Les Cahiers du GERAD ISSN: 0711–2440

A nonsmooth exact penalty method for equality-constrained optimization: Complexity and implementation

Y. Diouane, M. Gollier, D. Orban

G–2024–65

Septembre 2024

GERAD HEC Montréal
3000, chemin de la Côte-Sainte-Catherine
Montréal (Québec) Canada H3T 2A7

Tél. : 514 340-6053
Téléc. : 514 340-5665
info@gerad.ca
www.gerad.ca

A nonsmooth exact penalty method for equality-constrained optimization: Complexity and implementation

Youssef Diouane
Maxence Gollier
Dominique Orban

GERAD & Department of Mathematics and Industrial Engineering, Polytechnique Montréal, Montréal (Qc), Canada, H3T 1J4

youssef.diouane@polymtl.ca
maxence-2.gollier@polymtl.ca
dominique.orban@gerad.ca

Septembre 2024
Les Cahiers du GERAD
G–2024–65

Abstract : Penalty methods are a well known class of algorithms for constrained optimization. They transform a constrained problem into a sequence of unconstrained *penalized* problems in the hope that approximate solutions of the latter converge to a solution of the former. If Lagrange multipliers exist, exact penalty methods ensure that the penalty parameter only need increase a finite number of times, but are typically scorned in smooth optimization for the penalized problems are not smooth. This led researchers to consider the implementation of exact penalty methods inconvenient. Recently, advances in proximal methods have led to increasingly efficient solvers for nonsmooth optimization. We show that the exact ℓ_2-penalty method for equality-constrained optimization can in fact be implemented efficiently by solving the penalized problem with a proximal-type algorithm. We study the convergence of our algorithm and establish a worst-case complexity bound of $\mathcal{O}(\epsilon^{-2})$ to bring a stationarity measure below $\epsilon > 0$ under the Mangarasian-Fromowitz constraint qualification and Lipschitz continuity of the objective gradient and constraint Jacobian. In a degenerate scenario where the penalty parameter grows unbounded, the complexity becomes $\mathcal{O}(\epsilon^{-8})$, which is worse than another bound found in the literature. We justify the difference by arguing that our feasibility measure is properly scaled. Finally, we report numerical experience on small-scale problems from a standard collection and compare our solver with an augmented-Lagrangian and an SQP method. Our preliminary implementation is on par with the augmented Lagrangian in terms of robustness and efficiency. It is on par with the SQP method in terms of robustness, though the former remains ahead in terms of number of problem function evaluations.

Keywords : Equality-constrained optimization, penalty methods, proximal methods

Résumé: Les méthodes de pénalité constituent une classe bien connue d'algorithmes pour l'optimisation sous contraintes. Elles transforment un problème contraint en une séquence de problèmes sans contraintes dans l'espoir que les solutions approximatives de ces derniers convergent vers une solution du premier. S'il existe des multiplicateurs de Lagrange, les méthodes de pénalité exacte garantissent que le paramètre de pénalité ne doit augmenter qu'un nombre fini de fois, mais elles sont généralement ignorées dans l'optimisation lisse car les problèmes pénalisés ne sont pas lisses. Cela a conduit les chercheurs à considérer que l'implémentation des méthodes de pénalités exactes n'était pas pratique. Récemment, les progrès des méthodes proximales ont conduit à des solveurs de plus en plus efficaces pour l'optimisation non lisse. Nous montrons que la méthode de pénalité exacte ℓ_2 pour l'optimisation avec contraintes d'égalité peut en fait être implémentée efficacement en résolvant le problème pénalisé avec un algorithme de type proximal. Nous étudions la convergence de notre algorithme et établissons une borne de complexité dans le pire des cas de $\mathcal{O}(\epsilon^{-2})$ pour ramener une mesure de stationnarité en dessous de $\epsilon > 0$ sous la contrainte de qualification de Mangarasian-Fromowitz et la Lipschitz continuité du gradient de l'objectif et du jacobien de la contrainte. Dans un scénario dégénéré où le paramètre de pénalité croît sans limite, la complexité devient $\mathcal{O}(\epsilon^{-8})$, ce qui est pire qu'une autre borne trouvée dans la littérature. Nous justifions cette différence en faisant valoir que notre mesure de faisabilité est correctement échelonnée. Enfin, nous présentons une expérience numérique sur des problèmes à petite échelle provenant d'une collection standard et nous comparons notre solveur à une méthode de lagrangien augmentée et à une méthode SQP. Notre implémentation préliminaire est à la hauteur du Lagrangien augmenté en termes de robustesse et d'efficacité. Elle est à égalité avec la méthode SQP en termes de robustesse, bien que la première reste en tête en termes de nombre d'évaluations de la fonction du problème.

Keywords : Optimisation sous contraintes d'égalité, méthodes de pénalité, méthodes proximales

---

Acknowledgements: We would like to thank Tangi Migot and Alexis Montoison for their kind assistance with the numerical implementation of our algorithm. The research of Y. Diouane and D. Orban is partially supported by an NSERC Discovery Grant.

