On Constraints in First-Order Optimization: A View from Non-Smooth Dynamical Systems

Michael Muehlebach MICHAELM@TUEBINGEN.MPG.DE
Learning and Dynamical Systems Group
Max Planck Institute for Intelligent Systems
72076 Tübingen, Germany

Michael I. Jordan JORDAN@CS.BERKELEY.EDU
Department of Electrical Engineering and Computer Sciences
Department of Statistics
University of California
Berkeley, CA 94720, USA

Abstract

We introduce a class of first-order methods for smooth constrained optimization that are based on an analogy to non-smooth dynamical systems. Two distinctive features of our approach are that (i) projections or optimizations over the entire feasible set are avoided, in stark contrast to projected gradient methods or the Frank-Wolfe method, and (ii) iterates are allowed to become infeasible, which differs from active set or feasible direction methods, where the descent motion stops as soon as a new constraint is encountered. The resulting algorithmic procedure is simple to implement even when constraints are nonlinear, and is suitable for large-scale constrained optimization problems in which the feasible set fails to have a simple structure. The key underlying idea is that constraints are expressed in terms of velocities instead of positions, which has the algorithmic consequence that optimizations over feasible sets at each iteration are replaced with optimizations over local, sparse convex approximations. The result is a simplified suite of algorithms and an expanded range of possible applications in machine learning.

Keywords: Convex optimization, nonconvex optimization, constrained optimization, non-smooth dynamical systems, gradient-based optimization, convergence rate analysis

1. Introduction

Optimization has played an essential role in machine learning in recent years, providing a conceptual and practical platform on which algorithms, systems, and datasets can be brought together at unprecedented scales. This joint platform has led to high-impact applications, the discovery of new phenomena, and the development of new theory. One of the major themes that have catalyzed the interplay between optimization and learning is that "simple is good." Whereas classical optimization has tended to focus on relatively complex schemes for determining update directions and step sizes, the recent focus of research at the learning/optimization interface has been on algorithms that use simple, stochastic approximations to first-order operators and that set step sizes via simple averaging schemes, or even employ constant step sizes. The simplifications have worked well in practice and have triggered the development of commodity software systems that are increasingly general and robust. They have also, appealingly, created new challenges for theoreticians, who have begun to develop new tools to fill in the gaps that the absence of strong assumptions has opened up.

Somewhat overlooked in all of these developments is the treatment of *constraints* in machine-learning problems. Machine-learning practitioners often handle constraints on parameters and predictions via simple, adhoc reparameterizations. This reflects the "simple is good" dictum, but it also creates a need to develop special-case reparameterizations in many cases and it poses additional challenges for theory, as convergence rates can be affected by the reparameterizations. More significantly, it overlooks the broader potential role that constrained optimization can play in machine learning. Moving beyond pattern recognition, emerging problems involving decision-making in real-world, multi-agent settings often involve contextual-driven constraints. Control-theoretic problems generally involve interactions with physical, biological, and social systems, whose laws are often expressed in terms of fundamental constraints. Mathematically, constraints can simplify statements of existence and uniqueness, simplify the specification of sets of solutions, and allow duality principles to be brought to bear.

There is a nascent thread of research on constrained optimization in machine learning that has aimed to build on the success of first-order methods. It has focused primarily on *projected gradient algorithms* and the *Frank-Wolfe method*. Both of these methods involve an inner loop nested inside of the overall procedure—in the former case the optimization of a quadratic function and in the latter case the optimization of a linear function. In both cases the optimization is over the entire feasible set. From a theoretical point of view, these are relatively simple methods, providing hooks such that convergence analyses from the unconstrained case can be readily brought to bear. Moreover, they can be easy to implement when the feasible set has a simple structure, such as a norm ball or a low-dimensional hyperplane. In these cases it is often possible to obtain closed-form expressions for the inner loop. This simplicity can disappear entirely, however, when the feasible set fails to have a simple structure. In such cases, optimizing a quadratic or linear function over the entire feasible set becomes prohibitive, and the "simple is good" dictum provides no clear path forward.

When the structure of the feasible set fails to enable closed-form projections or closed-form solutions for Frank-Wolfe updates, optimization theorists often turn to interior point or sequential quadratic programming methods. The idea of interior point methods is to reduce the constrained optimization problem to an unconstrained one by using barrier functions that assign a high cost to points close to the boundary of the feasible set. In sequential quadratic programming, the underlying nonlinear problem is approximated by a series of quadratic programs. While both classes of methods have been proposed for applications in machine learning (see, e.g., Koh et al., 2007; Ferris and Munson, 2003; Domahidi et al., 2012), they are significantly more complex than the stochastic-gradient methods that have been so successful in unconstrained machine learning. There remains a need for a learning-friendly approach to constrained optimization.

In the current paper, we present a class of first-order methods that are applicable to a wide range of problems in machine learning. A notable simplification of these methods, relative to classical constrained optimization methods, including projection methods and Frank-Wolfe, is that our methods rely exclusively on *local approximations of the feasible set*. These local approximations are a natural generalization of Clarke's tangent cone and are well defined for feasible and infeasible points. Moreover, as we will show, they make possible a key algorithmic simplification—they yield algorithms that converge even with a constant step size. Technically, they handle the case when the iterates become infeasible. This makes the resulting algorithmic procedure simple to implement and also ensures that the descent motion is not necessarily stopped as soon as a new constraint is violated. Finally, while the entire feasible set might be described by a very large (or even infinite) number of *nonlinear* constraints, the local approximation typically only includes a small number

of *linear* constraints, which substantially reduces the amount of computation required for a single iteration.

We believe that these simplifications make our methods a natural candidate for large-scale constrained machine-learning problems. Our main goal in the current paper is to provide a theoretical foundation to support such a claim. We also present results from a preliminary set of numerical experiments, which include, for example, randomly generated high-dimensional quadratic programs. Comparing the new methods to the interior point solver CVXOPT of Andersen et al. (2011), we find that the complexity of the new methods scales roughly with n^2 (where n is the problem dimension), whereas the complexity of the interior point solver scales with n^3. When n is large, this may lead to speedups of several orders of magnitude.

As our discussion has hinted, while our methods are relatively simple to specify and deploy, their analysis brings new challenges. Our treatment builds on recent progress in using continuous-time dynamical systems tools to analyze discrete-time algorithms in gradient-based optimization (Su et al., 2016; Wibisono et al., 2016; Diakonikolas and Jordan, 2021; Krichene et al., 2015; França et al., 2020; Betancourt et al., 2018; Muehlebach and Jordan, 2019, 2020, 2021). Much of the work in this vein is focused on understanding accelerated first-order optimization methods, such as Nesterov's algorithm, where the understanding arises by exposing links between differential and symplectic geometry, dynamical systems, and mechanics. These links, which supply a mechanical interpretation of accelerated methods and provide a rigorous interpretation of concepts such as "momentum," are often easiest to derive in continuous time, making use of variational, Hamiltonian, and control-theoretic perspectives. Indeed, the most complex part of these analyses often arises in the conversion from continuous time to discrete time.

In line with this recent literature, our treatment of constrained optimization also straddles the boundary between continuous time and discrete time. As in the unconstrained setting, the continuous case is relatively straightforward and the major challenges arise in the conversion to discrete time. Indeed, the key novelty is that in our constrained setting, the discrete-time function that maps one iterate to the next is *discontinuous*. Thus, tools such as smooth Lyapunov functions or the theory of monotone operators that have been widely employed in the unconstrained setting are not applicable in our setting, and a new analysis framework is needed. We develop such a framework by making use of ideas from *non-smooth mechanics*. Indeed, as we will discuss in the following section, the closest point of contact with existing literature is the notion of *Moreau time-stepping* in non-smooth mechanics.

Related work: In the following paragraphs we highlight some of the connections of our approach to the existing literature. Due to the wealth of work on constrained optimization over the last several decades, a comprehensive summary seems out of reach. We will therefore focus on ideas that are most closely related to our approach and refer to the textbooks of Bertsekas (1999), Nesterov (2004), Nocedal and Wright (2006), or Luenberger and Ye (2016) for a broader overview.

Our approach is in the spirit of projected gradient methodology. The basic idea of the projected gradient method is to compute a step along the negative gradient of the objective function and to project the resulting point back to the feasible set (see, e.g., Bertsekas, 1999, Ch. 2.3). From a theoretical point of view, the analysis of projected gradients strongly parallels that of unconstrained gradient descent. Indeed, by generalizing the notion of gradients to the "gradient mapping" (Nesterov, 2004, p. 86), arguments can be readily translated from the unconstrained to the constrained case. More generally, projected gradients can be viewed as an instance of a proximal point algo-

rithm (Parikh and Boyd, 2013), which itself can be elegantly described with the theory of monotone operators (Bauschke and Combettes, 2011; Rockafellar, 1976).

The key difference between our approach and classical projected gradients is that our approach is based on a local approximation of the feasible set. This local approximation includes only the active constraints[1] and is guaranteed to be a convex cone even if the underlying set is nonconvex. Our approach can be viewed as an inexact projected gradient method, and as such has similarities to the work of Wang and Liu (2006) and Birgin et al. (2003). However, in contrast to this work, we do not impose a monotone decrease of the cost function by an appropriate line search. In fact, our approach converges even with a constant step size, whereby the objective function fails to be monotonically decreasing (in general).

While projected gradient approaches have been successfully applied in various machine learning problems (see, e.g., Beck and Teboulle, 2011; Bloom et al., 2016), an even simpler algorithm—the Frank-Wolfe algorithm—has also received considerable attention in recent years (Jaggi, 2013). At each iteration of the Frank-Wolfe algorithm, a feasible descent direction is computed by maximizing the inner product with the negative gradient. This reduces to the minimization of a linear objective function over the feasible set, which, compared to projected gradients, can lead to considerable simplification. The simplification is in accord with the "simple is good" dictum of machine learning, and indeed it has been found that the Frank-Wolfe algorithm provides a unified theoretical framework for many greedy machine learning algorithms, including support vector machines, online estimation of mixtures of probability densities, and boosting (Clarkson, 2010). Recent results extend the Frank-Wolfe algorithm to the stochastic setting (Hazan and Kale, 2012; Zhang et al., 2020), or improve on its relatively slow convergence rate (Combettes and Pokutta, 2020; Garber and Hazan, 2015).

As we have already discussed, alternatives to projected gradients and Frank-Wolfe include interior point methods and sequential quadratic programming. Interior point methods provide practical solutions to many problems in constrained optimization, and they are guaranteed to return approximate solutions to many convex nonlinear programming problems in polynomial time (Nesterov and Nemirovskii, 1994). They can be particularly efficient if the underlying Karush-Kuhn-Tucker system is sparse, which can be exploited for simplifying the Newton updates (Domahidi et al., 2012). Similarly, in sequential quadratic programming, the underlying Karush-Kuhn-Tucker system resembles the Newton update of interior point methods. There are many different flavors of sequential quadratic programming, depending on the type of line search, whether only approximate second order information is used, or whether equality constraints are eliminated. An implementation that is widely used to solve complex optimal control and planning problems is presented in Gill et al. (2005). Recent advances in sequential quadratic programming share some similarity with our approach; see, for example, Torrisi et al. (2018) and Häberle et al. (2021). Both of these methods involve linearizing both the active and inactive constraints. The fact that all constraints are taken into account at each iteration enables the algorithms to anticipate constraint violations and distinguishes these approaches from the methods that will be discussed herein.

Finally, a main goal of the current paper is to bring to the fore an analogy between constrained optimization and non-smooth mechanics. Indeed, from a certain point of view, finding stationary points of a constrained optimization problem is equivalent to computing equilibria of a cor-

1. We say that the ith constraint is active at the iterate x_k if $g_i(x_k) \leq 0$, where the smooth function $g : \mathbb{R}^n \to \mathbb{R}^{n_g}$ describes the feasible set as $\{x \in \mathbb{R}^n \mid g(x) \geq 0\}$. It is important to note that this definition of active constraints does not require the corresponding dual multipliers to be nonzero.

responding non-smooth mechanical system. The classical approach to simulating such systems is *event-based integration*, which is a relatively complex algorithm that switches between smooth and non-smooth motion. An alternative is Moreau time-stepping (Moreau, 1988), which is based on the discretization of a measure-differential inclusion that captures the smooth and non-smooth parts of the motion. Moreau's algorithm can handle multiple (or even an infinite number of) discontinuities that may all happen within one time step. Further background can be found in the texts of Glocker (2001) and Studer (2009). Recent work in this area includes extensions to continuum mechanics (Capobianco and Eugster, 2018) and higher-order integration schemes (Acary, 2012).

Although we will exploit analogies to the simulation of physical systems, the focus of our theoretical analysis is in developing algorithms that efficiently compute approximate local minima of constrained nonlinear programming problems. In this setting, it will be crucial to consider large time steps, to handle constraint violations (which are often ignored when simulating non-smooth mechanical systems), and to provide convergence guarantees in discrete time.

Compared to classical treatments of constrained optimization, our treatment exhibits a key feature that arises directly from the physical analogy. Rather than expressing constraints at the language of positions or configurations, as is standard in optimization, our constraints will be expressed in terms of velocities. Thus, we will distinguish between constraints on the "position level" and constraints on the "velocity level." Our focus on the latter will be seen to lead directly to a local, convex approximation of the feasible set. By a constraint on velocity level, we mean a constraint on the forward increment $\lim_{\mathrm{d}t\downarrow 0}(x(t+\mathrm{d}t)-x(t))/\mathrm{d}t$ in continuous time or the difference $(x_{k+1}-x_k)/T$ in discrete time, where T is the step size. In continuous time, a given position constraint can (in most cases) be reformulated as an equivalent velocity constraint. However, this equivalence breaks down in discrete time, which necessitates a careful analysis of the resulting discrete-time algorithms. We also note that there are (many) mechanical systems that have velocity constraints which cannot be formulated as position constraints. For example, while ice skater can move to any position in a skating rink, their velocity is constrained to lie parallel to the blades of the skates.

Notation: We follow standard notation from convex analysis. In particular, $\mathbb{R}$ denotes the real numbers, $\mathbb{R}_{\geq 0}$ the nonnegative real numbers, $\mathbb{R}_{\leq 0}$ the nonpositive real numbers, and $\mathbb{Z}$ the set of all integers. The notation $|\cdot|$ is reserved for the Euclidean norm or the cardinality of a set. The gradient of a function $h : \mathbb{R}^n \to \mathbb{R}^m$ is denoted by $\nabla h : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ and the indicator function of the set C is referred to as $\psi_C : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$, that is, $\psi_C(x)$ takes the value zero for $x \in C$ and ∞ otherwise. The subgradient of a convex function $g : \mathbb{R}^n \to \mathbb{R}$ evaluated at $x \in \mathbb{R}^n$ is denoted by $\partial g(x)$ and is defined as the set $\{v \in \mathbb{R}^n \mid v^{\mathsf{T}}(y-x) \leq g(y) - g(x),\ \forall y \in \mathbb{R}^n\}$. The tangent cone (in the sense of Clarke) at any point $x \in C$ is referred to as $T_C(x)$, that is, $\delta x \in T_C(x)$ if there exist two sequences $x_j \to x$, $x_j \in C$, $t_j \downarrow 0$, such that $(x_j - x)/t_j \to \delta x$. The corresponding normal cone is denoted by $N_C(x) := \{\lambda \in \mathbb{R}^n \mid \lambda^{\mathsf{T}}\delta x \leq 0, \forall \delta x \in T_C(x)\}$. Finally, we use subscripts to denote both single components of a vector and the iteration number of a discrete algorithm. The distinction will be made from context (we usually reserve the subscript k for the iteration number).

2. Overview of the Results

We consider the following optimization problem:

$$\min_{x\in\mathbb{R}^n} f(x), \quad \text{s.t.} \quad g(x) \geq 0, h(x) = 0, \tag{1}$$

where the function $f : \mathbb{R}^n \to \mathbb{R}$ defines the objective function, the functions $g : \mathbb{R}^n \to \mathbb{R}^{n_\mathrm{g}}$ and $h : \mathbb{R}^n \to \mathbb{R}^{n_\mathrm{h}}$ define the constraints, and where n, n_g, and n_h are positive integers. The function f is assumed to be such that $f(x) \to \infty$ for $|x| \to \infty$. We denote the set of all $x \in \mathbb{R}^n$ that satisfy the constraints $g(x) \geq 0$ and $h(x) = 0$ by C, which we assume to be non-empty and bounded. Combined with the properties of f this guarantees that the minimum in (1) is attained. The functions f, g, and h are continuously differentiable and have a Lipschitz continuous gradient.

Brief summary of the main contributions: In mathematical optimization constraints are typically treated by direct reference to positions, meaning that x_k or $x(t)$ are constrained to lie in C for all $k \geq 0$ or all $t \geq 0$, respectively. We adopt a fundamentally different point of view—instead of constraining $x(t)$ or x_k, we constrain the forward velocity $\dot{x}(t)^+ = \lim_{\mathrm{d}t \downarrow 0}(x(t+\mathrm{d}t) - x(t))/\mathrm{d}t$ or forward increments $(x_{k+1} - x_k)/T$. At a given position $x \in \mathbb{R}^n$, the set of all admissible velocities will be denoted by $V_\alpha(x) \subset \mathbb{R}^n$. When $x \in C$, the set $V_\alpha(x)$ corresponds to the tangent cone of the set C at x. We will introduce an appropriate generalization of $V_\alpha(x)$ in order to also capture cases in which $x \notin C$. The two different point of views on constraints are illustrated in Figure 1.

In continuous time, the resulting velocity constraint is equivalent to the original position constraint, assuming constraint qualification. However, this equivalence breaks down in discrete time, and may lead to infeasible iterates over the course of the optimization. One of our main results is a guarantee that the resulting discrete algorithm nonetheless converges to stationary points, despite the possibility of infeasible iterates and despite the discontinuous nature of the map from x_k to x_{k+1}. In addition to providing such a guarantee, we derive rates of convergence and we show that a formulation of constraints on the velocity level can lead to computational advantages. In particular, we show that at each iteration, only a *linear* and *convex* approximation of the original *nonlinear* and *nonconvex* feasible set needs to be considered. Moreover, the linear approximation includes only the constraints that are active at $x(t)$ or x_k. On randomly generated dense quadratic programs, for example, the complexity of the proposed method scales with n^2 (empirically), which contrasts with state-of-the-art implementations of an interior point method, which scale with n^3. Moreover, in many practical problems (for example, support vector machines) the proposed algorithm greatly reduces the number of constraints that must be considered at each iteration.

