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 generic equality or inequality constrained optimization problems, which is suited for shape optimization applications. We rely on a variant of the Ordinary Differential Equation (ODE) approach proposed by Yamashita [48] for equality constrained problems: the search direction is a combination of a null space step and a range space step, which are aimed to reduce 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, and which allows to identify the inequality constraints which should remain tangent to the optimization trajectory. Our second contribution is a formulation of our gradient flow in the context of—infinite-dimensional—Hilbert space settings. This allows to extend the method to quite general optimization sets equipped with a suitable manifold structure, and notably to sets of shapes as it occurs in shape optimization with the framework of Hadamard's boundary variation method. The cornerstone of this latter setting is the classical operation of extension and regularization of shape derivatives. Some numerical comparisons on simple academic examples are performed to illustrate the behavior of our algorithm. Its numerical efficiency and ease of implementation are finally demonstrated on more 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

1. **Introduction** 2
2. **Gradient flows for equality-constrained optimization in Hilbert spaces** 5
   2.1. Notations and first-order optimality conditions 5
   2.2. Definitions and properties of the null space and range space steps $\boldsymbol{\xi}_J$ and $\boldsymbol{\xi}_C$ 6
   2.3. Behavior of the trajectories of the flow 9
3. **Proposed extension to equality and inequality constraints** 11
   3.1. Notations and preliminaries 11
   3.2. The method of slack variables for inequality constraints 12
   3.3. The proposed algorithm 13
   3.4. Comparison between the proposed method and the use of slack variables 20
4. **Practical implementation details** 21
   4.1. Time step adaptation based on a merit function. 21
   4.2. Accounting for discontinuities near the inequality constraint barriers 22
5. **Illustrations and comparisons on academic test cases** 23

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

5.1. Test case 1 : unfeasible initialization with initial gradient aligned with the constraints. 24
5.2. Test case 2 : unfeasible initialization with initial gradient not aligned with the constraints. 25
5.3. Test case 3: a saturated inequality constraint becoming inactive along the optimization path 27
6. **Optimization on smooth manifolds: application to shape optimization** 29
6.1. Hadamard's framework for gradient based shape optimization 30
6.2. Implementation of the constrained gradient flow for level set based shape optimization 32
6.3. Illustrations on a multiple load structural shape optimization test case 33
7. **Conclusion and perspectives** 38
References 41

1. Introduction

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

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

where $\mathcal{X}$ is the optimization set, $J : \mathcal{X} \to \mathbb{R}$ is a differentiable objective function, $\boldsymbol{g} : \mathcal{X} \to \mathbb{R}^p$ and $\boldsymbol{h} : \mathcal{X} \to \mathbb{R}^q$ are differentiable functions accounting for p equality and q inequality differentiable 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) [14, 36, 49, 29], the Method of Moving Asymptotes (MMA) [44], the Method of Feasible Directions [51]. When dealing with a given application, the choice of one particular optimization method is partly determined by the difficulty of implementation. As is often the case in constrained optimization, it turns out that all the aforementioned techniques require fine tuning of the algorithm parameters in order to achieve convergence with a reasonable computational efficiency; 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 a slightly different spirit, a significant amount of work has been devoted to dynamical system approaches for addressing problems of the form (1.1): a solution is reached as the stationary point x^* of a continuous trajectory $x(t)$, solving a suitable Ordinary Differential Equation (ODE). When $\mathcal{X} = \mathbb{R}^k$ is a finite-dimensional vector space and in the particular, unconstrained case where constraints are omitted in (1.1), the most basic of these is the celebrated gradient flow:

$$\dot{x}(t) = -\nabla J(x(t)). \tag{1.2}$$

The value of the objective $t \mapsto J(x(t))$ is guaranteed to decrease along the optimization path, and the Euler step size Δt, which is the unique parameter involved in the discretization of (1.2), can always be made sufficiently small so that a reduction of the value of the objective function is observed. In this sense, such dynamical system approaches can be expected to be more reliable than the aforementioned iterative algorithms, although they might not achieve the fastest rate of convergence. Extensions of the standard gradient flow (1.2) have been proposed to account for equality constraints $\boldsymbol{g}(x) = 0$. For instance, Tanabe [45] proposed to replace the gradient $\nabla J(x)$ in (1.2) by its tangential projection $\boldsymbol{\xi}_J(x)$ onto the feasible set. Yamashita [48] suggested to add a Gauss-Newton direction $\boldsymbol{\xi}_C(x)$ to this ODE in order to smoothly lead the optimization path toward the feasible domain; the resulting dynamical system for equality constrained optimization reads:

$$\dot{x}(t) = -\alpha_J \boldsymbol{\xi}_J(x(t)) - \alpha_C \boldsymbol{\xi}_C(x(t)), \tag{1.3}$$

where $\boldsymbol{\xi}_J$ and $\boldsymbol{\xi}_C$ are respectively defined by

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

$$\boldsymbol{\xi}_C(x) = \mathrm{D}\boldsymbol{g}^T(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^T)^{-1}\boldsymbol{g}(x), \tag{1.5}$$

with I the identity mapping and $(\mathrm{D}\boldsymbol{g}(x))_{ij} = \partial_j g_i(x)$ the Jacobian matrix of the constraint function $\boldsymbol{g}(x)$ (we shall omit the dependence with respect to x when the context is clear). The descent direction $\dot{x}$ is a combination of the so-called 'null space' direction $\boldsymbol{\xi}_J(x) \in \mathrm{Ker}(\mathrm{D}\boldsymbol{g}(x))$ and 'range space' direction $\boldsymbol{\xi}_C(x) \in \mathrm{span}(\mathrm{D}\boldsymbol{g}^T(x))$, lying respectively in the null space of the constraints and in its orthogonal complement. For this reason, we call the ODE (1.3) a 'null space' gradient flow. In (1.3), $\alpha_J, \alpha_C > 0$ are two (facultative) parameters that we choose to introduce in order to scale how fast the objective function should be decreased which respect to the violation of the constraints. This decomposition of the descent direction ensures that the constraints vanish at an exponential rate, namely $\boldsymbol{g}(x(t)) = e^{-\alpha_C t}\boldsymbol{g}(x_0)$, while the objective function decreases along the direction $-\boldsymbol{\xi}_J(x(t))$ (see [48] and Proposition 1 hereafter). The discretization of the flow (1.3) can be related to SQP (see [38]) and to null space iterative methods [13, 14, 36] (which, to our knowledge, apply only to equality-constrained optimization).

When it comes to handling inequality constraints, the most common approach consists in introducing q slack variables $\{z_i\}_{1\leq i\leq q} \in \mathbb{R}^q$ so as to convert 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 then solving the ODE (1.3) in the augmented space $(x, z) \in \mathbb{R}^k \times \mathbb{R}^q$. It is even possible to eliminate the variables z_i from the latter ODE if the initialization is feasible, see [32, 41]. Schropp and Singer showed in [38] that stationary points of the resulting dynamical system fulfill partially the Karush Kuhn Tucker (KKT), necessary first-order optimality condition: at such a stationary point x^*, there exist multipliers $(\boldsymbol{\lambda}(x^*), \boldsymbol{\mu}(x^*)) \in \mathbb{R}^p \times \mathbb{R}^q$ such that

$$\nabla J(x^*) + \mathrm{D}\boldsymbol{g}(x^*)^T\boldsymbol{\lambda}(x^*) + \mathrm{D}\boldsymbol{h}(x^*)^T\boldsymbol{\mu}(x^*) = 0, \text{ and } h_i(x^*)\mu_i(x^*) = 0 \text{ for } i = 1, ..., q.$$

However, the complete set of KKT conditions may not be satisfied because of possible negative values of some components of the Lagrange multiplier $\mu_i(x^*) < 0$. Nevertheless, this strategy proves efficient in practice because among all possible stationary points, only those satisfying the complete KKT conditions are asymptotically stable; see again [38] about this point.

The contributions of the present article are twofold. First, we propose an extension of the flow (1.3) which is able to account for inequality constraints without the need to increase the size of the problem with auxiliary slack variables. Our approach relies on a suitable dual program for the resolution of the combinatorial character of the inequality constraints: we precisely identify to which subset of the active inequality constraints the optimization path is allowed to 'unstick' (thus re-entering into the feasible domain) and, conversely, to which inequality constraints it must remain tangent. The resulting dynamical system is slightly more robust than those based on slack variables, since we shall see that all its stationary points are true KKT points of (1.1), and conversely. More specifically, for a given subset of indices $I \subset \{1, \ldots, q\}$, denote $\boldsymbol{h}_I(x) := (h_i(x))_{i\in I}$ the corresponding inequality constraints and $\boldsymbol{C}_I(x)$ the matrix

$$\boldsymbol{C}_I(x) := \begin{bmatrix} \boldsymbol{g}(x) \\ \boldsymbol{h}_I(x) \end{bmatrix}. \tag{1.6}$$

Then, for inequality constrained problems, we define new directions $\boldsymbol{\xi}_J(x)$ and $\boldsymbol{\xi}_C(x)$ in (1.3) as follows:

$$\boldsymbol{\xi}_J(x) = (\mathrm{I} - \mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^T(\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^T)^{-1}\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)})\nabla J(x), \tag{1.7}$$

$$\boldsymbol{\xi}_C(x) = \mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}^T(\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}^T)^{-1}\boldsymbol{C}_{\widetilde{I}(x)}(x), \tag{1.8}$$

which involve two different subsets $\widehat{I}(x) \subset \widetilde{I}(x) \subset \{1, ..., q\}$: $\widetilde{I}(x)$ is the set of all saturated or violated constraints, defined by

$$\widetilde{I}(x) = \{i \in \{1, \ldots, q\} \,|\, h_i(x) \geq 0\}, \tag{1.9}$$

and $\widehat{I}(x)$ is a subset of $\widetilde{I}(x)$ identifying the constraints onto which the gradient of the objective function is projected tangentially. The set $\widehat{I}(x)$ shall be obtained by solving a dual quadratic optimization subproblem (equation (3.13) below) of the size of $\widetilde{I}(x)$. As detailed later in Proposition 4, this dual problem ensures that $\boldsymbol{\xi}_J(x)$ is the projection of the gradient $\nabla J(x)$ onto the cone of feasible directions. As a result, $-\boldsymbol{\xi}_J(x)$ is

always the best possible descent direction respecting locally both equality and inequality constraints. Since the sets $\widetilde{I}(x)$ or $\widehat{I}(x)$ are subject to change as soon as inequality constraints become active or inactive, or if not enough regularity holds, the ODE (1.3) has a discontinuous right-hand side and is defined only formally (note that a rigorous mathematical meaning could still be provided with the theory of non smooth ODEs, see [21, 28]). However and as we shall detail further on, its discretization makes sense and exhibits the same decreasing properties as its continuous counterpart for sufficiently small steps Δt.

Our second main contribution is the exposure of our dynamical system strategy in a setting that can handle quite general ranges of infinite-dimensional optimization sets, including for instance those involved in non parametric shape and topology optimization, which is our final goal. Here, a clear distinction needs to be made between the Fréchet derivative and the gradient of the objective and constraint functionals: if $\mathcal{X} = V$ is a Hilbert space, recall that the gradient is obtained by identifying the differential (a linear form on V) to an element in V via the Riesz representation theorem; see Definition 1. In order to account for the infinite dimensionality, we shall specify clearly how (1.7) and (1.8) must be computed with respect to such identification and used when the optimization set $\mathcal{X}$ is a Hilbert space or a more general set equipped with a suitable manifold structure. As we have mentioned, our ultimate motivations actually originate from the field of shape optimization based on the method of Hadamard, for which the minimization set $\mathcal{X}$ in (1.1) is the set of all possible open Lipschitz subdomains Ω enclosed in some 'hold-all' domain $D \subset \mathbb{R}^d$:

$$\mathcal{X} = \{\Omega \subset D \,|\, \Omega \text{ Lipschitz}\}. \tag{1.10}$$

This set is not a vector space, but it can be locally parameterized by the Sobolev space $W^{1,\infty}(D, \mathbb{R}^d)$. In order to minimize a shape functional $\Omega \mapsto J(\Omega)$, we determine the best local variation of the form $(\mathrm{I} + \boldsymbol{\theta})(\Omega)$ where $\boldsymbol{\theta} \in W^{1,\infty}(D, \mathbb{R}^d)$ can be interpreted as a (sufficiently small) displacement field [35, 42, 30]. As we shall review later on in Section 6.1, this endows $\mathcal{X}$ with a manifold structure which allows for gradient based optimization [6, 47]. In shape optimization, the identification of the gradient is achieved by solving an extension and regularization problem, which has some very important consequences in numerical algorithms, see e.g. [17, 19]. We shall detail below how this step is naturally and consistently included in our algorithm, a matter which so far does not seem fully clear in the literature concerned with constrained shape optimization: common approaches rather compute a descent direction *first*, before performing a regularization, see e.g. [23, 25].

In the open academic literature on shape optimization based on Hadamard's method, advanced mathematical programming methods are not frequently described. Rather, for simplicity of implementation, Penalty and Augmented Lagrangian Methods are often used, all the more when only one constraint is considered [6, 18]. Morin et. al. introduced a variant of SQP in [34] but treated a volume constraint with a Lagrange Multiplier method. For more complex applications, some authors have introduced adapted variants of Sequential Linear Programming [23] or of the Method of Feasible Direction [25]. However, the high dimensionality and complexity of the research space leaves very little intuition on how to select appropriately the parameter values featured in the implementation of these latter algorithms. As a result, a fair amount of trials and errors is often required in order to obtain satisfying minimizing sequences of shapes—a process which can be very time consuming, especially when every optimization step depends on the resolution of some physical model involving partial differential equations. As a result of the flexibility of the ODE approaches, our method depends truly only on the discretization step Δt, and on the physically interpretable dimensionless parameters α_J, α_C, which makes them relatively easy to tune for the user.

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. [12], who derived an iterative algorithm for equality constrained optimization which turns out to be a discretization of (1.3) with a variable scaling for the parameter α_C. For inequality constraints, the authors proposed (without convergence results) an active set strategy also based on the extraction of an appropriate subset of the active constraints. However their method relies on a different algorithm from ours, that yields generally a different set than $\widehat{I}(x)$, see Remark 4 below for more details. Finally, Yulin and Xiaoming [50] also suggested to project the gradient of the objective function onto the convex cone of feasible directions; nevertheless, they remained elusive regarding how the projection problem is solved or how violated constraints are tackled.

The present article is organized as follows. In Section 2, we review the definition and the properties of the gradient flow (1.3) for equality constrained optimization in the case where the minimisation set $\mathcal{X}$ 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 the "null space" gradient flow (1.3) towards points satisfying the full KKT condition. We detail algorithmic implementation aspects in Section 4. In Section 5, we provide pedagogical illustrations of our method on simple academic test cases, and we compare it to the method of slack variables for inequality constraints. We finally focus on shape optimization applications in Section 6. After clarifying the necessary adaptations required to extend the discretization of (1.3) to the set $\mathcal{X}$ defined by (1.10), we explain how our algorithm can be integrated within the level set method for shape optimization [47, 6, 5]. We then implement and demonstrate numerically this method on the optimal design of a bridge structure subject to multiple loads, which involves up to ten constraints. A conclusion and several perspectives are outlined in Section 7.

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

In this section, we consider the case where the optimization takes place on a Hilbert space $\mathcal{X} = V$ with inner product $a(\cdot,\cdot)$ and relative norm $||\cdot||_V = a(\cdot,\cdot)^{1/2}$; see Section 6 for the description of the more general situation associated to our shape optimization applications. The first focus of our study is the minimization problem (1.1) 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 (1.3) can be extended to the present Hilbertian setting. Let us emphasize that, although this section is elementary and not completely new, it is not easily found as is in the literature. Since it is key in understanding our technique for handling inequality constraints in Section 3, the present context is thoroughly detailed for the reader's convenience.

The section is organized as follows. We first recall the definitions of the differential, the gradient, and the transpose of the differential in the Hilbertian context. We then sketch briefly how the formulas (1.4) and (1.5) can be formally obtained. We state the properties of the null space step $\boldsymbol{\xi}_J(x)$ and its relation to Lagrange multipliers by means of a dual problem in Lemma 1. Finally, we review the decrease properties of the obtained dynamical system in Proposition 1.

2.1. Notations and first-order optimality conditions

We start by setting notations about differentiability and gradients in Hilbert spaces that we use throughout this article. Our notations may differ from those used by other authors because we need a clear distinction between gradient and Fréchet derivatives.

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 [15] ensures the existence of a unique vector $\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\mu} \in V$ satisfying

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

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 differentiable at $x \in V$, the Riesz representation theorem ensures the existence of a unique vector $\nabla J(x) \in V$ satisfying

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

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

Throughout the paper, we shall sometimes omit the explicit mention to x in the notations 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 $\mathbb{R}^k$, equipped with the standard inner product, the Fréchet derivative and the transpose of a vector function $\boldsymbol{g} : \mathbb{R}^k \to \mathbb{R}^p$ are respectively given by the Jacobian matrix $(\mathrm{D}\boldsymbol{g})_{ij} = \partial_j g_i$ and its transpose $(\mathrm{D}\boldsymbol{g}^{\mathcal{T}})_{ij} = (\mathrm{D}\boldsymbol{g}^T)_{ij} = \partial_i g_j$. 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 to denote the gradient of scalar functions $J : V \to \mathbb{R}$, and it holds that $\nabla J(x) = \mathrm{D}J(x)^{\mathcal{T}}1$. The curly 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.

(2) Still in the case where $V = \mathbb{R}^k$ is finite-dimensional and a is given by a symmetric definite positive matrix A (that is $a(\boldsymbol{\xi}, \boldsymbol{\xi}) = \boldsymbol{\xi}^T A \boldsymbol{\xi}$), the transpose of a $p \times k$ matrix $M : \mathbb{R}^k \to \mathbb{R}^p$ with respect to a is $M^{\mathcal{T}} = A^{-1}M^T$. As we shall see in Section 6, in shape optimization applications, a often stands for the bilinear form associated to an elliptic operator, hence the calligraphic transpose $\mathcal{T}$ encompasses the extension and regularization step of the shape derivative, see Section 6.1 below. If V is replaced by the tangent space to some Riemannian manifold, the bilinear form a can be interpreted as a metric and $\nabla J(x)$, as given by (2.4), is the covariant gradient with respect to this metric.