1 Introduction

We consider the problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; f(x) \quad \text{subject to} \;\; c(x) = 0, \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^m$ are $\mathcal{C}^1$ and both may be nonconvex. We solve (1) by solving a sequence of unconstrained, nonsmooth penalized problems

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; f(x) + h_\tau(c(x)), \tag{2}$$

where $h_\tau = \tau\|\cdot\|_2$ and $\tau > 0$ is the penalty parameter. This approach, was first proposed by Pietrzykowski [45], who used $h_\tau = \tau\|\cdot\|_1$. Penalty approaches such as (2) are attractive because, under standard assumptions, for all sufficiently large and finite τ, solutions of (1) are solutions of (2). However, the nonsmoothness of h_τ caused them to fall out of favor in practice, and other methods, such as the augmented-Lagrangian method [34, 46] were preferred.

Independently, attention was given recently to nonsmooth regularized problems

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; f(x) + h(x), \tag{3}$$

where f is as before, and h is proper and lower semi-continuous (lsc), and may be nonconvex. In particular, proximal methods [29, 38] have been studied intensively in the last decade and have led to increasingly efficient implementations.

We show that proximal methods can be used to implement exact penalty methods so they perform similarly to augmented-Lagrangian approaches. To do so, we study an intuitive algorithm that enacts the exact penalty scheme, but in which subproblems are solved by a first-order proximal method proposed in [2, 3], and a second-order proximal method proposed in [23]. We establish convergence of an appropriate stationarity measure to zero, and a worst-case complexity bound of $\mathcal{O}(\epsilon^{-2})$ to bring said measure below $\epsilon > 0$ under the Mangarasian-Fromowitz constraint qualification and Lipschitz continuity of the objective gradient and constraint Jacobian. In a degenerate scenario where the penalty parameter grows unbounded, the complexity becomes $\mathcal{O}(\epsilon^{-8})$, which is worse than the bound $\mathcal{O}(\epsilon^{-5})$ developed in [15, §3]. Although both bounds are correct, we justify the difference by arguing that our feasibility measure is properly scaled. As far as we are aware, ours is the first implementation of the exact penalty scheme based on proximal methods. We report numerical experience on small-scale problems from a standard collection and compare our solver with an augmented-Lagrangian and an SQP method. Our preliminary implementation is on par with the augmented Lagrangian in terms of robustness and efficiency. It is on par with the SQP method in terms of robustness, though the former remains ahead in terms of number of problem function evaluations.

The paper is organized as follows. In Section 2 we recall key background concepts and results. In Section 4, we present our algorithm and derive complexity bounds. In Section 5 we derive closed-form solutions of the proximal operators that arise in our algorithm. Additionally, we provide algorithms to evaluate the required proximal operators efficiently in practice. In Section 6 we show numerical results and experiments conducted on a variety of problems. Finally, we provide a closing discussion in Section 7

Related research

Pietrzykowski [45] first reported the advantage of using a nonsmooth penalty function. His analysis only covered the ℓ_1-penalty but subsequently, Charalambous [16], Han and Mangasarian [33], Coleman and Conn [18], Bazaraa and Goode [7], and Huang and Ng [36] strengthened his results. Cartis et al. [15] analyze general *composite* objectives, i.e., of the form $h(c(x))$, where h is convex and globally Lipschitz using a trust-region approach and a quadratic regularization variant. They then apply their results to

an exact penalty method similar to ours. Grapiglia et al. [32] generalize the results of [15] to various optimization frameworks under similar assumptions. Contrary to their algorithms, ours is concrete in the sense that we provide precise guidance on how to compute steps. Additionally, our convergence analysis holds under weaker assumptions.

Several authors reformulate (2) as a smooth inequality-constrained problem. For instance, (2) can be written equivalently

$$\underset{x\in\mathbb{R}^n, u\in\mathbb{R}^m}{\text{minimize}} \;\; f(x) + \tau u^T e \quad \text{subject to} \;\; -u \leq c(x) \leq u,$$

when $h_\tau(x) = \tau\|x\|_1$, where e is the vector of ones. That is what Gould et al. [31] do before applying an interior-point method. The literature review of [31] provides a number of related references.

Estrin et al. [26, 27] implement a smooth exact penalty function originally proposed by Fletcher [28]. They challenge the notion that evaluating the penalty function and its gradient is costlier than in other widely-accepted constrained frameworks, and detail an efficient implementation.

Originally proposed by Hestenes [34] and Powell [46], the augmented-Lagrangian approach may be defined as a quadratic penalty applied to the Lagrangian, and was viewed by Powell as a shifted quadratic penalty function. Two of its attractive features are that it is smooth, and it acts as an exact penalty once optimal multipliers have been identified. Bertsekas [8] analyzes its convergence thoroughly and improves the original results. We refer interested readers to [8] for further information. Renowned implementations of augmented-Lagrangian methods include LANCELOT [19, 20], MINOS [40, 41], and ALGENCAN [1, 10]. In our numerical experiments, we use the recent implementation of dos Santos and Siqueira [25], named Percival.

