(\Omega_n)), \tag{5.12}$$

where α_J and α_C of the update (5.6) have been replaced by dynamic coefficients $\alpha_{J,n}$ and $\alpha_{C,n}$.

The parameters $\alpha_{J,n}$ and $\alpha_{C,n}$ scaling the null space and range space steps $\boldsymbol{\xi}_J(\Omega_n)$ and $\boldsymbol{\xi}_C(\Omega_n)$ are updated dynamically in order to control the step size $||\boldsymbol{\theta}_n||_{L^\infty(D,\mathbb{R}^d)}$. Note that the $||\cdot||_{L^\infty(D,\mathbb{R}^d)}$ norm is considered because *all* values of the displacement $\boldsymbol{\theta}_n$ should be of the order of the mesh size. We consider A_J and A_C two user-defined parameters, which are expressed in terms of the minimum edge length `hmin` for a clearer intuitive meaning. The coefficients $\alpha_{J,n}$ and $\alpha_{C,n}$ are updated at every iteration according to the following rules:

$$\alpha_{J,n} := \begin{cases} \dfrac{A_J \texttt{hmin}}{||\boldsymbol{\xi}_J(\Omega_n)||_{L^\infty(D,\mathbb{R}^d)}} & \text{if } n < \texttt{n}_0 \\ \dfrac{A_J \texttt{hmin}}{\max(||\boldsymbol{\xi}_J(\Omega_n)||_{L^\infty(D,\mathbb{R}^d)}, ||\boldsymbol{\xi}_J(\Omega_{\texttt{n}_0})||_{L^\infty(D,\mathbb{R}^d)})} & \text{if } n \geq \texttt{n}_0 \end{cases} \tag{5.13}$$

$$\alpha_{C,n} := \min\left(0.9, \frac{A_C \texttt{hmin}}{\max\left(\texttt{1e-9}, ||\boldsymbol{\xi}_C(\Omega_n)||_{L^\infty(D,\mathbb{R}^d)}\right)}\right). \tag{5.14}$$

These normalizations ensure that the null space and range space steps always remain smaller than multiples A_J and A_C of the mesh size:

$$\forall n \geq 0,\ ||\alpha_{J,n}\boldsymbol{\xi}_J(\Omega_n)||_{L^\infty(D,\mathbb{R}^d)} \leq A_J \texttt{hmin} \text{ and } ||\alpha_{C,n}\boldsymbol{\xi}_C(\Omega_n)||_{L^\infty(D,\mathbb{R}^d)} \leq \min(0.9, A_C \texttt{hmin}).$$

Actually, the null space contribution $\alpha_{J,n}\boldsymbol{\xi}_J(\Omega_n)$ of $\boldsymbol{\theta}_n$ is scaled so that its size is exactly A_J`hmin` during the first $\texttt{n}_0$ iterations:

$$\forall 1 \leq n \leq \texttt{n}_0,\ ||\alpha_{J,n}\boldsymbol{\xi}_J(\Omega_n)||_{L^\infty(D,\mathbb{R}^d)} = A_J \texttt{hmin}.$$

Then, $\boldsymbol{\xi}_J(\Omega_n)$ is allowed to converge to 0 as $n \to \infty$.

The range step $\alpha_{C,n}\boldsymbol{\xi}_C(\Omega_n)$ is also set to remain smaller than the constant 0.9, in view of the stability condition $0 < \alpha_C \Delta t < 2$ (see Remark 13). The role of the constant `1e-9` is only to avoid division by 0 when no constraint is active.

Remark 8. Since we measure step sizes with the norm $||\boldsymbol{\theta}_n||_{L^\infty(D)}$ rather than with the Hilbertian norm $||\boldsymbol{\theta}_n||_V = ||\boldsymbol{\theta}_n||_{H^1(D,\mathbb{R}^d)}$, the tolerance bounds (4.3) need to be updated with respect to this norm as follows:

$$\epsilon_i := \texttt{hmin} \int_{\partial\Omega} |v_{C_i}(\Omega_n)| \mathrm{d}s,$$

where it is assumed that the shape derivative of each constraint functional $C_i(\Omega_n)$ can be written as a boundary integral, as in (5.4), featuring the scalar field $v_{C_i}(\Omega_n)$:

$$\mathrm{D}C_i(\Omega_n)(\boldsymbol{\theta}) := \int_{\partial\Omega} v_{C_i}(\Omega_n)\, \boldsymbol{\theta} \cdot \boldsymbol{n} \mathrm{d}s\,.$$

6. Applications to shape optimization in the design of mechanical structures

In this final section, we illustrate the efficiency of our optimization strategy with practical structural design examples. We first treat in Section 6.1 a structural design problem in thermoelasticity which was tackled in [64] with an Augmented Lagrangian Method. This case study is interesting because it features a situation where an initially violated inequality constraint is not saturated by the final design. Then, we treat in

Section 6.2 a multiple load bridge test case in order to show the ability of the method to handle multiple objective criteria or multiple constraint functions.

6.1. Minimum compliance problem in thermoelasticity: detection of unsaturated constraints

In this section, we reproduce a test case coming from Xia and Wang [64] concerned with compliance minimization in thermoelasticity. The objective of our study is to illustrate (i) the efficiency of our null space optimization scheme in contrast with the classical Augmented Lagrangian strategy used in [64] and (ii) the importance of the use of the dual problem (3.9) for discriminating the inequality constraints which must remain saturated in the course of the optimization process.

The structure $\Omega \subset D$ is sought within the fixed 'hold-all' domain $D = [0,2] \times [0,1]$. It is made of an elastic material characterized by Lamé parameters $\lambda = 11510$, $\mu = 7673$, thermoelastic coefficient $\alpha = 0.77$ and reference temperature $T_{\text{ref}} = 0$. A constant temperature field $T = T_{\text{ref}} + \Delta T$ is applied on the whole

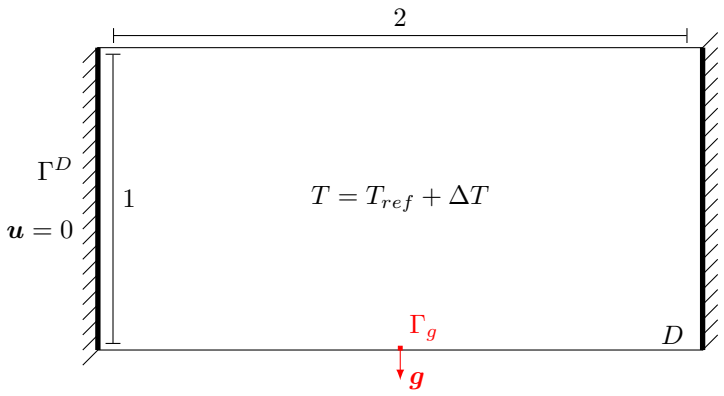


FIGURE 3. Setting for the thermoelastic compliance minimization problem of Section 6.1, issued from [64].

structure, which induces thermal expansion of the material. The boundary of the considered shape is divided into three parts:

$$\partial\Omega = \Gamma_D \cup \Gamma_g \cup \Gamma,$$

where:

- Γ_D is the reunion of the (non optimizable) left and right-hand boundaries of D on which the structure Ω is clamped.
- Γ_g is a (non optimizable) small portion of the middle of the bottom boundary on which a vertical traction load $\boldsymbol{g} = (0, -1/|\Gamma_g|)$ is applied. In our implementation, we set $|\Gamma_g| = 0.0125$.
- Γ is the remaining part of $\partial\Omega$ which is traction-free; it is the only part of $\partial\Omega$ which is subject to optimization.

The setting is reproduced on Figure 3. Both thermal and traction loads ΔT and $\boldsymbol{g}$ induce an elastic displacement $\boldsymbol{u} \in H^1(\Omega, \mathbb{R}^d)$ which is given by the solution of the linearized thermoelasticity system:

$$\begin{cases} -\text{div}(\sigma_s(\boldsymbol{u}, \Delta T)) = 0 \text{ in } \Omega \\ \sigma_s(\boldsymbol{u}, \Delta T)\boldsymbol{n} = \boldsymbol{g} \text{ on } \Gamma_g \\ \sigma_s(\boldsymbol{u}, \Delta T)\boldsymbol{n} = 0 \text{ on } \Gamma \\ \boldsymbol{u} = 0 \text{ on } \Gamma_D, \end{cases} \tag{6.1}$$

where $\sigma_s(\boldsymbol{u}, \Delta T)$ is the thermoelastic tensor given by

$$\sigma_s(\boldsymbol{u}, \Delta T) := Ae(\boldsymbol{u}) - \alpha \Delta T \text{div}(\boldsymbol{u})\text{I},$$

associated to the Hooke's law

$$Ae(\boldsymbol{u}) := 2\mu e(\boldsymbol{u}) + \lambda \text{Tr}(e(\boldsymbol{u}))\text{I}, \text{ with } e(\boldsymbol{u}) = (\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)/2.$$

Our goal is to minimize the compliance of the structure (that is, to maximize its rigidity) under a volume inequality constraint:

$$\begin{aligned} \min_{\Omega\in\mathcal{X}} \quad & J(\Omega,\boldsymbol{u}(\Omega)) := \int_\Omega Ae(\boldsymbol{u}) : e(\boldsymbol{u})\mathrm{d}x \\ s.t. \quad & \mathrm{Vol}(\Omega) := \int_\Omega \mathrm{d}x \leq V_{\mathrm{target}}. \end{aligned} \tag{6.2}$$

The shape derivatives of J and Vol are classically given by (see [64, 31]):

$$\mathrm{D}J(\Omega)(\boldsymbol{\theta}) = \int_\Gamma (-Ae(\boldsymbol{u}) : e(\boldsymbol{u}) + 2\alpha\Delta T\mathrm{div}(\boldsymbol{u}))\boldsymbol{\theta}\cdot\boldsymbol{n}\mathrm{d}s, \qquad \mathrm{DVol}(\Omega)(\boldsymbol{\theta}) = \int_\Gamma \boldsymbol{\theta}\cdot\boldsymbol{n}\mathrm{d}s,$$

where $\boldsymbol{n}$ denotes the normal vector to $\partial\Omega$, pointing outward Ω.