Detailed summary of the main contributions: In order to discuss the results in greater detail, we introduce the following definition and assumption, which will hold throughout the remainder of the article.

Definition 1 *The point $x \in \mathbb{R}^n$ satisfies the* Mangasarian-Fromovitz constraint qualification *if the columns of $\nabla h(x)$ are linearly independent and if there exists a vector $w \in \mathbb{R}^n$ such that $\nabla h(x)^\mathsf{T} w = 0$ and $\nabla g_i(x)^\mathsf{T} w > 0$ for all $i \in I_x$, where I_x denotes the set of active inequality constraints at x, i.e., $I_x := \{i \in \mathbb{Z} \mid g_i(x) \leq 0\}$.*[1]

Assumption 1 *The Mangasarian-Fromovitz constraint qualification is satisfied for all $x \in \mathbb{R}^n$.*

From the definition of $T_C(x)$ it follows that every $\delta x \in T_C(x)$ satisfies $\nabla h(x)\delta x = 0$ and $\nabla g_i(x)^\mathsf{T} \delta x \geq 0$, for all $i \in I_x$. Assumption 1 ensures that the converse is also true, which guarantees that all stationary points of (1) satisfy the corresponding Karush-Kuhn-Tucker conditions. We

1. We would like to point out that our definition of active constraints does not require constraints to have corresponding dual multipliers that are nonzero.

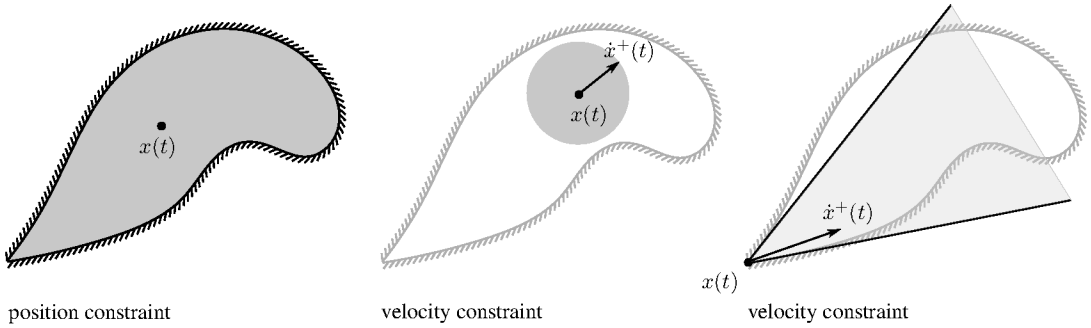

Figure 1: The figure contrasts position constraints with velocity constraints. The leftmost sketch illustrates the position constraint, where $x(t)$ is constrained to the feasible set as indicated by the shaded region. The center and right figure illustrate the induced constraints on the velocity $\dot{x}(t)^+$ (which will be precisely defined below). If $x(t)$ is in the interior of the feasible set, there are no restrictions on the forward velocity, as indicated with the shaded ball without border (center). The figure on the right illustrates the case where $x(t)$ lies on the boundary of the feasible set. As a result, $\dot{x}(t)^+$ is constrained to lie in the cone indicated by the shaded region. In the discrete-time case $\dot{x}(t)^+$ is replaced with $(x_{k+1} - x_k)/T$.

further introduce the set

$$V_\alpha(x) := \{v \in \mathbb{R}^n \mid \nabla h(x)^\mathsf{T} v + \alpha h(x) = 0, \quad \nabla g_i(x)^\mathsf{T} v + \alpha g_i(x) \geq 0, \ \ \forall i \in I_x\}, \tag{2}$$

where $\alpha \geq 0$ is a positive scalar. The role of α will be discussed below. As a result of the constraint qualification, the set $V_\alpha(x)$ reduces to the tangent cone $T_C(x)$ of the set C for any $x \in C$. Moreover, for a fixed $x \in \mathbb{R}^n$, $V_\alpha(x)$ is a convex polyhedral set, involving only the active constraints I_x.

With the notation in place, we are ready to state our main results. We start with a general framework based on a continuous-time gradient flow which will be used as a starting point for our discrete algorithm.

Proposition 2 *(constrained gradient flow) Let $x : [0, \infty) \to \mathbb{R}^n$ be an absolutely continuous trajectory, which has a piecewise continuous derivative. Then, for any $x(0) \in C$, the following are equivalent:*

$$\dot{x}(t) = -\nabla f(x(t)) + R(t), \quad -R(t) \in N_C(x(t)), \qquad \forall t \in [0, \infty) \text{ a.e.}, \tag{3}$$

$$\dot{x}(t)^+ = -\nabla f(x(t)) + R(t), \quad -R(t) \in \partial \psi_{V_\alpha(x(t))}(\dot{x}(t)^+), \quad \forall t \in [0, \infty), \tag{4}$$

$$\dot{x}(t)^+ = \operatorname*{argmin}_{v \in V_\alpha(x(t))} \frac{1}{2}|v + \nabla f(x(t))|^2, \qquad \forall t \in [0, \infty), \tag{5}$$

where $\dot{x}(t)^+$ denotes the right-hand derivative of x at t.

For any $x(0) \in \mathbb{R}^n$, (4) *and* (5) *are equivalent and lead to a unique trajectory $x(t)$ (if it exists), which is guaranteed to converge to the set of stationary points of* (1) *(for $\alpha > 0$); that is, $x(t) \to C$*

as $t \to \infty$ *and*

$$\lim_{t\to\infty} | - \nabla f(x(t)) + R(t)| = 0,$$

where $R(t)$ *is defined in* (4)*. Moreover, if the stationary points are isolated, the trajectory* $x(t)$ *converges to a single stationary point.*

When C *is convex,* f *is strongly convex with strong convexity constant* μ*, and* $\alpha \leq 2\mu$*, the trajectory satisfying* (4) *and* (5) *converges exponentially:*

$$(h(x(0)), \min\{0, g(x(0))\})^\mathsf{T} \lambda^* e^{-\alpha t} \leq f(x(t)) - f^* \leq (f(x(0)) - f^*) e^{-2\mu t}, \tag{6}$$

for all $x(0) \in \mathbb{R}^n$*, where* f^* *is the value of the minimizer in* (1) *and* λ^* *is a multiplier that satisfies the Karush-Kuhn-Tucker conditions.*

We make the following remarks:

- The first condition, (3), amounts to a differential inclusion, whereas (4) and (5) give rise to differential equations that have a discontinuous right-hand side. We restrict ourselves from the outset to piecewise smooth motion, that is, trajectories that are absolutely continuous and have a piecewise continuous derivative. Absolute continuity means that $x(t) - x(0)$ can be expressed as the Lebesgue integral over the velocity $\dot{x}$; that is, $x(t) = x(0) + \int_0^t \dot{x}(\tau) \mathrm{d}\tau$ for all $t \geq 0$. The assumption that $\dot{x}$ is piecewise continuous means that on any finite interval, $\dot{x}$ is continuous except at a finite number of points, where left and right limits, denoted by $\dot{x}(t_0)^-$ and $\dot{x}(t_0)^+$, are well-defined. The value $\dot{x}(t_0)$ at the discontinuity t_0 is of no interest and may or may not exist.

- The assumptions on $\dot{x}$ are used for establishing the equivalence between (3) and (4). Convergence results for (4) and (5) similar to those of Proposition 2 can still be obtained when the restrictions on $\dot{x}$ are relaxed. We also note that by applying the theory of Filippov (1988), (4) and (5) can be extended to a differential inclusion that is guaranteed to have an absolutely continuous solution. We refer the reader who is interested in existence results to the work of Filippov (1988) and Aubin and Cellina (1984). The equivalence between (3) and (4) under weaker assumptions on $\dot{x}$ is discussed in Brogliato et al. (2006), which also provides a short existence proof (requiring, however, that C is convex).

- The variable $R(t)$ in (3) can be regarded as a reaction force that imposes the constraint $x(t) \in C$ for all $t \in [0, \infty)$ (by definition, the normal cone is empty if $x(t) \notin C$). We therefore say that (3) includes the constraint on the position level. In contrast, the reaction force $R(t)$ in (4) enforces $\dot{x}(t)^+ \in V_\alpha(x(t))$ for all $t \in [0, \infty)$, which reduces to $\dot{x}(t)^+ \in T_C(x(t))$ for $x(t) \in C$. The condition $\dot{x}(t)^+ \in V_\alpha(x(t))$ can be viewed as an extension of $\dot{x}(t)^+ \in T_C(x(t))$ to allow also for $x(t) \notin C$. Interpreting (4) as a stationarity condition for $\dot{x}(t)^+$ yields (5). We therefore say that (4) and (5) impose the constraints on the velocity level.

- The intuition behind the equivalence of (3), (4), and (5) can be summarized in the following way. For an absolutely continuous trajectory $x(t)$, the constraint $x(t) \in C$ for all $t \in [0, \infty)$ is equivalent to $\dot{x}(t)^+ \in V_\alpha(x(t))$ for all $t \in [0, \infty)$, $x(0) \in C$, (Moreau, 1988, Remark 2.5).[1] If we think of $x(t)$ as the position of a point mass, and $\dot{x}(t)^+$ as its velocity, this can

1. Constraint qualification is needed for the equivalence to hold.

be stated as follows: A constraint on the position of the point mass induces a constraint on its velocity. Conversely, the constraint $\dot{x}(t)^+ \in V_\alpha(x(t))$ on the velocity ensures that the position constraint is satisfied for all times $t \geq 0$, provided that $x(0) \in C$.

- The reformulation (5) emphasizes that at each point in time, the velocity is chosen to match unconstrained gradient flow as closely as possible, subject to the velocity constraint $\dot{x}(t)^+ \in V_\alpha(x(t))$ (which for feasible $x(t)$ reduces to $\dot{x}(t)^+ \in T_C(x(t))$). This can be seen as an analogue of the principle of least constraint in mechanics (Glocker, 2001, Ch. 9).

- The set $V_\alpha(x)$ can be viewed as an extension of $T_C(x)$ to all of $\mathbb{R}^n$. This enables a generalization of constrained gradient flow, according to (4) and (5), which accounts for infeasible initial conditions. Imposing $\dot{x}(t)^+ \in V_\alpha(x(t))$ for all $t \in [0, \infty)$, concludes, by definition of the set $V_\alpha(x)$ and by applying Grönwall's inequality,

$$g_i(x(t)) \geq g_i(x(0))e^{-\alpha t}, \quad i \in I_{x(0)}, \quad h(x(t)) = h(x(0))e^{-\alpha t}, \tag{7}$$

  for all $t \in [0, \infty)$. Consequently, the constant α controls how quickly the constraint violations decay.

- By reformulating the constraint on the velocity level as in (4) and (5), the velocity $\dot{x}(t)^+$ can by computed by relying on a local and linear approximation of the set C via $\dot{x}(t)^+ \in V_\alpha(x(t))$, which includes only the active constraints $I_{x(t)}$. Hence, even for a nonconvex optimization problem such as (1), the optimization given by the right-hand side of (5) is convex.

By replacing $x(t)^+$ with $(x_{k+1} - x_k)/T$ and $x(t)$ with x_k in (4) or (5), we obtain the following discrete algorithm

$$x_{k+1} = x_k - T\nabla f(x_k) + TR_k, \quad -R_k \in \partial\psi_{V_\alpha(x_k)}((x_{k+1} - x_k)/T), \quad k = 0, 1, 2, \ldots, \tag{8}$$

which for any $x_0 \in \mathbb{R}^n$, leads to well-defined (unique) iterates, as long as the Mangasarian-Fromovitz constraint qualification is satisfied for all $x \in \mathbb{R}^n$. As in the continuous-time setting, the discrete algorithm relies on a local approximation of the feasible set at each iteration, which includes only the active constraints I_{x_k}. Projections or optimization over the entire feasible set C (at each iteration) are therefore avoided. While this reduces computation, it also complicates the analysis.

It is important to note that (8) can be reformulated in a number of equivalent ways. The choice made in Algorithm 1 is particularly suitable for numerical implementation.

The following definitions will be useful for characterizing the behavior and the convergence rate of (8). We start by introducing the function $v : \mathbb{R}^n \to \mathbb{R}^n$, which assigns the velocity $v(x)$ to each $x \in \mathbb{R}^n$:

$$v(x) := \operatorname*{argmin}_{v \in V_\alpha(x)} \frac{1}{2}|v + \nabla f(x)|^2. \tag{9}$$

Clearly, in continuous time, (4) and (5) evolve as $\dot{x}(t)^+ = v(x(t))$, whereas in discrete time, (8) imposes $(x_{k+1} - x_k)/T = v(x_k)$. As a result of the constraint qualification, strong duality holds and we obtain the following dual

$$d(x) := \max_{\lambda \in D_x} l(x, \lambda) - \frac{1}{2\alpha}|\nabla_x l(x, \lambda)|^2, \tag{10}$$

where ∇_x denotes the gradient with respect to x, and the Lagrangian $l : \mathbb{R}^n \times (\mathbb{R}^{n_\mathrm{h}} \times \mathbb{R}^{n_\mathrm{g}}_{\geq 0}) \to \mathbb{R}$ is defined as

$$l(x, \lambda) := f(x) - \lambda^\mathsf{T} \bar{g}(x), \tag{11}$$

with $\bar{g}(x) := (h(x), g(x))$. The set D_x in (10) is given by

$$D_x := \{\lambda \in \mathbb{R}^{n_\mathrm{h}} \times \mathbb{R}^{n_\mathrm{g}}_{\geq 0} \mid \lambda_{n_\mathrm{h}+i} = 0, \forall i \notin I_x\},$$

and includes only multipliers $\lambda_i \neq 0$ that correspond to equality constraints or active inequality constraints, defined by $i \in I_x$. The multipliers $\lambda_{n_\mathrm{h}+i}$, which correspond to inactive inequality constraints, i.e., $i \notin I_x$, are set to zero, and can therefore be eliminated from the outset when solving (10) (as is done in Algorithm 1). In general, there might be multiple $\lambda \in D_x$ that attain the maximum in (10). We will denote any one of them by $\lambda(x)$. As a consequence of Lagrange duality, $\lambda(x)$ is related to the minimizer of (9) by

$$v(x) = -\nabla_x l(x, \lambda(x)) = -\nabla f(x) + W(x)\lambda(x), \tag{12}$$

where $W(x) := \nabla \bar{g}(x)$. We also note that the variable $R(t)$ in (4) or R_k in (8) can therefore be expressed as $R(t) = W(x(t))\lambda(x(t))$ and $R_k = W(x_k)\lambda(x_k)$, respectively.

The maximum curvature of f (the Lipschitz constant of ∇f) limits the maximum admissible step size of gradient descent in the unconstrained case. We will see that the maximum curvature of $l(\cdot, \lambda)$ (for a fixed λ) will play a similar role for (8). We denote by $\bar{\mu}_l(\lambda)$ and $\bar{L}_l(\lambda)$ the smoothness and strong convexity constant of $l(\cdot, \lambda) : \mathbb{R}^n \to \mathbb{R}^n$ (for a fixed λ). In case C is convex and f is strongly convex, the strong convexity constant μ of f is a natural lower bound for $\bar{\mu}(\lambda)$, $\lambda \in \mathbb{R}^{n_\mathrm{h}} \times \mathbb{R}^{n_\mathrm{g}}_{\geq 0}$, which is attained for $\lambda = 0$.

We will also consider modifications of (1), where some inequality constraints are removed. The resulting optimal costs are denoted by

$$f_I^* := \min_{x \in \mathbb{R}^n} f(x) \quad \text{s.t.} \quad h(x) = 0, \quad g_i(x) \geq 0, \quad i \in I, \tag{13}$$

where I is any subset of $\{1, \ldots, n_\mathrm{g}\}$. The minimum in (13) is guaranteed to be attained, due to the assumptions on f and C. It is clear that $f^*_{\{\}} \leq f^*_I \leq f^*$ and we will denote any choice of multipliers that satisfy the Karush-Kuhn-Tucker conditions of (13) by λ^*_I.

With this notation in place, we are now ready to state the main results that characterize (8).

Proposition 3 *(constrained gradient descent) Let C be convex and let f be strongly convex with strong convexity constant μ. Then, for any $x_0 \in \mathbb{R}^n$, the iterates x_k of* (8) *are well-defined (unique) and guaranteed to converge to the minimizer of* (1) *for*

$$T \leq \frac{2}{L_l + \mu}, \quad \alpha < \mu,$$

where L_l is such that $L_l \geq \bar{L}_l(\lambda(x))$ for all $x \in \mathbb{R}^n$.[1]

The velocity $(x_{k+1} - x_k)/T$ converges with

$$\min_{j \in \{0,1,\ldots,k\}} |-\nabla f(x_j) + R_j|^2 \leq \frac{f^* - d(x_0)}{c_1(k+1)}, \quad \forall k \geq 0, \quad \forall x_0 \in \mathbb{R}^n,$$

1. In case g is affine, $L_l = L$, where L is the smoothness constant of f.

where $c_1 = T(\mu/\alpha - 1)(1 - \mu T/2) > 0$ *is constant.*

The sequence $d(x_k)$ *is monotonically increasing in* k *and it holds that* $d(x_k) \leq f^*_{I_{x_k}} \leq f^*$. *Each level set* $\{x \in \mathbb{R}^n \mid d(x) \geq f^*_I\}$, *where* I *is any subset of* $\{1, 2, \ldots, n_g\}$, *is closed, invariant and attractive. Provided that* $L_l \geq \bar{L}_l(\lambda^*_{I_{x_k}})$, *the trajectories converge at a linear rate:*

$$d(x_{k+1}) - f^*_{I_{x_k}} \geq (1 - c_2 T)(d(x_k) - f^*_{I_{x_k}}),$$

where $c_2 = 2\alpha(1 - \mu T/2)(\mu - \alpha)/(L_l - \alpha) > 0$ *is constant.*

Algorithm 1 Implementation of the gradient descent scheme (8).