The proximal-gradient (PG) method [29, 38] aims to solve problems with objectives of the form $f(x) + h(x)$ where f is $\mathcal{C}^1$ and h is proper and lsc. In that context, h often has regularizing power; it promotes solutions with desirable features, such as sparsity. Variants in the literature mainly differ in the assumptions on f and h. Numerous authors restrict their attention to convex f and/or h. Parikh and Boyd [44] review PG in the convex case with an insightful chapter on their interpretation. Lee et al. [37] study the convergence of a proximal Newton method in which steps are computed with PG where both f and h are convex. Bolte et al. [12] present a method for objectives of the form $g(x) + Q(x, y) + h(y)$ where both g and h are proper, lsc, and Q is $\mathcal{C}^1$. Tseng [51] proposes a general framework for accelerated PG. Proximal algorithms are related to augmented-Lagrangian methods [49, 50]. To the best of our knowledge, almost all proximal methods in the literature require evaluating a proximal operator—which involves the solution of a nonsmooth problem consisting of the sum of a squared norm and h—at each iteration. Evaluating that operator for h_τ in (2) would be impractical and prohibitively expensive. To compute a step, we use a variant of PG with adaptive step size developed by Aravkin et al. [3] that does not have convexity restrictions. Crucially, their method allows for a model of h to be used at each iteration, a feature that makes evaluating the proximal operator feasible. In a second stage, we use the recent proximal quasi-Newton method of Diouane et al. [23], which may be seen as a natural generalization of the method of [3].

Notation

The identity matrix of size n is I_n, or I if the context is clear. We use $J(x)$ to denote the Jacobian of c at x. For any matrix A, its Moore-Penrose pseudo-inverse is $A^\dagger$. The ℓ_p-norm of vectors is $\|\cdot\|_p$, and we use the shortcut $\|\cdot\|$ for the Euclidean norm. For matrices, $\|\cdot\|$ represents the operator norm. Similarly, $\mathbb{B}_p$ represents the unit ball, centered at the origin in the ℓ_p-norm and $\mathbb{B}$ is short for $\mathbb{B}_2$. For any $\Delta > 0$, $\Delta\mathbb{B}_p$ represents the ball of radius $\Delta > 0$ centered at the origin. If a set $\mathcal{S}$ is finite, $|\mathcal{S}|$ represents its cardinality. For any set $\mathcal{S} \subseteq \mathbb{R}^n$, $\chi(\cdot \mid \mathcal{S})$ is the *indicator* function, namely, for any $x \in \mathbb{R}^n$, $\chi(\cdot \mid \mathcal{S}) = 0$ if $x \in \mathcal{S}$ and $+\infty$ else. If g, h are two functions of $\epsilon > 0$, the notation $g = \mathcal{O}(h)$ means that there is $C > 0$ such that $\limsup_{\epsilon\to 0} g(\epsilon)/h(\epsilon) \leq C$.

2 Background

The Mangasarian-Fromowitz constraint qualification (MFCQ) holds at $x \in \mathbb{R}^n$ for (1) if $J(x)$ has full row rank.

The element $\bar{x} \in \mathbb{R}^n$ is a strict minimum of (1) if $c(\bar{x}) = 0$ and there is an open set $\mathcal{V}$ containing $\bar{x}$ such that $f(\bar{x}) < f(x)$ for all $x \in \mathcal{V}$ satisfying $c(x) = 0$.

Exactness of the ℓ_2-norm penalty relies on existence of Lagrange multipliers, and means the following.

Proposition 1 (33, Theorem 4.4). *If $\bar{x}$ is a strict minimum of (1) where the MFCQ holds, for every $\tau \geq \|\bar{y}\|$, $\bar{x}$ is a local minimum of (2), where $\bar{y}$ is the unique vector of Lagrange multipliers at $\bar{x}$.*

Consider $\phi : \mathbb{R}^n \to \overline{\mathbb{R}}$ and $\bar{x} \in \mathbb{R}^n$ where $\phi(\bar{x}) \in \mathbb{R}$. We say that $v \in \mathbb{R}^n$ is a regular subgradient of ϕ at $\bar{x}$, and we write $v \in \hat{\partial}\phi(\bar{x})$, if $\phi(x) \geq \phi(\bar{x}) + v^T(x - \bar{x}) + o(\|x - \bar{x}\|)$. The set of regular subgradients is called the Fréchet subdifferential. We say that v is a general subgradient of ϕ at $\bar{x}$, and we write $v \in \partial\phi(\bar{x})$, if there are $\{x_k\}$ and $\{v_k\}$ such that $\{x_k\} \to \bar{x}$, $\{\phi(x_k)\} \to \phi(\bar{x})$, $v_k \in \hat{\partial}\phi(x_k)$ for all k and $\{v_k\} \to v$. The set of general subgradients is called the limiting subdifferential [47, Definition 8.3].

If ϕ is $\mathcal{C}^1$ at x, $\partial\phi(x) = \{\nabla\phi(x)\}$ [47, §8.8]. In what follows, we rely on the following criticality property.

Proposition 2 (47, Theorem 10.1). *If $\phi : \mathbb{R}^n \to \mathbb{R}$ is proper and has a local minimum at $\bar{x}$, then $0 \in \hat{\partial}\phi(\bar{x}) \subseteq \partial\phi(\bar{x})$. If $\phi = f + h$, where f is $\mathcal{C}^1$ over a neighborhood of $\bar{x}$ and h is finite at $\bar{x}$, then $\partial\phi(\bar{x}) = \nabla f(\bar{x}) + \partial h(\bar{x})$.*

The element $\bar{x} \in \mathbb{R}^n$ is a KKT point of (1) if there is $\bar{y} \in \mathbb{R}^m$ such that $\nabla f(\bar{x}) = J(\bar{x})^T\bar{y}$ and $c(\bar{x}) = 0$.