The upper bound for the volume of the structure is set to $V_{\mathrm{target}} = 0.4$. This problem is particularly interesting when it comes to illustrate the relevance of our dual problem strategy in the determination of whether an active constraint is aligned with the minimization of the objective function or not. Indeed, as we shall see below, the optimized design may not saturate the volume constraint $\mathrm{Vol}(\Omega) \leq V_{\mathrm{target}}$ depending on the considered value of the parameter ΔT.

Following [64], the optimization problem is solved for four values of ΔT ($\Delta T = 0$, 5, 10 or 20). In order to illustrate the importance of the resolution of the dual problem (3.9), we also consider a strategy (labeled '*no-dual*') obtained by ignoring the corresponding step 4 in Algorithm 1 and by setting 'naively' $\widehat{I}_\epsilon(x_n) = \widetilde{I}_\epsilon(x_n)$ in the subsequent steps.

Results are shown on Figure 4 where the convergence histories for the objective function and the volume constraint are plotted for each test case. The numerical values for the final objective and constraint functionals are provided in Table 1, while the initial and optimized shapes are shown on Figs. 5 and 6. Note that our numerical values do not coincide exactly with those in [64] because their original physical parameters were multiplied by nondimensionalization constants which are more compatible with our setting. However we clearly retrieve very similar optimized shapes.

Notice the smooth and rather fast convergence of our optimization strategy, whereas in the original paper (Figure 8 of [64]), convergence is obtained in about 1000 iterations, after a lot of oscillations of the constraint function.

Interestingly, and as observed in [64], we retrieve the fact that the volume constraint $\mathrm{Vol}(\Omega) \leq V_{target}$ is saturated for the first two test cases $\Delta T = 0$ and $\Delta T = 5$, and is not saturated otherwise. In the first two cases, the sets $\widehat{I}_\epsilon(\Omega_n)$ and $\widetilde{I}_\epsilon(\Omega_n)$ remain identical, hence no difference is observed between the strict application of Algorithm 1 and its 'no-dual' variant (the corresponding convergence histories for the latter strategy are therefore not represented on Figure 4 in these cases). However, significant differences are observed for the two situations $\Delta T = 10$ and $\Delta T = 15$: the strategy labeled 'no-dual' proceeds by enforcing all saturated or violated constraints in $\widetilde{I}_\epsilon(\Omega_n)$, and not only those in $\widehat{I}_\epsilon(\Omega_n)$ (which is empty in the present case). As a result, it is not able to detect that a better descent direction could be obtained by allowing the constraint to become unsaturated (and as a matter of fact, to become 'better' satisfied): the constraint remains saturated and a worse optimal final design is obtained at convergence.

6.2. **Shape optimization of a bridge structure subjected to multiple loads**

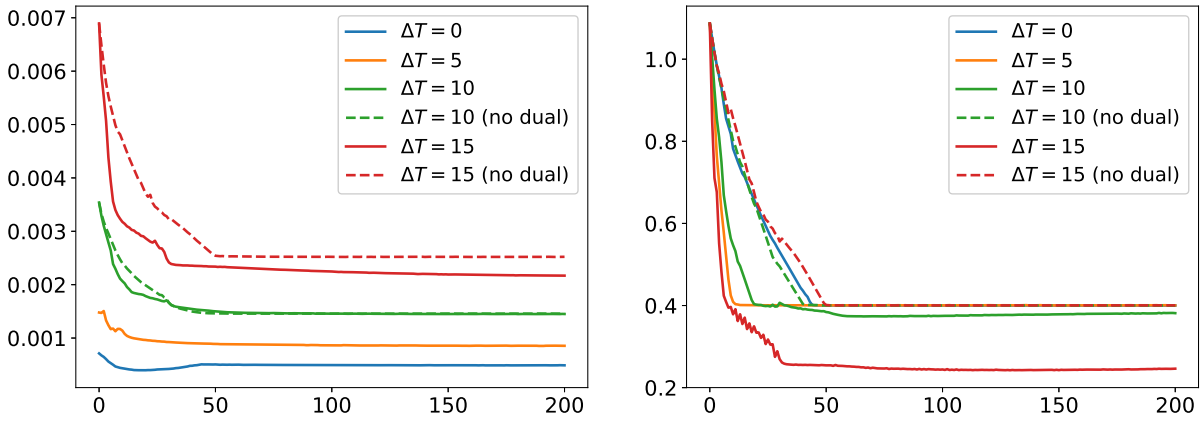
Let us now consider the shape optimization of a bridge-like structure Ω contained in a two-dimensional rectangular hold-all domain $D \subset \mathbb{R}^2$ with size 10×2. The purpose of this part is to show that the null space gradient flow is able to handle multiple constraints.

The boundary $\partial\Omega$ of the bridge is divided into disjoint regions as:

$$\partial\Omega = \Gamma \cup \Gamma_D \cup \bigcup_{i=0}^{8} \Gamma_i,$$

where

- Γ_D is a non-optimizable part of the boundary on which the structure Ω is clamped, made of two segments with unit length at the lower extremities of D.



(A) Objective function J.

(B) Volume constraint $\mathrm{Vol}(\Omega) \leq 0.4$.

FIGURE 4. Convergence histories for the thermoelasticity test case of Section 6.1.

ΔT	Final $J(\Omega, \boldsymbol{u}(\Omega))$	Final $\mathrm{Vol}(\Omega)$
0	0.0004891	0.4
5	0.0008546	0.4003
10	0.001451	0.3814
15	0.002169	0.2462
10 (no dual)	0.001457	0.4001
15 (no dual)	0.002522	0.4004

TABLE 1. Optimized compliance and volume values for the thermoelasticity test case of Section 6.1. The results are analogous to those of [62].

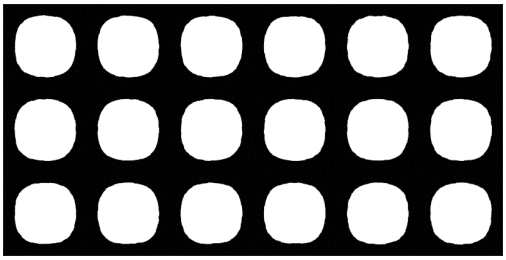
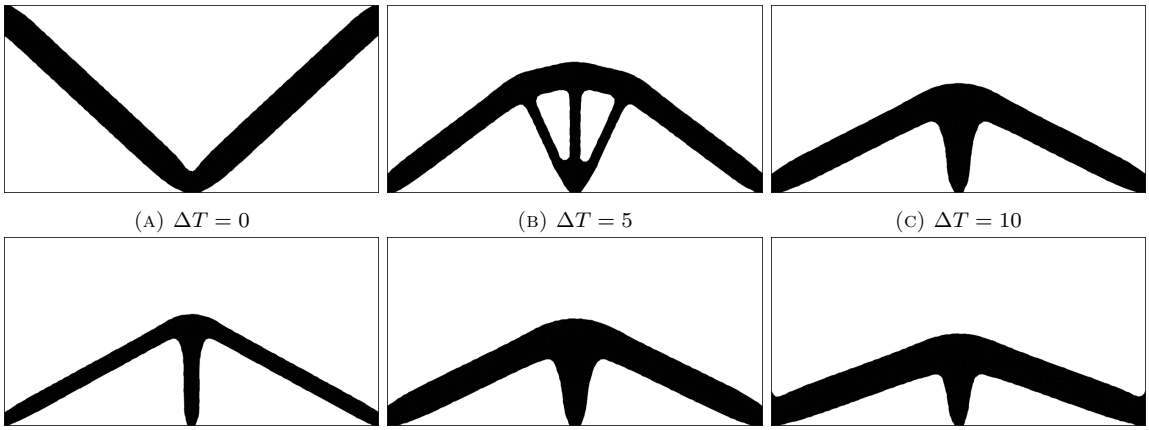


FIGURE 5. Initial design considered for each of the test cases of Section 6.1

- For $i = 0, ..., 8$, Γ_i is a non-optimizable subset of the upper side of D with respective abscissa $\left[i\frac{10}{9}, (i+1)\frac{10}{9}\right]$; Γ_i is subjected to a unit, vertical downward traction load $\boldsymbol{g}_i = (0, -1)$.
- The remaining region Γ is traction-free and it is the only region of $\partial\Omega$ which is subject to optimization.

Non-optimizable material layers of width 0.1 are additionally imposed on the upper part of the domain D and above each component of Γ_D and we impose that the structure do not infringe on a thin layer of void at the bottom of ∂D; see Figs. 7 and 8. We consider nine different load cases, that are obtained by applying successively and exclusively each of the loads $\boldsymbol{g}_i$ on the region Γ_i. In each situation, the corresponding elastic



(A) $\Delta T = 0$ (B) $\Delta T = 5$ (C) $\Delta T = 10$

(D) $\Delta T = 15$ (E) $\Delta T = 10$ (no dual) (F) $\Delta T = 15$ (no dual)

FIGURE 6. Final designs computed for each of the test cases of Section 6.1.