Require: $x_0 \in \mathbb{R}^n$, TOL, MAXITER, $T > 0$, $\alpha T \in (0, 1]$
  $k = 0$
  while $k <$ MAXITER **do**
    Determine the set of closed constraints I_{x_k}
    Define $W_k := (\nabla h(x_k), \nabla g_i(x_k)_{i \in I_{x_k}})$ and $D_k := \mathbb{R}^{n_\mathrm{h}} \times \mathbb{R}^{|I_{x_k}|}_{\geq 0}$
    Define $\bar{g}_k := (h(x_k), g_i(x_k)_{i \in I_{x_k}})$
    Find $\lambda_k \in D_k$ such that $-\lambda_k \in \partial \psi_{D_k}(W_k^\mathsf{T} W_k \lambda_k - W_k^\mathsf{T} \nabla f(x_k) + \alpha \bar{g}_k)$ (see Section 6, (27))
    Perform the update $x_{k+1} = x_k - T\nabla f(x_k) + T W_k \lambda_k$
    if $|x_{k+1} - x_k| \leq T \cdot$ TOL **then**
      return x_{k+1}
    end if
    $k \leftarrow k + 1$
  end while

The following remarks are important:

- Algorithm (8) does not anticipate any constraints that could potentially be violated at future iterations. Unlike in the continuous-time case, where constraint violations decrease exponentially over time (see (7)), a constraint may therefore open up, and close again a few iterations later. Nevertheless, the algorithm is guaranteed to converge at nearly a linear rate, which we find remarkable.

- The convergence rate is dimension independent, which distinguishes the algorithm from interior-point methods, for example, where $\mathcal{O}(\sqrt{n_\mathrm{g}})$ Newton-iterations are required to decrease the value of the objective function by a constant factor.

- In the important special case where constraints are affine, all the above results hold for $L_l = L$, where L is the smoothness constant of f. The constant L_l is related to the maximum curvature of the Lagrangian, which seems a natural generalization from the unconstrained to the constrained case.

- Another important special case is given for a single nonlinear inequality constraint ($n_\mathrm{g} = 1$, $n_\mathrm{h} = 0$). We then obtain

$$\lambda(x) = \begin{cases} \frac{\nabla g(x)^\mathsf{T} \nabla f(x) - \alpha g(x)}{|\nabla g(x)|^2} & \text{for } g(x) \leq 0, \;\; \nabla g(x)^\mathsf{T} \nabla f(x) - \alpha g(x) \geq 0, \\ 0 & \text{otherwise.} \end{cases}$$

In this case, the constant L_l is given by the largest eigenvalue of the Hessian $\mathrm{d}^2 l/\mathrm{d}x^2 = \mathrm{d}^2 f/\mathrm{d}x^2 - \lambda(x)\, \mathrm{d}^2 g/\mathrm{d}x^2$ over all $x \in \mathbb{R}^n$.

- The restriction $\alpha < \mu$ on the constant α is likely to be conservative. We observed in numerical experiments that a choice αT close to unity yields faster convergence. The restriction $\alpha T \leq 1$ is, however, necessary for convergence.

- The convergence analysis will point to immediate extensions and variants of (8), which include line-search strategies, or alternations between gradient updates of the Lagrangian with fixed multipliers (which are computationally inexpensive) and updates of the multipliers according to (10). These extensions will be discussed in Section 5.1.

The remainder of the article is concerned with proving Proposition 2 and Proposition 3, providing context for both algorithms, discussing a particular implementation of Algorithm 1, and illustrating the algorithms with numerical examples.

3. Motivation

The continuous-time formulation given in Proposition 2 can be motivated by drawing analogies to non-smooth mechanics. We will start by viewing the stationarity conditions of (1) as the static equilibrium of a mechanical system. We will then apply d'Alembert's principle (see, e.g., Lanczos, 1952), which relates this variational characterization of equilibria to the variational characterization of motion. In the context of optimization, this leads to the algorithm (3), and also enables generalizations to accelerated first-order methods or Newton-type methods. We further establish the equivalence between (3) and (4), which, in the context of mechanics, can be related to the equivalence between the principle of virtual work and the principle of virtual power. We then discuss various interpretations of (4), which lie at the heart of the discretization in (8).

We consider a mechanical system that consists of a point mass located at $x \in \mathbb{R}^n$ on which the external force $F := -\nabla f(x)$ acts. The point mass is constrained to the set C.[1] For a given $\bar{x} \in C$ we start by investigating whether the point mass is in static equilibrium; i.e., it does not move under the influence of the external force and the constraint $x \in C$. In order to do so, we isolate the point mass and replace the interaction with the constraint by a (constraint) force, $-R \in N_C(\bar{x})$. The corresponding graphical procedure, often referred to as free-body diagram, is illustrated in Figure 2. The principle of virtual work, which is the fundamental postulate of classical mechanics, can now be stated.

Postulate 1 *The point mass is in static equilibrium if and only if the virtual work vanishes for any virtual displacement $\delta x \in \mathbb{R}^n$. The* virtual work *is defined as $(F+R)^\mathsf{T} \delta x$, where F is the external force and $R \in -N_C(x)$ the constraint force.*

Due to the fact that arbitrary virtual displacements are allowed, Postulate 1 concludes that the point mass is in static equilibrium at $\bar{x} \in C$ if the following conditions are fulfilled

$$-\nabla f(\bar{x}) + R = 0, \quad -R \in N_C(\bar{x}). \tag{14}$$

1. From a physical perspective the constraint can be thought of as a second rigid body with infinite mass that consists of all points $\mathbb{R}^n \setminus C$. We seek to model the interaction between the point mass and the constraint.

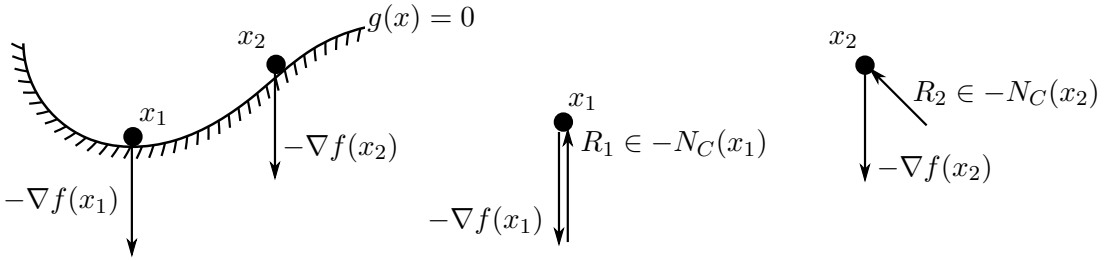

Figure 2: The figure illustrates the concept of a free-body diagram, where the geometric boundary condition $g(x) \geq 0$, as shown on the left, is replaced by the constraint forces, $-R \in N_C(x)$, as shown on the right. We note that x_1 is in static equilibrium, since $-\nabla f(x_1)$ and R_1 cancel, whereas x_2 is not.

By virtue of the constraint qualification, these are equivalent to the Karush-Kuhn-Tucker conditions of (1). Thus, with our choice $F := -\nabla f(x)$, we can relate the stationarity conditions of (1) to the static equilibrium of a mechanical system, as characterized by the principle of virtual work.

The connections to optimization are even more explicit when restricting ourselves to admissible virtual displacements; i.e., $\delta x \in T_C(\bar{x})$. By definition, constraint forces satisfy $-R^\mathsf{T}\delta x \leq 0$ for all $\delta x \in T_C(\bar{x})$ or, in the language of classical mechanics, constraint forces are such that their contribution to the virtual work is nonnegative.[1] This leads to the principle of d'Alembert-Lagrange, which represents the cornerstone of Lagrangian mechanics.

Corollary 4 *If the point mass located at $\bar{x} \in C$ is in static equilibrium, the virtual work of the external forces satisfies $F^\mathsf{T}\delta x \leq 0$ for all admissible variations $\delta x \in T_C(\bar{x})$.*

Through the lens of optimization, this means that $-\delta f = -\nabla f(\bar{x})^\mathsf{T}\delta x \leq 0$ for all admissible variations $\delta x \in T_C(\bar{x})$, or equivalently, $f(\bar{x}) \leq f(x)$ for all x in an open neighborhood of $\bar{x}$ with $x \in C$. The relations are summarized in Figure 3 (left).

The important insight from classical mechanics (essentially due to d'Alembert) is that the principle of virtual work, Postulate 1, and the principle of d'Alembert-Lagrange, Corollary 4, naturally extend from the static equilibrium to the dynamic equilibrium that characterizes the motion of a mechanical system. It suffices to add the "forces of inertia," which for the point mass amounts to adding $-m\ddot{x}$ to the external forces F (Lanczos, 1952, Ch. 4). We will apply these ideas to gradient flow, where the "forces of inertia" are given by $-\dot{x}$. This yields (3), which we restate as follows:

$$\dot{x}(t) = -\nabla f(x(t)) + R(t), \quad -R(t) \in N_C(x(t)), \quad \forall t \in [0, \infty) \text{ a.e.}$$

We restrict ourselves to trajectories $x : [0, \infty) \to \mathbb{R}^n$ that are absolutely continuous and that have a derivative (almost everywhere) which is piecewise continuous. The former requirement ensures that a change in position, $x(t) - x(0)$, can be expressed as the Lebesgue integral of the velocity:

1. In most classical textbooks only equality constraints are considered. In that case, constraint forces exert no virtual work.

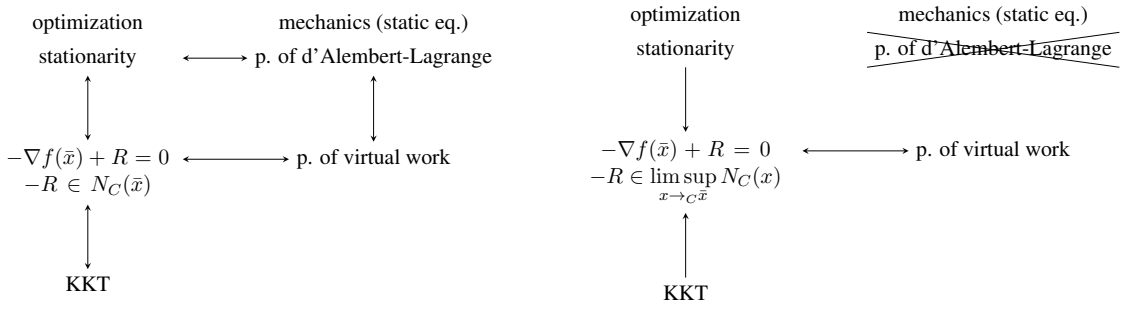

Figure 3: The figure summarizes the analogies between constrained optimization and non-smooth mechanics. On the left, constraint qualifications are assumed to hold ensuring that the set C is regular in the sense of Clarke. On the right, the set C fails to be regular, for example due to a reintrant (inward facing) corner. In that case, the notion of equilibrium needs to be extended by an appropriate closure of $N_C(x)$; see, for example, Rockafellar and Wets (1997, Ch. 6). The resulting equilibrium condition is no longer sufficient for stationarity and its equivalence to the Karush-Kuhn-Tucker conditions breaks down, (Rockafellar and Wets, 1997, Thm. 6.14). Moreover, the principle of d'Alembert-Lagrange is no longer a consequence of the principle of virtual work and therefore fails to characterize static equilibria when C is not regular (May and Panagiotopoulous, 1985). There are important examples of mechanical systems where C fails to be regular; see, for example, Glocker (2001, Ch. 11).

$x(t) = x(0) + \int_0^t \dot{x}(\tau)\mathrm{d}\tau$. The latter requirement implies that the velocity is almost everywhere continuous. On any finite interval, the velocity has a finite number of discontinuities, where left and right limits exist.

The condition (3) can still be viewed as a force balance between $\dot{x}(t) + \nabla f(x(t))$ and $R(t)$, whereby the reaction force $R(t)$ ensures that $x(t)$ remains feasible. Moreover, when the system is at rest, $\dot{x}$ vanishes and (3) reduces to the Karush-Kuhn-Tucker conditions (14). If $x(t)$ happens to be in the interior of C, the reaction force $R(t)$ vanishes, and $x(t)$ evolves according to unconstrained gradient flow. The almost everywhere quantifier is clearly needed—if $x(t)$ approaches the boundary of the set C, an instantaneous velocity jump might be required for ensuring that $x(t)$ remains in C (at the time instant of the velocity jump, $\dot{x}$ is no longer defined).

As mentioned in Section 2, the constraint $x(t) \in C$ for all $t \in [0, \infty)$ can be reformulated as a constraint on the velocity, i.e., $\dot{x}(t)^+ \in V_\alpha(x(t))$. This forms the basis for the equivalence between (3) and (4):

Proposition 5 *(Similar to Moreau (1988, Prop. 5.1), Glocker (2001, Ch. 7)) Let $x : [0, \infty) \to \mathbb{R}^n$, $x(0) \in C$, be an absolutely continuous trajectory that has a piecewise continuous derivative. Then, $x(t)$ satisfies* (3) *if and only if it satisfies* (4)*:*

$$\dot{x}(t)^+ = -\nabla f(x(t)) + R(t), \quad -R(t) \in \partial \psi_{V_\alpha(x(t))}(\dot{x}(t)^+), \quad \forall t \in [0, \infty).$$

Proof The proof is adapted from Moreau (1988, Prop. 5.1). We start by assuming that $x(t)$ satisfies (4). The fact that the subdifferential of the indicator function is non-empty implies that $\dot{x}(t)^+ \in$

$V_\alpha(x(t))$ for all $t \in [0, \infty)$. Combined with $x(0) \in C$, we therefore have $x(t) \in C$ for all $t \in [0, \infty)$, and $V_\alpha(x(t)) = T_C(x(t))$. In addition, it follows from the definition of the subdifferential that $-R(t)^\mathsf{T}(v - \dot{x}(t)^+) \leq 0$ for all $v \in T_C(x(t))$. Due to the fact that $T_C(x(t))$ is a cone, this implies $-R(t)^\mathsf{T} v \leq 0$ for all $v \in T_C(x(t))$ (otherwise we could derive a contradiction by scaling an appropriate $v \in T_C(x(t))$), or in other words, $-R(t) \in N_C(x(t))$. This shows that any $x(t)$ with $x(0) \in C$ satisfying (4) also satisfies (3).

In order to show the converse we start by assuming that $x(t)$ satisfies (3). We consider any interval (t_0, t_1) where $\dot{x}(t)$ is continuous. By definition of the tangent cone, we have $\lim_{\mathrm{d}t \to 0}(x(t+\mathrm{d}t) - x(t))/\mathrm{d}t = \dot{x}(t) \in T_C(x(t))$ and $\lim_{\mathrm{d}t \to 0}(x(t-\mathrm{d}t) - x(t)) = -\dot{x}(t) \in T_C(x(t))$ for all $t \in (t_0, t_1)$. Thus, from $-R(t) \in N_C(x(t))$ it follows that $-R(t)^\mathsf{T}\dot{x}(t) \leq 0$ and $R(t)^\mathsf{T}\dot{x}(t) \leq 0$, which implies that $-R(t)^\mathsf{T}\dot{x}(t) = 0$ for all $t \in (t_0, t_1)$. In addition, by definition of the normal cone, it follows that $-R(t)^\mathsf{T} v \leq 0$ for all $v \in T_C(x(t))$. Combining these two facts results in $-R(t)^\mathsf{T}(v - \dot{x}(t)) \leq 0$ for all $v \in T_C(x(t))$ and all $t \in (t_0, t_1)$. Hence, $-R(t) \in \partial\psi_{T_C(x(t))}(\dot{x}(t))$ for all $t \in (t_0, t_1)$, which implies (4) for any time interval where $\dot{x}(t)$ is continuous. By taking the right-limit $t \downarrow t_0$, we conclude that $-R(t_0)^+ \in \partial\psi_{T_C(x(t_0))}(\dot{x}(t_0)^+)$, $\dot{x}(t_0)^+ = -\nabla f(x(t_0)) + R(t_0)^+$, since $x(t)$ is continuous. Thus, (4) holds for $t = t_0$, and therefore also at any other time instant where $\dot{x}(t)$ is discontinuous. ■

Three important points are worth mentioning:

- The piecewise continuity assumptions on $\dot{x}$ are only used for showing that (3) implies (4); absolute continuity of x is enough for the converse to hold (provided the constraint qualifications are satisfied).
- When the solution $x(t)$ slides along the boundary of the constraint ($\dot{x}(t)$ is continuous), the reaction force is necessarily orthogonal to the velocity. From the point of view of classical mechanics, this means that the constraint reaction forces are passive and do not exert any power (at almost every time instant). This directly implies that the function $f(x(t))$ necessarily decreases along the trajectories of (3) or (4).
- The condition (4) describes the forward evolution of $x(t)$ by prescribing the right-hand derivative of $\dot{x}$ at each point in time. An equivalent formulation for the backwards evolution also exists. We will concentrate on the forward evolution, since we are interested in minimizing f.

The above proposition proves the equivalence between (3) and (4) as stated in Proposition 2. The equivalence between (4) and (5) follows by interpreting (4) as stationarity condition for $\dot{x}(t)^+$. The assumption that the Mangasarian-Fromovitz constraint qualification holds for all $x \in C$ implies that Slater's condition holds for $T_C(x)$ for all $x \in C$. A similar statement applies to the set $V_\alpha(x)$, as shown in the following proposition.