We call $h : \mathbb{R}^n \to \overline{\mathbb{R}}$ proper if $h(x) > -\infty$ for all $x \in \mathbb{R}^n$ and $h(x) < +\infty$ for at least one x. h is lsc at $\bar{x}$ if $\liminf_{x \to \bar{x}} h(x) = h(\bar{x})$.

For proper lsc h and step length $\nu > 0$, the proximal mapping of νh at x is

$$\operatorname*{prox}_{\nu h}(x) := \operatorname*{argmin}_u \tfrac{1}{2}\nu^{-1}\|u - x\|^2 + h(u). \tag{4}$$

Proposition 3 (47, Theorem 1.25). *Let $h : \mathbb{R}^n \to \overline{\mathbb{R}}$ be proper lsc and bounded below. For all $\nu > 0$ and all $x \in \mathbb{R}^n$, $\operatorname{prox}_{\nu h}(x)$ is nonempty and compact.*

The proximal mapping of νh is nonempty under more general conditions than stated in Proposition 3, e.g., prox-boundedness [47, Definition 1.23], but boundedness is sufficient for our purposes.

We now intentionally use the notation φ and ψ instead of f and h for reasons that become clear in subsequent sections. Inspired by (2), we consider the generic nonsmooth problem

$$\operatorname*{minimize}_s \varphi(s) + \psi(s), \tag{5}$$

where φ is $\mathcal{C}^1$, and ψ is proper, lsc, and bounded below. A popular method to solve (5) is the *proximal-gradient* (PG) method [29, 38]. The PG iteration is

$$s_{i+1} \in \operatorname*{prox}_{\nu_i \psi}(s_i - \nu_i \nabla\varphi(s_i)), \quad i \geq 0, \tag{6}$$

where $\nu_i > 0$ is a step length. Descent is guaranteed when $\nabla\varphi$ is Lipschitz-continuous and ν_i is chosen appropriately. Recall that $\phi : \mathbb{R}^{n_1} \to \mathbb{R}^{n_2}$ is Lipschitz-continuous with Lipschitz constant $L > 0$ whenever $\|\phi(x) - \phi(y)\| \leq L\|x - y\|$ for all $x, y \in \mathbb{R}^{n_1}$.

Proposition 4 (12, Lemma 2). *Let φ be $\mathcal{C}^1$, $\nabla\varphi$ be Lipschitz-continuous with Lipschitz constant $L \geq 0$, and ψ be proper, lsc and bounded below. For any $0 < \nu < L^{-1}$ and $s_0 \in \mathbb{R}^n$ where ψ is finite, (6) is such that $(\varphi + \psi)(s_{i+1}) \leq (\varphi + \psi)(s_i) - \frac{1}{2}(\nu^{-1} - L)\|s_{i+1} - s_i\|^2$ for all $i \geq 0$. If $L = 0$, such as arises when φ is linear, the above is taken to mean $L^{-1} = +\infty$.*

3 Models

For given $x \in \mathbb{R}^n$, consider models

$$\varphi(s;x) \approx f(x+s), \tag{7a}$$
$$\psi_\tau(s;x) \approx h_\tau(c(x+s)) = \tau\|c(x+s)\|_2, \tag{7b}$$
$$m_\tau(s;x) \coloneqq \varphi(s;x) + \psi_\tau(s;x), \tag{7c}$$

where $\approx$ means that the left-hand side is an approximation of the right-hand side. The type of approximation expected of (7) is formalized in Model Assumption 3.1.

Model Assumption 3.1. For any $x \in \mathbb{R}^n$, $\varphi(\cdot;x)$ is $\mathcal{C}^1$, and satisfies $\varphi(0;x) = f(x)$ and $\nabla_s\varphi(0;x) = \nabla f(x)$. For any $x \in \mathbb{R}^n$, ψ_τ is proper lsc, and satisfies $\psi_\tau(0;x) = h_\tau(c(x))$ and $\partial_s\psi_\tau(0;x) = \partial h_\tau(c(x))$.

In Model Assumption 3.1, we use the notation ∂_s to emphasize that the subdifferential of $\psi_\tau(s;x)$ is computed with respect to its variable s while keeping x fixed.

We first focus our attention on the first order models

$$\varphi(s;x) \coloneqq f(x) + \nabla f(x)^T s, \tag{8a}$$
$$\psi_\tau(s;x) \coloneqq h_\tau(c(x) + J(x)s) = \tau\|c(x) + J(x)s\|_2. \tag{8b}$$

For future reference, define

$$g_x : \mathbb{R}^n \to \mathbb{R}^m, \quad g_x(s) \coloneqq c(x) + J(x)s \quad (x \in \mathbb{R}^n), \tag{9a}$$
$$\eta : \mathbb{R}^m \to \mathbb{R}, \quad \eta(w) \coloneqq \|w\|. \tag{9b}$$

Hence, from [35, Section D.3],

$$\partial\eta(x) = \begin{cases} \mathbb{B}_2 & \text{if } x = 0, \\ x/\eta(x) & \text{otherwise.} \end{cases} \tag{10}$$

The following lemma gives the subdifferential of ψ_τ, which will be useful to derive a stationarity measure for (2).