(3) When V is a general Hilbert space, for a vector-valued function $\boldsymbol{g} : V \to \mathbb{R}^p$ with coordinates $\boldsymbol{g}(x) = (g_i(x))_{1 \le i \le p}$, $\mathrm{D}\boldsymbol{g} : V \to \mathbb{R}^p$ 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 $(\nabla g_i(x))_{1 \le i \le p}$ obtained by solving the p identification problems:

$$\forall \boldsymbol{\xi} \in V,\ a(\nabla g_i(x), \boldsymbol{\xi}) = \mathrm{D}g_i(x)\boldsymbol{\xi}; \tag{2.5}$$

more precisely:

$$\forall \boldsymbol{\mu} \in \mathbb{R}^d,\ \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\mu} = \sum_{i=1}^{p} \mu_i \nabla g_i(x).$$

In particular, the $p \times p$ matrix $\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}} \in \mathbb{R}^{p \times p}$ has entries

$$(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})_{ij} = a(\nabla g_i, \nabla g_j) = Dg_i(x)(\nabla g_j(x)).$$

Throughout this section, the equality constraints are said to be qualified at a point $x \in V$ if

$$\mathrm{rank}(\mathrm{D}\boldsymbol{g}(x)) = p, \text{ or equivalently } \mathrm{D}\boldsymbol{g}(x)\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}} \text{ is an invertible } p\text{-by-}p \text{ matrix.} \tag{2.6}$$

Note that (2.6) makes sense even at points $x \in V$ where $\boldsymbol{g}(x) \neq 0$, a fact that we shall use in the sequel. Under the above notations, let us recall the classical first-order necessary optimality conditions (KKT) for the equality-constrained problem (2.1) at some point $x^* \in V$ [14, 36] where the constraints are satisfied and qualified: there exists $\boldsymbol{\lambda}(x^*) \in \mathbb{R}^p$ such that,

$$\begin{cases} \nabla J(x^*) + \mathrm{D}\boldsymbol{g}(x^*)^{\mathcal{T}}\boldsymbol{\lambda}(x^*) = 0, \\ \boldsymbol{g}(x^*) = 0. \end{cases} \tag{2.7}$$

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

The defining formulas (1.4) and (1.5) for the steps $\boldsymbol{\xi}_J(x)$ and $\boldsymbol{\xi}_C(x)$, featured in the ODE (1.3), can be given a meaning in the present Hilbert space setting owing to the definition (2.3) of the transpose:

Definition 2. Consider the optimization problem (2.1). For any point $x \in V$ satisfying the constraint qualification condition (2.6), we define the *null space* and *range space* directions $\boldsymbol{\xi}_J(x)$ and $\boldsymbol{\xi}_C(x)$ by, respectively:

$$\boldsymbol{\xi}_J(x) := (\mathrm{I} - \mathrm{D}\boldsymbol{g}^{\mathcal{T}}(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\mathrm{D}\boldsymbol{g})\nabla J(x), \tag{2.8}$$

$$\boldsymbol{\xi}_C(x) := \mathrm{D}\boldsymbol{g}^{\mathcal{T}}(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\boldsymbol{g}(x). \tag{2.9}$$

Let us recall a formal intuition motivating the expressions (2.8), (2.9) and the dynamical system (1.3). Following the derivation of e.g. [11] and as is classical in Lagrange multiplier methods for optimization judging from the KKT optimality conditions (2.7), it is natural to search for an iterative optimization scheme (indexed by the subscript $_n$) of the form

$$x_{n+1} = x_n - \Delta t(\alpha_J \nabla J(x_n) + \mathrm{D}\boldsymbol{g}(x_n)^{\mathcal{T}}\boldsymbol{\lambda}_n), \tag{2.10}$$

where $\boldsymbol{\lambda}_n \in \mathbb{R}^p$ is a tentative value for the Lagrange multiplier $\boldsymbol{\lambda}$ in (2.7), α_J is a user-defined coefficient and Δt is the step increment between successive iterations. We determine the value of $\boldsymbol{\lambda}_n$ by imposing that the constraint $\boldsymbol{g}(x_{n+1})$ decreases by a factor $1 - \alpha_C\Delta t$ at the next iteration, up to some first-order error in Δt. Since

$$\boldsymbol{g}(x_{n+1}) = \boldsymbol{g}(x_n) - \Delta t\mathrm{D}\boldsymbol{g}(x_n)(\alpha_J \nabla J(x_n) + \mathrm{D}\boldsymbol{g}(x_n)^{\mathcal{T}}\boldsymbol{\lambda}_n) + o(\Delta t),$$

the requirement that $\boldsymbol{g}(x_{n+1}) \simeq (1 - \alpha_C\Delta t)\boldsymbol{g}(x_n)$ suggests the rule:

$$\boldsymbol{\lambda}_n = (\mathrm{D}\boldsymbol{g}(x_n)\mathrm{D}\boldsymbol{g}(x_n)^{\mathcal{T}})^{-1}\left(\alpha_C\boldsymbol{g}(x_n) - \alpha_J\mathrm{D}\boldsymbol{g}(x_n)\nabla J(x_n)\right). \tag{2.11}$$

We recognize then the scheme (2.25) (a time discretization of (1.3)) by replacing $\boldsymbol{\lambda}_n$ with the above value (2.11) in (2.10).

2.2.1. *Properties of the null space step $\boldsymbol{\xi}_J$*

In the finite-dimensional case where $V = \mathbb{R}^k$, it is well-known that the null space step $\boldsymbol{\xi}_J(x)$ defined by (1.4) 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 $\{y \in V \,|\, \boldsymbol{g}(y) = \boldsymbol{g}(x)\}$. Of course, this is still true when V is a Hilbert space, as recalled in the next lemma.

Lemma 1. *Let $x \in V$ be a point satisfying the qualification condition (2.6). The following properties hold:*

(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 we have introduced the range $\mathrm{Ran}(\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}) := \{\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda} \,|\, \boldsymbol{\lambda} \in \mathbb{R}^p\}$ of $\mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}$.
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 normalized feasible descent direction for J in the sense that

$$-\frac{\boldsymbol{\xi}_J(x)}{||\boldsymbol{\xi}_J(x)||_V} = \arg\min_{\boldsymbol{\xi} \in V} \quad \mathrm{D}J(x)\boldsymbol{\xi} \quad \text{s.t.} \begin{cases} \mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi} = 0 \\ a(\boldsymbol{\xi}, \boldsymbol{\xi}) \leq 1. \end{cases} \tag{2.12}$$

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

$$\boldsymbol{\xi}_J(x) = \nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda}^*(x), \tag{2.13}$$

where the Lagrange multiplier $\boldsymbol{\lambda}^*(x) := -(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\mathrm{D}\boldsymbol{g}\nabla J(x)$ is the unique solution to the following least squares problem that is the dual of (2.12):

$$\boldsymbol{\lambda}^*(x) = \arg\min_{\boldsymbol{\lambda} \in \mathbb{R}^p} ||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda}||_V. \tag{2.14}$$

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,

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

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

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

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 (2.12).

(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 $\nabla J(x)$ on $\mathrm{Ker}(\mathrm{D}\boldsymbol{g}(x))$. 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 (2.13) and the minimization property (2.14) follow. Note that the uniqueness of the solution $\boldsymbol{\lambda}^*(x)$ to (2.14) results from the qualification condition (2.6).
Finally, the optimization problem (2.12) can be rewritten as

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

Hence the (formal) dual problem of (2.12) reads:

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

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

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

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 minimization problem at the left-hand side of the above identity. This shows that (2.14) is the dual problem of (2.12).

□

2.2.2. *Properties of the range space step $\boldsymbol{\xi}_C$*

The next lemma is also classical in the literature. It characterizes the range space step $\boldsymbol{\xi}_C(x)$, defined by (2.9), as the unique Gauss-Newton direction for the minimization of the constraint function $\boldsymbol{g}(x)$ which is orthogonal to the (linearized) set of constraints:

Lemma 2. *Let $x \in V$ satisfy the condition (2.6); 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 $\mathrm{Ker}(\mathrm{D}\boldsymbol{g}(x))$:

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

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

$$\mathrm{D}\boldsymbol{g}(x)(-\boldsymbol{\xi}_C(x)) = -\boldsymbol{g}(x). \tag{2.15}$$

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

$$\min_{\boldsymbol{\xi}\in V} ||\boldsymbol{g}(x) + \mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi}||^2 \tag{2.16}$$

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

Proof.

(1) This easily follows from (2.3) and the calculation:

$$\forall \boldsymbol{\xi} \in \mathrm{Ker}(\mathrm{D}\boldsymbol{g}(x)), a(\boldsymbol{\xi}_C(x), \boldsymbol{\xi}) = ((\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\boldsymbol{g}(x))^T(\mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi}) = 0.$$

(2) This is an immediate consequence of the definition (2.9) of $\boldsymbol{\xi}_C(x)$. Note that (2.15) 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 $g_i(x) \leq 0$.

(3) Since (2.16) 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:

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

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, (2) states that $-\boldsymbol{\xi}_C(x)$ is one particular solution to the above equation; therefore, any two solutions of this problem differ by some $\boldsymbol{\zeta}$ such that $\mathrm{D}\boldsymbol{g}(x)\boldsymbol{\zeta} = 0$.

□

2.3. **Behavior of the trajectories of the flow**

The main features of the definitions of $\boldsymbol{\xi}_J(x)$ and $\boldsymbol{\xi}_C(x)$ are the facts that $\boldsymbol{\xi}_J$ 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 values of the constraint functional $\boldsymbol{g}(x(t))$ decrease to zero along the trajectories of the ODE (1.3), independently of the value of $\boldsymbol{\xi}_J(x)$. Then, as soon as the violation of the constraint becomes sufficiently small, the objective function J decreases without affecting the asymptotic vanishing of $\boldsymbol{g}(x(t))$. We review these properties in the next proposition, which was also observed in [48] in the finite-dimensional context.

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

$$\begin{cases} \dot{x} = -\alpha_J(\mathrm{I} - \mathrm{D}\boldsymbol{g}^{\mathcal{T}}(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\mathrm{D}\boldsymbol{g}(x))\nabla J(x) - \alpha_C\mathrm{D}\boldsymbol{g}^{\mathcal{T}}(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\boldsymbol{g}(x) \\ x(0) = x_0 \end{cases} \tag{2.17}$$

exist on some time interval $[0, T]$ *for* $T > 0$, *and that the qualification condition* (2.6) *holds at any point* $x(t)$, $t \in [0, T]$. *Then the following properties hold true:*

(1) The violation of the constraints decreases exponentially:

$$\forall t \in [0, T],\ \boldsymbol{g}(x(t)) = e^{-\alpha_C t}\boldsymbol{g}(x_0). \tag{2.18}$$

(2) Assume in addition 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*

$$\sup_{x\in K} ||\nabla J(x)||_V |\sigma_p^{-1}(x)| < +\infty, \tag{2.19}$$

where $\sigma_p(x)$ *is the smallest singular value of* $\mathrm{D}\boldsymbol{g}(x)$. *Then* J *decreases as long as the projection of its gradient on the set of constraints is large with respect to the decrease rate of the constraints given by* (2.18), *namely there exists a constant* $C > 0$ *such that*

$$\forall t \in [0, T],\ ||\Pi_{g(x)}(\nabla J(x(t)))||_V^2 > Ce^{-\alpha_C t} \Rightarrow \frac{\mathrm{d}}{\mathrm{d}t}J(x(t)) < 0. \tag{2.20}$$

(3) Any stationary point x^* of (2.17) satisfies the first-order KKT conditions (2.7) of the optimization program (2.1), that is:

$$\begin{cases} \boldsymbol{g}(x^*) = 0 \\ \exists \boldsymbol{\lambda}^* \in \mathbb{R}^p,\ \nabla J(x^*) + \mathrm{D}\boldsymbol{g}^{\mathcal{T}}(x^*)\boldsymbol{\lambda}^* = \Pi_{g(x^*)}(\nabla J(x^*)) = 0. \end{cases} \tag{2.21}$$

Proof.

(1) Using the definition (2.17), the decreasing property (2.15) 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 (2.18) 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 $\forall \boldsymbol{\xi} \in V$, $a(\boldsymbol{v}_i(x), \boldsymbol{\xi}) = \boldsymbol{v}_i(x)^\dagger\boldsymbol{\xi}$; 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 $\boldsymbol{a}(\boldsymbol{v}_i(x), \boldsymbol{v}_j(x)) = \boldsymbol{v}_i(x)^\dagger\boldsymbol{v}_j(x) = \delta_{ij}$. 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 we obtain the following inequality:

$$\forall x \in V,\ |\mathrm{D}J(x)\mathrm{D}\boldsymbol{g}^{\mathcal{T}}(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\boldsymbol{g}| \leq \sigma_p^{-1}(x)||\nabla J(x)||_V\,||\boldsymbol{g}(x)||. \tag{2.22}$$

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, from (2.18) and (2.22), the constant C in (2.20) can be selected as

$$C = p\frac{\alpha_C}{\alpha_J}||\boldsymbol{g}(x_0)|| \sup_{x \in K}\left[\sigma_p^{-1}(x)||\nabla J(x)||\right]. \tag{2.23}$$

(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 (2.17) must satisfy

$$\Pi_{g(x^*)}(\nabla J(x^*)) = 0, \text{ and } \mathrm{D}\boldsymbol{g}^{\mathcal{T}}(\mathrm{D}\boldsymbol{g}\mathrm{D}\boldsymbol{g}^{\mathcal{T}})^{-1}\boldsymbol{g}(x^*) = 0, \tag{2.24}$$

and so the first KKT condition in (2.7) 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 (2.24) implies $\boldsymbol{g}(x^*) = 0$, which completes the proof.

□

Remark 2. The solutions to the dynamical system (2.17) 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 $\mathcal{C}^2$ [22]. In the case where V is finite-dimensional, the assumption (2.19) 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 (2.17), similar properties to those of Proposition 1 hold for the discretized scheme

$$x_{n+1} = x_n - \Delta t(\alpha_J \boldsymbol{\xi}_J(x_n) + \alpha_C \boldsymbol{\xi}_C(x_n)), \tag{2.25}$$

which is sufficient for optimization. One can indeed verify that:

(1) At first order, the constraints decrease with a geometric rate: $\boldsymbol{g}(x_{n+1}) = (1 - \alpha_C \Delta t)\boldsymbol{g}(x_n) + o(\Delta t)$.
(2) x^* is an accumulation point of the sequence $(x_n)_{n\in\mathbb{N}}$ if and only if $\boldsymbol{g}(x^*) = 0$ and x^* is a KKT point of the problem (2.1), satisfying (2.7).

Remark 3. In our design of the update rule (1.3) to (1.5), it is possible to control more accurately the pace at which each of the constraints decreases: consider a diagonal matrix of positive coefficients $\boldsymbol{K} = \mathrm{diag}(\kappa_i)_{1\leq i\leq p}$ and replace the definition (1.5) or (2.9) 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 (2.17):

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

3. Proposed extension to equality and inequality constraints

We now proceed to extend the dynamical system (1.3) to handle inequality constraints as well. We consider from now on the full optimization problem

$$\begin{aligned} &\min_{x\in\mathcal{X}} \quad J(x) \\ &\text{s.t.} \begin{cases} \boldsymbol{g}(x) = 0 \\ \boldsymbol{h}(x) \leq 0, \end{cases} \end{aligned} \tag{3.1}$$

taking place over the Hilbert space $\mathcal{X} = V$ with inner product $a(\cdot,\cdot)$, and where $J : V \to \mathbb{R}$, $\boldsymbol{g} : V \to \mathbb{R}^p$ and $\boldsymbol{h} : V \to \mathbb{R}^q$ are differentiable functions.

This section is organized as follows: after setting notations in Section 3.1, we briefly review in Section 3.2 the classical method of slack variables for handling the inequality constraints in the problem (3.1). Section 3.3 then introduces another method for dealing with these constraints, which is original to the best of our knowledge. The essence of our method is the resolution of a dual quadratic subproblem for identifying the subset of constraints whose violation 'naturally' decreases in the course of the minimization of J, and those which should be enforced by projection of the gradient of J. The behavior of the induced flow in the context of the problem (3.1) is finally analyzed in Section 3.3.3 in the spirit of Proposition 1.

3.1. Notations and preliminaries

The set of indices corresponding to saturated or violated inequality constraints at $x \in V$ is denoted by $\widetilde{I}(x)$

$$\widetilde{I}(x) = \{i \in \{1, \dots, q\} \,|\, h_i(x) \geq 0\}, \tag{3.2}$$

and $\widetilde{q}(x) := \mathrm{Card}(\widetilde{I}(x))$ is the number of such constraints. Recall the notation $\boldsymbol{h}_I(x) = (h_i(x))_{i\in I}$ for the inequality constraints indexed by a subset $I \subset \widetilde{I}(x)$ and $\boldsymbol{C}_I(x)$, defined by (1.6), for the vector collecting the equality constraints $\boldsymbol{g}(x)$ and those selected inequality constraints $\boldsymbol{h}_I(x)$. The constraints are said to be qualified at $x \in V$, in the sense that the linearized saturated or violated constraints are independent:

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

If the point x satisfies the constraints, (3.3) is one usual qualification condition (of course, there are other possible qualification conditions, see [14, 36]). 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 $(\boldsymbol{\lambda}_I(x), \boldsymbol{\mu}_I(x)) \in \mathbb{R}^p \times \mathbb{R}_+^{\mathrm{Card}(I)}$ 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 equality and inequality constrained problem (3.1), the necessary first-order optimality conditions (the KKT conditions) for a given point $x^* \in V$, satisfying the qualification 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 [14, 36].