Proposition 6 *Let the Mangasarian-Fromovitz constraint qualification be satisfied for all $x \in \mathbb{R}^n$. This implies that Slater's condition holds for $V_\alpha(x)$; i.e., for any $x \in \mathbb{R}^n$, there exists a $v \in \mathbb{R}^n$ such that $\nabla h(x)^\mathsf{T} v + \alpha h(x) = 0$ and $\nabla g_i(x)^\mathsf{T} v + \alpha g_i(x) > 0$ for all $i \in I_x$.*

Proof We pick a $\bar{v} \in \mathbb{R}^n$ such that $\nabla h(x)^\mathsf{T}\bar{v} = -\alpha h(x)$. Due to the fact that the columns of $\nabla h(x)$ are linearly independent, such a $\bar{v}$ exists. Thus, for a sufficiently large constant $\xi > 0$, we have that

$\nabla h(x)^\mathsf{T}(\bar{v}+\xi w) = -\alpha h(x)$, $\nabla g_i(x)^\mathsf{T}(\bar{v}+\xi w) > -\alpha g_i(x)$ for all $i \in I_x$, where $w \in \mathbb{R}^n$ satisfies $\nabla h(x)^\mathsf{T} w = 0$ and $\nabla g_i(x)^\mathsf{T} w > 0$ for all $i \in I_x$. By assumption such a w exists. Thus, $v = \bar{v} + \xi w$ satisfies the required conditions. ■

Thus, (5) amounts to optimizing a strongly convex objective over a closed non-empty convex set, which implies that there exists a unique solution $\dot{x}(t)^+$ for all $t \in [0, \infty)$. The constraint qualification further implies that strong duality holds for all $t \in [0, \infty)$. The corresponding dual problem is stated in (10) and can be restated as

$$\max_{\lambda \in D_x} -\frac{1}{2}|W(x)\lambda - \nabla f(x)|^2 - \alpha \lambda^\mathsf{T} \bar{g}(x). \tag{15}$$

Solving the dual problem at time t for a given $x(t)$ yields the corresponding constraint force $R(t)$ via $R(t) = W(x(t))\lambda(t)$, where $\lambda(t)$ is a maximizer of (15) (or (10)). The quadratic term in the cost function in (15) leads to the following dual interpretation of (4) and (5): At each point in time, the constraint force (or dual variable) is roughly chosen to minimize the Euclidean norm of the forward velocity $\dot{x}(t)^+$ subject to the constraint $\lambda(t) \in D_{x(t)}$. The forward velocity can be viewed as the residual of the Karush-Kuhn-Tucker conditions evaluated at $x(t)$. The additional term $\lambda^\mathsf{T}\bar{g}(x)$ vanishes whenever $x \in C$. For $x \notin C$, $\lambda^\mathsf{T}\bar{g}(x)$ can be interpreted as a potential function of the constraint $x \in C$ (see, for example, Lanczos, 1952).

For $\alpha > 0$, maximizing (15) is equivalent to maximizing (10). When f is strongly convex with constant μ, $\alpha \leq \mu$, and C is convex, (10) maximizes a lower bound on f^*. More precisely, the strong convexity of f and the convexity of C imply that $l(\cdot, \lambda)$ is strongly convex with constant μ (for a fixed $\lambda \in D_x$), which means that

$$\begin{aligned} f^* \geq f_{I_x}^* \geq \inf_{z \in \mathbb{R}^n} l(z, \lambda) &\geq l(x, \lambda) - \frac{1}{2\mu}|\nabla_x l(x, \lambda)|^2, \quad \forall \lambda \in D_x, \\ &\geq l(x, \lambda) - \frac{1}{2\alpha}|\nabla_x l(x, \lambda)|^2, \quad \forall \lambda \in D_x, \end{aligned} \tag{16}$$

where the last inequality follows from $\alpha \leq \mu$. The lower bound in (16) corresponds to the cost function in (10), which shows that (10) (or (15)) indeed maximizes a lower bound on f^*. The bound (16) will also be useful for deriving convergence rates, as it relates the velocity $\dot{x}(t)^+ = -\nabla_x l(x(t), \lambda(t))$ to the difference of the Lagrangian, $l(x(t), \lambda(t)) - f^*$.

The stationarity condition for (15) (or (10)) is given by

$$W(x(t))^\mathsf{T} W(x(t))\lambda(t) - W(x(t))^\mathsf{T} \nabla f(x(t)) + \alpha \bar{g}(x(t)) \in \partial \psi_{D_{x(t)}}(\lambda(t)). \tag{17}$$

This turns out to be an ideal starting point for computing $\lambda(t)$ via fixed-point iteration, as will be discussed in detail in Section 6.

We close the section by proving the remaining statements of Proposition 2; i.e., showing that the solutions of (3), (4), and (5) converge to stationary points of (1) and deriving convergence rates in case C is convex and f is strongly convex.

We start by proving the following intermediate lemma, which will also be important for the discrete-time analysis.

Lemma 7 *Let the assumptions of Proposition 2 be satisfied. Then, for every $t_0 > 0$, there exists $\delta > 0$ such that $\lambda(t) \in D_{x(t_0)}$ for all $t \in (t_0 - \delta, t_0)$.*

Proof We fix $t_0 > 0$ and consider the set of inequality constraints that are inactive at t_0; that is, $g_i(x(t_0)) > 0$. Due to the continuity of x and g there exists an interval $(t_0 - \delta, t_0)$, where $\delta > 0$ is small enough, such that $g_i(x(t)) > 0$ for all $t \in (t_0 - \delta, t_0)$ and for all $i \notin I_{x(t_0)}$. As a result, $I_{x(t)} \subset I_{x(t_0)}$ for all $t \in (t_0 - \delta, t_0)$ and the result follows. ■

Claim 1 *Let the assumptions of Proposition 2 be satisfied. For any $x(0) \in \mathbb{R}^n$,* (4) *and* (5) *are equivalent and lead to a unique trajectory $x(t)$, which is guaranteed to converge to the set of stationary points of* (1) *(for $\alpha > 0$). Moreover, if the stationary points are isolated, the trajectory $x(t)$ converges to a single stationary point.*

Proof The equivalence between (4) and (5) follows from the fact that (4) corresponds to the stationarity condition of (5), which, by strong convexity and non-emptiness of $V_\alpha(x(t))$, uniquely defines $\dot{x}(t)^+$ for each $t \in (0, \infty)$. This concludes that $x(t)$ is unique.

We argue next that $x(t) \to C$ for $t \to \infty$, and that, as a result, $x(t)$ and $\lambda(t)$ are bounded. According to (7), the constraint violations at time t can be bounded by $g_i(x(t)) \geq g_i(x(0))e^{-\alpha t}$ for all $i \in I_{x(0)}$ and $|h(x(t))| \leq |h(x(0))|e^{-\alpha t}$. We therefore conclude that $x(t) \to C$ for $t \to \infty$. The fact that C is bounded and x is continuous implies that $x(t)$ is bounded for all $t \geq 0$. As a result, there exist bounded dual variables $\lambda(t)$ satisfying (15).

The stationarity condition (17) implies that

$$\lambda(t)^\mathsf{T} W(x(t))^\mathsf{T} \left[W(x(t))\lambda(t) - \nabla f(x(t))\right] + \alpha\lambda(t)^\mathsf{T}\bar{g}(x(t)) = 0,$$

due to complementary slackness. This can be restated as $-R(t)^\mathsf{T}\dot{x}(t)^+ = \alpha\lambda(t)^\mathsf{T}\bar{g}(x(t))$, which, in view of (4), yields

$$\frac{\mathrm{d}}{\mathrm{d}t} f(x(t))^+ = -|\dot{x}(t)^+|^2 - \alpha\lambda(t)^\mathsf{T}\bar{g}(x(t)). \tag{18}$$

This means that f necessarily decreases for large t, since $-\lambda(t)^\mathsf{T}\bar{g}(x(t))$ decreases exponentially. We further note that $f(x(t))$ is bounded below, which, by taking the integral of the right-hand side of (18), implies

$$\int_0^\infty -|\dot{x}(t)^+|^2 - \alpha\lambda(t)^\mathsf{T}\bar{g}(x(t))\mathrm{d}t > -\infty. \tag{19}$$

We note that the integrand is closely related to the objective function in (15), which we denote as $\xi_\mathrm{d}(t)$:

$$\xi_\mathrm{d}(t) := -\frac{1}{2}|\dot{x}(t)^+|^2 - \alpha\lambda(t)^\mathsf{T}\bar{g}(x(t)).$$

From the fact that $\lambda(t)$ is bounded and that $-\lambda(t)^\mathsf{T}\bar{g}(x(t))$ decays exponentially, we conclude that $\limsup_{t\to\infty} \xi_\mathrm{d}(t) \leq 0$. From (19) it also follows that the integral of ξ_d over $\mathbb{R}_{\geq 0}$ is bounded below.

We will now establish that $\lim_{t\to\infty} \xi_\mathrm{d}(t) = 0$ by applying a variant of Barbalat's lemma; see Lemma 11 in Appendix A. We start by observing that λ inherits the continuity properties of $\dot{x}^+$, due to the fact that $W(x(t))\lambda(t) = \nabla f(x(t)) + \dot{x}(t)^+$. This means that λ is piecewise continuous, and for each time $t_0 > 0$, $\lambda(t_0) = \lim_{t\downarrow t_0} \lambda(t)$. The same applies for ξ_d. We now characterize the discontinuities of ξ_d and provide a lower bound on its derivative, whenever it exists. We fix $t_0 > 0$.

By virtue of Lemma 7, we conclude that $\lambda(t)$ is a feasible candidate for (15) at time t_0 as long as $t \in (t_0 - \delta, t_0)$ for a small enough $\delta > 0$. This means

$$\begin{aligned}\xi_{\mathrm{d}}(t_0) &\geq -\frac{1}{2}|W(x(t_0))\lambda(t) - \nabla f(x(t_0))|^2 - \alpha \bar{g}(x(t_0))^{\mathsf{T}}\lambda(t), \\ &\geq \xi_{\mathrm{d}}(t) - r_1(t_0)|x(t_0) - x(t)| - r_2(t_0)|x(t_0) - x(t)|^2, \end{aligned} \tag{20}$$

for all $t \in (t_0 - \delta, t_0)$, where $r_1(t_0) \geq 0$ and $r_2(t_0) \geq 0$ are related to the remainder terms of a first-order Taylor expansion of $\nabla_x l(x, \lambda(t))$ and $\lambda(t)^{\mathsf{T}}\bar{g}(x)$ with respect to x at $(x(t), \lambda(t))$. The fact that $x(t)$ and $\lambda(t)$ are bounded implies that $r_1(t_0)$ and $r_2(t_0)$ are likewise bounded (uniformly) for all $t_0 > 0$. Furthermore, $\dot{x}(t)^+$ is bounded, which implies the existence of a constant $\bar{r}_1 > 0$ (independent of t_0) such that

$$\xi_{\mathrm{d}}(t_0) \geq \xi_{\mathrm{d}}(t) - \bar{r}_1 \delta,$$

for a small enough δ and all $t \in (t_0 - \delta, t_0)$. We can now distinguish two cases, depending on whether ξ_{d} is continuous at t_0 or not. If ξ_{d} is discontinuous at t_0, we obtain $\xi_{\mathrm{d}}(t_0) \geq \xi_{\mathrm{d}}(t_0)^-$. The other case yields $\xi_{\mathrm{d}}(t_2) \geq \xi_{\mathrm{d}}(t_1) - \bar{r}_1(t_2 - t_1)$, for all $t_2 \geq t_1$, as long as ξ_{d} is continuous on (t_1, t_2).

We are now ready to apply Lemma 11 (see Appendix A), which implies that $\lim_{t\to\infty} \xi_{\mathrm{d}}(t) = 0$. As a result of the exponential convergence of $\lambda(t)^{\mathsf{T}}\bar{g}(x(t))$, we obtain $\lim_{t\to\infty} |\dot{x}(t)^+| = 0$, and conclude that $x(t)$ necessarily converges to the union of all stationary points.

It remains to show that $x(t)$ converges to a single stationary point in case that the stationary points are isolated. To that extent, we consider the sequence $x(k)$, $k > 0$. Due to the fact that $\dot{x}(t)^+$ converges, we can find, for every $\epsilon > 0$, an integer $N > 0$ such that $|x(k+1) - x(k)| < \epsilon$ for all $k > N$. Choosing ϵ small enough implies that $x(k)$ necessarily converges to a single stationary point, which we denote by x_{s} (this would otherwise contradict the fact that the stationary points are isolated). Moreover, $|x(t) - x_{\mathrm{s}}| \leq |x(t) - x(k_t)| + |x(k_t) - x_{\mathrm{s}}|$, where k_t is the largest integer such that $k_t < t$. We conclude $\lim_{t\to\infty} x(t) = x_{\mathrm{s}}$ by observing that $|x(t) - x(k_t)|$ is bounded by the supremum of $|\dot{x}(\tau)^+|$ over $\tau \in (k_t, t)$, which becomes arbitrarily small for large t. ■

Claim 2 *Let the assumptions of Proposition 2 be satisfied and let C be convex and f strongly convex with strong convexity constant μ and $\alpha \leq 2\mu$. Then the following holds:*

$$(h(x(0)), \min\{0, g(x(0))\})^{\mathsf{T}}\lambda^* e^{-\alpha t} \leq f(x(t)) - f^* \leq (f(x(0)) - f^*)e^{-2\mu t},$$

for all $x(0) \in \mathbb{R}^n$, where $x(t)$ satisfies (4) *and* (5)*, f^* is the optimal cost in* (1) *and λ^* is a corresponding multiplier that satisfies the Karush-Kuhn-Tucker conditions.*

Proof We will use (18) as a starting point for deriving the upper bound. From (16) we conclude that

$$-|\dot{x}(t)^+|^2 \leq -2\mu(f(x(t)) - f^*) + 2\mu\lambda(t)^{\mathsf{T}}\bar{g}(t).$$

Thus, inserting the upper bound on $-|\dot{x}(t)^+|^2$ in (18), we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t} f(x(t))^+ \leq -2\mu(f(x(t)) - f(x^*)) + (2\mu - \alpha)\lambda(t)^{\mathsf{T}}\bar{g}(x(t)).$$

For $\alpha \leq 2\mu$, the term $(2\mu - \alpha)\lambda(t)^{\mathsf{T}}\bar{g}(x(t))$ is certainly negative (or vanishes completely if $x(0) \in C$), which readily proves the upper bound.

The lower bound follows from a perturbation analysis. For a given $x(0) \in C$, we define

$$f^*(t) := \min_{z \in \mathbb{R}^n} f(z), \quad \text{s.t.} \quad h(z) = h(x(0))e^{-\alpha t}, \quad g(z) \geq \min\{0, g(x(0))\}e^{-\alpha t},$$

which is of the form (1), with the sole difference that the right-hand side of the constraints has been replaced with the vector $(h(x(0)), \min\{0, g(x(0))\}) \exp(-\alpha t)$. The trajectory $x(t)$ is guaranteed to be feasible with respect to these modified constraints, which implies that $f^*(t) \leq f(x(t))$. The above minimum is attained for all $t \in [0, \infty)$, due to the fact that f is bounded below and the modified set of feasible points is closed. A multiplier λ^* satisfying the Karush-Kuhn-Tucker conditions of (1) captures the sensitivity of the cost function with respect to perturbations of the right-hand side of the constraints. More precisely, $-\lambda^*$ is guaranteed to satisfy the following inequality (see, e.g., Rockafellar, 1970, p. 277):

$$f^*(t) - f^* \geq (h(x(0)), \min\{0, g(x(0))\})^\mathsf{T} \lambda^* \exp(-\alpha t).$$

The lower bound of (6) in Proposition 2 then follows from the fact that $f(x(t)) \geq f^*(t)$ for all $t \in [0, \infty)$. ■

4. A First Example

We start by discussing an example that illustrates the behavior of (4) and (8). We consider the following problem:

$$\min_{x \in \mathbb{R}} \frac{1}{10}(x+1)^2, \quad \text{s.t.} \quad x \in [0, 2], \tag{21}$$

which has the unique minimum $x^* = 0$. The function f is therefore given by $(x+1)^2/10$, whereas $g_1(x) = x$ and $g_2(x) = 2 - x$. It will be instructive to plot the function $\nabla_x l(x, \lambda(x)) = \nabla f(x) - R(x)$, where the multiplier $\lambda(x)$ is obtained from (15). This yields a continuous-time gradient flow that is given by $\dot{x}(t)^+ = -\nabla_x l(x(t), \lambda(t))$, whereas the discrete-time version is given by $x_{k+1} - x_k = -T\nabla_x l(x_k, \lambda_k)$, where $\lambda(t)$ and λ_k are implicitly dependent on $x(t)$ and x_k, respectively. Furthermore, we can interpret $\nabla_x l(x, \lambda(x))$ as the gradient of a continuous function $F_\alpha : \mathbb{R} \to \mathbb{R}$, with $F_\alpha(0) = f^*$. We also plot the function $d(x)$ as defined in (10).

The plots are shown in Figure 4 for two different α. The left column is prototypical for $\alpha \leq 1/5$, the right column for $\alpha > 1/5$, where $1/5$ amounts to the Hessian of f. It is important to note that $\nabla_x l$ is discontinuous at the origin, but nonetheless unique. In the continuous-time case, the discontinuity at the origin is less of an issue, since the solutions to $\dot{x}(t)^+ = -\nabla_x l(x(t), \lambda(t))$ approach the origin either from $x(t) > 0$ or from $x(t) < 0$ and never cross the origin. When the solution approaches the origin from negative values, $x(t) < 0$, the velocity $\dot{x}(t)$ continuously reduces to zero for $t \to \infty$. If the solutions approach the origin from positive values, $x(t) > 0$, the velocity continuously reduces to $\dot{x}(t)^- = -0.2$ at which point it instantly drops to zero. Hence, if $x(t)$ approaches the origin from positive values, the convergence is in finite time. The origin is therefore a stable and attractive equilibrium in the sense of Lyapunov.