Lemma 1. *Let $\tau > 0$ and $x \in \mathbb{R}^n$. Then,*

$$\partial_s\psi_\tau(0;x) = \begin{cases} \tau J(x)^T\mathbb{B}_2 & \text{if } c(x) = 0, \\ \tau J(x)^T c(x)/\|c(x)\| & \text{otherwise,} \end{cases} \tag{11}$$

where ψ_τ is defined in (8b) and $\tau J(x)^T\mathbb{B}_2 \coloneqq \{\tau J(x)^T y \mid y \in \mathbb{B}_2\}$.

Proof. Let g_x and η be as in (9). Since $\operatorname{dom}\eta = \mathbb{R}^m$ and $c(x) \in \operatorname{Range}(g_x) \subseteq \mathbb{R}^m$, [48, Theorem 23.9] and (8b) yield $\partial_s\psi_\tau(s;x) = \tau J(x)^T\partial\eta(c(x) + J(x)s)$. Combining (10) with the latter and evaluating at $s = 0$ concludes. □

By the same reasoning as in the proof of Lemma 1, $\partial h_\tau(x) = \partial_s\psi_\tau(0;x)$, so that (8) satisfy Model Assumption 3.1.

4 Algorithm and convergence analysis

At iteration $k \in \mathbb{N}$, we solve (2) inexactly for fixed $\tau_k > 0$. The process of updating τ_k is the *outer* iterations. Solving (2) with $\tau = \tau_k$ is the k-th set of *inner* iterations with inner iterates $x_{k,j}$.

We solve (2) with method R2 [3, Algorithm 6.1]; a quadratic regularization method that uses models (8) and may be viewed as an adaptive variant of PG that does not require knowledge of the Lipschitz constant of the gradient. At inner iteration j, we compute

$$s_{k,j,\text{cp}} \in \operatorname*{argmin}_s \varphi(s; x_{k,j}) + \psi_{\tau_k}(s; x_{k,j}) + \tfrac{1}{2}\sigma_{k,j}\|s\|^2 \tag{12a}$$

$$= \operatorname*{prox}_{\sigma_{k,j}^{-1}\psi_{\tau_k}(\cdot;x_{k,j})}(-\sigma_{k,j}^{-1}\nabla\varphi(0; x_{k,j})), \tag{12b}$$

where $\sigma_{k,j} > 0$ is a regularization parameter, i.e., $s_{k,j,\text{cp}}$ is the first step of PG for the minimization of $\varphi(s; x_{k,j}) + \psi_{\tau_k}(s; x_{k,j})$ with step length $\sigma_{k,j}^{-1}$ initialized with $s_{k,j,0} = 0$ [3, §3.2]. The subscript "cp" stands for *Cauchy point*, of which $s_{k,j,\text{cp}}$ is an appropriate generalization to the nonsmooth context. We then set either $x_{k,j+1} \coloneqq x_{k,j} + s_{k,j,\text{cp}}$ or $x_{k,j+1} \coloneqq x_{k,j}$ depending on whether $s_{k,j,\text{cp}}$ results in sufficient decrease in the objective of (2) or not. We refer the reader to [2, 3] for the complete algorithm and further details. In particular, the quantity

$$\xi(x_{k,j}; \sigma_{k,j}, \tau_k) \coloneqq f(x_{k,j}) + h_{\tau_k}(c(x_{k,j})) - (\varphi + \psi_{\tau_k})(s_{k,j,\text{cp}}; x_{k,j}) \tag{13}$$

is key to defining the stationarity measure [2, 3]

$$\sigma_{k,j}^{1/2}\xi(x_{k,j}; \sigma_{k,j}, \tau_k)^{1/2}. \tag{14}$$

Proposition 5. *Let $\sigma > 0$, $\tau > 0$ and $\bar{x} \in \mathbb{R}^n$. If $\xi(\bar{x}; \sigma, \tau) = 0$, there is $\bar{y} \in \tau\partial\eta(c(x))$ such that $\nabla f(\bar{x}) = J(\bar{x})^T\bar{y}$. Moreover, if $c(\bar{x}) = 0$ $\bar{x}$ is a KKT point of (1).*

Proof. In view of [3, Lemma 6.1], $\xi(\bar{x}, \sigma, \tau) = 0$ is equivalent to

$$0 \in \arg\min_s \varphi(s; \bar{x}) + \psi_\tau(s; \bar{x}) + \tfrac{1}{2}\sigma\|s\|^2. \tag{15}$$

The first-order conditions of (15) and Proposition 2 yield

$$0 \in \nabla_s\varphi(0; \bar{x}) + \partial_s\psi_\tau(0; \bar{x}). \tag{16}$$

Thus, by Model Assumption 3.1, Lemma 1 and (16), there is $y \in \partial\eta(c(\bar{x}))$ such that $\nabla f(\bar{x}) + \tau J(\bar{x})^T y = 0$, and the result holds with $\bar{y} \coloneqq \tau y$. If additionally, $c(\bar{x}) = 0$, $\bar{x}$ is a KKT point of (1). □

The following assumption ensures convergence of the inner iterations.