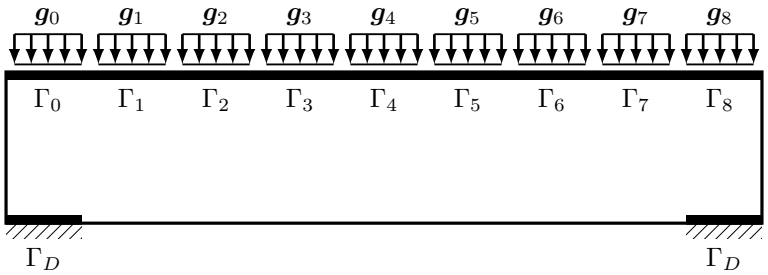


FIGURE 7. Geometric setting for the multiple load case test case

displacement $\boldsymbol{u}_i$ is the unique solution in $H^1(\Omega, \mathbb{R}^d)$ to the linearized elasticity system:

$$\begin{cases} -\mathrm{div}(Ae(\boldsymbol{u}_i)) = 0 & \text{in } \Omega \\ Ae(\boldsymbol{u}_i)\boldsymbol{n} = 0 & \text{on } \Gamma \\ Ae(\boldsymbol{u}_i)\boldsymbol{n} = \boldsymbol{g}_i & \text{on } \Gamma_i \\ Ae(\boldsymbol{u}_i)\boldsymbol{n} = 0 & \text{on } \Gamma_j \text{ for } j \neq i \\ \boldsymbol{u}_i = 0 & \text{on } \Gamma_D. \end{cases} \tag{6.3}$$

The Young's modulus and the Poisson's ratio are set to $E = 15$ and $\nu = 0.35$, which corresponds to $\lambda = 12.96$ and $\mu = 5.56$. Let us emphasize once again (see Section 5.3.1) that the shape is exactly meshed at each iteration (see Figure 10 below), so that each state equation (6.3) is solved by means of a standard finite element method on the meshed subdomain Ω_n (without resorting to ersatz material approaches as in e.g. [9]).

Starting from the initial structure Ω_0 depicted in Figure 8, we minimize the volume $\mathrm{Vol}(\Omega)$ of the structure Ω and maximize the collection of compliances $C_i(\Omega)$ (for each load case $\boldsymbol{g}_i$), which are defined by:

$$C_i(\Omega) := \int_\Omega Ae(\boldsymbol{u}_i) : e(\boldsymbol{u}_i)\mathrm{d}x. \tag{6.4}$$

Their shape derivatives read (see e.g. [9, 35]):

$$\mathrm{DVol}(\Omega)(\boldsymbol{\theta}) = \int_\Gamma \boldsymbol{\theta}\cdot\boldsymbol{n}\mathrm{d}s, \quad \mathrm{D}C_i(\Omega)(\boldsymbol{\theta}) = -\int_\Gamma Ae(\boldsymbol{u}_i) : e(\boldsymbol{u}_i)\,\boldsymbol{\theta}\cdot\boldsymbol{n}\mathrm{d}s. \tag{6.5}$$

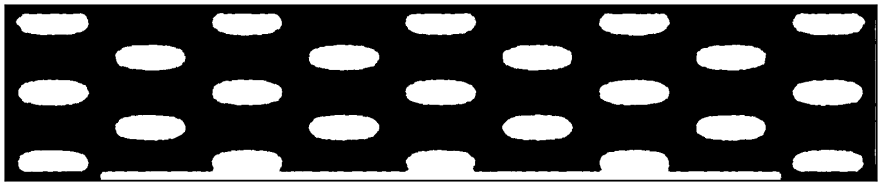


FIGURE 8. Initialisation Ω_0 (solid in *black*) for the shape optimization examples of Section 5. The thin white layer at the bottom is a non optimizable part of the domain.

In what follows, two possible configurations are investigated for the shape optimization of the bridge structure, featuring either multiple constraint functions (in Section 6.2.1) or multiple objective criteria (in Section 6.2.2).

6.2.1. *Volume minimization with maximum compliance constraint*

At first, the volume $\mathrm{Vol}(\Omega)$ is minimized and we require that each individual compliance $C_i(\Omega)$ do not exceed a given threshold C:

$$\begin{aligned} \min_{\Omega\in\mathcal{X}} \quad & \mathrm{Vol}(\Omega) \\ \text{s.t.} \quad & C_i(\Omega) \le C \quad \text{for all } i \in I \end{aligned} \tag{6.6}$$

where $I \subset \{0, 1, \dots, 8\}$ is a set of indices selecting the considered load cases. We solve (6.6) in the following three configurations:

(1) *Case 1: single load case:* $I = \{4\}$ (only the central load $\boldsymbol{g}_4$ is applied)
(2) *Case 2: three load case:* $I = \{0, 4, 8\}$ (only the central load $\boldsymbol{g}_4$ and the two extreme loads $\boldsymbol{g}_0$ and $\boldsymbol{g}_8$ are applied).
(3) *Case 3: all load cases:* $I = \{0, 1, \dots, 8\}$ (all nine loads are considered).

The value of C in (6.6) is set to a fraction of the maximum of the compliances $C_i(\Omega_0)$ of the initial design Ω_0 (reported on Figure 8):

$$C = 0.7 \max_{i=0,\dots,8} \int_{\Omega_0} Ae(\boldsymbol{u}_i) : e(\boldsymbol{u}_i)\mathrm{d}x. \tag{6.7}$$

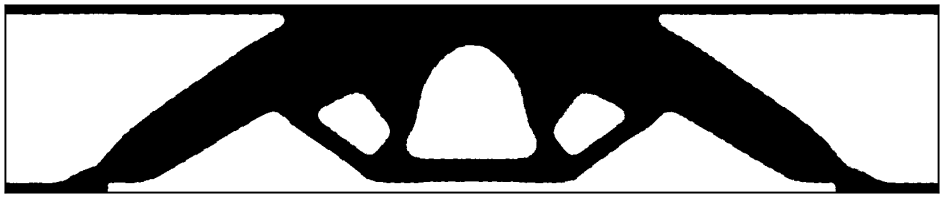
Let us emphasize that for this example (and the next ones), no fine tuning of the algorithm parameters A_J and A_C (determining the update of the values of $\alpha_{J,n}$ and $\alpha_{C,n}$ in (5.12)) of Section 5.3.2 is required. The only intuition guiding our choice for this particular test case is that the value of A_J should be set lower than A_C. Indeed, a too large value of A_J might entail a too fast decrease of the volume, which would incur dramatic topological changes violating the rigidity constraints. Therefore these parameters are set to $A_J = 1$ and $A_C = 2$ for this test case. The minimum mesh size is $\texttt{hmin} = 0.03$.

The optimized shapes obtained in the three aforementioned situations are represented on Figure 9. The meshes of the initial and final designs, as well as several intermediate shapes corresponding to the nine load test-case are shown on Figs. 10 and 11. The convergence histories in the three situations are reported on Figs. 12 to 14. They allow to verify the decrease of the objective function even after the saturation of the constraints. Note that for this example and the one to follow, the sets $\widehat{I}_\epsilon(\Omega_n)$ and $\widetilde{I}_\epsilon(\Omega_n)$ (see Algorithm 1) happen to coincide at every iteration. As expected, the optimum value found for the volume of the solid distribution increases with the number of constraints. The major structural change between the different situations is the addition of extra vertical bars of material near the extremities of the structure.

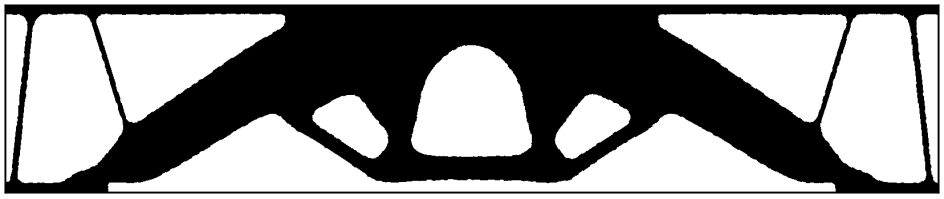
6.2.2. *Min/Max compliance optimization with a volume constraint*

Now, the maximum value of the compliances $C_i(\Omega)$ is minimized with an equality volume constraint:

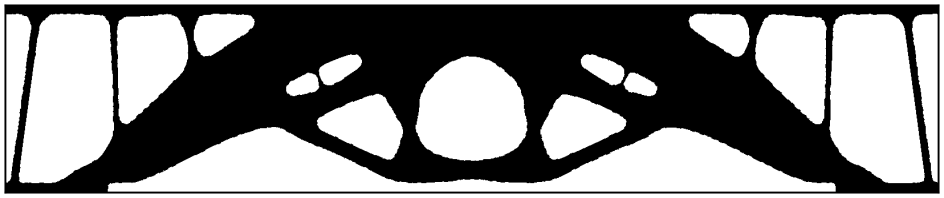
$$\begin{aligned} \min_{\Omega\in\mathcal{X}} \quad & \max_{i\in I} \quad C_i(\Omega) \\ \text{s.t.} \quad & \mathrm{Vol}(\Omega) = \rho_0 \mathrm{Vol}(D) \end{aligned} \tag{6.8}$$



(A) One load (only $\boldsymbol{g}_4$ is considered).



(B) Three loads (only $\boldsymbol{g}_0, \boldsymbol{g}_4, \boldsymbol{g}_8$ are considered).



(C) All nine loads.