3.2. The method of slack variables for inequality constraints

It is classical to introduce slack variables so as to turn inequality constraints into equality constraints of an augmented problem including these additional variables, see [38] for the present gradient flow in the finite dimensional context. In other words, problem (3.1) is replaced with the following equivalent one, involving q extra variables $(z_1, \ldots, z_q) \in \mathbb{R}^q$:

$$\begin{aligned} \min_{\substack{x \in V \\ z \in \mathbb{R}^q}} \quad & J(x) \\ \text{s.t.} \quad & \boldsymbol{C}(x, z) = 0, \end{aligned} \tag{3.7}$$

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 (3.7) is an equality constrained optimization problem of the form (2.1), set over the Hilbert space

$$\widetilde{V} := V \times \mathbb{R}^q, \ \text{ with inner product } \widetilde{a}((x, z), (x', z')) := a(x, x') + z^T z'.$$

It can be solved thanks to the proposed algorithm in Section 2; the associated gradient flow for (3.7) reads:

$$\begin{cases} \begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} = -\alpha_J (\mathrm{I} - \mathrm{D}\boldsymbol{C}^{\mathcal{T}} (\mathrm{D}\boldsymbol{C} \mathrm{D}\boldsymbol{C}^{\mathcal{T}})^{-1} \mathrm{D}\boldsymbol{C}) \begin{bmatrix} \nabla J(x(t)) \\ 0 \end{bmatrix} - \alpha_C \mathrm{D}\boldsymbol{C}^{\mathcal{T}} (\mathrm{D}\boldsymbol{C} \mathrm{D}\boldsymbol{C}^{\mathcal{T}})^{-1} \boldsymbol{C}(x(t), z(t)), \\ \begin{bmatrix} x(0) \\ z(0) \end{bmatrix} = \begin{bmatrix} x_0 \\ z_0 \end{bmatrix}, \end{cases} \tag{3.8}$$

where $x_0 \in V$ is the considered initial point in the resolution of (3.1), and the variable z is initialized with a value $z_0 \in \mathbb{R}^q$ in such a way that the inequality constraints of (3.1) 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 (3.7):

$$\forall i \in \{1, \ldots, q\}, \ z_{0,i} = \sqrt{2|h_i(x_0)|}.$$

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 [38] that:

(i) Stationary points of the extended flow (3.8) are exactly critical points of (3.1), that is points x^* satisfying (3.6) but with $\boldsymbol{\mu}(x^*) \in \mathbb{R}^q$ possibly negative.

(ii) Among all possible critical points, 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 (3.8) converges in practice to a KKT point for problem (3.1); see also Section 3.4 about this point.

In the following, we shall introduce a different approach that does not use slack variables, but rather relies on a suitable selection of the active or violated constraints. The main differences with the method of slack variables lies in that inactive constraints are ignored, and that all stationary points of our proposed gradient flow are KKT points (satisfying all conditions (3.6)).

3.3. **The proposed algorithm**

Inspired by the methodology developed in Section 2, we still propose to solve the equality and inequality constrained problem (3.1) thanks to a dynamical system of the form:

$$\begin{cases} \dot{x}(t) = -\alpha_J \boldsymbol{\xi}_J(x(t)) - \alpha_C \boldsymbol{\xi}_C(x(t)) \\ x(0) = x_0, \end{cases} \tag{3.9}$$

whose discretized version reads:

$$x_{n+1} = x_n - \Delta t(\alpha_J \boldsymbol{\xi}_J(x_n) + \alpha_C \boldsymbol{\xi}_C(x_n)). \tag{3.10}$$

In the next subsections, we define $\boldsymbol{\xi}_J(x)$ and $\boldsymbol{\xi}_C(x)$ from formulas analogous to (1.7) and (1.8), which involve a procedure discriminating a relevant subset $\widehat{I}(x) \subset \widetilde{I}(x)$ of the saturated or violated constraints. Finally, we establish the properties of the flow (3.9) in Proposition 5.

3.3.1. *Definition of the range step direction*

Definition 3. For the optimization problem (3.1), the range step $\boldsymbol{\xi}_C(x)$ is defined by

$$\boldsymbol{\xi}_C(x) := \mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widetilde{I}(x)}(\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}\mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widetilde{I}(x)})^{-1}\boldsymbol{C}_{\widetilde{I}(x)}(x), \tag{3.11}$$

where $\widetilde{I}(x)$ is the subset of saturated or violated constraints, defined by (3.2).

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

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

The definition of the null space direction $\boldsymbol{\xi}_J(x)$ is slightly more involved as it is not obtained by replacing $\mathrm{D}\boldsymbol{g}(x)$ by $\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}$ in (2.8). It requires the introduction of a different subset $\widehat{I}(x) \subset \widetilde{I}(x)$, which is now detailed.

3.3.2. *Definition and characterizations of the null space direction $\boldsymbol{\xi}_J(x)$*

Inspired by the characterization of the null space direction of Lemma 1 for equality constrained problems, the null space direction $\boldsymbol{\xi}_J(x)$ is now sought, up to a change of sign, as a best normalized descent direction diminishing violated or saturated inequality constraints, i.e. $-\boldsymbol{\xi}_J(x)$ shall be set positively proportional to the solution of the following minimization problem:

$$\begin{aligned} \min_{\boldsymbol{\xi} \in V} \quad & \mathrm{D}J(x)\boldsymbol{\xi} \\ \text{s.t.} \quad & \begin{cases} \mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi} = 0 \\ \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)\boldsymbol{\xi} \leq 0 \\ ||\boldsymbol{\xi}||_V \leq 1. \end{cases} \end{aligned} \tag{3.12}$$

The purpose of this subsection is to characterize explicitly the minimizer $\boldsymbol{\xi}^*(x)$ of (3.12) by relying on its dual problem. As a consequence an explicit formula for the null space direction $\boldsymbol{\xi}_J(x)$, in the form of (1.7), will be given in Definition 4 below.

We now introduce the dual optimization problem to (3.12), which is analogous to the dual problem (2.14) of the previous section.

Proposition 2. *Let $x \in V$ satisfy the qualification condition (3.3). There exists a unique couple of multipliers $\boldsymbol{\lambda}^*(x) \in \mathbb{R}^p$ and $\boldsymbol{\mu}^*(x) \in \mathbb{R}_+^{\widetilde{q}(x)}$ solution to the following quadratic optimization problem which is the dual of (3.12):*

$$(\boldsymbol{\lambda}^*(x), \boldsymbol{\mu}^*(x)) := \arg \min_{\substack{\boldsymbol{\lambda} \in \mathbb{R}^p \\ \boldsymbol{\mu} \in \mathbb{R}^{\widetilde{q}(x)},\, \boldsymbol{\mu} \geq 0}} ||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}} \boldsymbol{\lambda} + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}} \boldsymbol{\mu}||_V. \tag{3.13}$$

Proof. Problem (3.12) is equivalent to the following min-max formulation:

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

Inverting formally the min and the max and performing the maximization with respect to $\boldsymbol{\xi}$ as in the proof of Lemma 1 yields that (3.13) is the dual problem of (3.12) up to a change of sign (the duality gap between (3.13) and (3.12) will be shown to vanish in Proposition 3). The program (3.13) brings into play the closed convex set $\mathbb{R}^p \times \mathbb{R}_+^{\widetilde{q}(x)}$ and the least squares functional

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

The latter is strictly convex over $\mathbb{R}^p \times \mathbb{R}_+^{\widetilde{q}(x)}$ by virtue of (3.3). Hence, (3.13) has a unique solution. □

The optimization problem (3.13) belongs to the class of non negative least squares problems; it can be solved efficiently with a number of dedicated solvers, such as `cvxopt` [8] or `IPOPT` [46]. One nice feature of (3.13) lies in that its dimension is the number $p + \widetilde{q}(x)$ of saturated or violated constraints, which can be small for many practical cases, as e.g. in our shape optimization applications of Section 6. It is also possible to exploit the sparsity of the constraints if $p + \widetilde{q}(x)$ is large, see Remark 5 below.

In the next proposition, we relate the optimal values and the solutions $\boldsymbol{\xi}^*(x)$ and $(\boldsymbol{\lambda}^*(x), \boldsymbol{\mu}^*(x))$ of the primal and dual problems (3.12) and (3.13). In essence, we show that the optimal feasible descent direction $\boldsymbol{\xi}^*(x)$ of (3.15) is the projection of the gradient $\nabla J(x)$ onto the cone of feasible directions. The proof follows classical arguments of linear programming duality theory and it is detailed for the convenience of the reader.

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

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

the value of the dual problem (3.13). Then the value of the primal problem (3.12) 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 (3.1) holds with (necessarily unique) Lagrange multipliers $(\boldsymbol{\lambda}^*(x), \boldsymbol{\mu}^*(x)) \in \mathbb{R}^p \times \mathbb{R}_+^{\widetilde{q}(x)}$:

$$\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}} \boldsymbol{\lambda}^*(x) + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}} \boldsymbol{\mu}^*(x) = 0. \tag{3.14}$$

One particular minimizer of (3.12) is $\boldsymbol{\xi}^*(x) = 0$.

(2) $m^*(x) > 0$: (3.14) does not hold and there exists a unique minimizer $\boldsymbol{\xi}^*(x)$ to (3.12), given by

$$\boldsymbol{\xi}^*(x) = -\frac{\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}} \boldsymbol{\lambda}^*(x) + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}} \boldsymbol{\mu}^*(x)}{||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}} \boldsymbol{\lambda}^*(x) + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}} \boldsymbol{\mu}^*(x)||_V}. \tag{3.15}$$

Proof. Let $\boldsymbol{\xi} \in V$ be a feasible direction for the problem (3.12), 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

$$\begin{aligned} \mathrm{D}J(x)\boldsymbol{\xi} \geq & \quad \mathrm{D}J(x)\boldsymbol{\xi} + \boldsymbol{\lambda}^T \mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi} + \boldsymbol{\mu}^T \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)\boldsymbol{\xi} \\ = & \quad a(\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda} + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}}\boldsymbol{\mu}, \boldsymbol{\xi}) \\ \geq & \quad -||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda} + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}}\boldsymbol{\mu}||_V \end{aligned} \tag{3.16}$$

Since (3.16) holds for any feasible direction $\boldsymbol{\xi}$ for (3.12), and for any $(\boldsymbol{\lambda}, \boldsymbol{\mu}) \in \mathbb{R}^p \times \mathbb{R}_+^{\widetilde{q}(x)}$, it follows:

$$\min_{\substack{\boldsymbol{\xi} \in V \\ \boldsymbol{\xi} \text{ feasible for (3.12)}}} \mathrm{D}J(x)\boldsymbol{\xi} \geq - \min_{\substack{\boldsymbol{\lambda} \in \mathbb{R}^p \\ \boldsymbol{\mu} \in \mathbb{R}^{\widetilde{q}(x)}, \boldsymbol{\mu} \geq 0}} ||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda} + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}}\boldsymbol{\mu}||_V. \tag{3.17}$$

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 (3.17) implies $p^*(x) \geq 0$. Therefore, the value of (3.12) is $p^*(x) = -m^*(x) = 0$, attained in particular at $\boldsymbol{\xi}^* = 0$, and more generally at any feasible $\boldsymbol{\xi}^* \in V$ satisfying $\boldsymbol{\mu}^*(x)^T \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)\boldsymbol{\xi}^* = 0$, as follows readily from the KKT conditions for (3.12). Furthermore, the KKT equation (3.14) is satisfied by definition of $m^*(x) = 0$.

(2) Assume now $m^*(x) > 0$. The KKT condition for (3.12) 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,

$$\forall \boldsymbol{\xi} \in V,\ (\mathrm{D}J(x) + \boldsymbol{\lambda}'^T \mathrm{D}\boldsymbol{g}(x) + \boldsymbol{\mu}'^T \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x))\boldsymbol{\xi} = -\alpha a(\boldsymbol{\xi}', \boldsymbol{\xi}). \tag{3.18}$$

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 $\alpha > 0$. The complementarity condition $\alpha(a(\boldsymbol{\xi}', \boldsymbol{\xi}') - 1) = 0$ 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 (3.12) implies $\boldsymbol{\mu}'^T \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)\boldsymbol{\xi}' = 0$. Therefore it holds that

$$\begin{aligned} \mathrm{D}J(x)\boldsymbol{\xi}' &= \mathrm{D}J(x)\boldsymbol{\xi}' + \boldsymbol{\lambda}'^T \mathrm{D}\boldsymbol{g}(x)\boldsymbol{\xi}' + \boldsymbol{\mu}'^T \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)\boldsymbol{\xi}' \\ &= a(\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda}' + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}}\boldsymbol{\mu}', \boldsymbol{\xi}') \\ &= -||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda}' + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}}\boldsymbol{\mu}'||_V. \end{aligned} \tag{3.19}$$

The previous equation together with the inequality (3.16) with $\boldsymbol{\xi} = \boldsymbol{\xi}'$ then implies that $(\boldsymbol{\lambda}', \boldsymbol{\mu}')$ achieves the minimum of (3.13). By uniqueness, this implies $\boldsymbol{\lambda}' = \boldsymbol{\lambda}^*(x)$ and $\boldsymbol{\mu}' = \boldsymbol{\mu}^*(x)$, hence $\boldsymbol{\xi}' = \boldsymbol{\xi}^*(x)$. 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 expression of the optimal descent direction $\boldsymbol{\xi}^*(x)$ from the signs of the multiplier $\boldsymbol{\mu}^*(x)$, and highlights in which sense the problem (3.12) is combinatorial. Recall definitions (3.4) and (3.5) for the projection operator Π_{C_I} and the multipliers $(\boldsymbol{\lambda}_I(x), \boldsymbol{\mu}_I(x))$.

Proposition 4. *In the context of point (2) in Proposition 3, let $\boldsymbol{\xi}^*(x)$ and $(\boldsymbol{\lambda}^*(x), \boldsymbol{\mu}^*(x))$ be the minimizers of the primal and dual problems (3.12) and (3.13). Define the subset $\widehat{I}(x) \subset \widetilde{I}(x)$ by*

$$\widehat{I}(x) := \{i \in \widetilde{I}(x) \,|\, \mu_i^*(x) > 0\}. \tag{3.20}$$

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

$$\begin{bmatrix} \boldsymbol{\lambda}^*(x) \\ \widehat{\boldsymbol{\mu}}^*(x) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\lambda}_{\widehat{I}(x)}(x) \\ \boldsymbol{\mu}_{\widehat{I}(x)}(x) \end{bmatrix} = -(\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^{\mathcal{T}})^{-1}\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}\nabla J(x), \tag{3.21}$$

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

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 either of the following discrete optimization problems:*

$$\begin{aligned} \widehat{I}(x) = \arg\max_{I\subset\widetilde{I}(x)} \quad & ||\Pi_{\boldsymbol{C}_I}(\nabla J(x))||_V \\ s.t. \quad & \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)\Pi_{C_I}(\nabla J(x)) \geq 0, \end{aligned} \tag{3.23}$$

$$\begin{aligned} \widehat{I}(x) = \arg\min_{I\subset\widetilde{I}(x)} \quad & ||\Pi_{C_I}(\nabla J(x))||_V \\ s.t. \quad & \boldsymbol{\mu}_I(x) \geq 0. \end{aligned} \tag{3.24}$$

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 complementary condition for the primal and dual problems (3.12) and (3.13) reads

$$\forall i \in \widetilde{I}(x), \quad \mu_i^*(x)\mathrm{D}h_i(x)\boldsymbol{\xi}^*(x) = 0. \tag{3.25}$$

Therefore, $\mathrm{D}h_i(x)\boldsymbol{\xi}^*(x) = 0$ for all indices $i \in \widehat{I}(x)$, which implies that as $\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}(x)\boldsymbol{\xi}^*(x) = 0$. Then, after left multiplication of (3.15) 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 (3.21), whence (3.22) follows.

(2) Let $I \subset \widetilde{I}(x)$ a subset satisfying $\mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)\Pi_{C_I}(\nabla J(x)) \geq 0$. This implies that

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

is feasible for the primal problem (3.12), and we obtain by definition of $\boldsymbol{\xi}^*(x)$ that

$$-||\Pi_{\boldsymbol{C}_{\widehat{I}(x)}}(\nabla J(x))||_V = \mathrm{D}J(x)\boldsymbol{\xi}^*(x) \leq \mathrm{D}J(x)\boldsymbol{\xi} = -||\Pi_{\boldsymbol{C}_I}(\nabla J(x))||_V, \tag{3.26}$$

whence the maximization property (3.23).

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 (3.13) by taking $\boldsymbol{\mu}$ to be equal to $\boldsymbol{\mu}_I$ on the indices of I and extended by 0 in the complementary subset $\widetilde{I}(x) \setminus I$. Then the optimality of $(\boldsymbol{\lambda}^*(x), \boldsymbol{\mu}^*(x))$ for this dual problem reads:

$$\begin{aligned} ||\Pi_{\boldsymbol{C}_{\widehat{I}(x)}}(\nabla J(x))||_V &= ||\nabla J + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda}^*(x) + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}(x)^{\mathcal{T}}\boldsymbol{\mu}^*(x)||_V \\ &\leq ||\nabla J(x) + \mathrm{D}\boldsymbol{g}(x)^{\mathcal{T}}\boldsymbol{\lambda} + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x)}^{\mathcal{T}}\boldsymbol{\mu}||_V = ||\Pi_{\boldsymbol{C}_I}(\nabla J(x))||_V, \end{aligned} \tag{3.27}$$

whence the minimization property (3.24).

□

In view of (3.20), the optimal multiplier $\boldsymbol{\mu}^*(x)$ can be interpreted as an indicator variable 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 (3.12)) 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 all constant at first order, i.e. $\mathrm{D}h_i(x)\boldsymbol{\xi} = 0$, (see Remark 6 for more details). It is therefore more efficient to project $\nabla J(x)$ only on those constraints associated to the indices $i \in \widehat{I}(x)$,

thus allowing the remaining ones (associated to $i \in \widetilde{I}(x) \setminus \widehat{I}(x)$, indicating vanishing multipliers $\mu_i^*(x) = 0$) to decrease since the calculated descent direction ensures that $\mathrm{D}h_i(x)\boldsymbol{\xi}^*(x) \leq 0$ holds for all $i = 1, ..., q$.

Note that actually, the use of a dual problem such as (3.13) in order to obtain information about which constraints should remain active is classical in active sets methods, see e.g. [16, 31, 36].

In principle, the subset $\widehat{I}(x)$ could be found by solving the discrete problems (3.23) or (3.24). However, we expect that in practice, it is more efficient to rely on iterative solvers relying on gradient descents for solving the dual problem (3.13), e.g. a cone programming solver or a non negative least squares algorithm such as [16]. This is what we do in the sequel.