Step Assumption 4.1 (3, Step Assumption 6.1)**.** For each outer iteration k of Algorithm 1, there is $\kappa_{m,k} > 0$ such that for each corresponding inner iteration j, $s_{k,j,\text{cp}}$ generated according to (12) satisfies

$$|f(x_{k,j} + s_{k,j,\text{cp}}) + h_{\tau_k}(c(x_{k,j} + s_{k,j,\text{cp}})) - (\varphi + \psi_{\tau_k})(s_{k,j,\text{cp}}; x_{k,j})| \le \kappa_{m,k}\|s_{k,j,\text{cp}}\|^2. \tag{17}$$

We make the following additional assumption on the growth of $\kappa_{m,k}$ along the outer iterations.

Step Assumption 4.2. There exist $\kappa_{m_1} > 0$ and $\kappa_{m_2} > 0$ such that for each outer iteration k of Algorithm 1, $\kappa_{m,k} \le \kappa_{m_1}\tau_k + \kappa_{m_2}$, where $\kappa_{m,k} > 0$ is defined in Step Assumption 4.1.

Step Assumption 4.2 is satisfied when f and c have Lipschitz gradient and Jacobian, respectively, and when h is Lipschitz-continuous, which is the case with h_τ. Indeed, if ∇f and J have Lipschitz constants L_g and L_J, respectively, then for any x, $s \in \mathbb{R}^n$ and $\tau > 0$, $|f(x+s) + h_\tau(c(x+s)) - \varphi(s; x) - \psi_\tau(s; x)| \le \frac{1}{2}L_g\|s\|^2 + \tau\frac{1}{2}L_J\|s\|^2$, where we used the triangle inequality, the descent lemma for smooth functions [12, Lemma 1] on f and c, and the fact that h_τ is Lipschitz-continuous with constant τ.

In view of Proposition 5, the aim of the outer iterations is to find a feasible point for (1). We focus on the feasibility problem

$$\operatorname*{minimize}_{x\in\mathbb{R}^n} 0 \quad \text{subject to } c(x) = 0. \tag{18}$$

In the same fashion as we did for ξ, in the case where $f = 0$, we define a feasibility measure for the outer iterates based on

$$\theta(x) \coloneqq h_1(c(x)) - \psi_1(s^*; x), \quad s^* \in \operatorname*{prox}_{1\psi_1}(0). \tag{19}$$

Note that $\theta(x)$ is $\xi(x; 1, 1)$ in the case where $f = 0$. Therefore, results for ξ equally apply to θ. In particular, Proposition 5 applies, so that $\theta(x) = 0$ implies $J(x)^T y = 0$ for some $y \in \partial\eta(c(x))$, where η is defined in (9b). By (10), $\|y\| \leq 1$. In addition, $\|y\| < 1 \implies c(x) = 0$. In particular, whenever the MFCQ holds, $J(x)^T y = 0$ implies $y = 0$, so that x is feasible. On the contrary, $y \neq 0$ indicates a dependency between the constraint gradients at x, whether x is feasible or not.

By analogy with (14), we define our feasiblity measure as $\theta(x)^{1/2}$. Algorithm 1 summarizes the outer iteration. For a tolerance $\epsilon > 0$, we wish to find the number of inner and outer iterations of Algorithm 1 until

$$\theta(x_k)^{1/2} \leq \epsilon. \tag{20}$$

Algorithm 1 **Exact penalty algorithm.**

1: Choose $x_0 \in \mathbb{R}^n$, $\beta_1 > 0$ and $\tau_0 > 0$.
2: Choose initial and final tolerances $\epsilon_0 \geq \epsilon > 0$, $0 < \beta_2 < 1$, and β_3, $\beta_4 > 0$.
3: **for** $k = 0, 1, \ldots$ **do**
4: Compute an approximate solution of (2) with $\tau \coloneqq \tau_k$ starting from x_k with initial step size $\sigma_{k,0} = \max(\beta_3\tau_k, \beta_4)$ and minimal regularization parameter $\sigma_{k,\min} = \beta_4$ Stop at the first iteration j such that

$$\sigma_{k,j}^{1/2}\xi(x_{k,j}; \sigma_{k,j}, \tau_k)^{1/2} \leq \epsilon_k.$$

5: Set $x_{k+1} \coloneqq x_{k,j}$.
6: If $\theta^{1/2}(x_{k+1}) > \epsilon_k$, choose $\tau_{k+1} \geq \tau_k + \beta_1$ and set $\epsilon_{k+1} \coloneqq \epsilon_k$. Otherwise, set $\tau_{k+1} \coloneqq \tau_k$ and $\epsilon_{k+1} \coloneqq \beta_2\epsilon_k$.
7: **end for**

Algorithm 1 is similar to [15, Algorithm 3.1], except that we do not use "steering"; a procedure that guarantees that the outer and inner measures decrease at the same rate. Instead, we use an increasing accuracy strategy of the inner measure on Line 6.