FIGURE 9. Optimized shapes for three possible configurations of the volume minimization problem subject to maximum compliance constraint (Section 6.2.1).

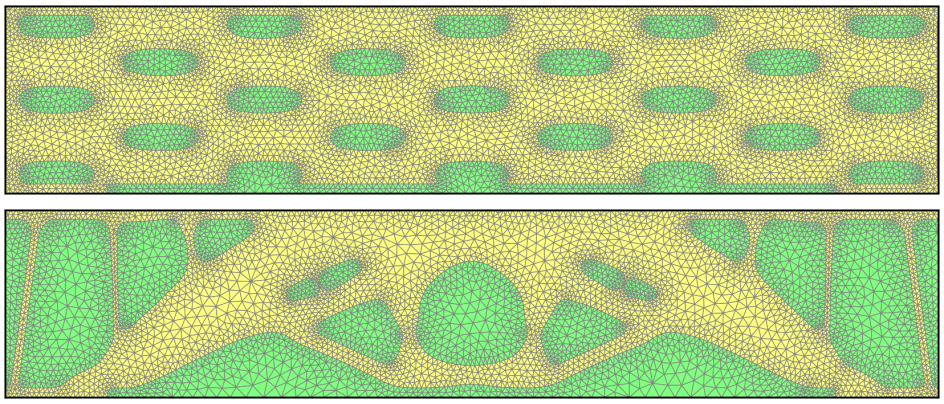


FIGURE 10. Meshes of the initial and final shapes for the nine load case of Figure 9c (Section 6.2.1).

for a target volume fraction $\rho_0 = 0.5$ of elastic material and for the three load sets I introduced in the previous subsection. This problem may be given the form (1.1) after introducing a slack variable m:

$$\min_{(\Omega,m)\in\mathcal{X}\times\mathbb{R}} \quad m$$
$$\text{s.t.} \begin{cases} \mathrm{Vol}(\Omega) = \rho_0 \mathrm{Vol}(D) \\ C_i(\Omega) \leq m \quad \text{for all } i \in I. \end{cases} \tag{6.9}$$

The optimization is now performed with respect to both the slack variable m and the domain geometry Ω, which demands minor adaptations of our optimization algorithm (similar e.g. to those in Section 3.5): the optimization set $\mathcal{X} \times \mathbb{R}$ is equipped with the tensorized tangent space $\widetilde{V} = V \times \mathbb{R}$ and differentials are

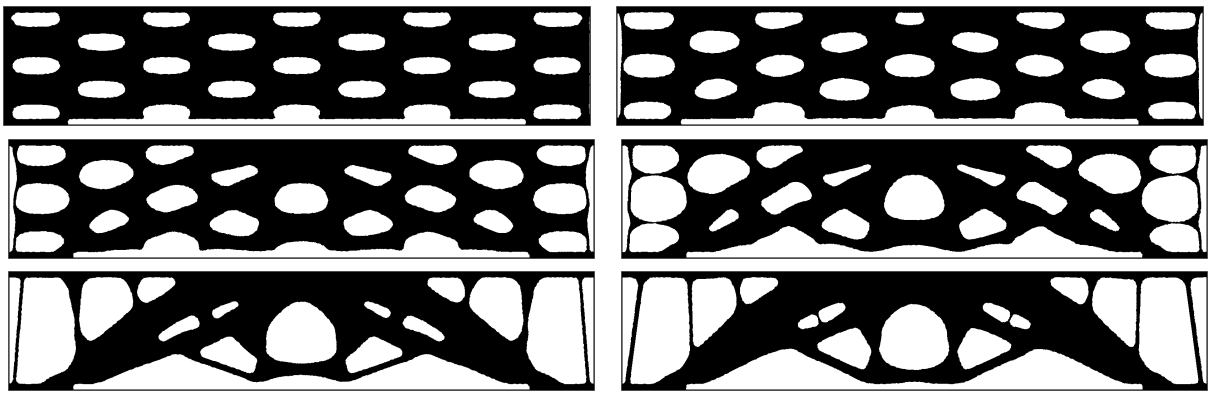
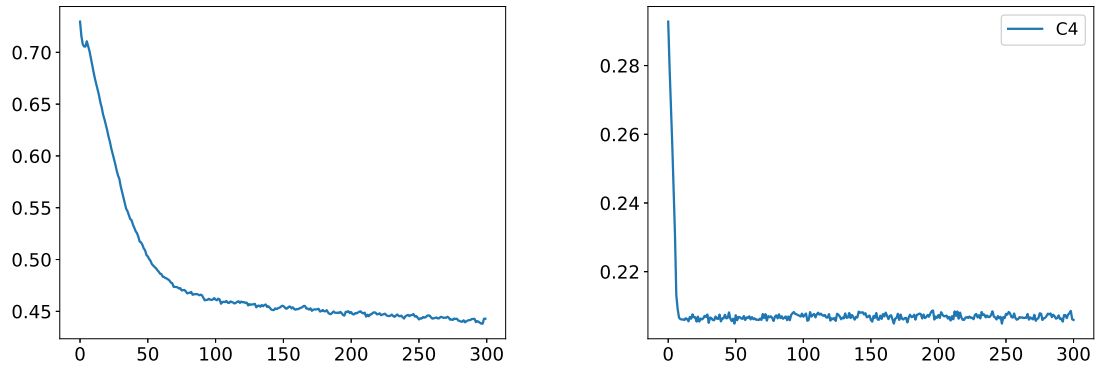


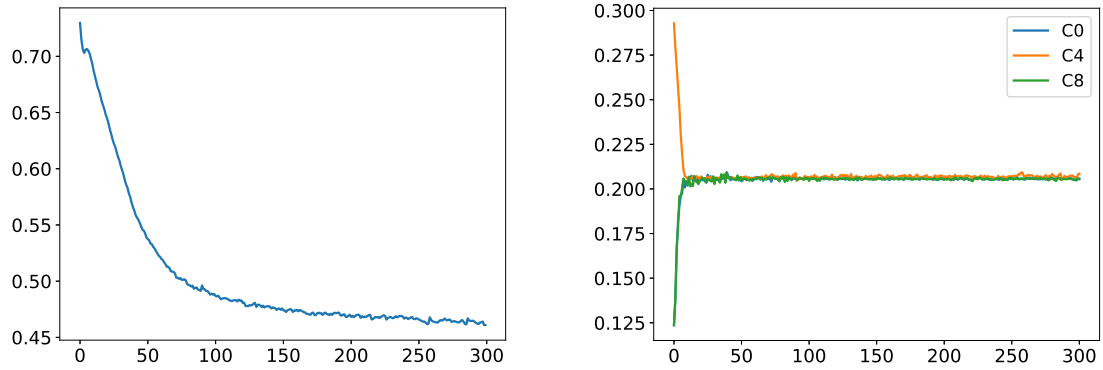
FIGURE 11. Intermediate minimizing shapes for the nine load case of the volume minimization problem of Section 6.2.1 (iterations 0, 5, 10, 20, 80, and 300).



(A) Objective function $J(\Omega) = \mathrm{Vol}(\Omega)$. Final value: $\mathrm{Vol}(\Omega) = 0.44\mathrm{Vol}(D)$.

(B) Constraints (compliance values C_i). Final values: $C_4 = 1.29$.

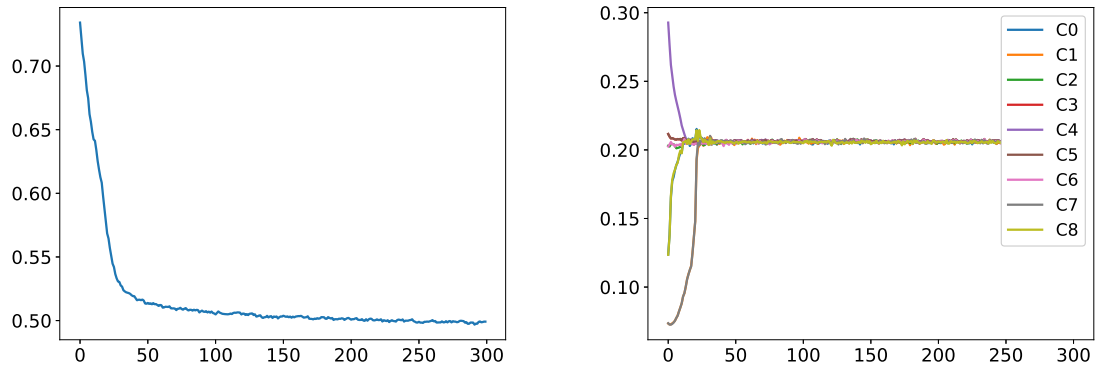
FIGURE 12. Convergence histories for the single load case of Section 6.2.1.



(A) Objective function $J(\Omega) = \mathrm{Vol}(\Omega)$. Final value: $\mathrm{Vol}(\Omega) = 0.46\mathrm{Vol}(D)$.

(B) Constraints (compliance values C_i). Final values: $C_0 = 1.29$, $C_4 = 1.30$, $C_8 = 1.28$.

FIGURE 13. Convergence history curves for the three load case of Section 6.2.1.



(A) Objective function $J(\Omega) = \mathrm{Vol}(\Omega)$. Final value: $\mathrm{Vol}(\Omega) = 0.50\mathrm{Vol}(D)$.

(B) Constraints (compliance values C_i). Final values: $C_0 = 1.29$, $C_1 = 1.28$, $C_2 = 1.28$, $C_3 = 1.29$, $C_4 = 1.29$, $C_5 = 1.29$, $C_6 = 1.29$, $C_7 = 1.30$, $C_8 = 1.29$.