In discrete time, the situation changes drastically. Starting from a generic initial condition, $x_0 > 0$, the solution to $x_{k+1} = x_k - T\nabla_x l(x_k, \lambda_k)$ crosses the origin and eventually always approaches the origin from $x_k < 0$ (provided that α and T are small enough). For small α and T, the origin

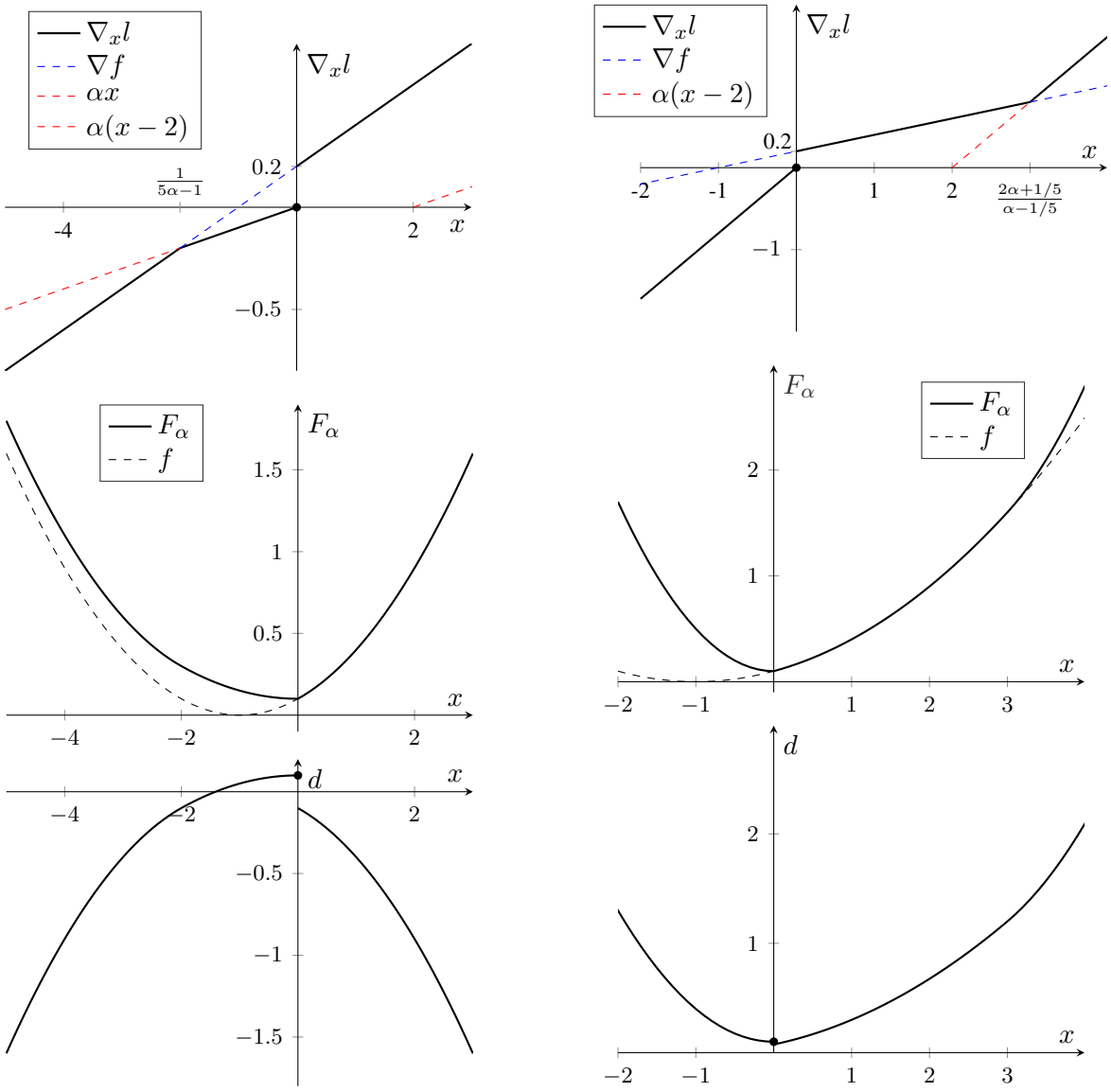

Figure 4: This figure shows the values of $\nabla_x l$, F_α, and d for $\alpha = 1/10$ (left column) and $\alpha = 4/5$ (right column). Top row: The solid thick black line represents $\nabla_x l$, which is discontinuous at the origin, where it takes the value zero (the origin is the minimizer of (21)). For values $x \leq 0$, $\nabla_x l$ is given by $\min\{\nabla f(x), \alpha x\}$ and for values $x \geq 2$, $\nabla_x l$ is given by $\max\{\nabla f(x), \alpha(x-2)\}$, which is represented by the dashed lines in blue and in red. Middle row: The solid thick black line represents F_α, which is continuous and has its minimum at the origin (the origin is the minimizer of (21)). The objective function f is indicated with dashed lines. Last row: The function d is discontinuous at the origin for $\alpha \neq 1/5$, unbounded below for $\alpha < 1/5$, and unbounded above for $\alpha > 1/5$. For $\alpha < 1/5$, $d(x)$ is upper bounded by $f^*_{I_x}$, that is, $d(x) \leq f^* = f^*_{\{1\}} = 0.1$ for $x \leq 0$ and $d(x) \leq f^*_{\{\}} = f^*_{\{2\}} = 0$ for $x > 0$, where $g_1(x) = x$ and $g_2(x) = 2 - x$. As we will show in Section 5, this holds more generally provided that f and C are convex.

can therefore be viewed as a semi-permeable membrane; solutions cross from $x_k > 0$ to $x_{k+1} < 0$, but not vice versa. The origin is *not* a stable equilibrium, since trajectories starting arbitrarily close to the origin will jump to a negative x_1, such that $|x_1| \geq |0.2T - (1 - 0.2T)x_0| \approx 0.2T$. (Hence, no matter how small we choose $\delta > 0$, there exists an initial condition x_0 with $|x_0| < \delta$ such that $|x_k| \geq 0.1T$ for some $k \geq 0$.) We therefore conclude that any attempt to find a continuous Lyapunov function for proving convergence in discrete time is doomed to fail. Indeed, as we will show in the following, proving convergence of (8) hinges on the analysis of the *discontinuous* function $d(x)$, which can be shown to be monotonically increasing along trajectories x_k for small enough α and T. The analysis can also be interpreted as choosing an appropriate sequence of nested invariant sets, which generalizes the above discussion of the origin acting as a semi-permeable membrane. Each of these invariant sets can then be shown to be attractive, whereby trajectories converge at a linear rate.

We would like to emphasize that even though the origin is *not* stable in the sense of Lyapunov (in discrete time), it is still *attractive*; that is, x_k converges to origin for small enough α and T. From Figure 4, it follows that $\alpha T \leq 1$ is necessary for ensuring that trajectories approach the origin from $x_k < 0$ for large k. If $\alpha T > 1$, we observe oscillations about the origin. We further note that already the analysis of a two-dimensional problem with multiple linear constraints appears to be very challenging due to the discontinuity of $\nabla_x l$ and the discrete nature of (8), which results in a multitude of different constraints that may or may not become active over the course of the optimization.

5. The Discrete-Time Case

This section analyzes the convergence of algorithm (8) to stationary points of (1). In contrast to the continuous-time setting, where a trajectory starting from $x(0) \in C$ is guaranteed to remain feasible, a discrete trajectory x_k may become infeasible in the course of the optimization, even if $x_0 \in C$. This is due to the finite length of each step of the discrete algorithm and the fact that only the active constraints I_{x_k} are taken into account. While this potentially saves computation and distinguishes our algorithm from other methods, it also complicates the analysis. As we discussed in the previous section, while trajectories still converge to the minimizer of (1) (assuming convexity and appropriately chosen parameters T and α), the minimizer may not correspond to a stable equilibrium in the sense of Lyapunov.

In Section 4, we saw that for $\alpha T \leq 1$, the solutions x_k of algorithm (8) cross the origin from $x_k > 0$ to $x_{k+1} < 0$, but not vice versa. The property is crucial for guaranteeing convergence, as it excludes oscillations about the origin. We can therefore visualize the boundary of the feasible set as a semi-permeable membrane; trajectories can pass from the feasible to the infeasible region, but not the other way. The following lemma will be the first step in making this observation precise.

Lemma 8 *Let C be convex. Provided that $\alpha T \leq 1$, the inequality constraints at time k for which the corresponding λ_{ki} is nonzero, will remain active at time $k+1$. In other words, $\lambda_{ki} > 0$ implies $g_i(x_{k+1}) \leq 0$.*

Proof The stationarity condition (17), which applies in the same way to the discrete algorithm (8) (it suffices to replace $x(t)$ with x_k, $\lambda(t)$ by λ_k, and $\dot{x}(t)^+ = (x_{k+1} - x_k)/T$), implies that $\lambda_{ki} W_i(x_k)^\mathsf{T}(x_{k+1} - x_k) = -\alpha T \bar{g}_i(x_k)\lambda_{ki}$ for all $i \in \{1, 2, \ldots, n_\mathrm{h} + n_\mathrm{g}\}$ (complementary slackness). Due to the fact that C is convex, which means that h is linear and g is concave, it follows

that

$$h(x_{k+1}) = h(x_k) + T\nabla h(x_k)^\mathsf{T}(x_{k+1} - x_k),$$
$$g(x_{k+1}) \leq g(x_k) + T\nabla g(x_k)^\mathsf{T}(x_{k+1} - x_k).$$

Combined with the fact that $\lambda_k \in \mathbb{R}^{n_\mathrm{h}} \times \mathbb{R}^{n_\mathrm{g}}_{\geq 0}$, this implies

$$\lambda_{ki}\bar{g}_i(x_{k+1}) \leq (1 - \alpha T)\lambda_{ki}\bar{g}_i(x_k),$$

for any $i \in \{1, 2, \ldots, n_\mathrm{h} + n_\mathrm{g}\}$. The result follows by noting that $\lambda_{ki}\bar{g}_i(x_k) \leq 0$ and $1 - \alpha T \geq 0$. ■

Lemma 8 implies that $\lambda_k \in D_{x_{k+1}}$, ensuring that λ_k is a feasible candidate for (10), or (15), at time $k+1$. Lemma 8 also concludes that $D_{x_x} \subseteq D_{x_{k+1}}$ and can therefore be viewed as the discrete-time version of Lemma 7. As in the continuous-time case, Lemma 8 will be of paramount importance for proving convergence.

The convergence proof will also rely on the following bounds for $d(x)$.

Lemma 9 *Let C be convex. For $0 \leq \alpha \leq \mu$ and any $x \in \mathbb{R}^n$ the following upper and lower bounds on $d(x)$ hold*

$$f^*_{I_x} - \frac{1}{2\alpha}\frac{L_l}{\alpha}\left(1 - \frac{\alpha}{L_l}\right)|v(x)|^2 \quad \leq \quad d(x) \quad \leq \quad f^*_{I_x} - \frac{1}{2\alpha}\left(1 - \frac{\alpha}{\mu}\right)|v(x)|^2 \quad \leq \quad f^*,$$

where $L_l \geq \bar{L}_l(\lambda^*_{I_x})$.

Proof The upper bound follows directly from (16). In order to obtain the lower bound, we first note that the smoothness of $l(\cdot, \lambda^*_{I_x})$ (where $\lambda^*_{I_x}$ is fixed) implies

$$f^*_{I_x} = \inf_{z \in \mathbb{R}^n} l(z, \lambda^*_{I_x}) \leq l(x, \lambda^*_{I_x}) - \frac{1}{2L_l}|\nabla_x l(x, \lambda^*_{I_x})|^2. \tag{22}$$

We further consider the modified primal and dual problems (where α is replaced by L_l):

$$v_\mathrm{m}(x) = \operatorname*{argmin}_{v \in V_{L_l}(x)} \frac{1}{2}|v + \nabla f(x)|^2, \quad \lambda_\mathrm{m}(x) \in \operatorname*{argmax}_{\lambda \in D_x} l(x, \lambda) - \frac{1}{2L_l}|\nabla_x l(x, \lambda)|^2,$$

and note that $v(x)L_l/\alpha \in V_{L_l}(x)$, hence $v(x)L_l/\alpha$ is a feasible candidate for minimization over $V_{L_l}(x)$. This means that

$$\frac{1}{2}|v_m(x)|^2 + v_m(x)^\mathsf{T}\nabla f(x) \leq \frac{1}{2}\frac{L_l^2}{\alpha^2}|v(x)|^2 + \frac{L_l}{\alpha}v(x)^\mathsf{T}\nabla f(x).$$

Complementary slackness implies that $v(x)^\mathsf{T}\nabla f(x) = -|v(x)|^2 - \alpha\bar{g}(x)^\mathsf{T}\lambda(x)$ and similarly $v_\mathrm{m}(x)^\mathsf{T}\nabla f(x) = -|v_\mathrm{m}(x)|^2 - L_l\bar{g}(x)^\mathsf{T}\lambda_\mathrm{m}(x)$, and yields therefore

$$-\frac{1}{2}|v_\mathrm{m}(x)|^2 - L_l\bar{g}(x)^\mathsf{T}\lambda_\mathrm{m}(x) \leq \frac{1}{2}\frac{L_l}{\alpha}\left(\frac{L_l}{\alpha} - 2\right)|v(x)|^2 - L_l\bar{g}(x)^\mathsf{T}\lambda(x).$$

Dividing by L_l and adding $f(x)$ on both sides implies that

$$\max_{\lambda \in D_x} l(x,\lambda) - \frac{1}{2L_l}|\nabla_x l(x,\lambda)|^2 \leq d(x) + \frac{1}{2\alpha}\left(\frac{L_l}{\alpha} - 1\right)|v(x)|^2.$$

The left-hand side includes a maximum over λ, which means that for $\lambda^*_{I_x} \in D_x$, we have:

$$l(x,\lambda^*_{I_x}) - \frac{1}{2L_l}|\nabla_x l(x,\lambda^*_{I_x})|^2 \leq d(x) + \frac{1}{2\alpha}\left(\frac{L_l}{\alpha} - 1\right)|v(x)|^2.$$

Combining the previous inequality with (22) yields the desired lower bound. ■

We are now ready to prove Proposition 3. We will divide the proof into several smaller claims:

Claim 3 *Let the assumption of Proposition 3 be satisfied. Then, the sequence $d(x_k)$ is monotonically increasing and bounded above by f^*.*

Proof The fact that $d(x_k)$ is bounded above by f^* follows from Lemma 9. We note that due to Lemma 8, the multiplier λ_k is a feasible candidate for the dual (15) (or (10)) at time $k+1$; that is, $\lambda_k \in D_{x_{k+1}}$. This means that

$$d(x_{k+1}) \geq l(x_{k+1},\lambda_k) - \frac{1}{2\alpha}|\nabla_x l(x_{k+1},\lambda_k)|^2.$$

Due to the strong convexity of $l(\cdot,\lambda_k)$ (for a fixed λ_k) it follows that

$$l(x_{k+1},\lambda_k) \geq l(x_k,\lambda_k) + T\nabla_x l(x_k,\lambda_k)^\mathsf{T} v_k + \frac{\mu}{2}T^2|v_k|^2 = l(x_k,\lambda_k) - T|v_k|^2 + \frac{\mu}{2}T^2|v_k|^2.$$

Moreover, by using Taylor's theorem, we can relate the gradient $\nabla_x l(x_{k+1},\lambda_k)$ to the gradient $\nabla_x l(x_k,\lambda_k)$ in the following way:

$$\nabla_x l(x_{k+1},\lambda_k) = \nabla_x l(x_k,\lambda_k) + T\Delta_x l(\xi_k,\lambda_k)\ \ v_k,$$

where $\Delta_x l$ denotes the second derivative of l with respect to x, and ξ_k lies between x_k and x_{k+1}. Hence, we obtain the following lower bound for $d(x_{k+1})$:

$$d(x_{k+1}) \geq d(x_k) + \frac{T}{\alpha}v_k^\mathsf{T}\Delta_x l\ v_k - \frac{T^2}{2\alpha}v_k^\mathsf{T}(\Delta_x l)^2\ \ v_k - T|v_k|^2 + \frac{\mu}{2}T^2|v_k|^2,$$

where the arguments of the Hessian $\Delta_x l(\xi_k, x_k)$ have been omitted to simplify notation. We note that the Hessian $\Delta_x l$ is positive definite due to the convexity of $l(\cdot,\lambda_k)$ and has eigenvalues that are lower bounded by μ and upper bounded by L_l. Moreover, the matrix $(\Delta_x l)^2$ has the same eigenvectors as $\Delta_x l$, which means that

$$v_k^\mathsf{T}\left(\Delta_x l T - \frac{1}{2}\Delta_x l^2 T^2\right)v_k \geq |v_k|^2 \min_{s\in[\mu T, L_l T]} s - s^2/2.$$

It can be shown that this minimum is lower bounded by $\mu T(1-\mu T/2)$ as long as $T \leq 2/(L_l+\mu)$.[1] This yields

$$d(x_{k+1}) \geq d(x_k) + \underbrace{T\left(1-\frac{\mu T}{2}\right)\left(\frac{\mu}{\alpha}-1\right)}_{=c_1}|v_k|^2. \tag{23}$$

1. The choice $T = 2/(L_l+\mu)$ corresponds to the maximizer of $\max_T \min_{s\in[\mu T, L_l T]} s - s^2/2$.

From $T \leq 2/(L_l + \mu)$ and $\alpha < \mu$ we conclude that $c_1 > 0$, which proves the claim. ■

Claim 4 *Let the assumptions of Proposition 3 be satisfied. The velocity $(x_{k+1} - x_k)/T$ is guaranteed to converge with*

$$\min_{j \in \{0,1,\dots,k\}} | - \nabla f(x_j) + R_j|^2 \leq \frac{f^* - d(x_0)}{c_1(k+1)}, \quad \forall k \geq 0, \quad \forall x_0 \in \mathbb{R}^n,$$

where $c_1 = T(\mu/\alpha - 1)(1 - \mu T/2) > 0$ is constant.

Proof The result follows from Claim 3 by expanding $d(x_{k+1})$ as a telescoping sum

$$\begin{aligned} f^* \geq d(x_{k+1}) &= d(x_0) + \sum_{j=0}^{k} d(x_{j+1}) - d(x_j) \\ &\geq d(x_0) + c_1(k+1) \min_{j \in \{0,1,\dots,k\}} | - \nabla f(x_j) + R_j|^2, \end{aligned}$$

where (23) has been used for the last step. ■

Claim 5 *Let the assumptions of Proposition 3 be satisfied. Each level set $\{x \in \mathbb{R}^n \mid d(x) \geq f_I^*\}$, where I is any subset of $\{1, 2, \dots, n_g\}$, is closed, invariant and attractive. Provided that $L_l \geq \bar{L}_l(\lambda^*_{I_{x_k}})$, trajectories converge at a linear rate, that is,*

$$d(x_{k+1}) - f^*_{I_{x_k}} \geq (1 - c_2 T)(d(x_k) - f^*_{I_{x_k}}),$$

where $c_2 = 2\alpha(1 - \mu T/2)(\mu - \alpha)/(L_l - \alpha) > 0$ is constant.