Remark 4. With our notations, the optimization scheme proposed by Barbarosie et. al. [11, 12] reads

$$\begin{cases} x_{n+1} = x_n - \Delta t \nabla J(x_n) - \mathrm{D}\boldsymbol{C}_{I(x_n)}^{\mathcal{T}} \boldsymbol{\nu}_n \\ \quad \boldsymbol{\nu}_n = -\Delta t (\mathrm{D}\boldsymbol{C}_{I(x_n)} \mathrm{D}\boldsymbol{C}_{I(x_n)}^{\mathcal{T}})^{-1} \mathrm{D}\boldsymbol{C}_{I(x_n)} \nabla J(x_n) + \mathrm{D}\boldsymbol{C}_{I(x_n)}^{\mathcal{T}} (\mathrm{D}\boldsymbol{C}_{I(x_n)} \mathrm{D}\boldsymbol{C}_{I(x_n)}^{\mathcal{T}})^{-1} \boldsymbol{C}_{I(x_n)}, \end{cases} \tag{3.28}$$

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 becomes non negative. Therefore, the set $I(x_n)$ used in this strategy and that $\widehat{I}(x_n)$ featured in our strategy, given by (3.20), do not coincide in general; one could think of configurations where the procedure of [12] 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 about this procedure.

Having introduced the subset $\widehat{I}(x)$ (defined in (3.20)), we are now able 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 (3.22).

Definition 4. For any point $x \in V$ satisfying the constraint qualification (3.3), the null space direction $\boldsymbol{\xi}_J(x)$ at x for the optimization problem (3.1) is defined by:

$$\boldsymbol{\xi}_J(x) := \Pi_{\boldsymbol{C}_{\widehat{I}(x)}}(\nabla J(x)) = (I - \mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}(x)^{\mathcal{T}} (\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)} \mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^{\mathcal{T}})^{-1} \mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}) \nabla J(x), \tag{3.29}$$

where $\widehat{I}(x)$ is the set defined by (3.20).

The main point in Definition 4 is that, while all violated and saturated constraints are taken into account in the Gauss-Newton direction $\boldsymbol{\xi}_C(x)$ defined by (3.11), 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 5. Let us discuss two extreme cases related to the involved computational effort in the numerical implementation of (3.29). 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. For instance, this occurs in the case of bound constraints on the optimization variable $x = (x_1, ..., x_k)$, e.g. constraints of the form $\alpha_i \leq x_i \leq \beta_i$, $i = 1, ..., k$. Recalling from Remark 1 that in this setting, $\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^{\mathcal{T}} = A^{-1} \mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}^{T}$, it can be verified that the vector

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

can be computed as the solution of the sparse linear system

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

where $\Lambda \in \mathbb{R}^{p+\mathrm{Card}(\widehat{I}(x))}$ is an extra slack variable, which yields the null space directions $\boldsymbol{\xi}_J(x) = \nabla J(x) - X$. A similar strategy can be used to compute the range space direction $\boldsymbol{\xi}_C(x)$ of (3.11), or to solve the dual quadratic subproblem (3.13) by exploiting the sparsity of A and $\mathrm{D}\boldsymbol{C}_{\widehat{I}(x)}$.

Remark 6. As we have already mentioned, the Lagrange multiplier $\boldsymbol{\mu}^*(x)$ given by (3.21) may be understood as an indicator of which inequality constraints are aligned with the gradient of J. To further highlight this, it is instructive to consider the particular situation where the gradients of the constraint functions are orthogonal, i.e.:

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

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

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

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

Therefore the minimization problem (3.13) is separable with respect to the variables $(\boldsymbol{\lambda}, \boldsymbol{\mu}) \in \mathbb{R}^p \times \mathbb{R}_+^{\widetilde{q}(x)}$: $(\lambda_i^*(x))_{1 \leq i \leq p}$ and $(\mu_i^*(x))_{i \in \widetilde{I}(x)}$ are the respective solutions to the minimization problems:

$$\begin{aligned} &\forall i \in 1 \ldots p, \ \ \lambda_i^*(x) = \arg\min_{t \in \mathbb{R}} \left(t^2 ||\nabla g_i(x)||_V^2 + 2ta(\nabla J(x), \nabla g_i(x))\right), \\ &\forall i \in \widetilde{I}(x), \ \ \mu_i^*(x) = \arg\min_{\substack{t \in \mathbb{R} \\ t \geq 0}} \left(t^2 ||\nabla h_i(x)||_V^2 + 2ta(\nabla J(x), \nabla h_i(x))\right), \end{aligned}$$

which yields eventually:

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

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 (3.13) accounts for the combinatorics behind 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 worst, stay constant).

3.3.3. *Behavior of the trajectories of the flow*

The following proposition is the counterpart of Proposition 1 in the case of the equality and inequality constrained optimization problem (3.1).

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

$$\begin{cases} \dot{x}(t) = -\alpha_J \boldsymbol{\xi}_J(x(t)) - \alpha_C \boldsymbol{\xi}_C(x(t)) \\ x(0) = x_0, \end{cases} \tag{3.30}$$

with $\boldsymbol{\xi}_J$ and $\boldsymbol{\xi}_C$ given by (3.11) and (3.29) exist on some interval $[0,T]$ for $T > 0$ and are such that:
(a) The set $\widetilde{I}(x(t))$ defined in (3.2) 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 (3.3).

Then the following properties hold true:

(1) The violation of the constraints decreases exponentially:

$$\forall t \in [0,T],\ \boldsymbol{g}(x(t)) = e^{-\alpha_C t}\boldsymbol{g}(x_0)\ \text{and}\ \boldsymbol{h}(x(t)) \leq e^{-\alpha_C t}\boldsymbol{h}(x_0). \tag{3.31}$$

(2) Assume that $\operatorname{rank}(\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x_0)}(x))$ *is maximal for all x in* $K = \{x \in V \mid ||\boldsymbol{C}_{\widetilde{I}(x_0)}(x)||_\infty \leq ||\boldsymbol{C}_{\widetilde{I}(x_0)}(x_0)||_\infty\}$ *and*

$$\sup_{x\in K} ||\nabla J(x)||_V |\sigma_p^{-1}(x)| < +\infty. \tag{3.32}$$

where $\sigma_p(x)$ is the smallest singular value of $\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}(x)$. Then J decreases as soon as the projected gradient is large with respect to the violation of the constraints (which goes to 0 according to (3.31)), namely there exists a constant $C > 0$ such that

$$\forall t \in [0,T],\ ||\Pi_{C_{\widetilde{I}(x(t))}}(\nabla J(x(t)))||_V^2 > Ce^{-\alpha_C t} \Rightarrow \frac{\mathrm{d}}{\mathrm{d}t}J(x(t)) < 0. \tag{3.33}$$

(3) Any stationary point x^* of the flow (3.30) satisfies the KKT optimality conditions (3.6) which equivalently rewrite:

$$\begin{cases} \nabla J(x^*) + \mathrm{D}\boldsymbol{g}(x^*)^\mathcal{T}\boldsymbol{\lambda}^*(x^*) + \mathrm{D}\boldsymbol{h}_{\widetilde{I}(x^*)}(x^*)^\mathcal{T}\boldsymbol{\mu}^*(x^*) = 0, \\ \quad \boldsymbol{g}(x^*) = 0\ \text{and}\ \boldsymbol{h}_{\widetilde{I}(x^*)}(x^*) = 0 \Leftrightarrow \boldsymbol{C}_{\widetilde{I}(x^*)}(x^*) = 0, \end{cases} \tag{3.34}$$

where $(\boldsymbol{\lambda}^*(x^*), \boldsymbol{\mu}^*(x^*)) \in \mathbb{R}^p \times \mathbb{R}_+^{\widetilde{q}(x^*)}$ are defined in (3.13) or (3.21).

Proof.

(1) Definition (3.11) of $\boldsymbol{\xi}_C(x(t))$ implies $\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x(t))}\boldsymbol{\xi}_C(x(t)) = \boldsymbol{C}_{\widetilde{I}(x(t))}(x(t))$, 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

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{C}_{\widehat{I}(x(t))}(x(t)) = -\alpha_C\boldsymbol{C}_{\widehat{I}(x(t))}(x(t))\ \text{and}\ \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{h}_{\widetilde{I}(x(t))\setminus\widehat{I}(x(t))}(x(t)) \leq -\alpha_C\boldsymbol{h}_{\widetilde{I}(x_0)\setminus\widehat{I}(x_0)}(x(t)) \tag{3.35}$$

from which (3.31) follows by application of Gronwall's lemma.

(2) The proof is identical to that of Proposition 1.

(3) A stationary point x^* of (3.30) satisfies by definition $-\alpha_J\boldsymbol{\xi}_J(x^*) - \alpha_C\boldsymbol{\xi}_C(x^*) = 0$. Left multiplication of this identity by $\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x^*)}(x^*)$ yields:

$$-\alpha_J\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x^*)}(x^*)\boldsymbol{\xi}_J(x^*) - \alpha_C\boldsymbol{C}_{\widetilde{I}(x^*)}(x^*) = 0. \tag{3.36}$$

Remembering now that from definition (3.12),

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

equality in (3.36) 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 the fact that $-\alpha_J\boldsymbol{\xi}_J(x^*) - \alpha_C\boldsymbol{\xi}_C(x^*) = 0$, we obtain that $\boldsymbol{\xi}_J(x^*) = 0$, which corresponds to the first line in (3.6). This completes the proof.

□

Remark 7. 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 (3.30) is continuous. Indeed, in such a case, the range space direction $\boldsymbol{\xi}_C(x(t))$ is continuous by its definition (3.11), 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 (3.13) are continuous functions. 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 (3.30) with the new index set $\widetilde{I}(x(T))$. From (1) in Proposition 5, the bound $\boldsymbol{h}(x(t)) \leq e^{-\alpha_C t}\boldsymbol{h}(x(0))$ still holds after the time T for all constraints $i \in \{1, \dots, q\}$: constraints are asymptotically satisfied. Properties (2) and (3) remain true, up to an adjustment of the constant C in (3.33) (which can be taken global since there are finitely many possible sets $\widetilde{I}(x(t))$). There may exist situations where the set of asymptotically saturated constraints $\widetilde{I}(x(t))$ could oscillate indefinitely. However (2) states that $x(t)$ always keeps improving (in the sense of (3.33)), and (3) states that if $x(t)$ eventually converges, it is necessarily towards a KKT point.

Remark 8. In practice, the analysis of Proposition 5 is sufficient because, similarly to the conclusions of Remark 2, analogous properties hold for the discrete scheme

$$x_{n+1} = x_n - \Delta t\big(\alpha_J \boldsymbol{\xi}_J(x_n) + \alpha_C \boldsymbol{\xi}_C(x_n)\big). \tag{3.37}$$

Indeed, one can easily check that:

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

$$\boldsymbol{C}(x_{n+1}) = (1 - \alpha_C \Delta t)\boldsymbol{C}(x_n) + o(\Delta t). \tag{3.38}$$

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

Finally, note that 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.

3.4. **Comparison between the proposed method and the use of slack variables**

The main differences between the slack variable approach of Section 3.2 and the proposed flow (3.30) in Section 3.3 for dealing with equality and inequality constrained problems can be summarized as follows.

(1) Any point x^{crit} satisfying the constraints ($\boldsymbol{C}_{\widetilde{I}(x^{\text{crit}})}(x^{\text{crit}}) = 0$) and $\Pi_{\boldsymbol{C}_{\widetilde{I}(x^{\text{crit}})}}(\nabla J(x^{\text{crit}})) = 0$ is a stationary point of the extended dynamical system (3.8), although it might violate the full KKT condition (because (3.5) may yield possible negative values of the multiplier $\boldsymbol{\mu}_{\widetilde{I}(x^{\text{crit}})}(x^{\text{crit}})$). In contrast, x^* is a stationary points of the flow (3.30) if and only if it is a true feasible KKT point, see Proposition 5.

(2) The computation of $\boldsymbol{\xi}_J(x)$ and $\boldsymbol{\xi}_C(x)$ in our flow (3.30) requires to invert a matrix of size at most $(p + \widetilde{q}(x))$-by-$(p + \widetilde{q}(x))$ with $\widetilde{q}(x)$ the number of active or violated constraints at x. The process of equalizing inequality constraints as in [38, 41] 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 (3.30) follows the best locally admissible descent direction (with respect to the norm of V). This is not the case for the extended ODE (3.8). Therefore, from a common feasible point x, the ODE (3.30) always decreases the objective function with a steeper slope $\mathrm{d}J/\mathrm{d}s$ with respect to the parameterization induced by the path length s, defined as a function of the time t by

$$s(t) = \int_0^t ||\dot{x}(\alpha)||_V \mathrm{d}\alpha. \tag{3.39}$$

This property is illustrated in the academic examples of Section 5, and in particular on Figure 8a below.

All in all, our observations based on the simple numerical examples of the next section tend to illustrate that both flows (3.8) and (3.30) may have equivalent performances for solving the non linear optimization problem (1.1), this performance being measured in term of the total length

$$S = \int_0^{+\infty} ||\dot{x}||_V \mathrm{d}t$$

covered by the optimization path to reach the optimum. However, the two ODEs (3.8) and (3.30) yield optimization paths of essentially different natures. Our null space flow (3.30) 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 (3.8) yields smoother trajectories that more likely stay away from the constraints, at the cost of inverting at every step the full matrix $\mathrm{D}\boldsymbol{C}(x,z)\mathrm{D}\boldsymbol{C}(x,z)^{\mathcal{T}}$ of the size of the total number of constraints (active and inactive).

4. Practical implementation details

In this section, the ODE (1.3) is discretized by an explicit scheme and we propose a generic algorithm for adapting the time step Δt. We also discuss small adaptations for accounting for discontinuous changes of the right-hand side $-(\alpha_J \boldsymbol{\xi}_J + \alpha_C \boldsymbol{\xi}_C)$.

4.1. Time step adaptation based on a merit function.

The ODE (1.3) is discretized by an explicit scheme of the form:

$$x_{n+1} = x_n - \Delta t_n \big(\alpha_J \boldsymbol{\xi}_J(x_n) + \alpha_C \boldsymbol{\xi}_C(x_n)\big), \tag{4.1}$$

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 that 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*

$$\mathtt{merit}_{x_n}(x) := \alpha_J \left(J(x) + \boldsymbol{\Lambda}(x_n)^T \boldsymbol{C}_{\widetilde{I}(x_n)}(x)\right) + \frac{\alpha_C}{2} \boldsymbol{C}_{\widetilde{I}(x_n)}(x)^T \boldsymbol{S}(x_n) \boldsymbol{C}_{\widetilde{I}(x_n)}(x) \tag{4.2}$$

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 (3.13) (see (3.21)) 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 (4.1) is a gradient step for decreasing 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 (4.2). □

A possible implementation of an optimization strategy of the form (4.1) 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 (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 (3.13).

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 tunings 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 is invariant under multiplication of the constraints by arbitrary positive constants (up to the machine precision for the step 3); a preliminary rescaling of the constraints is therefore not required from the user.

Algorithm 1 Discretization of the null space gradient flow (3.30), based on a merit function.

for $n = 1 \dots \texttt{maxiter}$ **do**
  Determine $\widetilde{I}(x_n) = \{i \in \{1, \dots, q\} \,|\, h_i(x_n) \geq 0\}$ the set of active or violated constraints.
  Solve the dual problem (3.13) to obtain the Lagrange multiplier $\boldsymbol{\mu}^*(x_n)$. Infer the subset $\widehat{I}(x_n) \subset \widetilde{I}(x_n)$ of Proposition 4, indicating which constraints must be active in (3.12):

$$\widehat{I}(x_n) = \{i \in \widetilde{I}(x_n) \,|\, \mu_i^*(x_n) > \texttt{tolLag}\}. \tag{4.3}$$

  Extract the vectors $\boldsymbol{C}_{\widehat{I}(x_n)}(x_n)$ and $\boldsymbol{C}_{\widetilde{I}(x_n)}(x_n)$ (defined by (1.6)) and compute

$$\begin{aligned} \boldsymbol{\xi}_J(x_n) &= (\mathrm{I} - \mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widehat{I}(x_n)}(\mathrm{D}\boldsymbol{C}_{\widehat{I}(x_n)}\mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widehat{I}(x_n)})^{-1}\mathrm{D}\boldsymbol{C}_{\widehat{I}(x_n)})\nabla J(x_n), \\ \boldsymbol{\xi}_C(x_n) &= \mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widetilde{I}(x_n)}(\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x_n)}\mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widetilde{I}(x_n)})^{-1}\boldsymbol{C}_{\widetilde{I}(x_n)}. \end{aligned} \tag{4.4}$$

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

4.2. **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 (3.11) and (3.29) are characterized by 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 (3.29) 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. [21] 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 (1.9) (step 3 of Algorithm 1) with the set $\widetilde{I}_\epsilon(x_n)$ of inequality constraints violated "up to ϵ_i":

$$\widetilde{I}_\epsilon(x_n) = \{i \in \{1, \dots, q\} \,|\, h_i(x_n) \geq -\epsilon_i\}. \tag{4.5}$$

The tolerances $\epsilon_i > 0$ can be estimated in an automatic fashion, independent of an arbitrary rescaling of the constraints, thanks to an posteriori bound we now detail. Let $\mathtt{h}$ be a user-defined parameter accounting for the distance from the optimization path at which the constraints should be felt. This characteristic length $\mathtt{h}$ should be defined in accordance with the typical distance $||\Delta x||_V = ||x_{n+1} - x_n||_V$ between two successive iterations; in the academic examples in Section 5 below considering optimization in $\mathbb{R}^k$, we may set e.g. $\mathtt{h} = 0.01$ for a typical increment size $||\Delta x||_V \simeq 0.1$. For our shape optimization applications in Section 6, $\mathtt{h}$ is typically of the order of the discretization mesh size, see Section 6.2.2 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:

$$h_i(x_n) = |h_i(x_n) - h_i(x_n^*)| \simeq |\mathrm{D}h_i(x_n)(x_n^* - x_n)| \leq ||\nabla h_i(x_n)||_V \mathtt{h}. \tag{4.6}$$

It is therefore natural to set

$$\epsilon_i := ||\nabla h_i(x_n)||_V \mathtt{h} \tag{4.7}$$

for the value of ϵ_i in (4.5). Note that more generally, the a posteriori bound (4.6) 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 (3.13) 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 crossing the barrier $\boldsymbol{h} = 0$. The null space and range space steps $\boldsymbol{\xi}_J(x_n)$ and $\boldsymbol{\xi}_C(x_n)$ in step 4 of Algorithm 1 are finally replaced with $\boldsymbol{\xi}_{J,\epsilon}(x_n)$ and $\boldsymbol{\xi}_{C,\epsilon}(x_n)$ computed as follows:

$$\boldsymbol{\xi}_{J,\epsilon}(x_n) := (\mathrm{I} - \mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widehat{I}_\epsilon(x_n)}(\mathrm{D}\boldsymbol{C}_{\widehat{I}_\epsilon(x_n)}\mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widehat{I}_\epsilon(x_n)})^{-1}\mathrm{D}\boldsymbol{C}_{\widehat{I}_\epsilon(x_n)})\nabla J(x_n), \tag{4.8}$$

$$\boldsymbol{\xi}_{C,\epsilon} := \mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{I^*_\epsilon(x_n)}(\mathrm{D}\boldsymbol{C}_{I^*_\epsilon(x_n)}\mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{I^*_\epsilon(x_n)})^{-1}\boldsymbol{C}_{I^*_\epsilon(x_n)}(x_n), \tag{4.9}$$

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 on a small layer near the boundary of the feasible set. As a result, no abrupt discontinuity occurs anymore for $\boldsymbol{\xi}_{J,\epsilon}$ and $\boldsymbol{\xi}_{C,\epsilon}$ when crossing the boundary of the feasible domain while remaining in this layer. Including constraints $i \in \widehat{I}_\epsilon(x_n)$ 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.