FIGURE 14. Convergence history curves for the nine load case of Section 6.2.1.

identified to gradients thanks to the inner product $\langle \cdot, \cdot \rangle_{\widetilde{V}}$ defined by

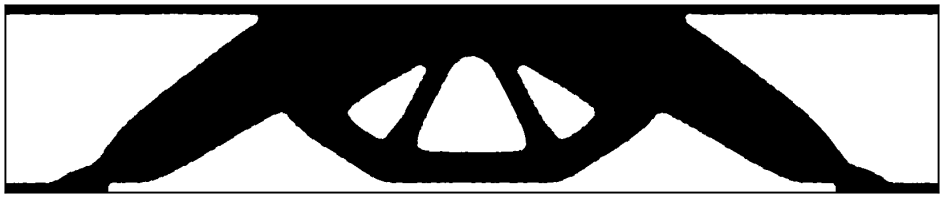
$$\forall (v,w) \in H^1(D,\mathbb{R}) \times H^1(D,\mathbb{R}),\ (l,m) \in \mathbb{R} \times \mathbb{R}, \quad \langle (v,l),(w,m) \rangle_{\widetilde{V}} := \langle v, w \rangle_V + lm, \tag{6.10}$$

where $\langle \cdot, \cdot \rangle_V$ is the scalar product of (5.5). The slack variable m is initialized with the maximum value of the compliance of the initial structure Ω_0 over all the considered loads:

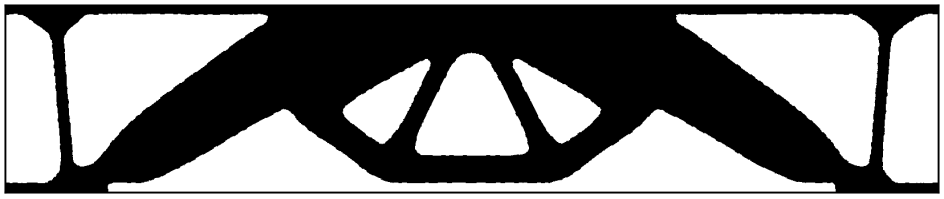
$$m_0 := \max_{i \in I} \quad C_i(\Omega_0), \tag{6.11}$$

and its values m_n are then updated along with the shape Ω_n according to Algorithm 1.

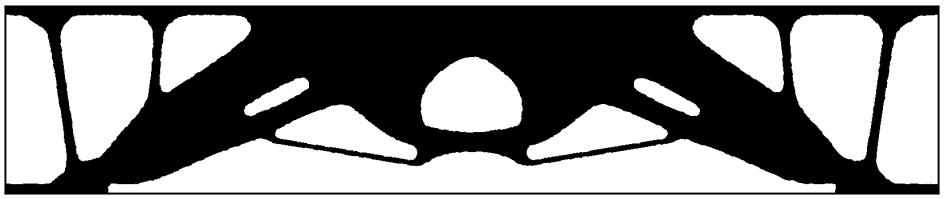
The resulting optimized structures are shown on Figure 15 for each of the three considered configurations and the associated convergence histories are displayed on Figs. 16 to 18 for the single, triple and nine load cases respectively. Note that sudden, abrupt peaks on the constraint curves correspond to topological changes (e.g. at iteration 38 for the nine load case) for which the displacements corresponding to the extremal loads $\boldsymbol{g}_0$ and $\boldsymbol{g}_8$ are especially sensitive. We observe the decrease of all the functionals $C_i(\Omega)$ even after all the inequality constraints have been saturated, which occurs as soon as the compliances reach a common value. As expected, the optimized design found for the nine load minimum compliance case (Figure 9c) is similar (up to a few bars) to the corresponding one found for the volume minimization (Figure 15c): indeed, both cases reach at convergence a volume fraction $\mathrm{Vol}(\Omega) = 0.5\mathrm{Vol}(D)$ and a maximum compliance $\max C_i(\Omega) \simeq 1.30$.



(A) One load (only $\boldsymbol{g}_4$ is considered).

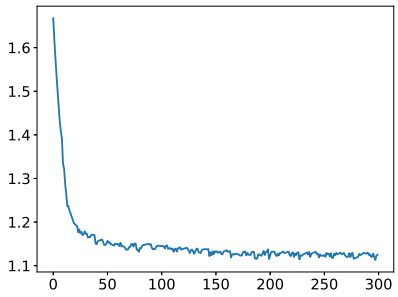


(B) Three loads (only $\boldsymbol{g}_0, \boldsymbol{g}_4, \boldsymbol{g}_8$ are considered).

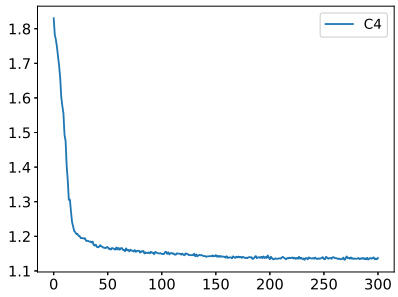


(C) All nine loads.

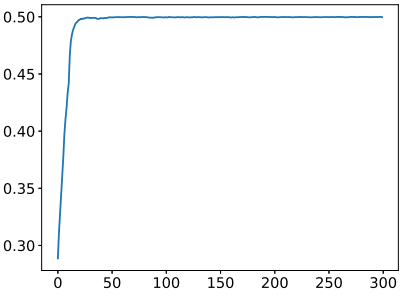
FIGURE 15. Optimized shapes for three possible configurations of the min/max optimization problem (6.9) of Section 6.2.2.



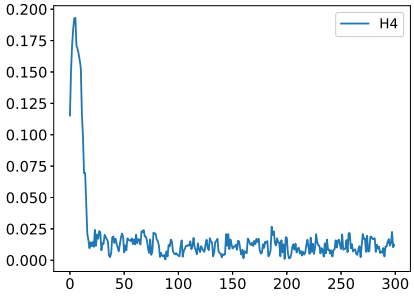
(A) Objective function m. Final value $m = 1.13$.



(B) Compliance $C_i(\Omega)$. Final value: $C_4 = 1.14$.

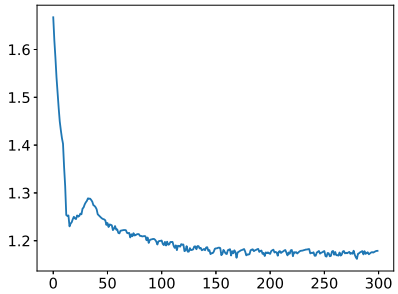


(C) Volume fraction $\mathrm{Vol}(\Omega)/\mathrm{Vol}(D)$. Final value: $\mathrm{Vol}(\Omega)/\mathrm{Vol}(D)$=0.50.

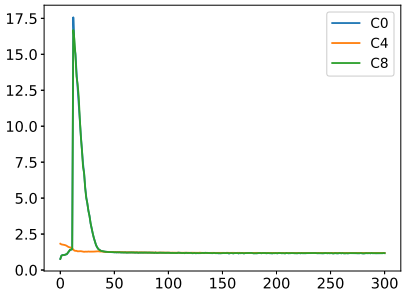


(D) Inequality constraint $H_i = C_i - m$

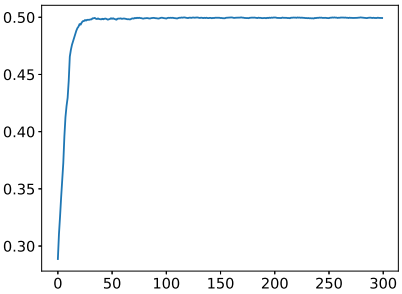
FIGURE 16. Convergence history curves for one load case of Section 6.2.2.



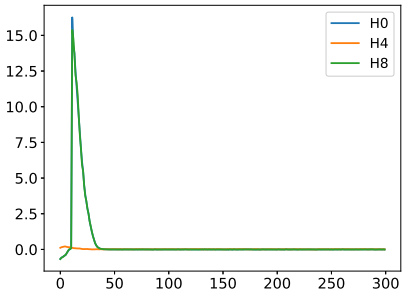
(A) Objective function m. Final value $m = 1.18$.



(B) Compliance $C_i(\Omega)$. Final values: $C_0 = 1.17$, $C_4 = 1.19$, $C_8 = 1.17$.

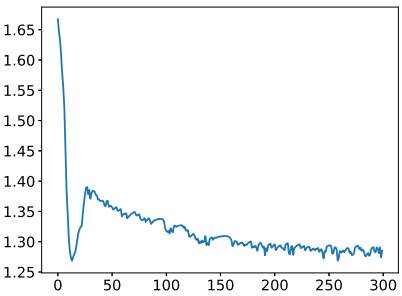


(C) Volume fraction $\mathrm{Vol}(\Omega)/\mathrm{Vol}(D)$. Final value: $\mathrm{Vol}(\Omega)/\mathrm{Vol}(D)$=0.50.

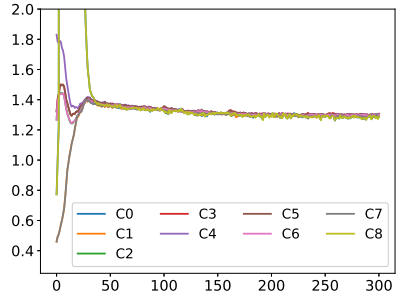


(D) Inequality constraints values: $H_i = C_i - m$.