Proof We conclude from Rockafellar and Wets (1997, Theorem 1.17, p. 16) that d is upper semi-continuous, which means that the level sets $\{x \in \mathbb{R}^n \mid d(x) \geq f_I^*\}$ are closed. The fact that these are attractive and invariant follows directly from Claim 3. For obtaining the linear rate, we start from (23) and apply the lower bound on $d(x_k)$ provided by Lemma 9. This yields

$$d(x_{k+1}) \geq d(x_k) + \frac{c_1}{L_l/(2\alpha^2)(1 - \alpha/L_l)}(f^*_{I_{x_k}} - d(x_k)).$$

Substracting $f^*_{I_{x_k}}$ on both sides yields the desired result (we note that $d(x_k) \leq f^*_{I_{x_k}}$). ■

The last claim provides a geometrical picture of the convergence of (8). At any iteration j, the algorithm converges to the level set $\{x \in \mathbb{R}^n \mid d(x) \geq f^*_{I_1}\}$, where $I_1 = I_{x_j}$. At each iteration, $d(x_k) - f^*_{I_1}$ decreases at least by a constant factor. Once this set is reached, the trajectory is guaranteed to remain inside, and will converge to the next smaller level set (where I_1 is replaced with I_{x_k}). The process continues until finally $d(x_k)$ approaches f^* from below.

With this geometrical picture in mind, we will discuss two extensions of (8). The resulting trajectories can be shown to converge to the minimizer of (1) with the same arguments as used for Claim 3 - Claim 5.

5.1 Extensions

The convergence proof hinges on the following two properties of (8): (i) the multiplier λ_k is feasible for the dual (15) at time $k+1$, and (ii) the function $l(x_k, \lambda_k) - |\nabla_x l(x_k, \lambda_k)|^2/2\alpha$ increases sufficiently from x_k to x_{k+1} (for a fixed λ_k). We can therefore extend (8) by including the following line-search mechanism:

$$T := \operatorname*{argmax}_{\tau>0, \alpha\tau\leq 1} l(x_k + \tau v_k, \lambda_k) - \frac{1}{2\alpha}|\nabla_x l(x_k + \tau v_k, \lambda_k)|^2, \qquad x_{k+1} = x_k + T v_k,$$

where the velocity v_k is determined by solving (9), as before. As an alternative, we can alternate between updating λ_k via (15) and applying gradient steps (with λ_k fixed):

$$x_{j+1} = x_j - T\nabla_x l(x_j, \lambda_k), \quad j = k, k+1, \ldots,$$

as long as $g_i(x_{j+1}) \leq 0$ for all $i \in I_{x_k}$ with corresponding multipliers $\lambda_{ik} > 0$ (constraints that were active and had a nonzero multiplier λ_k at time k are not allowed to open up). As is immediate from the arguments of Claim 3, each of these gradient steps increases $l(x, \lambda_k) - |\nabla_x l(x, \lambda_k)|^2/(2\alpha)$ by $c_1|\nabla l_x(x_j, \lambda_k)|^2$. Evaluating $\nabla_x l$ for a fixed λ_k is computationally cheap and requires only the evaluation of ∇f and $W(x)$.

6. Computational Aspects

This section highlights two important aspects of the implementation of the discrete-time algorithm (8): (i) the computation of the constraint force $R_k = W(x_k)\lambda_k$, and (ii) how to deal with round-off errors and inaccuracies.

6.1 Computing the constraint force R_k

The constraint forces are determined by the dual problem (15), which can be solved with various algorithms. The simple nature of the set D_{x_k} makes (accelerated) projected gradient descent schemes appealing. In the following, we present a procedure that is inspired by the method of successive over-relaxation, and solves (15) very efficiently. The procedure is useful for solving large linear complementary problems and is commonly used in the non-smooth mechanics community (see, e.g., Studer, 2009). For completeness, we give a rough overview of the main points and refer the reader to the work of Cottle et al. (2009) for further details. The stationarity conditions of (15) are given by

$$W_k^\mathsf{T} W_k \lambda_k - W_k^\mathsf{T} \nabla f(x_k) + \alpha \bar{g}(x_k) + \partial\psi_{D_{x_k}}(\lambda_k) \ni 0, \tag{24}$$

where we used the notation introduced in Algorithm 1.[1] The underlying idea relies on a suitable splitting of the matrix $W_k^\mathsf{T} W_k$ that enables fixed-point iteration. We introduce λ^j as the approximation of λ_k at iteration j, $j = 0, 1, \ldots$ and further suppress the subscript k for ease of notation. We denote the strictly upper triangular part of $W^\mathsf{T} W$ by U and the diagonal by D. The matrix $W^\mathsf{T} W$ is therefore given by $U^\mathsf{T} + D + U$, where the diagonal elements are guaranteed to be strictly positive. We can split the matrix $W^\mathsf{T} W$ into $U^\mathsf{T} + \omega^{-1} D$ and $U + (1 - \omega^{-1})D$, where $\omega \in (0, 2)$ is fixed, leading to

$$(U^\mathsf{T} + \omega^{-1} D)\lambda^{j+1} + \partial\psi_{D_x}(\lambda^{j+1}) + (U + (1 - \omega^{-1})D)\lambda^j - W^\mathsf{T}\nabla f(x) + \alpha\bar{g}(x) \ni 0, \tag{25}$$

1. Compared to the notation in (15), for example, we exclude all multipliers λ_i that correspond to inactive inequality constraints; that is, $i \notin I_{x_k}$.

where we have omitted the subscript k; hence, $x = x_k$, $W = W_k = W(x_k)$, etc. The role of the variable ω as a tuning parameter will become apparent below. We note that (25) reduces to (24) for $\lambda^{j+1} = \lambda^j$. As a result of the fact that U^T is strictly lower triangular, (25) reduces to the following inclusion for a single component: λ_i^{j+1}

$$\omega^{-1} D_{ii} \lambda_i^{j+1} + \partial \psi_{\mathbb{R}}(\lambda_i^{j+1}) + *_i \ni 0, \quad \text{or} \quad \omega^{-1} D_{ii} \lambda_i^{j+1} + \partial \psi_{\mathbb{R}_{\geq 0}}(\lambda_i^{j+1}) + *_i \ni 0, \tag{26}$$

depending whether $i \leq n_\mathrm{h}$ or $i > n_\mathrm{h}$, where $*_i$ is a placeholder for all remaining terms that are constant or only depend on λ^j and $\lambda_1^{j+1}, \dots \lambda_{i-1}^{j+1}$. The inclusion in (26) can be seen as a stationarity conditions for λ_i^{j+1}, which uniquely determines λ_i^{j+1} from λ^j and $\lambda_1^{j+1}, \dots, \lambda_{i-1}^{j+1}$. We can therefore express (25) as

$$\lambda^{j+1} = \mathrm{prox}_{D_x}\left(\lambda^j - \omega D^{-1}(U^\mathsf{T}\lambda^{j+1} + (D+U)\lambda^j - W^\mathsf{T}\nabla f(x) + \alpha \bar{g}(x))\right), \tag{27}$$

where $\mathrm{prox}_{D_x} : \mathbb{R}^{n_\mathrm{h}} \times \mathbb{R}^{|I_x|} \to \mathbb{R}^{n_\mathrm{h}} \times \mathbb{R}^{|I_x|}_{\geq 0}$ is defined as

$$\begin{aligned} (\mathrm{prox}_{D_x}(\xi))_i &= \xi_i, & i &= 1, \dots, n_\mathrm{h}, \\ (\mathrm{prox}_{D_x}(\xi))_i &= \max\{\xi_i, 0\}, & i &= n_\mathrm{h}+1, \dots n_\mathrm{h} + |I_x|, \end{aligned}$$

and where we have used the fact that $\omega^{-1} D_{ii} > 0$. It is important to note that (27) provides an explicit rule for computing λ^{j+1} from λ^j, since U^T is strictly lower triangular. In particular, by substituting the newly computed elements λ^{j+1} directly in the right-hand side of (27), i.e., overwriting λ_i^j with λ_i^{j+1} as soon as it becomes available, the expression on the right-hand side of (27) reduces to

$$\mathrm{prox}_{D_x}\left(\lambda^j - \omega D^{-1}(W^\mathsf{T} W \lambda^j - W^\mathsf{T}\nabla f(x) + \alpha \bar{g}(x))\right),$$

which becomes very convenient for a computer implementation. The expression (27) can therefore be viewed as an extension of the method of successive over-relaxation that accounts for the complementary slackness induced by the inequality constraints. The method reduces to a variant of the Gauss-Seidel method for $\omega = 1$. The following proposition due to Cottle et al. (2009, p. 400) ensures convergence of the $\lambda^j \to \lambda_k$ as long as $\omega \in (0, 2)$. The proof follows Cottle et al. (2009, p. 400) and is included in Appendix C for completeness.

Proposition 10 *Cottle et al. (2009, p. 400) The sequence λ^j, defined according to* (27)*, converges for $\omega \in (0, 2)$. The resulting multiplier $\lim_{j\to\infty} \lambda^j = \lambda_k$ satisfies* (24) *and therefore maximizes* (15)*.*

In our numerical experiments, the choice $\omega = 1$ (Gauss-Seidel variant) yielded good results.

6.2 Dealing with round-off errors and inaccuracies in the computation of R_k

In Section 4 and Section 5 we noted that the minimizer of (1) is typically not a stable equilibrium in the sense of Lyapunov for (8). If we revisit the example of Section 4 we realize that a trajectory initialized at $x_0 = \epsilon > 0$, where $\epsilon > 0$ is arbitrarily small, will make a relatively large step to $x_1 < 0$ before approaching the origin from $x_k < 0$. Thus, if we set the constraint force R_k to be slightly too large by mistake, when approaching the origin from $x_k < 0$, this might push x_k again to positive values ($x_k > 0$), at which point the cycle would start again. For a practical implementation of (8), it is therefore important to address and discuss the effect of round-off errors and inexact computations of R_k.

We can address the problem with a combination of the following two strategies:

(i) Slightly extending the infeasible set: We extend the set I_x to $\{i \in \mathbb{Z} \mid g_i(x) \leq \epsilon_\mathrm{g}\}$, where $\epsilon_\mathrm{g} > 0$ is a user-specified tolerance for constraint satisfaction. Provided that x^*, the minimizer of (1), lies on the boundary of the feasible set, this has the effect that in a neighborhood about x^* inequality constraints are treated as equality constraints, which prevents x_k from cycling even in the presence of round-off errors and inexact computations of R_k. We illustrate the situation with the example of Section 4, where Figure 5 shows the gradient $\nabla_x l$. The introduction of the parameter ϵ_g slightly extends the infeasible region, and moves the discontinuity of $\nabla_x l$ from x^* to $x^* + \epsilon_\mathrm{g}$. This renders the origin stable in the sense of Lyapunov and therefore mitigates the effect of small round-off errors and slight inaccuracies in the computation of R_k.

(ii) Adapting the stopping criteria of (27)**:** In continuous time, the complementary slackness states that $\lambda_i > 0$ implies $\mathrm{d}g_i(x(t))/\mathrm{d}t + \alpha g_i(x(t)) = 0$ (constraint i remains active), whereas $\mathrm{d}g_i(x(t))/\mathrm{d}t + \alpha g_i(x(t)) \geq 0$ for $\lambda_i = 0$ (constraint i might open up). Since we are solving the complementary slackness conditions only approximately, it might happen that even for $\lambda_i > 0$, $\mathrm{d}g(x(t))_i/\mathrm{d}t$ becomes too large such that the constraint incorrectly opens up in the next iteration of our discrete approximation. This can be avoided by stopping the iteration (27) only if for each inequality constraint i with $\lambda_i > 0$, we have

$$\underbrace{(W_k^\mathsf{T} W_k \lambda - W_k^\mathsf{T} \nabla f(x_k))_i}_{\approx \mathrm{d}g_i(x(t))/\mathrm{d}t} + \alpha g_i(x_k) \leq \epsilon_\mathrm{g} \alpha / 2. \tag{28}$$

For convex constraints (g is concave) this inequality ensures that

$$g_i(x_{k+1}) \leq (1 - \alpha T) g_i(x_k) + \epsilon_\mathrm{g} \alpha T / 2,$$

for all constraints where the corresponding multiplier λ_i is strictly positive. The fact that $g_i(x_k) \leq \epsilon_\mathrm{g}$ (see point (i) above) and $0 < \alpha T \leq 1$ guarantees that $g_i(x_{k+1}) < \epsilon_\mathrm{g}$, which means that the constraint remains active.

Algorithm 2 summarizes the discussions of the two previous sections. The next section will be concerned with the empirical evaluation of Algorithm 2 on various examples. The exact implementation in Python and C++ will be made available as supplementary material.

7. Numerical Examples

The following section illustrates the application of Algorithm 2 to the following problems: (i) Randomly generated quadratic programs, (ii) trust region optimization, (iii) ν-support vector machines, and (iv) computing a catenary subject to nonlinear constraints. The examples (i)-(iii) lead to convex quadratic programs or convex second-order cone programs, whereas the last example is a nonconvex problem. Algorithm 2 is implemented in C++ and we use pybind11 (Jakob et al., 2017), as a Python interface. The experiments are conducted on a Dell Precision Tower 3620 that runs Ubuntu 20.04LTS and is equipped with an Intel Core i7-6700 processor (8x3.4GHz) and 64GB of random access memory. All matrices are stored in compressed row storage for exploiting sparsity. The parameters of Algorithm 2, which are used for the experiments are summarized in Table 1.

The section will also highlight that Algorithm 2 is competitive with the state-of-the-art interior point solver CVXOPT (Andersen et al., 2011) for larger problem instances.

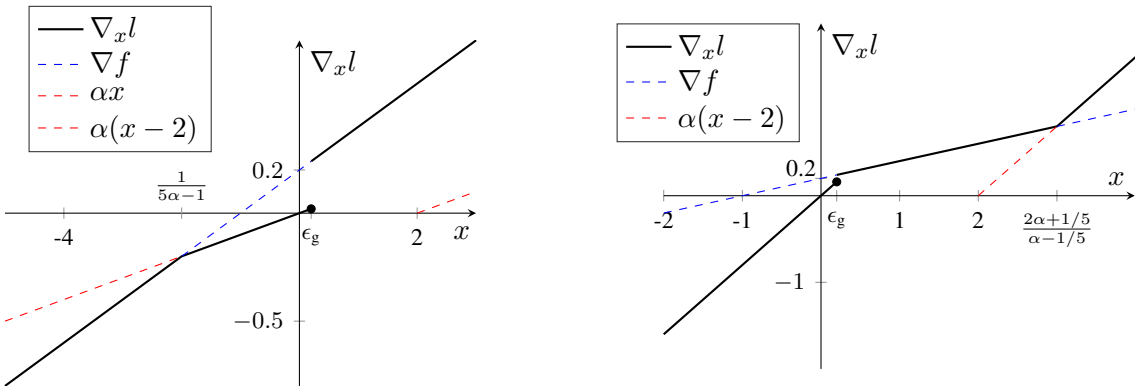

Figure 5: This figure shows the values of $\nabla_x l$ (solid thick line) for $\alpha = 1/10$ (left) and $\alpha = 4/5$ (right), where $\epsilon_g = 0.2$. We note that the discontinuity of $\nabla_x l$ is now at $\epsilon_g > 0$, which means that the origin is an asymptotically stable equilibrium in the sense of Lyapunov. The parameter ϵ_g has no effect on the constraint $x \geq 2$. The original gradient ∇f is again shown in blue (dashed) and the functions αx and $\alpha(x-2)$ are shown in red (dashed).

Algorithm 2 Implementation of the gradient descent scheme (8).

Require: $x_0 \in \mathbb{R}^n$, $T > 0$, $\alpha T \in (0, 1]$, $\epsilon_g > 0$, $\omega \in (0, 2)$,
TOL, MAXITER, MAXITER_PROX, TOL_PROX

$k = 0$
while $k <$ MAXITER **do**
  Determine the set of closed constraints $I_k = \{i \in \mathbb{Z} \mid g_i(x_k) \leq \epsilon_g\}$
  Define $W_k := (\nabla h(x_k), \nabla g_i(x_k)_{i \in I_k})$ and $D_k := \mathbb{R}^{n_h} \times \mathbb{R}^{|I_k|}_{\geq 0}$
  Define $\bar{g}_k := (h(x_k), g_i(x_k)_{i \in I_k})$

  $j = 0$, $\lambda^0 = 0$ ▷ initialization with λ_{k-1} is also possible
  while $j <$ MAXITER_PROX **do**
    $\lambda^{j+1} = \text{prox}_{D_k}\left(\lambda^j - \omega D^{-1}(U^\mathsf{T}\lambda^{j+1} + \lambda^j - W_k^\mathsf{T}\nabla f(x_k) + \alpha \bar{g}_k)\right)$
    if $|\lambda^{j+1} - \lambda^j| \leq$ TOL_PROX **and** $\forall i > n_h : \lambda_i > 0$,
        $(W_k^\mathsf{T} W_k \lambda^{j+1} - W_k^\mathsf{T}\nabla f(x_k))_i + \alpha \bar{g}_{ki} \leq \epsilon_g \alpha T/2$, **then**
      break
    end if
  end while
  $\lambda_k = \lambda^{j+1}$

  Perform the update $x_{k+1} = x_k - T\nabla f(x_k) + T W_k \lambda_k$
  if $|x_{k+1} - x_k| \leq T \cdot$ TOL **then**
    return x_{k+1}
  end if
  $k \leftarrow k + 1$
end while

Parameters	1) Rand.QP	2) Trust region	3) ν-SVM	4) Catenary
T	$2/(L+\mu)$	$2/(\bar{L}_l+\mu)$	$2/(L+\mu)$	$2/n$
αT	0.4	0.4	0.4	0.8
ϵ_{g}	1e-6	1e-6	1e-6	1e-6
ω	1	1	1	1
TOL	1e-6	1e-6	1e-6	1e-6
MAXITER	1000	1000	1000	10000
MAXITER_PROX	200	200	200	10000
TOL_PROX	1e-6	1e-6	1e-6	1e-8

Table 1: Parameters of Algorithm 2 used for the experiments, where L and μ refer to the smoothness and strong convexity constants of f. The variable n denotes the number of chain links of the catenary, as defined in Section 7.4.

7.1 Randomly generated quadratic programs

We generate quadratic programs of the following form:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & \frac{1}{2}x^{\mathsf{T}}Qx + c^{\mathsf{T}}x, \\ \text{s.t.} \quad & Ax + b \in \mathbb{R}^{n/2}_{\geq 0} \times \mathbb{R}^{n/4}, \end{aligned}$$

where the entries of A and b are independent samples from a normal distribution with zero mean and unit variance, the entries of c are independent samples of a uniform distribution supported on $[-1, 1]$, and Q is a diagonal matrix. The first two diagonal elements of Q are set to 1/20 and 1, respectively, whereas the remaining elements are independent samples of a uniform distribution in $[1/20, 1]$. The condition number of Q is therefore fixed to 20. The problem dimension n is chosen such that $n/4$ (the number of equality constraints) and $n/2$ the number (of inequality constraints) are integers. We initialize Algorithm 2 with $x_0 = 0$, $\lambda_0 = 0$.