5. Illustrations and comparisons on academic test cases

In this section, we consider simple and illustrative academic examples in order to compare qualitatively the following three strategies for dealing with inequality constraints in optimization problems:

- The method of Section 3.2 for equalizing inequality constraints by means of slack variables, see (3.30); this strategy is hereafter labelled as 'SLACK'.
- The proposed null space flow (3.30) in Section 3.3, based on the dual problem (3.13) for solving the combinatorial character of the constraints, which is labelled as 'NLSPACE'.
- An alternative, naive version of (3.30) which does not take advantage of the use of a dual problem, and simply projects $\nabla J(x(t))$ on all the violated constraints:

$$\begin{cases} \dot{x} = -\alpha_J \widetilde{\boldsymbol{\xi}}_J(x(t)) - \alpha_C \boldsymbol{\xi}_C(x(t)) \\ \widetilde{\boldsymbol{\xi}}_J(x) := (\mathrm{I} - \mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widetilde{I}(x)}(\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}\mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widetilde{I}(x)})^{-1}\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)})\nabla J(x) \\ \boldsymbol{\xi}_C(x) := \mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widetilde{I}(x)}(\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x)}\mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widetilde{I}(x)})^{-1}\boldsymbol{C}_{\widetilde{I}(x)}(x). \end{cases} \tag{5.1}$$

  In other words, all the violated or saturated constraints are taken into account in the computation of both the null space and range space directions $\boldsymbol{\xi}_J(x)$ and $\boldsymbol{\xi}_C(x)$. This strategy is labelled as 'NLSPACE (no dual)'.

To achieve our comparison purpose, Algorithm 1 is implemented for the discretization of (3.8), (3.30) and (5.1), with straightforward adaptations for equalizing slack variables or disabling the resolution of the dual problem. In all the following cases considered, we have set the values of α_J and α_C such that $\alpha_J/\alpha_C = 5/3$. The step size Δt was chosen sufficiently small to compute continuous paths with satisfying accuracy. Our discussion is exclusively focused on the continuous trajectories of the considered ODEs. In particular, we do neither discuss the issue of the selection of the time step, nor the efficiency of these methods in terms of the needed number of iterations required to achieve convergence.

In order to compare the three methods without bias, we consider the arc length $s(t)$ (defined in (3.39)) as the common reference time for the three ODEs (3.8), (3.30) and (5.1); recall indeed that this quantity is invariant under any monotone parameterization change of the time t. In the convergence figures below, optimized quantities are then plotted with respect to the pseudo time $s(t)$ in abscissa, for example we plot the graph $t \mapsto (s(t), J(s(t)))$ in order to account for the evolution of the objective function J.

We shall also plot the evolution of the Lagrange multipliers $s \mapsto \mu(x(s))$ associated with $\boldsymbol{\xi}_J(x(s))$ or $\widetilde{\boldsymbol{\xi}}_J(x(s))$ for the ODEs (3.30) and (5.1). For that purpose, these Lagrange multipliers are defined on the violated indices $i \in \widetilde{I}(x(s))$ by (3.21) for the null space flow (3.30), and by

$$\boldsymbol{\mu}(x(s)) := -(\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x(s))}\mathrm{D}\boldsymbol{C}^{\mathcal{T}}_{\widetilde{I}(x(s))})^{-1}\mathrm{D}\boldsymbol{C}_{\widetilde{I}(x(s))}\nabla J(x(s)) \tag{5.2}$$

for the flow (5.1) that does not use the dual problem (3.13). For the indices $i \in \{1, \dots, q\} \setminus \widetilde{I}(x(s))$, the value of the Lagrange multiplier is set to $\mu_i(x(s)) := 0$ by convention. We do not plot Lagrange multipliers for the ODE (3.8) using slack variables because these are defined with respect to the extended variables $(x(s), z(s))$.

The examples of this section take place in the optimization set $\mathcal{X} = \mathbb{R}^2$, which is equipped with the usual euclidean inner product; the Hilbert transposition $\mathcal{T} = T$ coincides with the usual transposition operator (see Definition 1). For simplicity, these examples only involve inequality constraints; we consider the following three scenarios:

- In Section 5.1, the initial point is unfeasible and the gradient of the objective function $\nabla J(x)$ is always aligned with the directions of the constraints;
- In Section 5.2 also, the initial point is unfeasible, but the gradient $\nabla J(x)$ may not be aligned with the direction of constraints;
- In Section 5.3, one of the constraints becomes inactive in the course of the optimization path.

Further comparisons with other iterative optimization algorithms will be presented in the PhD thesis [26].

5.1. **Test case 1 : unfeasible initialization with initial gradient aligned with the constraints.**

Our first example features the following problem, reproduced from [25]:

$$\begin{aligned} \min_{(x_1,x_2)\in\mathbb{R}^2} \quad & J(x_1,x_2) := x_2 + 0.3x_1 \\ \text{s.t.} \quad & \begin{cases} h_1(x_1,x_2) := -x_2 + \dfrac{1}{x_1} & \leq 0, \\ h_2(x_1,x_2) := x_1 + x_2 - 3 & \leq 0. \end{cases} \end{aligned} \tag{5.3}$$

This test case is designed so that for the chosen initial point $x_0 = (1.5, 2.25)$, the gradient of the objective function $\nabla J(x)$ is 'aligned' with the linear constraint h_2, in the sense that

$$-\nabla h_2(x_0) \cdot \nabla J(x_0) < 0.$$

Hence at least for small times (in fact during the whole optimization path), the constraint h_2 can be ignored since the minimization of J is naturally concurrent with a decrease of the value of h_2.

The optimization paths taken by the solutions of the three ODEs (3.8), (3.30) and (5.1) are plotted on Figure 1. The associated convergence histories for the values of the objective and constraint functions are displayed on Figure 2.

Let us comment the trajectory followed by the null space flow (3.30) in details. The gradient of the objective function remains aligned with the constraint h_2, which is associated with a zero Lagrange multiplier $\mu_2(x(s))$ (see Figure 3). During the first part of the optimization, the first constraint h_1 is not violated, hence the multiplier $\mu_1(x(s))$ is also set to zero. As a result, both constraints are ignored when computing the null space direction, which is set equal to the gradient: $\boldsymbol{\xi}_J(x(s)) = \nabla J(x(s))$. The optimization path $x(s)$ follows then almost the direction of the gradient $\nabla J(x(s))$ (without projection), up to a small deviation induced by the non zero Gauss-Newton direction $\boldsymbol{\xi}_C(x(s))$ in the unfeasible domain. When the hyperbolic constraint represented by h_1 becomes violated, the gradient is not aligned anymore with this constraint and the dual problem (3.13) yields a non-zero Lagrange multiplier $\mu_1(x(s))$ (near $s = 1.4$). From this point, the gradient $\nabla J(x(s))$ is then projected tangentially to the constraint h_1 till the optimum is attained.

In contrast, the path of the ODE (5.1) (no dual problem) fails to find the optimum as it is unable to unstick from the first saturated constraint. Notably, the gradient $\nabla J(x)$ is kept being projected tangentially to the violated constraint h_2 while it should not, which could have been detected from the negativity of the computed Lagrange multiplier μ_2 (see Figure 3).

Finally, the extended ODE (3.8) making use of slack variables feels the constraint h_1 from distance, inducing a deviation of the trajectory in the direction of the optimum before reaching the saturation of the constraint. This allows the trajectory $x(s)$ to find globally a slightly shorter path than our null space flow (3.30).

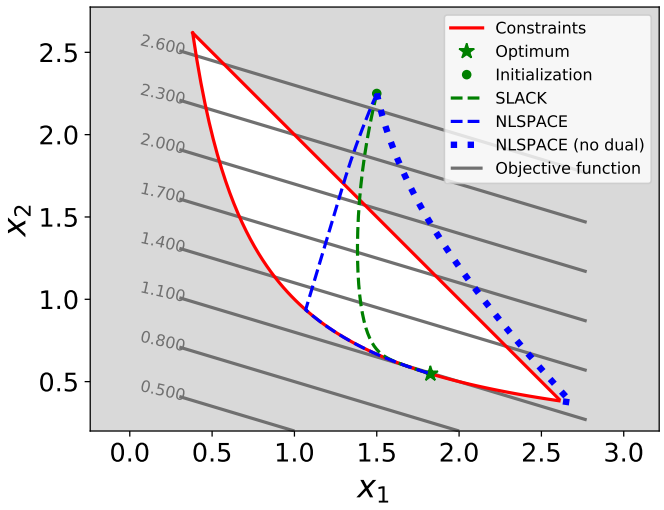
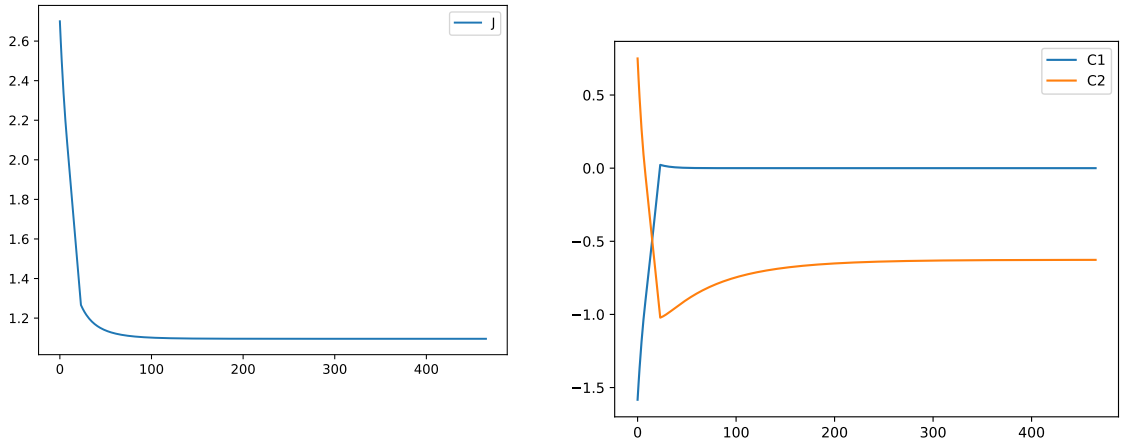


FIGURE 1. Optimization problem of Section 5.1: optimization paths for an unfeasible initialization x_0 with $\nabla J(x_0)$ aligned in the direction of the constraints.



(A) Objective function J (B) Constraints $\boldsymbol{h}$

FIGURE 2. History curves for the optimization problem of Section 5.1.

5.2. **Test case 2 : unfeasible initialization with initial gradient not aligned with the constraints.**

We now devise a test case where the gradient of the initialization is not aligned with the constraints. The feasible domain is the same as in the previous test case but the objective function is different:

$$\min_{(x_1,x_2)\in\mathbb{R}^2} \quad J(x_1,x_2) := (x_1-2)^2 + (x_2-2)^2$$
$$s.t. \begin{cases} h_1(x_1,x_2) := -x_2 + \dfrac{1}{x_1} \leq 0 \\ h_1(x_1,x_2) := x_1 + x_2 - 3 \leq 0. \end{cases} \tag{5.4}$$

We keep the same initialization $x_0 = (1.5, 2.25)$. Corresponding optimization paths and convergence curves are displayed on Figs. 4 and 5.

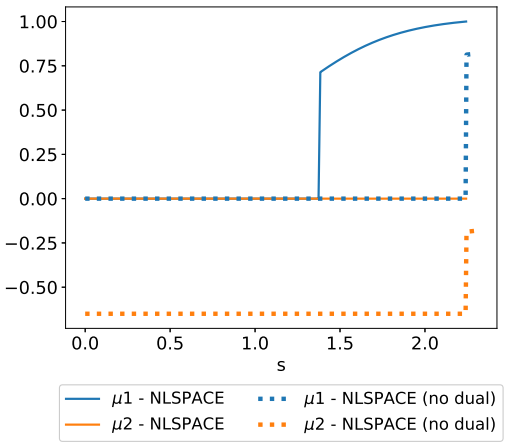


FIGURE 3. Evolution of the Lagrange multipliers $\mu_1(x(s)), \mu_2(x(s))$ for the optimization test case of Section 5.1.

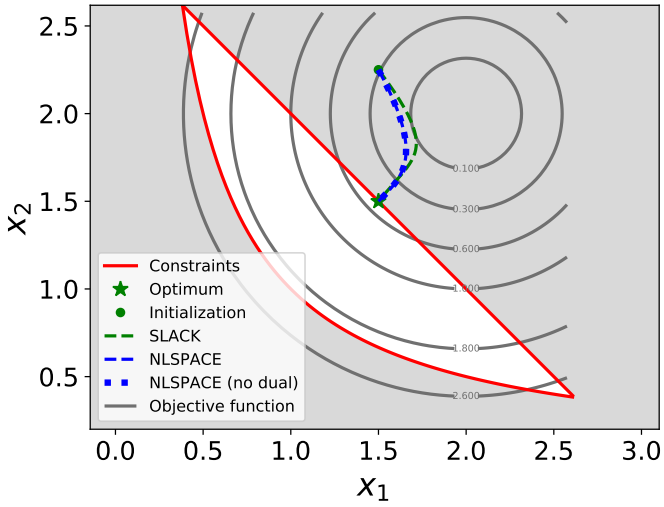
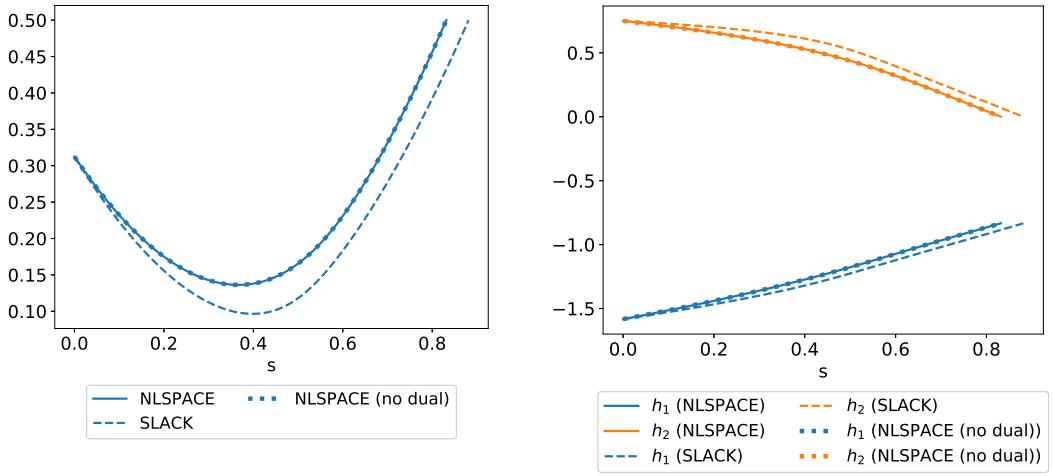


FIGURE 4. Optimization problem of Section 5.2: unfeasible initialization x_0 with $\nabla J(x_0)$ not aligned in the direction of the constraints.

For this example, the linear constraint h_2 is not aligned with the gradient along the optimization path of the null space gradient flow (3.30). This is associated with a non-zero Lagrange multiplier $\mu_2(x(s)) > 0$ (see Figure 6): the gradient $\nabla J(x(s))$ is kept being projected tangentially to the constraint h_2 when computing $\boldsymbol{\xi}_J(x(s))$. For this case, the combination with the Gauss-Newton direction $\boldsymbol{\xi}_C(x(s))$ allows to decrease simultaneously the objective function and the violation of the constraints, which enables the optimization path to reach directly the optimum when hitting the feasible set. Note that the convergence curve Figure 5a depicts a monotonically increasing objective function J throughout the optimization path (although we are minimizing J); this is of course due to the fact that constraints are never satisfied.

Since the constraint h_2 remains active from initialization to the optimum, the optimization path is unchanged when disabling the dual problem (ODE (5.1)). Finally, the path selected by the extended flow (3.8) to reach the optimum is very similar to the one of our method, although slightly longer.



(A) Objective function J (B) Constraints $\boldsymbol{h}$

FIGURE 5. History curves for the optimization problem of Section 5.2.

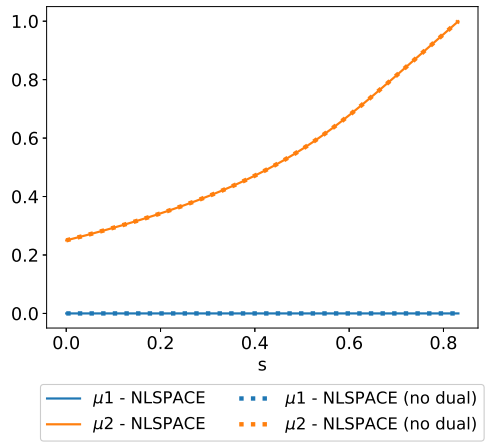


FIGURE 6. Evolution of the Lagrange multipliers $\mu_1(x(s)), \mu_2(x(s))$ for the optimization test case of Section 5.2.