We use the following notation in the complexity analysis:

$$j_k(\epsilon_k) \coloneqq \min\{j \in \mathbb{N} \mid \sigma_{k,j}^{1/2}\xi(x_{k,j}; \sigma_{k,j}, \tau_k)^{1/2} \leq \epsilon_k\} \quad (k \in \mathbb{N}), \tag{21}$$

$$k(\epsilon) \coloneqq \min\{k \in \mathbb{N} \mid \theta(x_k)^{1/2} \leq \epsilon\}. \tag{22}$$

The total number of iterations we are looking for is then $\sum_{k=0}^{k(\epsilon)} j_k(\epsilon_k)$.

The next result states [3, Theorem 6.4] with the bound given by Aravkin et al. [2].

Theorem 1 (3, Theorem 6.4 and 2). *Let Model Assumption 3.1 and Step Assumption 4.1 hold and f be bounded below by f_{low}. Let $0 < \eta_1 \leq \eta_2 < 1$ and $0 < \gamma_3 \leq 1 < \gamma_2$ be the parameters of R2. Then,*

$$j_k(\epsilon_k) \leq \alpha_1\sigma_{\max,k}\frac{f(x_{k,0}) + h_{\tau_k}(c(x_{k,0})) - f_{\text{low}}}{\epsilon_k^2} + \alpha_2\left|\log\left(\frac{\sigma_{\max,k}}{\sigma_{k,0}}\right)\right|, \tag{23}$$

where $\sigma_{\max,k} = \max(\sigma_{k,0}, \alpha_3\kappa_{m,k})$, and

$$\alpha_1 \coloneqq (1 + |\log_{\gamma_1}(\gamma_3)|) > 0, \quad \alpha_2 \coloneqq |1/\log(\gamma_1)| > 0, \quad \alpha_3 \coloneqq 2\gamma_2/(1 - \eta_2) > 2. \tag{24}$$

Because our approach is similar to that of [15], the next lemmas give estimates between our measures, based on ξ and θ, and theirs. For all $x \in \mathbb{R}^n$ and $\tau > 0$, their measures [15, Equations (3.1) and (3.10)] are

$$\xi_{\text{TR}}(x, \tau) \coloneqq f(x) + h_\tau(c(x)) - \min_{\|s\| \leq 1}(\varphi + \psi_\tau)(s; x), \tag{25a}$$

$$\theta_{\text{TR}}(x) \coloneqq h_1(c(x)) - \min_{\|s\| \leq 1}\psi_1(s; x). \tag{25b}$$

Lemma 2. *For any $x \in \mathbb{R}^n$, $\tau > 0$ and $\sigma > 0$,*

$$\xi(x;\sigma,\tau) \geq \tfrac{1}{2}\min(1,\, \sigma^{-1}\xi_{\mathrm{TR}}(x,\tau))\,\xi_{\mathrm{TR}}(x,\tau). \tag{26}$$

In particular, for any $\epsilon > 0$, if $\sigma^{1/2}\xi(x;\sigma,\tau)^{1/2} \leq \epsilon$, then $\sigma \geq \sqrt{2}\epsilon$ implies

$$\sigma^{1/2}\xi(x;\sigma,\tau)^{1/2} \geq \tfrac{1}{\sqrt{2}}\xi_{\mathrm{TR}}(x,\tau). \tag{27}$$

Proof. Since f and c are $\mathcal{C}^1$, and h_τ is convex and globally Lipschitz continuous, [15, Lemma 2.5] gives $\xi(x,\sigma,\tau) - \frac{1}{2}\sigma\|s\|^2 \geq \frac{1}{2}\min(1,\sigma^{-1}\xi_{\mathrm{TR}}(x,\tau))\xi_{\mathrm{TR}}(x,\tau)$. Since $\xi(x,\sigma,\tau) \geq \xi(x,\sigma,\tau) - \frac{1}{2}\sigma\|s\|^2$, (26) holds.

For the second part, let $\epsilon > 0$ and assume that $\sigma^{1/2}\xi(x;\sigma,\tau)^{1/2} \leq \epsilon$ and $\sigma \geq \sqrt{2}\epsilon$. We now show that $\sigma^{-1}\xi_{\mathrm{TR}}(x,\tau) \leq 1$, as that will imply (27) by way of (26). Assume by contradiction that $\sigma^{-1}\xi_{\mathrm{TR}}(x,\tau) > 1$. Our assumption and (26) imply $\epsilon^2\sigma^{-1} \geq \xi(x;\sigma,\tau) \geq \frac{1}{2}\xi_{\mathrm{TR}}(x,\tau)$. We multiply both sides by σ^{-1}, and obtain $\epsilon^2\sigma^{-2} \geq \frac{1}{2}\sigma^{-1}\xi_{\mathrm{TR}}(x,\tau) > \frac{1}{2}$, which contradicts our assumption that $\sigma \geq \sqrt{2}\epsilon$. □

Lemma 3. *For any $x \in \mathbb{R}^n$, $\theta_{\mathrm{TR}}(x) \geq \min(\frac{1}{\sqrt{2}},\, \theta(x)^{1/2})\,\theta(x)^{1/2}$.*

Proof. For any $s^* \in \mathrm{prox}_{1\psi_1}(0) = \mathrm{argmin}_s\, \psi_1(s;x) + \frac{1}{2}\|s\|^2$, Proposition 4 implies

$$\theta(x) = \psi_1(0;x) - \psi_1(s^*;x) \geq \tfrac{1}{2}\|s^*\|^2. \tag{28}$$

Assume first that $\theta(x) \leq \frac{1}{2}$. Then, (28) implies $\|s^*\| \leq 1$. Hence,

$$\min_{\|s\|\leq 1} h_1(c(x) + J(x)s) = \min_{\|s\|\leq 1} \psi_1(s;x) \leq \psi_1(s^*;x),$$

which, together with (19) and (25b), implies that $\theta_{\mathrm{TR}}(x) \geq \theta(x)$.