The results for a randomly generated quadratic program of size $n = 1000$ are shown in Figure 6. We observe very little difference between different randomly generated programs. We also observe little change when increasing n; even though the computational complexity increases, the number of iterations required for convergence remains at about 35, the maximum number of iterations that are required for computing λ_k remains at about 70, and only about 50% of the inequality constraints are active. Figure 7 compares the runtime of Algorithm 2 to the interior point solver CVXOPT.[1] The execution time of Algorithm 2 scales favorably in the problem dimension n. For $n = 20,000$ the execution time is roughly reduced by a factor of five; larger improvements seem possible when increasing n further.

1. We ran CVXOPT by exploiting sparsity of the Hessian and standard tolerance settings.

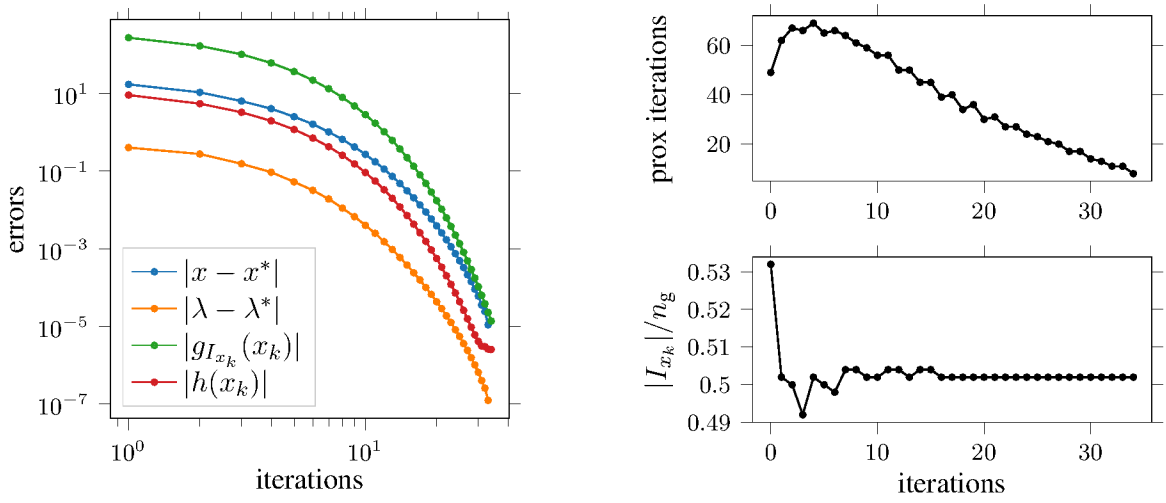

Figure 6: Trajectories for a single randomly generated convex quadratic program with $n = 1000$. The figure on the left indicates linear convergence of the iterate x_k, the multiplier λ_k, and the constraint violations. The figures on the right display the number of iterations of the inner loop of Algorithm 2 (top) and the ratio of constraints that enter $|I_{x_k}|$ (bottom).

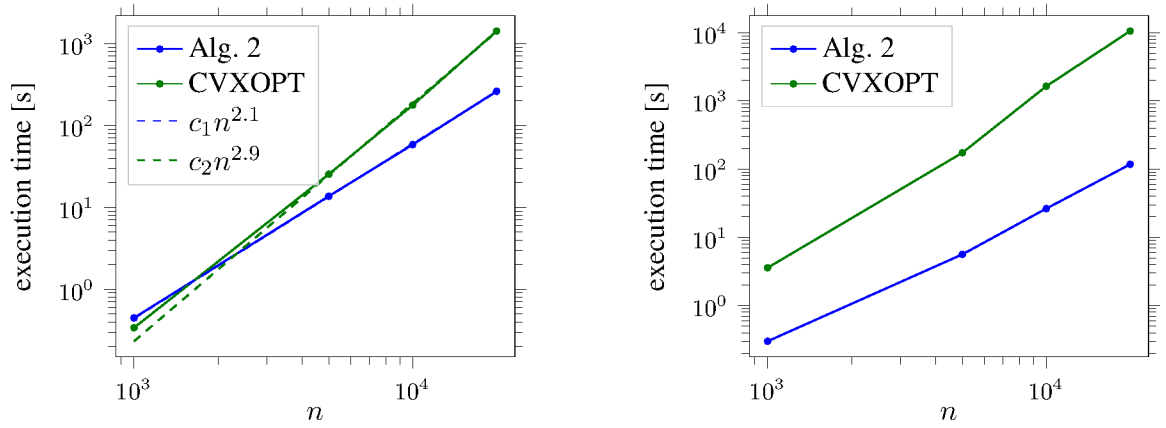

Figure 7: This plot shows the results obtained for randomly generated quadratic programs (left) and the randomly generated trust region problems (right). In the case of quadratic programs (left), Algorithm 2 seems to achieve a better scaling with respect to the problem size n (an exponent of 2.1 instead of 2.9), leading to a speedup of a factor of roughly 5.5 for $n = 2 \cdot 10^4$. For the trust region problems (right), Algorithm 2 achieves a speedup of roughly two orders of magnitude for large n; the scaling with n seems similar, however (the execution time of Algorithm 2 scales roughly with $n^{2.2}$).

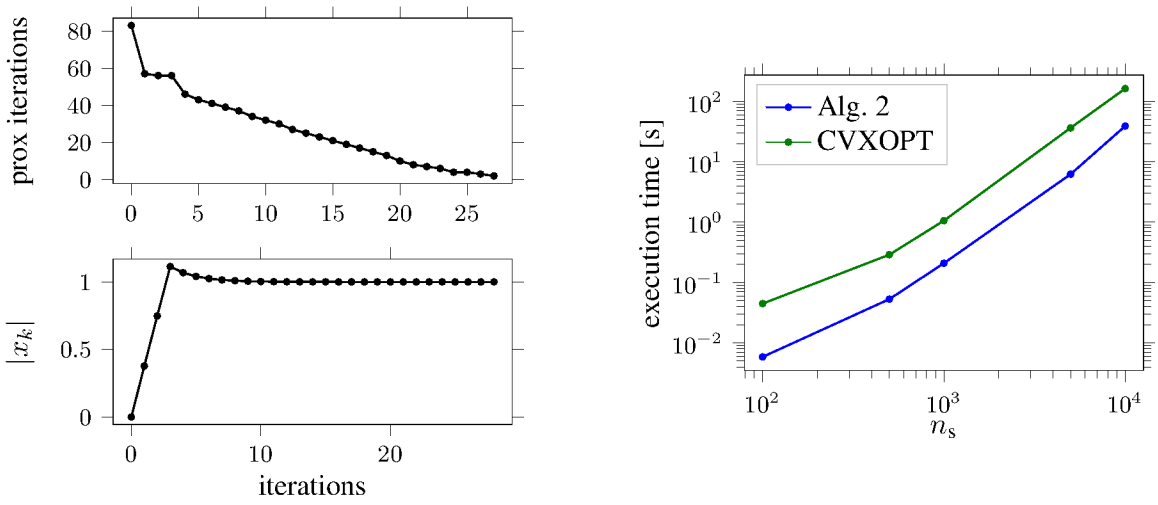

Figure 8: The left panel shows a trajectory of Algorithm 2 for the trust region problem. The top right indicates the number of iterations required in the inner loop for computing the multiplier λ_k, which decreases steadily. The constraint $|x_k| \leq 1$ is initially not active, leading to a violation at the fourth iteration. The violation then decreases at a linear rate, which parallels the continuous-time case. The right graph shows the execution times for the ν -support vector machine when varying n. Compared to CVXOPT, a constant speedup of roughly a factor of five can be observed across all problem instances.

7.2 Trust-region optimization

In order to demonstrate that Algorithm 2 can efficiently handle nonlinear constraints, we extend the example of the previous section and consider the trust-region optimization

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \frac{1}{2} x^\mathsf{T} Q x + c^\mathsf{T} x, \\ \text{s.t.} \quad & Ax \in \mathbb{R}^{n/2}_{\geq 0} \times \mathbb{R}^{n/4}, \quad |x|^2 \leq 1, \end{aligned}$$

where the matrices Q and A and the vector c are generated as in Section 7.1. According to Appendix B, the constant L_l can be upper bounded as

$$L_l \leq \bar{L}_l := \alpha + L(2 + |Q^{-1}c|\sqrt{2}/2).$$

We choose $T = 2/(\bar{L}_l + \mu)$ and $\alpha T = 0.4$, which parallels the previous section. Figure 8 (left) shows the number of iterations needed for computing λ_k and the evolution of $|x_k|$ on an example with $n = 1000$. The iterations of the inner loop are comparable to Section 7.1. The constraint $|x_k| < 1$ is initially not active leading to a violation at the fourth iteration. At this point, the constraint enters the set $|I_{x_k}|$ and its violation decreases linearly over the remaining iterations. Figure 7 (right) shows how the execution time scales with the problem size n. Compared to CVXOPT, a speedup of up to two orders of magnitude is achieved.

7.3 ν - support vector machine

We use the support vector machine formulation suggested by Schölkopf et al. (2000), which leads to the following quadratic program:

$$\begin{aligned} \min_{x \in \mathbb{R}^{n_\mathrm{s}}} \quad & \frac{1}{2} \sum_{i=1}^{n_\mathrm{s}} \sum_{j=1}^{n_\mathrm{s}} x_i x_j l_i l_j k(r_i, r_j) + \frac{\nu_1}{2} |x|^2 \\ \text{s.t.} \quad & 0 \leq x_i \leq 1/n_\mathrm{s}, \quad \sum_{n=1}^{n_\mathrm{s}} x_i l_i = 0, \quad \sum_{i=0}^{n_\mathrm{s}} x_i \geq \nu_2, \end{aligned}$$

where $r_i \in \mathbb{R}^2$ are the training samples with labels $l_i \in \{-1, 1\}$, $i = 1, \ldots, n_\mathrm{s}$, the integer $n_\mathrm{s} > 0$ denotes the number training samples, ν_1 and ν_2 are regularization parameters, and $k : \mathbb{R}^2 \to \mathbb{R}^2$ is the kernel function. The kernel is chosen to be a radial basis function kernel with unit standard deviation. We set $\nu_1 = 0.1 \mu_\mathrm{k}$ and $\nu_2 = 0.1$, where μ_k denotes the smallest eigenvalue of the kernel matrix $k(r_i, r_j)$. The parameter ν_1 therefore improves the conditioning of the Hessian, whereas the parameter ν_2 can be interpreted as an upper bound on the fraction of margin errors; i.e., the training samples which lie on the "wrong" side of the boundary. It is clear that Algorithm 2, which is based on gradient descent, has difficulties with ill-conditioned objective functions (its rate that scales with $1/\kappa$). The purpose of the regularization with ν_1 is to reduce these effects.

We generate the training samples in the following way: The points with label +1 are generated in polar coordinates where the radius is sampled from a normal distribution with mean two and standard deviation 0.5, and the angle is uniformly sampled in $[0, 2\pi)$. The points with label -1 are likewise generated in polar coordinates where the radius is sampled from a normal distribution with mean zero and standard deviation 0.5, and the angle is uniformly sampled in $[0, 2\pi)$. As an example, the training data and the resulting classifier are shown in Figure 9 for $n_\mathrm{s} = 1000$. Due to the nature of the problem, only very few inequality constraints tend to be active at the optimum (in this case just one). The numerical results indicate that Algorithm 2 can indeed take advantage of this fact and identifies the correct active inequality constraint after very few iterations (in this case just one). The number of constraints that enter the computation of the reaction force R_k is therefore largely reduced after the first iterations enabling a rapid convergence of the inner loop of Algorithm 2.

Figure 8 shows how the execution time scales with the problem dimension n_s. Compared to CVXOPT, we observe a speedup of a factor of five across all problem instances. The scaling with n_s seems similar.

7.4 Catenary

We consider an idealized chain of length two, which has n chain links and is suspended at the points $(0, 0)$ and $(1, 0)$ (in a two-dimensional coordinate system). The aim is to solve the following problem:

$$\begin{aligned} \min_{(x,y) \in \mathbb{R}^{n+1} \times \mathbb{R}^{n+1}} \quad & \frac{9.81}{n+1} \sum_{i=2}^{n} y_i \\ \text{s.t.} \quad & |x_i - x_{i+1}|^2 + |y_i - y_{i+1}|^2 = 4/n^2, \\ & |x_i - 0.5|^2 + |y_i + 0.8|^2 \geq 0.5^2, \quad i = 1, \ldots, n \\ & (x_1, y_1) = (0, 0), \quad (x_{n+1}, y_{n+1}) = (1, 0). \end{aligned}$$

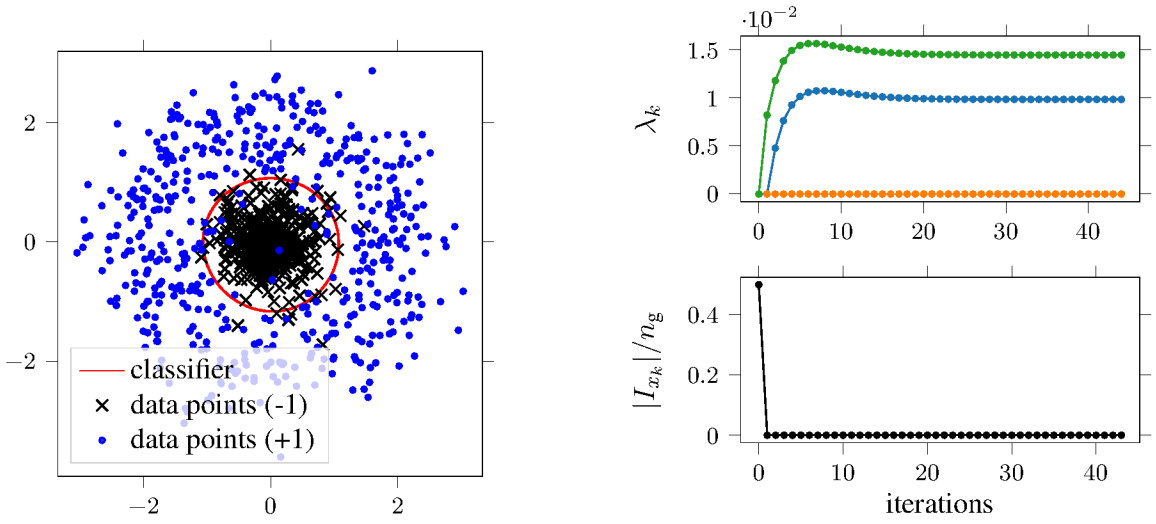

Figure 9: This figure shows the training data and the resulting classifier (left), as well as the dual variables λ_k (top right) and the percentage of constraints that are active (bottom right). Only two dual variables are nonzero: The first one corresponds to the equality constraints (blue line), whereas the second one corresponds to the support constraint (green line). All remaining constraints are inactive.

The position of the ith joint is described by the tuple (x_i, y_i) and we have included the nonlinear constraint that each joint is required to lie outside a circle centered at $(0.5, -0.8)$ with radius 0.5 (the chain therefore lies on a circular object). The cost function captures the potential energy of the chain. We found that a time step of $T = 2/n$ works well, which can be motivated by the fact that L_l is roughly $\mathcal{O}(n)$ (considering the continuous limit of the chain). The results for a chain of length $n = 40$ can be found in Figure 10. Starting from a random initialization that violates the equality constraints (see Figure 10 (left, black)) the solution evolves and finds a local minimum that satisfies all the constraints. We note that the cost has a plateau at about iteration 1000, which corresponds to a symmetric shape, where the chain lies on top of the round object (see Figure 10 (left, green)). This corresponds to an unstable equilibrium, since the slightest deviation will cause the chain to slide down either to the left or the right. This is precisely what we observe in our numerical results, leading to the final solution shown in red.

8. Conclusions

We have presented a new class of primal first-order algorithms for smooth constrained optimization. The key feature of these algorithms is that at each iteration, a low-dimensional, local, and convex approximation of the feasible set is constructed and used for computing the next iterate. The local approximation is a natural generalization of the tangent cone (in the sense of Clarke) to include infeasible points. It can be motivated by drawing analogies to non-smooth mechanical systems and can be viewed as a reformulation of constraint optimization on the velocity level. That is, the algorithm imposes a constraint on $x_{k+1} - x_k$ rather than on x_k. While in our continuous-time

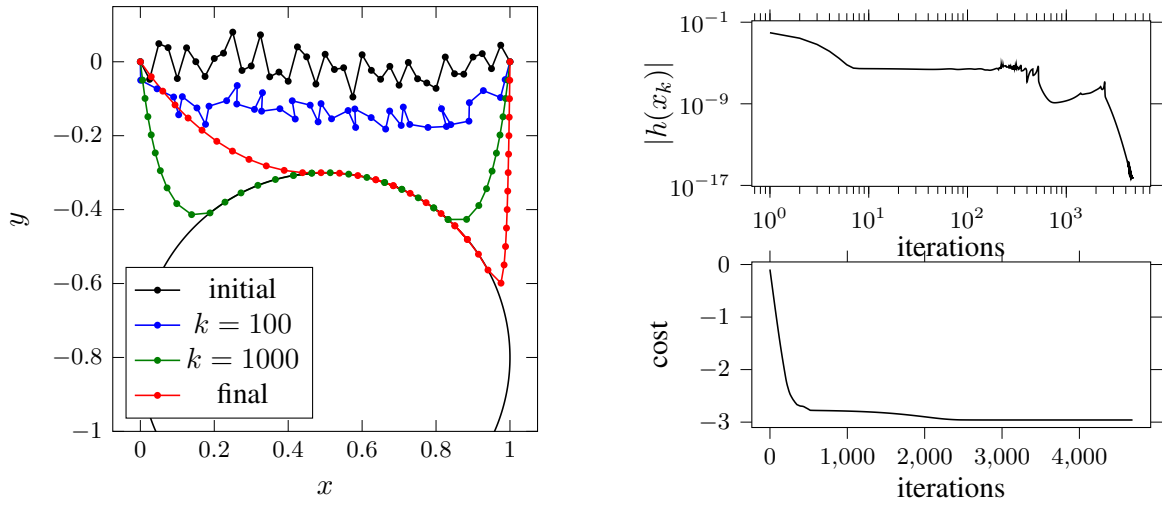

Figure 10: This figure shows the evolution of the solution of the catenary problem (left) as well as the violation of the equality constraints and the evolution of the cost (right). We can clearly see that the symmetric shape (roughly corresponding to the solution at time $k = 1000$) is suboptimal, and unstable from a physics perspective. Thus, the chain slides to the right and reaches a lower energy state. (We suspect that the random initialization and the finite precision brakes the symmetry.)

formulation constraints on the position level and the velocity level are equivalent, this is no longer true for the resulting discrete-time algorithms. We found that a formulation of constraints on the velocity level leads to efficient first-order algorithms that avoid projection or optimization over the entire feasible set at each iteration. This simplification does a more complex theoretical analysis, but, as we have shown, that analysis can be carried out with a blend of ideas from dynamical systems and mathematical optimization.