5.3. Test case 3: a saturated inequality constraint becoming inactive along the optimization path

This last optimization test case is designed to illustrate the usefulness of the dual problem for detecting when a saturated inequality constraint becomes unsaturated. We consider a disconnected unfeasible domain made from the reunion of a half-space and the interior region of a parabola:

$$\begin{aligned} \min_{(x_1,x_2)\in\mathbb{R}^2} \quad & J(x_1,x_2) = x_1^2 + (x_2+3)^2 \\ \text{s.t.} \quad & \begin{cases} h_1(x_1,x_2) = -x_1^2 + x_2 & \leq 0 \\ h_2(x_1,x_2) = -x_1 - x_2 - 2 & \leq 0 \end{cases} \end{aligned} \tag{5.5}$$

The feasible domain and optimization paths starting from the initialization $x_0 = (3,3)$ are displayed on Figure 7. Associated convergence curves are reported on Figure 8.

For the null space flow (3.30), four different stages occur as is visible on the evolution of the Lagrange multipliers $\mu_1(x(s)), \mu_2(x(s))$ reported on Figure 9:

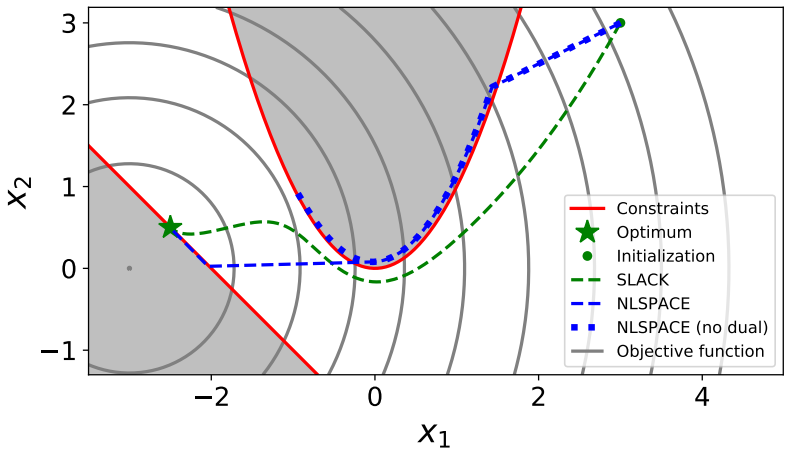
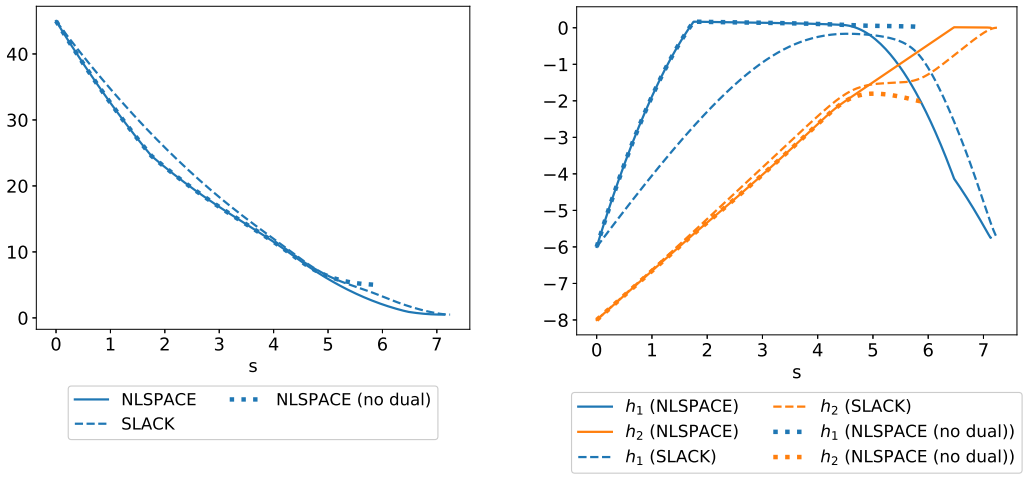


FIGURE 7. Optimization problem of Section 5.3 : feasible initialization x_0 but the optimization has to find a path across the parabolic domain.



(A) Objective function J (B) Constraints $\boldsymbol{h}$

FIGURE 8. History curves of the null space algorithm for the optimization problem of Section 5.3.

(1) From $s = 0$ to $s = 1.73$, the trajectory $x(s)$ remains in the feasible domain. Lagrange multipliers $\mu_1(x(s)) = \mu_2(x(s)) = 0$ are set to 0 and the null space direction $\boldsymbol{\xi}_J(x(s)) = \nabla J(x(s))$ coincides with the gradient of the objective function, until $x(s)$ hits the parabolic domain, which corresponds to the saturation of the first constraint h_1.
(2) From $s = 1.73$ to $s = 4.4$, the resolution of the dual problem yields a non zero multiplier $\mu_1(x(s)) > 0$. The optimization trajectory $x(s)$ remains tangent to the first constraint, until reaching a limit point such that $\nabla J(x(s)) \cdot \nabla h_1(x(s)) = 0$. At this moment, it is not necessary to project the gradient tangentially to this constraint any more, and the values of both Lagrange multipliers $\mu_1(x(s)) = \mu_2(x(s))$ are equal to 0.
(3) From $s = 4.4$ to $s = 6.5$, both constraints h_1 and h_2 are ignored and the trajectory $x(s)$ follows the gradient $-\nabla J(x(s))$, till the saturation of h_2.
(4) From $s = 6.5$ to $s = 7.1$, the second Lagrange multiplier $\mu_2(x(s)) > 0$ has a positive value; $x(s)$ evolves then tangentially to this constraint till the optimum is attained.

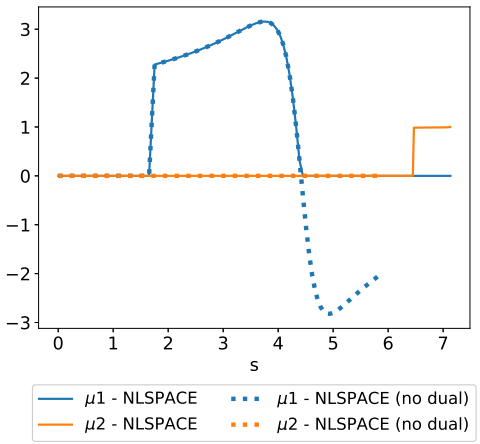


FIGURE 9. Evolution of Lagrange multipliers $\mu_1(x(s)), \mu_2(x(s))$ for the optimization problem of Section 5.3.

As illustrated on Figure 7, the use of the dual problem is key in the detection of the moment when the optimization trajectory $x(s)$ needs to be released from active inequality constraints. Because of the discrete nature of the time stepping, the path followed by the ODE (5.1) necessary enters slightly the violated parabolic domain. Since it does not use the information provided by the dual problem (3.13), the gradient $\nabla J(x(s))$ is kept being projected tangentially to the constraint h_1 till $x(s)$ converges to some stationary point (which is not a KKT point). As can be seen on Figure 9, the optimization trajectory followed by this ODE coincides with the one of the flow (3.30), till the instant $s = 4.4$ at which the Lagrange multiplier $\mu_1(x(s))$ becomes negative (which violates the feasibility condition of the dual problem (3.13)). Note that using larger steps could have allowed the trajectory $x(s)$ to exit "by chance" the unfeasible domain, in that case convergence to the optimum would have been obtained, however this would not reflect the actual behavior of the continuous solutions of (5.1).

Finally, the extended ODE (3.8) using slack variables finds a smooth path to the optimum. Since inactive constraints are felt from distance, the trajectory $x(s)$ is able to remain more strictly in the feasible domain for all times. The total length of the optimization path is almost the same than the one of the null space flow (3.30) (note the steeper descending slopes $\mathrm{d}J/\mathrm{d}s$ for the latter at intersection points of the two trajectories, see the point (3) in the discussion of Section 3.2).

6. Optimization on smooth manifolds: application to shape optimization

As we have already mentioned, our ultimate goal is to apply our optimization strategy to shape and topology optimization problems. Recalling (1.10), the optimization set $\mathcal{X}$ in this section is therefore a set of shapes in $\mathbb{R}^d$ ($d = 2$ or 3 in standard applications):

$$\mathcal{X} = \{\Omega \subset D \,|\, \Omega \text{ Lipschitz}\}, \tag{6.1}$$

where $D \subset \mathbb{R}^d$ is an enclosing 'hold-all' domain. Since $\mathcal{X}$ is not a Hilbert space, the present context does not fall into the optimization framework described in Sections 2 and 3. However, $\mathcal{X}$ may be endowed with a manifold structure, which makes it possible to extend our dynamical system (3.9) to this context, up to small adaptations that we now describe.

Building upon the framework of Hadamard's method of boundary variations, this manifold structure on the set $\mathcal{X}$ is first defined in Section 6.1. Then, Section 6.2 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 $\mathcal{T}$. Finally, Section 6.3 is devoted to numerical illustrations of our constrained optimization algorithm on the model example of the shape optimization of a bridge structure subjected to multiple load cases.

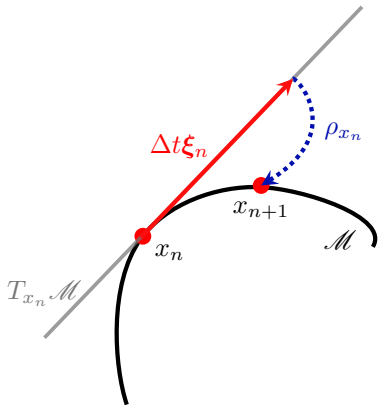


FIGURE 10. Optimization on a manifold $\mathscr{M}$: a retraction map ρ_{x_n} is used to project a tangential motion $\Delta t\boldsymbol{\xi}_n \in T_{x_n}\mathscr{M}$ from $x_n \in \mathscr{M}$ back onto the optimization domain $\mathscr{M}$.

6.1. **Hadamard's framework for gradient based shape optimization**

Our extension of the previous material to the shape optimization context is inspired by 'classical' optimization strategies on a smooth embedded manifold $\mathscr{M} \subset \mathbb{R}^k$. In this 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. [24, 1]. Then one relies on a *retraction* ρ_{x_n}, that is 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 of the actual point x_n on $\mathscr{M}$:

$$x_{n+1} := \rho_{x_n}(\Delta t\boldsymbol{\xi}_n), \tag{6.2}$$

where $\Delta t > 0$ is the descent step; see [2] and Figure 10. Since the new point x_{n+1} belongs to $\mathscr{M}$, this procedure can be repeated iteratively.

The same idea can be used to apply the methods of Sections 2 and 3 to the optimization problem (1.1), set over the set of shapes $\mathcal{X}$. To this end, we rely on Hadamard's method (see for instance [3, 30, 35, 43]), which considers variations of a shape $\Omega \in \mathcal{X}$ of the form

$$\rho_\Omega(\boldsymbol{\theta}) := (\mathrm{I} + \boldsymbol{\theta})(\Omega), \text{ for } \boldsymbol{\theta} \in W^{1,\infty}(D, \mathbb{R}^d) \text{ with } ||\boldsymbol{\theta}||_{W^{1,\infty}(D,\mathbb{R}^d)} < 1. \tag{6.3}$$

Formally, the set $W^{1,\infty}(D, \mathbb{R}^d)$ may be interpreted as the tangent space of $\mathcal{X}$ at Ω and the mapping ρ_Ω, which is defined by (6.3) on a neighborhood of 0 in $W^{1,\infty}(D, \mathbb{R}^d)$, plays the role of a retraction. Other definitions are possible for such a transformation dictating how a shape should evolve according to a vector field $\boldsymbol{\theta}$, see [4, 20] for discussions regarding that matter Note also that more rigorous manifold structures on shape spaces can be formulated, see e.g. [9, 39].

Usually, in the context of a general embedded manifold $\mathscr{M} \subset \mathbb{R}^k$, a differential structure on $\mathscr{M}$ is defined first (inducing a notion of derivative on $\mathscr{M}$), and the definition of a suitable retraction is inferred accordingly. In the framework of Hadamard's method however, it is the retraction ρ_Ω itself, that is the parametrization (6.3) by deformation fields $\boldsymbol{\theta}$, that is used to define the notion of derivative.

Definition 5. A function $\mathcal{X} \ni \Omega \mapsto F(\Omega) \in \mathbb{R}$ is *shape differentiable* at $\Omega \in \mathcal{X}$ if the underlying mapping $F \circ \rho_\Omega \,:\, \boldsymbol{\theta} \mapsto F(\rho_\Omega(\boldsymbol{\theta}))$, from $W^{1,\infty}(D, \mathbb{R}^d)$ into $\mathbb{R}$, is Fréchet differentiable at $\boldsymbol{\theta} = 0$. The corresponding derivative

$$\mathrm{D}F(\Omega) := \mathrm{D}(F \circ \rho_\Omega) \,:\, W^{1,\infty}(D, \mathbb{R}^d) \to \mathbb{R}$$

is called the *shape derivative* of F at Ω and the following expansion holds in the vicinity of $\boldsymbol{\theta} = 0$:

$$F(\rho_\Omega(\boldsymbol{\theta})) = F(\Omega) + \mathrm{D}F(\Omega)(\boldsymbol{\theta}) + o(\boldsymbol{\theta}), \text{ where } \frac{|o(\boldsymbol{\theta})|}{||\boldsymbol{\theta}||_{W^{1,\infty}(D,\mathbb{R}^d)}} \xrightarrow{\boldsymbol{\theta}\to 0} 0. \tag{6.4}$$

Remark 9. In practice, the set $\mathcal{X}$ of considered shapes is often a subset $\mathcal{U}_{\mathrm{ad}} \subset \mathcal{X}$ of admissible shapes (e.g. smooth shapes containing some non optimizable regions). The considered deformations are accordingly restrained to a subset $\Theta_{\mathrm{ad}} \subset W^{1,\infty}(D,\mathbb{R}^d)$, so that variations of shapes $\Omega \in \mathcal{U}_{\mathrm{ad}}$ remain in $\mathcal{U}_{\mathrm{ad}}$. To keep notations simple, we ignore these details in the presentation; see nevertheless the examples in Section 6.3.

For the shape optimization problem (1.1), we consider objective and constraint functions $J : \mathcal{X} \to \mathbb{R}$, $\boldsymbol{g} : \mathcal{X} \to \mathbb{R}^p$ and $\boldsymbol{h} : \mathcal{X} \to \mathbb{R}^q$ 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 $J : \mathcal{X} \to \mathbb{R}$ (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 $a(\cdot,\cdot)$, 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 $\mathrm{D}\boldsymbol{g}^T(\Omega) : \mathbb{R}^p \to V$, $\mathrm{D}\boldsymbol{h}^T(\Omega) : \mathbb{R}^q \to V$ with respect to the inner product a make sense; see Definition 1. For instance, the gradient $\nabla J(\Omega) \in V$ is obtained by solving the identification problem:

$$\forall \boldsymbol{\theta} \in V, \ a(\nabla J(\Omega), \boldsymbol{\theta}) = \mathrm{D}J(\Omega)(\boldsymbol{\theta}). \tag{6.5}$$

Intuitively, the bilinear form a can be interpreted as a metric on the 'manifold of shapes' $\mathcal{X}$, see e.g. [39, 40]. As for the choice of the Hilbert space $V \subset W^{1,\infty}(D,\mathbb{R}^d)$ used in the identification (6.5), one can for instance take 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 [15]). In this case, the identification problem (6.5) boils down to a linear elliptic problem of order $2m$.

Let us recall that, under mild regularity assumptions on the objective function $J(\Omega)$ and the state, 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* [30, 35, 42]). Namely, there exists $v_J(\Omega) \in L^1(\partial\Omega)$ such that

$$\forall \boldsymbol{\theta} \in W^{1,\infty}(D,\mathbb{R}^d), \ \mathrm{D}J(\Omega)\boldsymbol{\theta} = \int_{\partial\Omega} v_J(\Omega)\, \boldsymbol{\theta}\cdot\boldsymbol{n}\mathrm{d}s. \tag{6.6}$$

A very common strategy in the literature (see for instance [7, 10, 17, 27, 19, 33]) consists in taking simply $H^1(D,\mathbb{R}^d)$ as for the Hilbert space V, equipped with the inner product

$$\forall \boldsymbol{\theta}, \boldsymbol{\theta}' \in V, \ a(\boldsymbol{\theta}, \boldsymbol{\theta}') = \int_D (\gamma^2 \nabla\boldsymbol{\theta} : \nabla\boldsymbol{\theta}' + \boldsymbol{\theta}\cdot\boldsymbol{\theta}')\mathrm{d}x, \tag{6.7}$$

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, $\gamma = 3\,$`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: it is only formal since V 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 6.3), the identification problem (6.5) is still well-posed because (6.6) defines a continuous linear form on $H^1(D,\mathbb{R}^d)$. In such a situation, the identification (6.5) to (6.7) is interpreted as an extension and regularization of the normal velocity $v_J(\Omega)$ to the whole domain D. This step and its consistency with respect to optimization is very classical in shape optimization, see [7, 10, 17, 19, 33]. In particular, variants can be considered for tuning more finely the smoothness of such extensions, or to prescribe non optimizable boundaries by prescribing a zero Dirichlet boundary condition in (6.6). Last, 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 (3.9) 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 (3.11) and (3.29). 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 (6.5). In particular, steps 1 to 4 of Algorithm 1 including the resolution of the dual problem (3.13) are achieved from the knowledge of the Fréchet derivatives and of their transposes.

(2) The update (3.10) of the design from one iteration of the process to the next is performed by using the retraction map ρ_Ω as in (6.2):

$$\Omega_{n+1} := \rho_{\Omega_n}(-\Delta t(\alpha_J \boldsymbol{\xi}_J(\Omega_n) + \alpha_C \boldsymbol{\xi}_C(\Omega_n))); \tag{6.8}$$

the step 6 of Algorithm 1 is adapted accordingly.

The numerical implementation of the retraction map ρ_Ω, that is, the procedure used to effectively deform a shape Ω according to a prescribed deformation field $\boldsymbol{\theta}$, is presented in the next subsection.

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

Our level set framework for numerical shape and topology optimization is recalled in Section 6.2.1. Further technical details about the practical implementation of Algorithm 1 are then presented in Section 6.2.2.