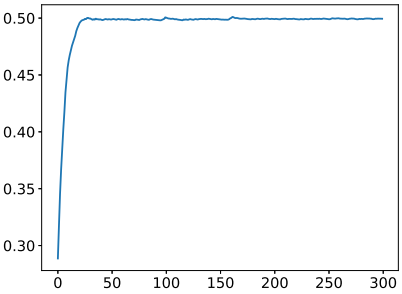
FIGURE 17. Convergence history curves for three load case of Section 6.2.2.



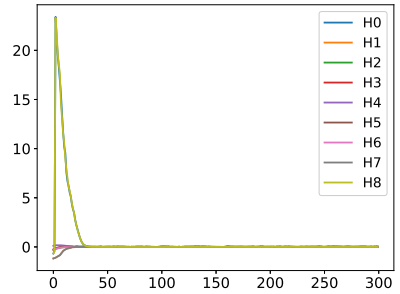
(A) Objective function m (eqn. (6.9))



(B) Compliance values $C_i(\Omega)$. Final values: $C_0 = 1.29$, $C_1 = 1.29$, $C_2 = 1.30$, $C_3 = 1.31$, $C_4 = 1.31$, $C_5 = 1.31$, $C_6 = 1.30$, $C_7 = 1.29$, $C_8 = 1.29$.



(C) Volume fraction $\mathrm{Vol}(\Omega)/\mathrm{Vol}(D)$. Final value: $\mathrm{Vol}(\Omega)/\mathrm{Vol}(D)$=0.50.



(D) Inequality constraints $H_i = C_i - m$

FIGURE 18. Convergence history curves for nine load case of Section 6.2.2.

Appendix A. Proofs and further remarks about trajectory flows

In this section, we provide the proofs of the two theoretical results in the main text devoted to the convergence properties of the flow defined by (1.2). We start with the proof of Proposition 1.

Proof.

(1) Using the definition (2.15) of the flow, the decay property (2.13) together with the fact that $\boldsymbol{\xi}_J(x)$ is orthogonal to $\mathrm{Ker}(\mathrm{D}\boldsymbol{g}(x))$, we obtain:

$$\frac{\mathrm{d}}{\mathrm{d}t}(\boldsymbol{g}(x(t))) = -\alpha_C \boldsymbol{g}(x(t)),$$

whence (2.16) follows easily.

(2) Let us introduce the eigenvalue decomposition

$$\mathrm{D}\boldsymbol{g}(x)\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}} = \sum_{i=1}^{p} \sigma_i(x)^2 \boldsymbol{u}_i(x)\boldsymbol{u}_i(x)^T, \text{ where } \sigma_1(x) \geq \ldots \geq \sigma_p(x) > 0, \ \boldsymbol{u}_i(x)^T\boldsymbol{u}_j(x) = \delta_{ij},$$

of the symmetric, positive definite $p \times p$ matrix $\mathrm{D}\boldsymbol{g}(x)\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}$. Let then $\boldsymbol{v}_i(x)^\dagger : V \to \mathbb{R}$ be the linear form defined for any $\boldsymbol{\xi} \in V$ by $\boldsymbol{v}_i(x)^\dagger\boldsymbol{\xi} = \sigma_i(x)^{-1}\boldsymbol{u}_i(x)^T\mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi}$ and $\boldsymbol{v}_i(x)$ be the vector in V such that $\forall \boldsymbol{\xi} \in V$, $\langle \boldsymbol{v}_i(x), \boldsymbol{\xi}\rangle_V = \boldsymbol{v}_i(x)^\dagger\boldsymbol{\xi}$; more explicitly, $\boldsymbol{v}_i(x) = \sigma_i(x)^{-1}\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{u}_i(x)$. These definitions allow to write a singular value decomposition for $\mathrm{D}\boldsymbol{g}(x)$; it is indeed easily verified from the definitions of $\boldsymbol{u}_i(x)$ and $\boldsymbol{v}_i(x)$ that:

$$\mathrm{D}\boldsymbol{g}(x) = \sum_{i=1}^{p} \sigma_i(x)\boldsymbol{u}_i(x)\boldsymbol{v}_i(x)^\dagger, \text{ and } \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}} = \sum_{i=1}^{p} \sigma_i(x)\boldsymbol{v}_i(x)\boldsymbol{u}_i(x)^T$$

with $\langle \boldsymbol{v}_i(x), \boldsymbol{v}_j(x)\rangle_V = \boldsymbol{v}_i(x)^\dagger\boldsymbol{v}_j(x) = \delta_{ij}$. We now calculate:

$$\mathrm{D}\boldsymbol{g}^{\mathcal{T}}(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\boldsymbol{g}(x) = \sum_{i=1}^{p} \sigma_i^{-1}(x)(\boldsymbol{u}_i(x)^T\boldsymbol{g}(x))\boldsymbol{v}_i(x),$$

whence the following inequality results:

$$\forall x \in V, \ |\mathrm{D}J(x)\mathrm{D}\boldsymbol{g}^{\mathcal{T}}(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\boldsymbol{g}| \leq \sigma_p^{-1}(x)||\nabla J(x)||_V \, ||\boldsymbol{g}(x)||. \tag{A.1}$$

Since

$$\frac{\mathrm{d}}{\mathrm{d}t}J(x(t)) = -\alpha_J \mathrm{D}J(x(t))\boldsymbol{\xi}_J(x(t)) - \alpha_C \mathrm{D}J(x(t))\boldsymbol{\xi}_C(x(t)),$$

it follows that $\frac{\mathrm{d}}{\mathrm{d}t}J(x(t)) < 0$ as soon as $\alpha_J|\mathrm{D}J(x(t))\boldsymbol{\xi}_J(x(t))| > \alpha_C|\mathrm{D}J(x(t))\boldsymbol{\xi}_C(x(t))|$. Thus, (2.18) holds true, and using (2.16) and (A.1), the constant C in there may be selected as

$$C = p\frac{\alpha_C}{\alpha_J}||\boldsymbol{g}(x_0)|| \sup_{x\in K}\left[\sigma_p^{-1}(x)||\nabla J(x)||\right]. \tag{A.2}$$

(3) Since the vectors $\boldsymbol{\xi}_J(x)$ and $\boldsymbol{\xi}_C(x)$ are orthogonal for any point $x \in V$, a stationary point x^* of (2.15) must satisfy

$$\Pi_{g(x^*)}(\nabla J(x^*)) = 0, \text{ and } \mathrm{D}\boldsymbol{g}^{\mathcal{T}}(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\boldsymbol{g}(x^*) = 0, \tag{A.3}$$

and so the first KKT condition in (2.6) is satisfied with the value $\boldsymbol{\lambda} = -(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\mathrm{D}\boldsymbol{g}(x^*)\nabla J(x^*)$ of the Lagrange multiplier. Then left multiplication by $\mathrm{D}\boldsymbol{g}$ in the second identity in (A.3) implies $\boldsymbol{g}(x^*) = 0$, which completes the proof.

□

Remark 9. The solutions to the dynamical system (2.15) are defined for small times if $\boldsymbol{\xi}_J$ and $\boldsymbol{\xi}_C$ are locally Lipschitz vector fields, which is the case if e.g. J and $\boldsymbol{g}$ are of class $\mathcal{C}^2$ [26]. In the case where V is finite-dimensional, the assumption (2.17) is satisfied if the set $K = \{x \in V | \boldsymbol{g}(x) \leq \boldsymbol{g}(x_0)\}$ is bounded and the functions J and $\boldsymbol{g}$ are $\mathcal{C}^1$ functions. It is worth noting that even if not enough regularity assumptions hold to ensure the existence of solutions to the continuous (2.15), similar properties to those of Proposition 1 can be proved for the discretize version

$$x_{n+1} = x_n - \Delta t(\alpha_J\boldsymbol{\xi}_J(x_n) + \alpha_C\boldsymbol{\xi}_C(x_n)), \tag{A.4}$$

which is sufficient for optimization purposes. One can indeed verify that, in the latter context:

(1) At first order, the constraints vanish at a geometric rate: $\boldsymbol{g}(x_{n+1}) = (1 - \alpha_C \Delta t)\boldsymbol{g}(x_n) + o(\Delta t)$.
(2) If x^* is an accumulation point of the sequence $(x_n)_{n\in\mathbb{N}}$, then $\boldsymbol{g}(x^*) = 0$ and x^* is a KKT point of the problem (2.1), satisfying (2.6).

Remark 10. In our design of the update rule (2.15) with (2.7) and (2.8), it is possible to control more accurately the pace at which each of the constraints decreases: let us indeed introduce a diagonal matrix of positive coefficients $\boldsymbol{K} = \mathrm{diag}(\kappa_i)_{1\leq i\leq p}$ and replace the definition (2.8) of $\boldsymbol{\xi}_C(x)$ by

$$\boldsymbol{\xi}_C(x) := \mathrm{D}\boldsymbol{g}^{\mathcal{T}}(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\boldsymbol{K}\boldsymbol{g}(x).$$

Then it can be shown along the lines of the previous discussion that each constraint function g_i decreases at its own rate $\kappa_i\alpha_C$ along the solution $x(t)$ of (2.15):

$$\forall t \in [0, T],\ g_i(x(t)) = e^{-\kappa_i\alpha_C t} g_i(x_0).$$

Now we prove Proposition 5.

Proof.