The purpose of the article was to highlight and explain our different point of view on constrained optimization. Many aspects deserve a more thorough treatment. For example, we have not discussed existence of solutions to the non-smooth differential equations or the differential inclusions that were introduced. Similarly, the strong convexity assumptions on the objective function for proving convergence of our discrete algorithm can most likely be relaxed, and the numerical experiments do not include an extensive comparison to different state-of-the-art solvers. We also acknowledge that there are software packages that are tailored to, for example, support vector machines, which would most likely outperform our method by orders of magnitudes. However, we felt that a thorough discussion of all these issues is outside of the scope of the present article and would distract the reader from the main points.

There are many opportunities for further research in this vein. In particular, the analogies to non-smooth mechanical systems that are made throughout the article enable extensions to Newton-type methods or accelerated first-order methods. We hope that our perspective helps to trigger further developments at the intersection between non-smooth dynamics, constrained optimization, and machine learning.

Acknowledgments

We thank the German Research Foundation and the Branco Weiss Fellowship, administered by ETH Zurich, for the generous support. We also thank the Office of Naval Research under grant number N00014-18-1-2764.

Appendix A. Barbalat's Lemma

Lemma 11 *(Variant of Barbalat's lemma) Let $\xi : \mathbb{R}_{\geq 0} \to \mathbb{R}$ be piecewise continuous, such that*

$$-\infty < \int_0^\infty \xi(\tau)\mathrm{d}\tau, \quad \xi(t)^+ \geq \xi(t)^-, \quad \xi(t_2) - \xi(t_1) \geq -\bar{r}(t_2 - t_1),$$

for any $t_2 \geq t_1 > 0$ such that ξ is continuous on (t_1, t_2) and any $t \geq 0$. If $\bar{\xi} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$, with $\bar{\xi}(x) := \max\{\xi(x), 0\}$, is integrable and such that $\lim_{t\to\infty} \bar{\xi}(t) = 0$, then $\lim_{t\to\infty} \xi(t) = 0$ holds.

Proof The proof follows a standard argument, which is also used for proving Barbalat's lemma (see, e.g., Sastry, 1999, p. 204). We start by assuming that $\lim_{t\to\infty} \xi(t) \neq 0$ (provided that $\lim_{t\to\infty} \xi(t)$ exists) and show that this leads to a contradiction. This means that there exists an $\epsilon > 0$ and a sequence $t_k \geq 0$, such that $\xi(t_k) < -\epsilon$ for all $k > 0$ (taking into account that $\lim_{t\to\infty} \bar{\xi}(t) = 0$). However, since $\xi(t)^+ \geq \xi(t)^-$ at every t where ξ is discontinuous, we conclude that $\xi(t) \leq \xi(t_1) + \bar{r}(t_1 - t)$ for all $t \leq t_1$, where $t_1 > 0$ is arbitrary (looking backwards in time, the function increases by a slope of at most $\bar{r}$). For each t_k, we thus conclude $\xi(t) < -\epsilon/2$ as long as $t \in (t_k - \epsilon/(2\bar{r}), t_k)$. This means that for any subsequence t_{kj}, $j = 1, 2, \ldots$ such that $t_{k(j+1)} > t_{kj} + \epsilon/(2\bar{r})$,

$$\int_0^\infty \xi(\tau)\mathrm{d}\tau = \sum_{j=1}^\infty \int_{t_{k(j-1)}}^{t_{kj}-\epsilon/(2\bar{r})} \xi(\tau)\mathrm{d}\tau + \int_{t_{kj}-\epsilon/(2\bar{r})}^{t_{kj}} \xi(\tau)\mathrm{d}\tau \leq \int_0^\infty \bar{\xi}(\tau)\mathrm{d}\tau - \sum_{j=1}^\infty \epsilon^2/(4\bar{r}),$$

where t_{k0} is defined as $t_{k0} = 0$ for notational convenience. The right-hand side is unbounded below leading to the desired contradiction. ■

Appendix B. Nonlinear constraints

When estimating the constant L_l, a bound on λ is often useful. The following proposition, which can be generalized to multiple constraints by a similar argument, establishes such a bound.

Proposition 12 *Let $g : \mathbb{R}^n \to \mathbb{R}$ be a scalar L_g-smooth and μ_g-strongly concave function. Then, in the absence of any other constraints, the corresponding multiplier $\lambda > 0$ is bounded by*

$$\lambda \leq \frac{1}{\mu_g}\left(\alpha + L(1 + |x_f - x_g|\sqrt{L_g/(2g(x_g))})\right),$$

where x_f is the (unconstrained) minimizer of f, x_g the (unconstrained) maximizer of g, L the smoothness constant of f, and $\kappa_g := L_g/\mu_g$.

Proof It follows from (5) that

$$\frac{1}{2}|W(x)\lambda|^2 = \frac{1}{2}|v(x) + \nabla f(x)|^2 \leq \frac{1}{2}|v_{\mathrm{f}} + \nabla f(x)|^2,$$

for any $v_{\mathrm{f}} \in V_\alpha(x)$. In particular, we can set $v_{\mathrm{f}} = \alpha(x_{\mathrm{g}} - x)$, which satisfies $v_{\mathrm{f}} \in V_\alpha(x)$, due to the concavity of g. Moreover, from $W(x) = \nabla g(x)$ and the strong concavity of g we conclude

$$\mu_{\mathrm{g}}|x - x_{\mathrm{g}}|\lambda \leq |\alpha(x_{\mathrm{g}} - x) + \nabla f(x)| \leq \alpha|x - x_{\mathrm{g}}| + L|x - x_{\mathrm{f}}|,$$

where $\lambda > 0$ by definition of λ. This yields the following bound on the dual variable

$$\lambda \leq \sup_{g(x)\leq 0} \frac{1}{\mu_{\mathrm{g}}} \left(\alpha + \frac{L|x - x_{\mathrm{f}}|}{|x - x_{\mathrm{g}}|} \right),$$

which can be further simplified to

$$\lambda \leq \frac{1}{\mu_{\mathrm{g}}} \left(\alpha + L + L|x_{\mathrm{g}} - x_{\mathrm{f}}| \sup_{g(x)\leq 0} \frac{1}{|x - x_{\mathrm{g}}|} \right).$$

Due to the strong concavity of g, it follows that $g(x) \geq g(x_{\mathrm{g}}) - L_{\mathrm{g}}|x - x_{\mathrm{g}}|^2/2$ for all $x \in \mathbb{R}^n$. As a consequence, $L_{\mathrm{g}}|x - x_{\mathrm{c}}|^2/2 \geq g(x_{\mathrm{c}})$, for all $x \in \mathbb{R}^n$ such that $g(x) \leq 0$, which means that the last supremum is bounded by $\sqrt{L_{\mathrm{g}}/(2g(x_{\mathrm{g}}))}$. ■

Appendix C. Proof of Proposition 10

Proof The proof follows the presentation of Cottle et al. (2009, p. 400). In order to simplify the notation we define $G := W_k^{\mathsf{T}} W_k$, $q := -W_k^{\mathsf{T}} \nabla f(x_k) + \alpha \bar{g}(x_k)$, $B := U^{\mathsf{T}} + \omega^{-1} D$, $C := U + (1 - \omega^{-1})D$, and omit the subscript k. We can therefore express (24) concisely as

$$G\lambda + q + \partial \psi_{D_x}(\lambda) \ni 0.$$

Furthermore, by virtue of the conjugate subgradient theorem, (25) is equivalent to

$$\lambda^{j+1} \in D_x, \quad -B\lambda^{j+1} - C\lambda^j - q \in D_x^*, \quad {\lambda^{j+1}}^{\mathsf{T}}(-B\lambda^{j+1} - C\lambda^j - q) = 0, \tag{29}$$

where $D_x^* := \{0\}^{n_{\mathrm{h}}} \times \mathbb{R}_{\leq 0}^{|I_x|}$ is the polar cone of D_x. We further introduce the function $\tilde{d} : D_x \to \mathbb{R}$, $\tilde{d}(\lambda) = \lambda^{\mathsf{T}} G\lambda/2 + \lambda^{\mathsf{T}} q$. Due to the fact that G is positive semi-definite, $\tilde{d}$ is convex and can be shown to be bounded below for $\lambda \in D_x$. We further have that

$$\tilde{d}(\lambda^j) - \tilde{d}(\lambda^{j+1}) = (\lambda^j - \lambda^{j+1})^{\mathsf{T}}(q + G\lambda^{j+1}) + \frac{1}{2}(\lambda^j - \lambda^{j+1})^{\mathsf{T}} G(\lambda^j - \lambda^{j+1}).$$

As a consequence of (29) and some elementary manipulations, the decrease in $\tilde{d}$ can be expressed as

$$\begin{aligned}\tilde{d}(\lambda^j) - \tilde{d}(\lambda^{j+1}) &= {\lambda^j}^{\mathsf{T}}(q + B\lambda^{j+1} + C\lambda^j) + \frac{1}{2}(\lambda^j - \lambda^{j+1})^{\mathsf{T}}(B - C)(\lambda^j - \lambda^{j+1}) \\ &\geq \frac{1}{2}(\lambda^j - \lambda^{j+1})^{\mathsf{T}}(B - C)(\lambda^j - \lambda^{j+1}).\end{aligned}$$

For the last inequality we have used $-B\lambda^{j+1} - C\lambda^j - q \in D_x^*$ and $\lambda^j \in D_x$, which ensures that $(q + B\lambda^{j+1} + C\lambda^j)^\mathsf{T}\lambda^j \geq 0$. The symmetric part of $B - C$ is given by $(2\omega^{-1} - 1)D$, which is guaranteed to be positive definite for $\omega \in (0, 2)$ (the elements of D are given by $|\nabla g_i(x)|^2 > 0$). This concludes that $\tilde{d}(\lambda^j)$ is a monotonically decreasing sequence, which therefore converges. Thus, the above inequality implies, in the limit as $j \to \infty$,

$$0 = \lim_{j\to\infty} \frac{1}{2}(\lambda^j - \lambda^{j+1})^\mathsf{T}(B - C)(\lambda^j - \lambda^{j+1}),$$

which, due to the positive definiteness of the symmetric part of $B - C$, implies that λ^j converges. Moreover, $\lim_{j\to\infty} \lambda^j$ satisfies (24) by construction. ■

References

Vincent Acary. Higher order event capturing time-stepping schemes for nonsmooth multibody systems with unilateral constraints and impacts. *Applied Numerical Mathematics*, 62(10):1259–1275, 2012.

Martin Andersen, Joachim Dahl, Zhang Liu, and Lieven Vandenberghe. Interior-point methods for large-scale cone programming. In *Optimization for Machine Learning*, pages 55–79. The MIT Press, 2011.

J. P. Aubin and A. Cellina. *Differential Inclusions*. Springer, 1984.

Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2011.

Amir Beck and Marc Teboulle. Gradient-based algorithms with applications to signal-recovery problems. In *Convex Optimization in Signal Processing and Communications*, pages 42–88. Cambridge University Press, 2011.

Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, second edition, 1999.

Michael Betancourt, Michael I. Jordan, and Ashia C. Wilson. On symplectic optimization. *arXiv:1802.03653v2*, pages 1–20, 2018.

Ernesto G. Birgin, José Mario Martínez, and Marcos Raydan. Inexact spectral projected gradient methods on convex sets. *IMA Journal of Numerical Analysis*, 23:539–559, 2003.

Veronica Bloom, Igor Griva, and Fabio Quijada. Fast projected gradient method for support vector machines. *Optimization and Engineering*, 17(4):651–662, 2016.

B. Brogliato, A. Daniilidis, C. Lemaréchal, and V. Acary. On the equivalence between complementarity systems, projected systems and differential inclusions. *System and Control Letters*, 55(1): 45–51, 2006.

G. Capobianco and S. R. Eugster. Time finite element based Moreau-type integrators. *International Journal for Numerical Methods in Engineering*, 114(3):215–231, 2018.

Kenneth L. Clarkson. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Transactions on Algorithms*, 6(4):1–30, 2010.

Cyrille W. Combettes and Sebastian Pokutta. Boosting Frank-Wolfe by chasing gradients. *Proceedings of Machine Learning Research*, 119:2111–2121, 2020.

Richard W. Cottle, Jong-Shi Pang, and Richard E. Stone. *The Linear Complementarity Problem*. Society for Industrial and Applied Mathematics, 2009.

Jelena Diakonikolas and Michael I. Jordan. Generalized momentum-based methods: A Hamiltonian perspective. *SIAM Journal on Optimization*, 31(1):915–944, 2021.

Alexander Domahidi, Aldo U. Zgraggen, Melanie N. Zeilinger, Manfred Morari, and Colin N. Jones. Efficient interior point methods for multistage problems arising in receding horizon control. *Proceedings of the International Conference on Decision and Control*, pages 668–674, 2012.

Michael C. Ferris and Todd S. Munson. Interior point methods for massive support vector machines. *SIAM Journal on Optimization*, 13(3):783–804, 2003.

A. F. Filippov. *Differential Equations with Discontinuous Righthand Sides*. Springer, 1988.

Guilherme França, Jeremias Sulam, Daniel P. Robinson, and René Vidal. Conformal symplectic and relativistic optimization. *Journal of Statistical Mechanics: Theory and Experiment*, 2020 (12):1–30, 2020.

Dan Garber and Elad Hazan. Faster rates for the Frank-Wolfe method over strongly-convex sets. *Proceedings of Machine Learning Research*, 37:541–549, 2015.

Philip E. Gill, Walter Murray, and Michael A. Saunders. SNOPT: an SQP algorithm for large-scale constrained optimization. *SIAM Review*, 47(1):99–131, 2005.

Christoph Glocker. *Set-Valued Force Laws*. Springer, 2001.

Verena Häberle, Adrean Hauswirth, Lukas Ortmann, Saverio Bolognani, and Florian Dörfler. Nonconvex feedback optimization with input and output constraints. *IEEE Control Systems Letters*, 5(1):343–348, 2021.

Elad Hazan and Satyen Kale. Projection-free online learning. *Proceeding of the International Conference on Machine Learning*, pages 1–8, 2012.

Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. *Proceedings of Machine Learning Research*, 28(1):427–435, 2013.

Wenzel Jakob, Jason Rhinelander, and Dean Moldovan. pybind11 – seamless operability between C++11 and Python, 2017. https://github.com/pybind/pybind11.

Kwangmoo Koh, Seung-Jean Kim, and Stephen Boyd. An interior-point method for large-scale ℓ_1-regularized logistic regression. *Journal of Machine Learning Research*, 8:1519–1555, 2007.

Walid Krichene, Alexandre M. Bayen, and Peter L. Bartlett. Accelerated mirror descent in continuous and discrete time. *Advances in Neural Information Processing Systems 28*, pages 2845–2853, 2015.

Cornelius Lanczos. *The Variational Principles of Mechanics*. Oxford University Press, second edition, 1952.

David G. Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*. Springer, fourth edition, 2016.

H. May and P. D. Panagiotopoulous. F. H. Clarke's generalized gradient and Fourier's principle. *Journal of Applied Mathematics and Mechanics*, 65(2):125–126, 1985.

J. J. Moreau. Unilateral contact and dry friction in finite freedom dynamics. In *Nonsmooth Mechanics and Applications*, pages 1–88. Springer, 1988.

Michael Muehlebach and Michael I. Jordan. A dynamical systems perspective on Nesterov acceleration. *Proceedings of Machine Learning Research*, 97:4656–4662, 2019.

Michael Muehlebach and Michael I. Jordan. Continuous-time lower bounds for gradient-based algorithms. *Proceedings of Machine Learning Research*, 119:7088–7096, 2020.

Michael Muehlebach and Michael I. Jordan. Optimization with momentum: Dynamical, control-theoretic, and symplectic perspectives. *Journal of Machine Learning Research*, 22(73):1–50, 2021.

Yurii Nesterov. *Introductory Lectures on Convex Optimization - A Basic Course*. Springer Science+Business Media, LLC, 2004.

Yurii Nesterov and Arkadii Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research, second edition, 2006.

Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1 (3):123–231, 2013.

R. Tyrrell Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

R. Tyrrell Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

R. Tyrrell Rockafellar and Roger J-B Wets. *Variational Analysis*. Springer, 1997.

Shankar Sastry. *Nonlinear Systems*. Springer, 1999.

Bernhard Schölkopf, Alex J. Smola, Robert C. Williamson, and Peter L. Bartlett. New support vector algorithms. *Neural Computation*, 12(5):1207–1245, 2000.

Christian Studer. *Numerics of Unilateral Contacts and Friction*. Springer, 2009.

Weijie Su, Stephen Boyd, and Emmanuel J. Candès. A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights. *Journal of Machine Learning Research*, 17 (153):1–43, 2016.

Giampaolo Torrisi, Sergio Grammatico, Roy S. Smith, and Manfred Morari. A projected gradient and constraint linearization method for nonlinear model predictive control. *SIAM Journal on Control and Optimization*, 56(3):1968–1999, 2018.

Changyu Wang and Qian Liu. Convergence properties of inexact projected gradient methods. *Optimization*, 55(3):301–310, 2006.

Andre Wibisono, Ashia C. Wilson, and Michael I. Jordan. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 113(47):E7351–E7358, 2016.

Mingrui Zhang, Zebang Shen, Arzan Mokhtari, Hamed Hassani, and Amin Karbasi. One sample stochastic Frank-Wolfe. *Proceedings of Machine Learning Research*, 108:4012–4023, 2020.