6.2.1. *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 [37], then introduced in the shape optimization context in [6, 47]. 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:

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

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 vector velocity field $\boldsymbol{\theta}(x)$ translates in terms of an associated level set function $\phi(t, x)$ by the following advection equation:

$$\begin{cases} \dfrac{\partial \phi}{\partial t}(t,x) + \boldsymbol{\theta}(x) \cdot \nabla \phi(t,x) = 0, & t \in (0,t),\ \ x \in d, \\ \phi(0,x) = \phi_0(x), & x \in d, \end{cases} \tag{6.9}$$

where ϕ_0 is one level set function for ω. This approach, which is very convenient for describing the evolution of domains at the discrete level, corresponds actually to the use of a retraction $\widetilde{\rho}_\Omega$ which is slightly different from that ρ_Ω defined in (6.3):

$$\widetilde{\rho}_\Omega(\boldsymbol{\theta}) := \{x \in D \,|\, \phi(1,x) < 0\},$$

where $\phi(t, x)$ is the solution to (6.9). Observe that the mappings $\widetilde{\rho}_\Omega(\boldsymbol{\theta})$ and $\rho_\Omega(\boldsymbol{\theta})$ differ only from the second order in $\boldsymbol{\theta}$; therefore, the whole optimization process remains consistent in spite of this practical substitution.

In our applications, this method is carried out by relying on the mesh evolution technique of our previous works [5, 27]. At every iteration n, the current shape Ω_n is explicitly discretized as a submesh of a triangulated mesh $\mathcal{T}_n$ of D (see e.g. Figure 14 below). A descent direction $\boldsymbol{\theta}_n(x)$ is computed by estimating

$$\boldsymbol{\theta}_n := -(\alpha_{J,n} \boldsymbol{\xi}_J(\Omega_n) + \alpha_{C,n} \boldsymbol{\xi}_C(\Omega_n)), \tag{6.10}$$

where α_J and α_C of the update (6.8) have been replaced by dynamic coefficients $\alpha_{J,n}$ and $\alpha_{C,n}$ (this slight modification of Algorithm 1 is detailed in Section 6.2.2 below). A level set function ϕ_n is computed for the current shape Ω_n, which is then updated by solving equation (6.9) on the current mesh $\mathcal{T}_n$ with $\boldsymbol{\theta} = \boldsymbol{\theta}_n$:

$$\Omega_{n+1} = \widetilde{\rho}_{\Omega_n}(\Delta t \boldsymbol{\theta}_n).$$

In a last stage, $\mathcal{T}_n$ is remeshed adaptively into a new mesh $\mathcal{T}_{n+1}$ featuring a discretization of Ω_{n+1} as a submesh.

Remark 10. In our method, the deformation $\boldsymbol{\theta}(x)$ is a vector field, in contrast with more classical level set methods [6, 47] that rather rely on a non linear Hamilton-Jacobi equation different to (6.9) involving only the normal component of $\boldsymbol{\theta}$. In such settings and more generally, it is convenient to regularize only only the normal component $\boldsymbol{\theta} \cdot \boldsymbol{n}$ (a scalar field) of the shape derivative since it reduces the size of the identification

problem (6.6). These operations can be achieved consistently with respect to the whole optimization method up to an update of the parameterization space V and of the inner product a, see [26] for more details.

6.2.2. *Determination of the tolerance bounds (4.7) and settings of the parameters $\alpha_{J,n}$ and $\alpha_{C,n}$*

We rely on Algorithm 1 for our implementation of the null space flow (3.9) for numerical shape optimization, with the variant introduced in Section 4.2. A few comments are in order regarding the appropriate scaling of the null and range space steps in relation with the size of the mesh discretization; we define accordingly variable coefficients $\alpha_{J,n}$ and $\alpha_{C,n}$ for the descent direction $\boldsymbol{\theta}_n$ in (6.10).

For stability reasons, all vertices of the current shape Ω_n should move by only a few mesh elements when reaching 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` of the discussion in Section 4.2. Since *all* values of the displacement $\boldsymbol{\theta}_n$ should be controlled by `h`, we measure step sizes with the infinity norm $||\boldsymbol{\theta}_n||_{L^\infty(D)}$ rather than with the Hilbertian norm $||\boldsymbol{\theta}_n||_V = ||\boldsymbol{\theta}_n||_{H^1(D,\mathbb{R}^d)}$ as in Section 4.2. In order to take these changes into account, the tolerance bounds (4.7) need to be updated with respect to this norm as follows:

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

where $v_{C_i}(\Omega_n) \in L^1(\partial\Omega)$ is the scalar field featured in the shape derivative of the constraint $C_i(\Omega_n)$ as in (6.6).

Second, the step size $||\boldsymbol{\theta}_n||_{L^\infty(D,\mathbb{R}^d)}$ is effectively controlled by updating dynamically the parameters $\alpha_{J,n}$ and $\alpha_{C,n}$ scaling the null space and range space steps $\boldsymbol{\xi}_J(\Omega_n)$ and $\boldsymbol{\xi}_C(\Omega_n)$ in (6.10). Since only the products $\alpha_{J,n}\Delta t$ and $\alpha_{C,n}\Delta t$ matter, the step size is kept constant and equal to $\Delta t = 1$. Then, given A_J and A_C two user-defined parameters, which are expressed in terms of the mesh element size `hmin` for a clearer intuitive meaning, $\alpha_{J,n}$ and $\alpha_{C,n}$ are updated at every iteration according to the following rules:

$$\alpha_{J,n} := \begin{cases} \dfrac{A_J \mathtt{hmin}}{||\boldsymbol{\xi}_J(\Omega_n)||_{L^\infty(D,\mathbb{R}^d)}} & \text{if } n < \mathtt{n}_0 \\ \\ \dfrac{A_J \mathtt{hmin}}{\max(||\boldsymbol{\xi}_J(\Omega_n)||_{L^\infty(D,\mathbb{R}^d)}, ||\boldsymbol{\xi}_J(\Omega_{\mathtt{n}_0})||_{L^\infty(D,\mathbb{R}^d)})} & \text{if } n \geq \mathtt{n}_0 \end{cases} \tag{6.11}$$

$$\alpha_{C,n} := \min\left(0.9, \frac{A_C \mathtt{hmin}}{\max\left(\mathtt{1e\text{-}9}, ||\boldsymbol{\xi}_C(\Omega_n)||_{L^\infty(D,\mathbb{R}^d)}\right)}\right). \tag{6.12}$$

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

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

Note that 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 8, the role of the constant `1e-9` is only to avoid division by 0 when no constraint is active). In (6.11) and (6.12), the threshold $\mathtt{n}_0$ is an iteration number indicating for how many steps the normalization by the norm $||\boldsymbol{\xi}_J(\Omega_n)||_{L^\infty(D,\mathbb{R}^d)}$ should be done. By doing so, the null space direction $\boldsymbol{\xi}_J(\Omega_n)$ is allowed to converge to 0 as $n \to \infty$.

6.3. **Illustrations on a multiple load structural shape optimization test case**

In this final section, we illustrate the efficiency of our optimization strategy on a practical structural design problem. The physical model and the shape optimization setting are presented in Section 6.3.1, then numerical results are discussed for two possible test cases, featuring multiple objective criteria or multiple constraint functions, in Sections 6.3.2 and 6.3.3.

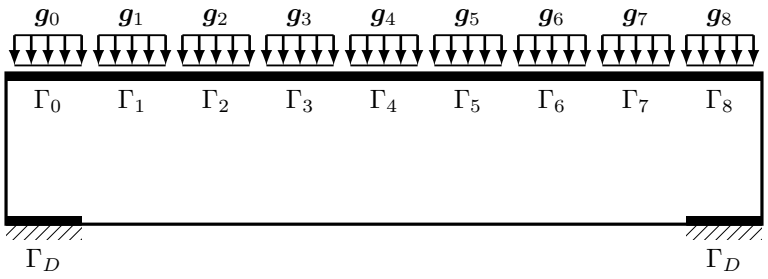


FIGURE 11. Geometric setting for the multiple load case test case

6.3.1. *Shape optimization setting*

We 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 boundary of $\partial\Omega$ 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.
- 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; see Figure 11. 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 displacement $\boldsymbol{u}_i$ is the unique solution in $H^1(\Omega, \mathbb{R}^d)$ to the linearized elasticity system:

$$\begin{cases} -\mathrm{div}(Ae(\boldsymbol{u}_i)) = 0 & \text{in } \Omega \\ Ae(\boldsymbol{u}_i)\boldsymbol{n} = 0 & \text{on } \Gamma \\ Ae(\boldsymbol{u}_i)\boldsymbol{n} = \boldsymbol{g}_i & \text{on } \Gamma_i \\ Ae(\boldsymbol{u}_i)\boldsymbol{n} = 0 & \text{on } \Gamma_j \text{ for } j \neq i \\ \boldsymbol{u}_i = 0 & \text{on } \Gamma_D, \end{cases} \tag{6.13}$$

where $e(\boldsymbol{u}) = (\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^T)/2$ is the strain tensor associated to the displacement $\boldsymbol{u}$ and $Ae(\boldsymbol{u}) = 2\mu e(\boldsymbol{u}) + \lambda \mathrm{Tr}(e(\boldsymbol{u}))\mathrm{I}$ is the corresponding stress tensor, involving the Hooke's law A. The Young modulus and the Poisson ratio are set to $E = 15$ and $\nu = 0.35$, which corresponds to $\lambda = 12.96$ and $\mu = 5.56$. As we have hinted at in Section 6.2.1, the shape is exactly meshed at each iteration (see Figure 14 below), so that each state equation (6.13) 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. [6]).

Starting from the initial structure Ω_0 depicted in Figure 12, we minimize the volume $\mathrm{Vol}(\Omega)$ of the structure Ω and maximize the collection of compliances $C_i(\Omega)$ (for each load case $\boldsymbol{g}_i$), as a measure of its global rigidity. These quantities are defined by:

$$\mathrm{Vol}(\Omega) := \int_\Omega \mathrm{d}x, \quad C_i(\Omega) := \int_\Omega Ae(\boldsymbol{u}_i) : e(\boldsymbol{u}_i)\mathrm{d}x, \tag{6.14}$$

and their shape derivatives read (see e.g. [6, 30]):

$$\mathrm{DVol}(\Omega)(\boldsymbol{\theta}) = \int_\Gamma \boldsymbol{\theta}\cdot\boldsymbol{n}\mathrm{d}s, \quad \mathrm{D}C_i(\Omega)(\boldsymbol{\theta}) = -\int_\Gamma Ae(\boldsymbol{u}_i) : e(\boldsymbol{u}_i)\,\boldsymbol{\theta}\cdot\boldsymbol{n}\mathrm{d}s. \tag{6.15}$$

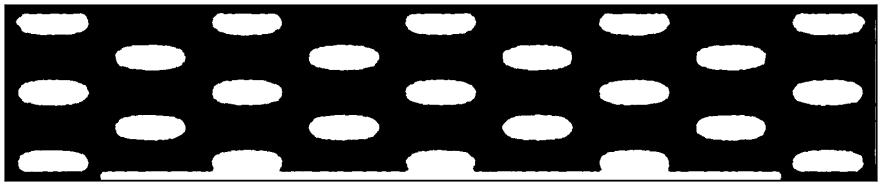


FIGURE 12. Initialisation Ω_0 (solid in *black*) for the shape optimization examples of Section 6. The thin white layer at the bottom is a non optimizable part of the domain.

6.3.2. *Volume minimization with maximum compliance constraint*

At first, the volume $\mathrm{Vol}(\Omega)$ is minimized while requiring that each individual compliance $C_i(\Omega)$ do not exceed a given prescribed value C:

$$\begin{aligned} \min_{\Omega\in\mathcal{X}} \quad & \mathrm{Vol}(\Omega) \\ \text{s.t.} \quad & C_i(\Omega) \le C \quad \text{for all } i \in I \end{aligned} \tag{6.16}$$

where $I \subset \{0, 1, \dots, 8\}$ is a set of indices selecting the considered load cases. We solve (6.16) 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 the threshold C in (6.16) is set to a fraction of the maximum of the compliances $C_i(\Omega_0)$ of the initial design Ω_0 (reported on Figure 12):

$$C = 0.7 \max_{i=0,\dots,8} \int_{\Omega_0} Ae(\boldsymbol{u}_i) : e(\boldsymbol{u}_i) \mathrm{d}x. \tag{6.17}$$

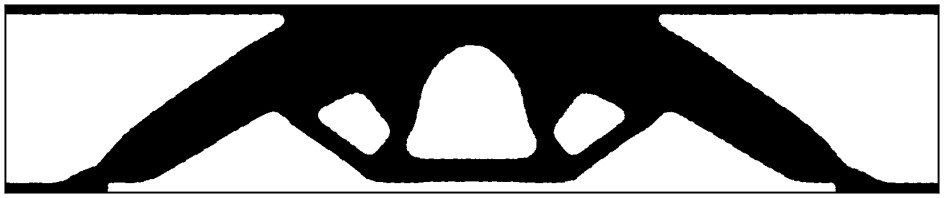
Let us emphasize that for this example (and the next ones), no fine tuning of the algorithm parameters A_J and A_C (determining the update of the values of $\alpha_{J,n}$ and $\alpha_{C,n}$ in (6.10)) of Section 6.2.2 is required. The only intuition guiding our choice for this particular test case is that the value of A_J should be set lower than A_C. Indeed, a too high value of A_J might entail a too quick decrease of the volume, which would incur dramatic topological changes violating the rigidity constraints. Therefore these parameters were 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 shown on Figure 13. 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. 14 and 15. The convergence histories in the three situations are reported on Figs. 16 to 18. 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, we observed that $\widehat{I}(\Omega_n)$ coincides with $\widetilde{I}(\Omega_n)$ at every iteration, however this situation is very specific to this test case and does not reproduce in generality, see [26] for different shape optimization applications featuring $\widetilde{I}(\Omega_n) \neq \widehat{I}(\Omega_n)$. 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.

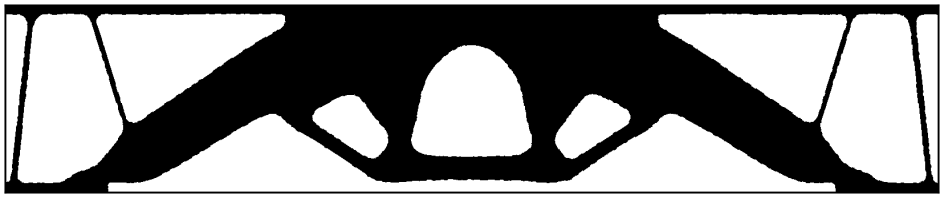
6.3.3. *Min/Max compliance optimization with a volume constraint*

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

$$\begin{aligned} \min_{\Omega\in\mathcal{X}} \quad & \max_{i\in I} \quad C_i(\Omega) \\ \text{s.t.} \quad & \mathrm{Vol}(\Omega) = \rho_0 \mathrm{Vol}(D) \end{aligned} \tag{6.18}$$



(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 13. Optimized shapes for three possible configurations of the volume minimization problem subject to maximum compliance constraint (Section 6.3.2).

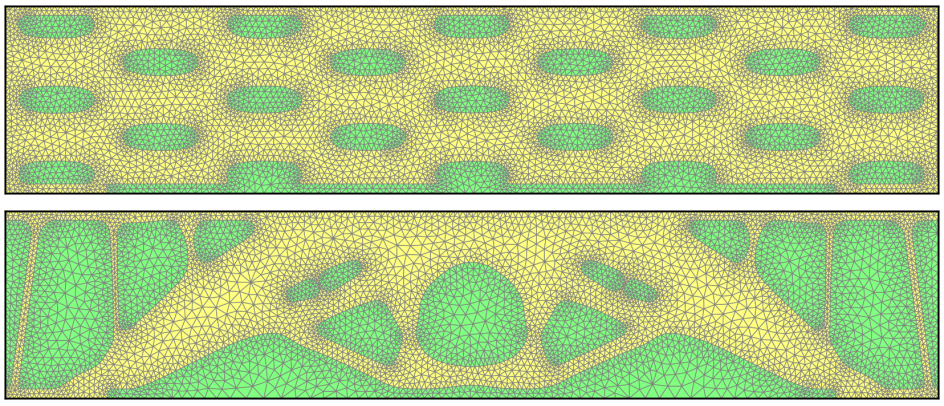


FIGURE 14. Meshes of the initialization and final shapes for the nine load case of Figure 13c (Section 6.3.2).

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:

$$\min_{(\Omega,m)\in\mathcal{X}\times\mathbb{R}} \quad m$$
$$\text{s.t.} \begin{cases} \text{Vol}(\Omega) = \rho_0 \text{Vol}(D) \\ C_i(\Omega) \leq m \quad \text{for all } i \in I. \end{cases} \tag{6.19}$$

The optimization must now be 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 3.2): the

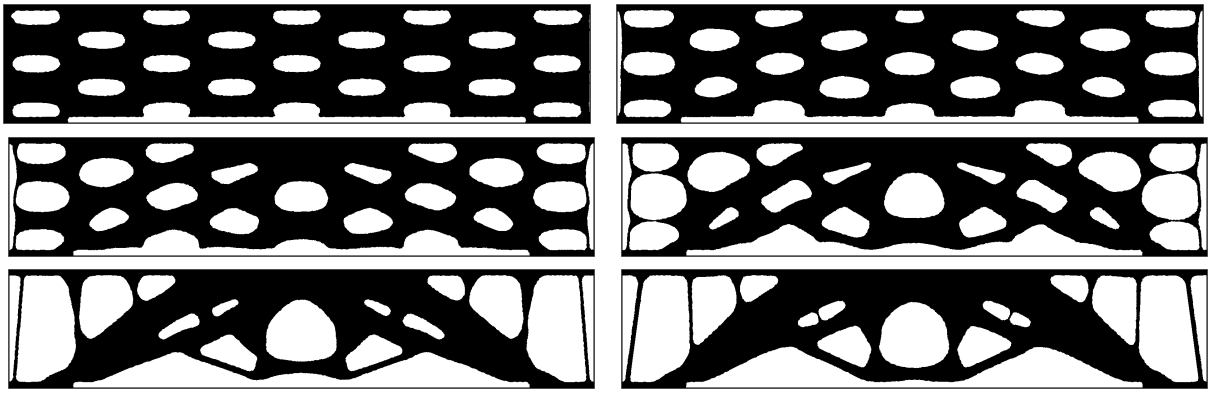
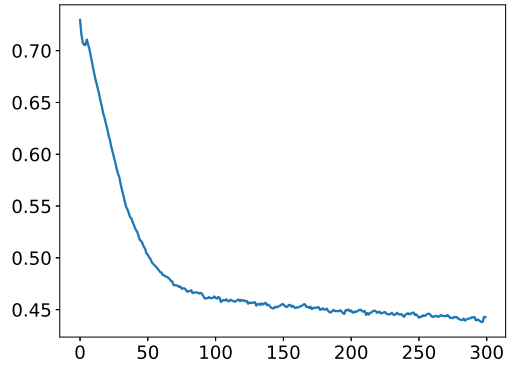
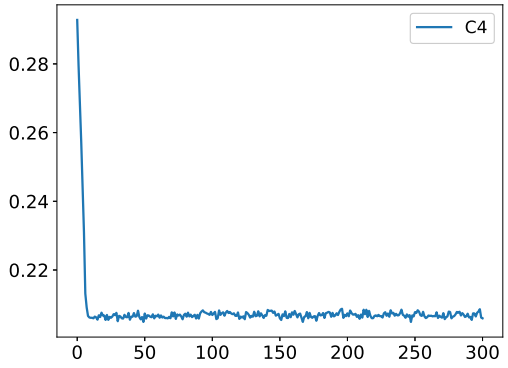


FIGURE 15. Intermediate minimizing shapes for the nine load case of the volume minimization problem of Section 6.3.2 (iterations 0, 5, 10, 20, 80, and 300).