(1) The definition (3.7) of $\boldsymbol{\xi}_C(x(t))$ implies that $\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x(t))}\boldsymbol{\xi}_C(x(t)) = \boldsymbol{C}_{\widetilde{I}(x(t))}(x(t))$, and since $-\boldsymbol{\xi}_J(x(t))$ is positively proportional to $\boldsymbol{\xi}^*(x(t))$ (Proposition 3), it holds

$$\mathrm{D}\boldsymbol{C}_{\widehat{I}(x(t))}\boldsymbol{\xi}_J(x(t)) = 0,\ -\mathrm{D}\boldsymbol{h}_{\widetilde{I}(x(t))\setminus\widehat{I}(x(t))}(x(t))\boldsymbol{\xi}_J(x(t)) \leq 0.$$

Therefore we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{C}_{\widehat{I}(x(t))}(x(t)) = -\alpha_C\boldsymbol{C}_{\widehat{I}(x(t))}(x(t)) \text{ and } \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{h}_{\widetilde{I}(x(t))\setminus\widehat{I}(x(t))}(x(t)) \leq -\alpha_C\boldsymbol{h}_{\widetilde{I}(x_0)\setminus\widehat{I}(x_0)}(x(t)) \tag{A.5}$$

whence (3.27) follows by application of Gronwall's lemma.
(2) The proof is identical to that of Proposition 1.
(3) A stationary point x^* of (3.26) satisfies by definition

$$-\alpha_J\boldsymbol{\xi}_J(x^*) - \alpha_C\boldsymbol{\xi}_C(x^*) = 0. \tag{A.6}$$

Left multiplication of this identity by $\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x^*)}(x^*)$ yields:

$$-\alpha_J\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x^*)}(x^*)\boldsymbol{\xi}_J(x^*) - \alpha_C\boldsymbol{C}_{\widetilde{I}(x^*)}(x^*) = 0. \tag{A.7}$$

Remembering now that from definition (3.8),

$$-\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x^*)}\boldsymbol{\xi}_J(x^*) \leq 0 \text{ and } \boldsymbol{C}_{\widetilde{I}(x^*)}(x^*) \geq 0,$$

equality in (A.7) can hold only if both terms vanish. In particular, we infer that $\boldsymbol{C}_{\widetilde{I}(x^*)}(x^*) = 0$, a fact which implies $\boldsymbol{\xi}_C(x^*) = 0$ and which encompasses the last two lines of the KKT conditions (3.6). Returning to (A.6), we obtain that $\boldsymbol{\xi}_J(x^*) = 0$, which is the first line in (3.6). This completes the proof.

□

Remark 11. The assumption (a) in Proposition 5, whereby the index set $\widetilde{I}(x(t))$ remains constant is essentially made to ensure that the right-hand side of the flow (3.26) is continuous. Indeed, in such a case, the range space direction $\boldsymbol{\xi}_C(x(t))$ is continuous by its definition (3.7), while the null space step $\boldsymbol{\xi}_J(x(t))$ is continuous because

$$\boldsymbol{\xi}_J(x(t)) = \nabla J(x(t)) + \mathrm{D}\boldsymbol{C}_{\widetilde{I}(x(t))}\begin{bmatrix}\boldsymbol{\lambda}^*(x(t))\\ \boldsymbol{\mu}^*(x(t))\end{bmatrix}$$

and it can be shown that the multipliers $(\boldsymbol{\lambda}^*(x(t)), \boldsymbol{\mu}^*(x(t)))$ defined by (3.9) are continuous functions. When the sets $\widetilde{I}(x)$ or $\widehat{I}(x)$ are subject to change (corresponding to inequality constraints becoming active or inactive), the ODE (3.26) has a discontinuous right-hand side and is only defined formally; we conjecture a rigorous mathematical meaning could still be provided in a weaker sense with the theory of non smooth ODEs, see e.g. [25, 33] or [11]. At a time T corresponding to a sudden change of the index set $\widetilde{I}(x(t))$, we assume that the solution $x(t)$ can be extended by restarting the ODE (3.26) with the new index set $\widetilde{I}(x(T))$. Under this

circumstance and by construction of $\boldsymbol{\xi}_J(x(t))$ and $\boldsymbol{\xi}_C(x(t))$, the bound $\boldsymbol{h}_{\widetilde{I}(x_0)}(x(t)) \leq e^{-\alpha_C t}\boldsymbol{h}_{\widetilde{I}(x_0)}(x(0))$ still holds while the other constraints remain saturated or satisfied for $t \geq T$: $\boldsymbol{h}_{\widetilde{I}(x(t))\setminus\widetilde{I}(x_0)} = 0$. Hence, assuming this procedure can be extended for all times, all constraints are asymptotically satisfied. Properties (2) and (3) then remain true, up to an adjustment of the constant C in (3.29) (which can be taken global since there are finitely many possible sets $\widetilde{I}(x(t))$). There might exist situations where the set of saturated constraints $\widetilde{I}(x(t))$ could oscillate indefinitely. However (2) states that $x(t)$ always keeps improving (in the sense of (3.29)), and (3) states that if $x(t)$ eventually converges, it is necessarily towards a KKT point.

Remark 12. Assuming the index set $\widehat{I}(x(t))$ eventually becomes stationary (which is expected to be the case under suitable regularity assumptions on the admissible set), the ODE (3.26) reduces to the one considered in [51, 65] involving only equality constraints. In such circumstances, it is possible to prove that all limit points of $x(t)$ are stationary points (satisfying in our case the KKT condition (3.30)), see Theorem 2.3 of [51].

Remark 13. In practice, the analysis of Proposition 5 is sufficient because, similarly to the conclusions of Remark 9, analogous properties hold for the discrete scheme

$$x_{n+1} = x_n - \Delta t\big(\alpha_J \boldsymbol{\xi}_J(x_n) + \alpha_C \boldsymbol{\xi}_C(x_n)\big). \tag{A.8}$$

Indeed, one can easily check that:

(1) Up to first order, the violation of the constraints vanish at a geometric rate:

$$\boldsymbol{C}(x_{n+1}) = (1 - \alpha_C \Delta t)\boldsymbol{C}(x_n) + o(\Delta t). \tag{A.9}$$

This suggests that in order to obtain a stable scheme, one must a priori select α_C and Δt such that $0 < \alpha_C \Delta t < 2$.

(2) If x^* is an accumulation point of the sequence $(x_n)_{n\in\mathbb{N}}$, then x^* is feasible, i.e. $\boldsymbol{C}_{\widetilde{I}(x^*)}(x^*) = 0$ and it is a KKT point for (1.1).

A flexibility of this ODE approach is that at the continuous level, the results of Proposition 5 do not depend on the values of the parameters $\alpha_J > 0$ and $\alpha_C > 0$. Therefore the convergence of the discrete scheme towards the continuous trajectory should hold as soon as the discretization step size $\Delta t > 0$ is sufficiently small. These arguments however remain formal: a rigorous statement is out of the scope of this work and would require a careful analysis and suitable regularity assumptions; see e.g. [43] for an instance of work discussing the relations between the properties of continuous ODEs and those of their associated discretization schemes.

Acknowledgements. This work was supported by the Association Nationale de la Recherche et de la Technologie (ANRT) [grant number CIFRE 2017/0024] and by the project ANR-18-CE40-0013 SHAPO financed by the French Agence Nationale de la Recherche (ANR). F.F. is a CIFRE PhD student, funded by SAFRAN, the support of which is kindly acknowledged. G. A. is a member of the DEFI project at INRIA Saclay Ile-de-France. The work of C.D. is partially supported by the IRS-CAOS grant from Université Grenoble-Alpes. We thank Alexis Faure for sharing his optimization experience with us.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*, Princeton University Press, 2009.

[2] P.-A. Absil and J. Malick, *Projection-like retractions on matrix manifolds*, SIAM Journal on Optimization, 22 (2012), pp. 135–158.

[3] G. Allaire, *Conception optimale de structures, volume 58 of Mathématiques & Applications (Berlin)[Mathematics & Applications]*, Springer-Verlag, Berlin, 2007.

[4] G. Allaire, C. Dapogny, R. Estevez, A. Faure, and G. Michailidis, *Structural optimization under overhang constraints imposed by additive manufacturing technologies*, Journal of Computational Physics, 351 (2017), pp. 295–328.

[5] G. Allaire, C. Dapogny, and P. Frey, *Shape optimization with a level set based mesh evolution method*, Computer Methods in Applied Mechanics and Engineering, 282 (2014), pp. 22–53.

[6] G. Allaire and F. Jouve, *Minimum stress optimal design with the level set method*, Engineering analysis with boundary elements, 32 (2008), pp. 909–918.

[7] G. Allaire, F. Jouve, and G. Michailidis, *Casting constraints in structural optimization via a level-set method*, in 10th world congress on structural and multidisciplinary optimization, 2013.

[8] ———, *Thickness control in structural optimization via a level set method*, Structural and Multidisciplinary Optimization, 53 (2016), pp. 1349–1382.

[9] G. Allaire, F. Jouve, and A.-M. Toader, *Structural optimization using sensitivity analysis and a level-set method*, Journal of computational physics, 194 (2004), pp. 363–393.

[10] G. Allaire and O. Pantz, *Structural optimization with freefem++*, Structural and Multidisciplinary Optimization, 32 (2006), pp. 173–181.

[11] L. Ambrosio, M. Colombo, and A. Figalli, *Existence and uniqueness of maximal regular flows for non-smooth vector fields*, Archive for Rational Mechanics and Analysis, 218 (2015), pp. 1043–1081.

[12] M. Andersen, J. Dahl, and L. Vandenberghe, *CVXOPT: A Python package for convex optimization*, Available at http://cvxopt.org/, (2012).

[13] S. Arguillère, E. Trélat, A. Trouvé, and L. Younes, *Shape deformation analysis from the optimal control viewpoint*, J. Math. Pures Appl. (9), 104 (2015), pp. 139–178.

[14] H. Azegami and Z. C. Wu, *Domain optimization analysis in linear elastic problems: approach using traction method*, JSME international journal. Ser. A, Mechanics and material engineering, 39 (1996), pp. 272–278.

[15] C. Barbarosie and S. Lopes, *A gradient-type algorithm for optimization with constraints*, submitted for publication, see also Pre-Print CMAF Pre-2011-001 at http://cmaf. ptmat. fc. ul. pt/preprints. html, (2011).

[16] C. Barbarosie, A.-M. Toader, and S. Lopes, *A gradient-type algorithm for constrained optimization with application to microstructure optimization*, Discrete & Continuous Dynamical Systems-B, 22 (2017), p. 0.

[17] J.-F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal, *Numerical optimization: theoretical and practical aspects*, Springer Science & Business Media, 2006.

[18] H. Brezis, *Functional analysis, Sobolev spaces and partial differential equations*, Springer Science & Business Media, 2010.

[19] R. Bro and S. De Jong, *A fast non-negativity-constrained least squares algorithm*, Journal of Chemometrics: A Journal of the Chemometrics Society, 11 (1997), pp. 393–401.

[20] C. Bui, C. Dapogny, and P. Frey, *An accurate anisotropic adaptation method for solving the level set advection equation*, International Journal for Numerical Methods in Fluids, 70 (2012), pp. 899–922.

[21] M. Burger, *A framework for the construction of level set methods for shape optimization and reconstruction*, Interfaces and Free boundaries, 5 (2003), pp. 301–329.

[22] C. Dapogny, C. Dobrzynski, and P. Frey, *Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems*, Journal of computational physics, 262 (2014), pp. 358–378.

[23] C. Dapogny, P. Frey, F. Omnès, and Y. Privat, *Geometrical shape optimization in fluid mechanics using FreeFem++*, Structural and Multidisciplinary Optimization, (2017), pp. 1–28.

[24] F. De Gournay, *Velocity extension for the level-set method and multiple eigenvalues in shape optimization*, SIAM journal on control and optimization, 45 (2006), pp. 343–367.

[25] L. Dieci and L. Lopez, *A survey of numerical methods for ivps of odes with discontinuous right-hand side*, Journal of Computational and Applied Mathematics, 236 (2012), pp. 3967–3991.

[26] J. Dieudonné, *Foundations of modern analysis*, Academic press, New York and London, 1960.

[27] P. D. Dunning and H. A. Kim, *Introducing the sequential linear programming level-set method for topology optimization*, Structural and Multidisciplinary Optimization, 51 (2015), pp. 631–643.

[28] P. Duysinx and M. P. Bendsøe, *Topology optimization of continuum structures with local stress constraints*, International journal for numerical methods in engineering, 43 (1998), pp. 1453–1478.

[29] A. Edelman, T. A. Arias, and S. T. Smith, *The geometry of algorithms with orthogonality constraints*, SIAM journal on Matrix Analysis and Applications, 20 (1998), pp. 303–353.

[30] A. Faure, *Optimisation de forme de matériaux et structures architecturés par la méthode des lignes de niveaux avec prise en compte des interfaces graduées*, PhD thesis, Grenoble Alpes, 2017.

[31] F. Feppon, *Shape and topology optimization of multiphysics systems*, PhD thesis, Thèse de doctorat de l'Université Paris Saclay préparée à l'École polytechnique, 2019.

[32] F. Feppon, G. Allaire, F. Bordeu, J. Cortial, and C. Dapogny, *Shape optimization of a coupled thermal fluid-structure problem in a level set mesh evolution framework*, SeMA, 76 (2019), pp. 413–458.

[33] A. F. Filippov, *Differential equations with discontinuous righthand sides: control systems*, vol. 18, Springer Science & Business Media, 2013.

[34] R. Fletcher, *Practical methods of optimization*, John Wiley & Sons, 2013.

[35] A. Henrot and M. Pierre, *Shape variation and optimization*, vol. 28 of EMS Tracts in Mathematics, European Mathematical Society (EMS), Zürich, 2018. A geometrical analysis, English version of the French publication [MR2512810] with additions and updates.

[36] M. Hintermüller, K. Ito, and K. Kunisch, *The primal-dual active set strategy as a semismooth newton method*, SIAM Journal on Optimization, 13 (2002), pp. 865–888.

[37] H. T. Jongen and O. Stein, *On the complexity of equalizing inequalities*, Journal of Global Optimization, 27 (2003), pp. 367–374.

[38] H. T. Jongen and O. Stein, *Constrained global optimization: adaptive gradient flows*, in Frontiers in global optimization, vol. 74 of Nonconvex Optim. Appl., Kluwer Acad. Publ., Boston, MA, 2004, pp. 223–236.

[39] J. Jost, *Chapter 7 Morse Theory and Floer Homology*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 327–417.

[40] C. Le, J. Norato, T. Bruns, C. Ha, and D. Tortorelli, *Stress-based topology optimization for continua*, Structural and Multidisciplinary Optimization, 41 (2010), pp. 605–620.

[41] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, *Gradient descent only converges to minimizers*, in 29th Annual Conference on Learning Theory, V. Feldman, A. Rakhlin, and O. Shamir, eds., vol. 49 of Proceedings of Machine Learning Research, Columbia University, New York, New York, USA, 23–26 Jun 2016, PMLR, pp. 1246–1257.
[42] J. Liu and Y. Ma, *A survey of manufacturing oriented topology optimization methods*, Advances in Engineering Software, 100 (2016), pp. 161–175.
[43] L. Ljung, *Analysis of recursive stochastic algorithms*, IEEE transactions on automatic control, 22 (1977), pp. 551–575.
[44] D. G. Luenberger, *The gradient projection method along geodesics*, Management Sci., 18 (1972), pp. 620–631.
[45] B. Mohammadi and O. Pironneau, *Applied shape optimization for fluids*, Oxford University Press, 2010.
[46] P. Morin, R. Nochetto, M. Pauletti, and M. Verani, *Adaptive sqp method for shape optimization*, in Numerical Mathematics and Advanced Applications 2009, Springer, 2010, pp. 663–673.
[47] F. Murat and J. Simon, *Sur le contrôle par un domaine géométrique, publications du Laboratoire d'Analyse Numérique*, Université Pierre et Marie Curie, (1976).
[48] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer Science, 35 (1999).
[49] S. Osher and J. A. Sethian, *Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations*, Journal of computational physics, 79 (1988), pp. 12–49.
[50] I. Panageas and G. Piliouras, *Gradient descent only converges to minimizers: Non-isolated critical points and invariant regions*, arXiv preprint arXiv:1605.00405, (2016).
[51] J. Schropp and I. Singer, *A dynamical systems approach to constrained minimization*, Numerical functional analysis and optimization, 21 (2000), pp. 537–551.
[52] V. H. Schulz, *A Riemannian view on shape optimization*, Found. Comput. Math., 14 (2014), pp. 483–501.
[53] V. H. Schulz, M. Siebenborn, and K. Welker, *Efficient pde constrained shape optimization based on steklov–poincaré-type metrics*, SIAM Journal on Optimization, 26 (2016), pp. 2800–2819.
[54] V. Shikhman and O. Stein, *Constrained optimization: projected gradient flows*, Journal of optimization theory and applications, 140 (2009), pp. 117–130.
[55] O. Sigmund, *Manufacturing tolerant topology optimization*, Acta Mechanica Sinica, 25 (2009), pp. 227–239.
[56] J. Sokolowski and J.-P. Zolesio, *Introduction to shape optimization*, in Introduction to Shape Optimization, Springer, 1992, pp. 5–12.
[57] J. Sokolowski and J.-P. Zolésio, *Introduction to shape optimization*, vol. 16 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1992. Shape sensitivity analysis.
[58] K. Svanberg, *The method of moving asymptotes—a new method for structural optimization*, Internat. J. Numer. Methods Engrg., 24 (1987), pp. 359–373.
[59] K. Tanabe, *A geometric method in nonlinear programming*, Journal of Optimization Theory and Applications, 30 (1980), pp. 181–210.
[60] G. N. Vanderplaats and F. Moses, *Structural optimization by methods of feasible directions*, Computers & Structures, 3 (1973), pp. 739–755.
[61] A. Wächter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Mathematical programming, 106 (2006), pp. 25–57.
[62] M. Y. Wang, X. Wang, and D. Guo, *A level set method for structural topology optimization*, Computer methods in applied mechanics and engineering, 192 (2003), pp. 227–246.
[63] Q. Xia, T. Shi, M. Y. Wang, and S. Liu, *A level set based method for the optimization of cast part*, Structural and Multidisciplinary Optimization, 41 (2010), pp. 735–747.
[64] Q. Xia and M. Y. Wang, *Topology optimization of thermoelastic structures using level set method*, Computational Mechanics, 42 (2008), pp. 837–857.
[65] H. Yamashita, *A differential equation approach to nonlinear programming*, Mathematical Programming, 18 (1980), pp. 155–168.
[66] Y.-x. Yuan, *A review of trust region algorithms for optimization*, in Iciam, vol. 99, Citeseer, 2000, pp. 271–282.
[67] M. Yulin and W. Xiaoming, *A level set method for structural topology optimization with multi-constraints and multi-materials*, Acta Mechanica Sinica, 20 (2004), pp. 507–518.
[68] G. Zoutendijk, *Methods of feasible directions: A study in linear and non-linear programming*, Elsevier Publishing Co., Amsterdam-London-New York-Princeton, N.J., 1960.