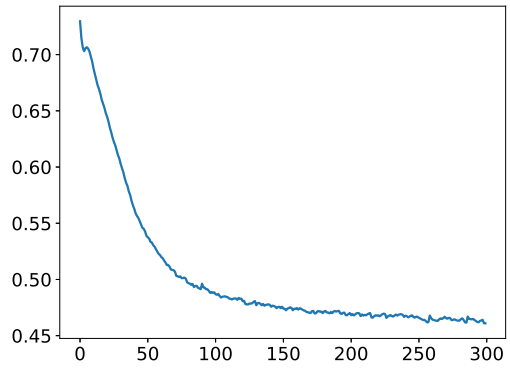


(A) Objective function $J(\Omega) = \mathrm{Vol}(\Omega)$. Final value: $\mathrm{Vol}(\Omega) = 0.44\mathrm{Vol}(D)$.

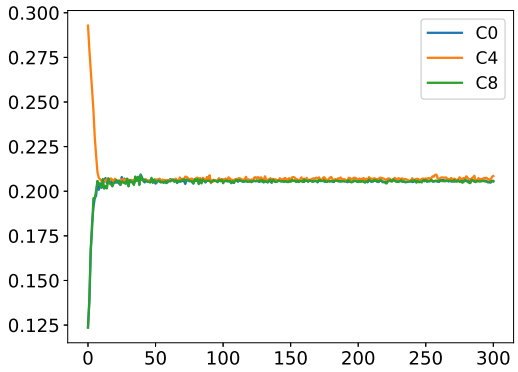


(B) Constraints (compliance values C_i). Final values: $C_4 = 1.29$.

FIGURE 16. Convergence history curves for the single load case of Section 6.3.2.

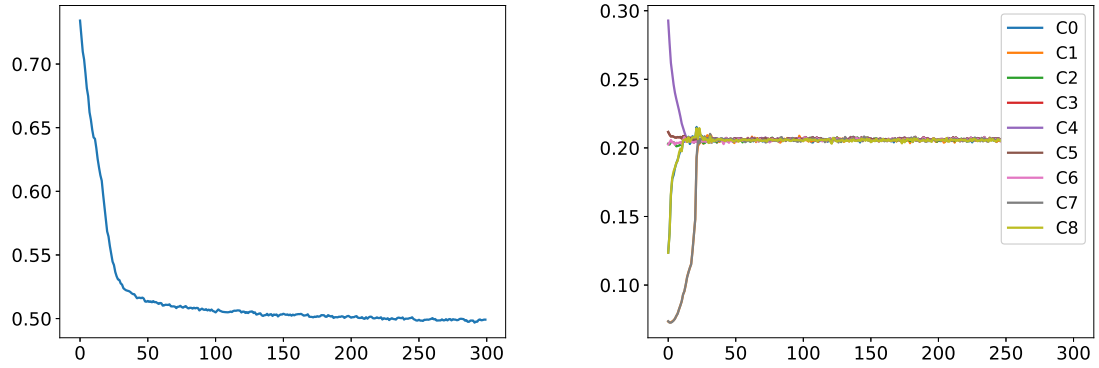


(A) Objective function $J(\Omega) = \mathrm{Vol}(\Omega)$. Final value: $\mathrm{Vol}(\Omega) = 0.46\mathrm{Vol}(D)$.



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

FIGURE 17. Convergence history curves for the three load case of Section 6.3.2.



(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 18. Convergence history curves for the nine load case of Section 6.3.2.

optimization domain $\mathcal{X} \times \mathbb{R}$ is equipped with the tensorized tangent space $V = H^1(D, \mathbb{R}) \times \mathbb{R}$ and differentials are identified to gradients thanks to the bilinear form $\widetilde{a} : H^1(D, \mathbb{R}) \times \mathbb{R} \to H^1(D, \mathbb{R}) \times \mathbb{R}$ defined by

$$\forall (v, w) \in H^1(D, \mathbb{R}) \times H^1(D, \mathbb{R}),\ (l, m) \in \mathbb{R} \times \mathbb{R}, \quad \widetilde{a}((v, l), (w, m)) = a(v, w) + lm, \tag{6.20}$$

where a is the $H^1(D, \mathbb{R}^d)$ scalar product of (6.7). The slack variable m is initialized with the maximum value of the compliance of the initial structure Ω_0 over all the considered loads:

$$m_0 := \max_{i \in I} \quad C_i(\Omega_0), \tag{6.21}$$

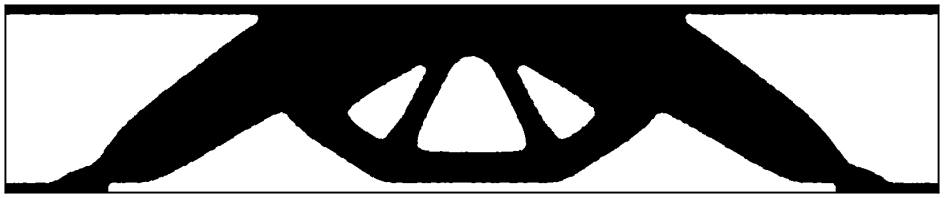
and its values m_n are then updated along with the shape Ω_n according to Algorithm 1.

The resulting optimized structures are shown on Figure 19 for each of the three considered configurations and the associated convergence histories are displayed on Figs. 20 to 22 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 compliances even after all the inequality constraints are saturated, which occurs as soon as where all compliances achieve a common value. As expected, the optimum design found for the nine load minimum compliance case (Figure 13c) is similar (up to a few bars) to the corresponding one found for the volume minimization (Figure 19c): indeed, both cases reach at convergence a volume fraction $\mathrm{Vol}(\Omega) = 0.5\mathrm{Vol}(D)$ and a maximum compliance $\max C_i(\Omega) \simeq 1.30$.

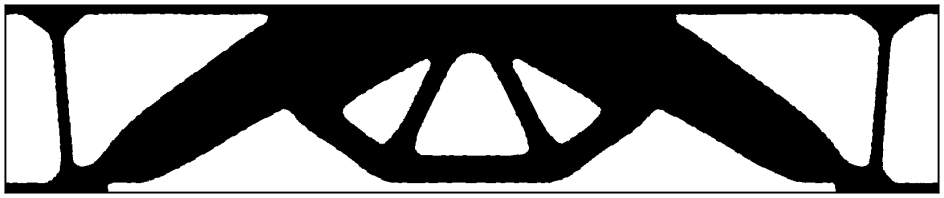
7. Conclusion and perspectives

In this paper, we have introduced a novel gradient flow for constrained minimization which does not rely on the use of slack variables but on a dual problem allowing to detect on which subset of the constraints the gradient of the objective function should be projected. We insisted on the clear distinction required between differentials and gradients in order to perform such minimization in Hilbert spaces. We provided pedagogical illustrations regarding the behavior of optimization trajectories. Finally, we demonstrated how this flow provides a reliable and comprehensive method for systematic mathematical programming in shape optimization. Further shape optimization applications built upon this algorithm will be provided in the PhD dissertation [26].

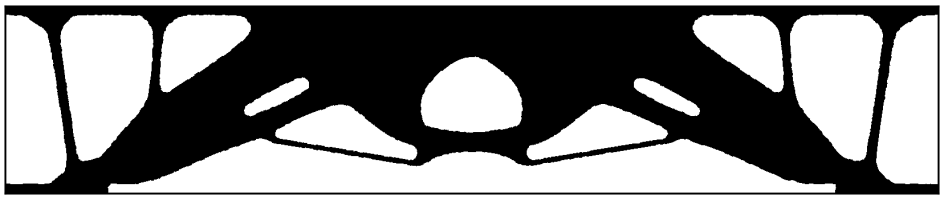
Acknowledgements. This work was supported by the Association Nationale de la Recherche et de la Technologie (ANRT) [grant number CIFRE 2017/0024]. G. A. is a member of the DEFI project at INRIA Saclay Ile-de-France. The work of C.D. is partially supported by the IRS-CAOS grant from Université Grenoble-Alpes. We thank Alexis Faure for gratefully providing his optimization and plotting routines, which was helpful for making the plots of Section 5.



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

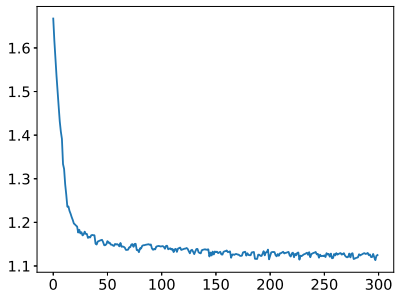


(B) Three loads (only $\boldsymbol{g}_0, \boldsymbol{g}_4, \boldsymbol{g}_8$ are considered).

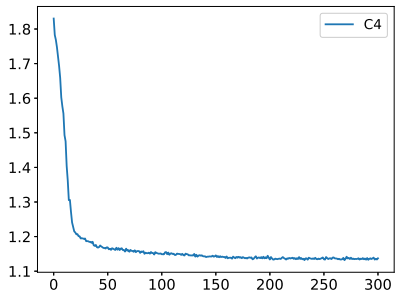


(C) All nine loads.

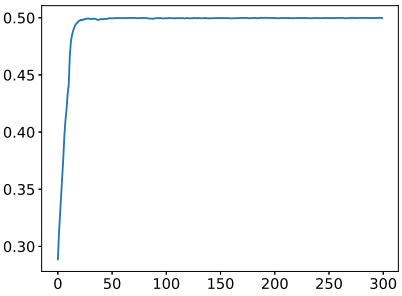
FIGURE 19. Optimized shapes for three possible configurations of the min/max optimization problem (6.19) of Section 6.3.3.



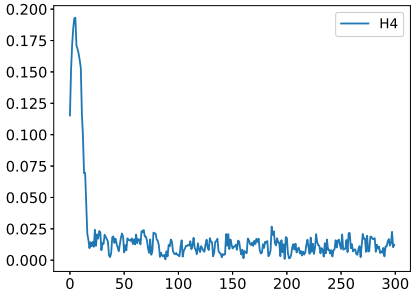
(A) Objective function m. Final value $m = 1.13$.



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

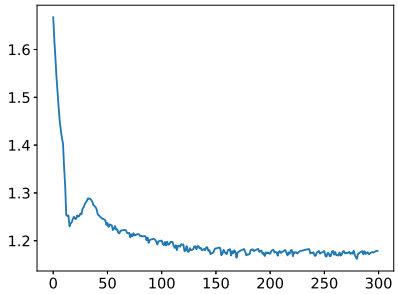


(C) Volume fraction Vol(Ω)/Vol(D). Final value: Vol(Ω)/Vol(D)=0.50.

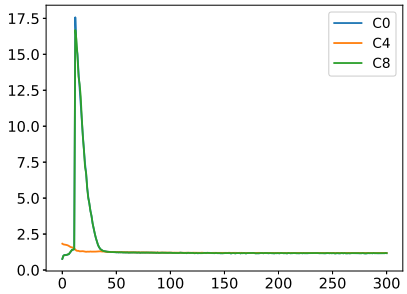


(D) Inequality constraint $H_i = C_i - m$

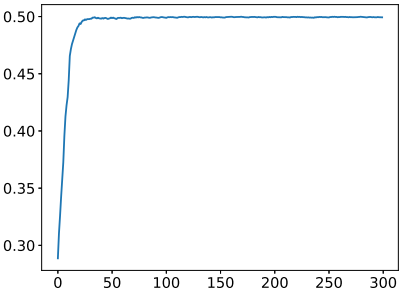
FIGURE 20. Convergence history curves for one load case of Section 6.3.3.



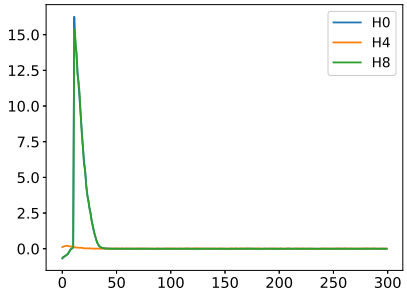
(A) Objective function m. Final value $m = 1.18$.



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

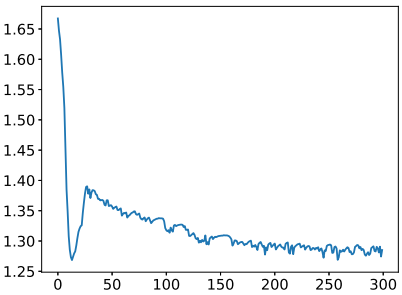


(C) Volume fraction Vol(Ω)/Vol(D). Final value: Vol(Ω)/Vol(D)=0.50.

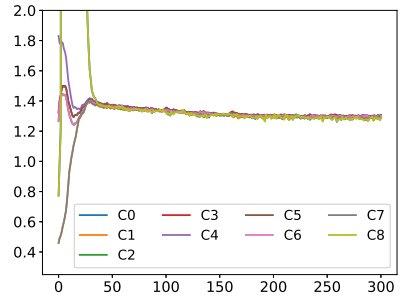


(D) Inequality constraints values: $H_i = C_i - m$.

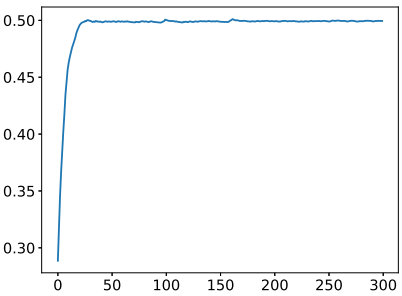
FIGURE 21. Convergence history curves for three load case of Section 6.3.3.



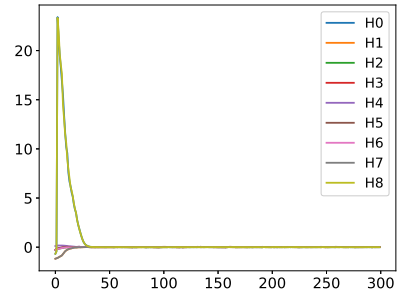
(A) Objective function m (eqn. (6.19))



(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 Vol(Ω)/Vol(D). Final value: Vol(Ω)/Vol(D)=0.50.



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

FIGURE 22. Convergence history curves for nine load case of Section 6.3.3.

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, E. Cancès, and J.-L. Vié, *Second-order shape derivatives along normal trajectories, governed by hamilton-jacobi equations*, Structural and Multidisciplinary Optimization, 54 (2016), pp. 1245–1266.

[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, 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.

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

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

[9] 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.

[10] 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.

[11] 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).

[12] C. Barbarosie, S. Lopes, and A.-M. Toader, *A gradient-type algorithm for constrained optimization with applications to multi-objective optimization of auxetic materials*, arXiv preprint arXiv:1711.04863, (2017).

[13] L. T. Biegler, *Nonlinear programming*, vol. 10 of MOS-SIAM Series on Optimization, Society for Industrial and Applied Mathematics (SIAM), 2010.

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

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] M. C. Delfour and J.-P. Zolâsio, *Shapes and geometries: metrics, analysis, differential calculus, and optimization*, vol. 22, Siam, 2011.

[21] 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.

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

[23] 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.

[24] 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.

[25] 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.

[26] F. Feppon, *Numerical methods for shape optimization of multiphysics systems*, PhD thesis, École polytechnique, In preparation.

[27] 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*, HAL preprint hal-01686770, (2018).

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

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

[30] 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.

[31] 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.

[32] 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.

[33] B. Mohammadi and O. Pironneau, *Applied shape optimization for fluids*, Oxford University Press, 2010.
[34] 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.
[35] F. Murat and J. Simon, *Sur le contrôle par un domaine géométrique, publications du Laboratoire d'Analyse Numérique*, Université Pierre et Marie Curie, (1976).
[36] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer Science, 35 (1999).
[37] 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.
[38] J. Schropp and I. Singer, *A dynamical systems approach to constrained minimization*, Numerical functional analysis and optimization, 21 (2000), pp. 537–551.
[39] V. H. Schulz, *A Riemannian view on shape optimization*, Found. Comput. Math., 14 (2014), pp. 483–501.
[40] 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.
[41] V. Shikhman and O. Stein, *Constrained optimization: projected gradient flows*, Journal of optimization theory and applications, 140 (2009), pp. 117–130.
[42] J. Sokolowski and J.-P. Zolesio, *Introduction to shape optimization*, in Introduction to Shape Optimization, Springer, 1992, pp. 5–12.
[43] 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.
[44] K. Svanberg, *The method of moving asymptotes—a new method for structural optimization*, Internat. J. Numer. Methods Engrg., 24 (1987), pp. 359–373.
[45] K. Tanabe, *A geometric method in nonlinear programming*, Journal of Optimization Theory and Applications, 30 (1980), pp. 181–210.
[46] 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.
[47] 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.
[48] H. Yamashita, *A differential equation approach to nonlinear programming*, Mathematical Programming, 18 (1980), pp. 155–168.
[49] Y.-x. Yuan, *A review of trust region algorithms for optimization*, in Iciam, vol. 99, Citeseer, 2000, pp. 271–282.
[50] 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.
[51] 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.