Assume next that $\theta(x) > \frac{1}{2}$. Note that (28) may be written $\|\frac{1}{\sqrt{2}}\theta(x)^{-1/2}s^*\| \leq 1$. Since $s \to h_1(c(x) + J(x)s)$ is convex and $\frac{1}{\sqrt{2}}\theta(x)^{-1/2} < 1$,

$$\begin{aligned}\min_{\|s\|\leq 1} h_1(c(x) + J(x)s) &\leq h_1(c(x) + \tfrac{1}{\sqrt{2}}\theta(x)^{-1/2}J(x)s^*) \\ &\leq \left(1 - \tfrac{1}{\sqrt{2}}\theta(x)^{-1/2}\right) h_1(c(x)) + \tfrac{1}{\sqrt{2}}\theta(x)^{-1/2}h_1(c(x) + J(x)s^*).\end{aligned}$$

We then have from (19) and (25b), $\theta_{\mathrm{TR}}(x) \geq \frac{1}{\sqrt{2}}\theta(x)^{-1/2}\theta(x) = \frac{1}{\sqrt{2}}\theta(x)^{1/2}$. □

Theorem 2. *Assume that Step Assumption 4.2 holds and that f is bounded below by f_{low}.*

Assume that there is $\bar{\tau} \geq 0$, independent of ϵ, such that $\theta(x_{k+1})^{1/2} < \epsilon$ whenever $\tau_k \geq \bar{\tau}$. Then Algorithm 1 with stopping criterion (20) terminates either with an approximate KKT point of (1) or with an infeasible critical point of the feasibility measure (19) in at most

$$\left(\left\lceil \frac{\bar{\tau} - \tau_0}{\beta_1} \right\rceil + \lceil \log_{\beta_2}(\epsilon/\epsilon_0) \rceil\right)\left(\frac{\alpha_1(\kappa_h\bar{\tau} + \kappa_f)(\alpha_4\bar{\tau} + \alpha_5)}{\epsilon^2} + \alpha_2\left|\log\left(\frac{\alpha_4\tau_0 + \alpha_5}{\beta_3\tau_0}\right)\right|\right)$$

inner iterations, which is an overall complexity of $\mathcal{O}((\bar{\tau} + |\log(\epsilon/\epsilon_0)|)\,\bar{\tau}^2\epsilon^{-2})$, where α_1, α_2, and α_3 are defined in (24), κ_{m_1} and κ_{m_2} are defined in Step Assumption 4.2, and κ_f, κ_h and α_4 are defined as

$$\kappa_h \coloneqq \max_{k<k(\epsilon)} h_1(c(x_{k,0})) > 0, \qquad \kappa_f \coloneqq \max_{k<k(\epsilon)} f(x_{k,0}) - f_{\mathrm{low}} \geq 0, \tag{29a}$$

$$\alpha_4 \coloneqq \max(\beta_3, \alpha_3\kappa_{m_1}) > 0, \qquad \alpha_5 \coloneqq \max(\beta_4, \alpha_3\kappa_{m,2}) > 0. \tag{29b}$$

Alternatively, assume τ_k grows unbounded with k. Assume further that there is $\kappa_g \geq 0$ such that $\|\nabla f(x_k)\| \leq \kappa_g$ for all k. Then, Algorithm 1 with stopping criterion (20) terminates either with an approximate KKT point of (1) or with an infeasible critical point of the feasibility measure (19) in as many iterations as in the first case, but replacing $\bar{\tau}$ with $(\kappa_g + 1)\epsilon^{-2} + 1$, which is an overall complexity of $\mathcal{O}(\epsilon^{-8})$.

Proof. Consider the first part of the theorem. From Line 6 of Algorithm 1, there are at most $\lceil \log(\epsilon/\epsilon_0)/\log(\beta_2) \rceil = \lceil \log_{\beta_2}(\epsilon/\epsilon_0) \rceil$ outer iterations where τ_k is not increased until $\theta(x_k)^{1/2}$ attains or drops below ϵ. Similarly, there are at most $\lceil (\bar{\tau} - \tau_0)/\beta_1 \rceil$ outer iterations where τ_k is increased until it reaches or attains $\bar{\tau}$. Hence, τ_k attains or exceeds $\bar{\tau}$ in

$$k(\epsilon) \leq \left\lceil \frac{\bar{\tau} - \tau_0}{\beta_1} \right\rceil + \lceil \log_{\beta_2}(\epsilon/\epsilon_0) \rceil \tag{30}$$

outer iterations. We now find an estimate on $j_k(\epsilon_k)$. By Step Assumptions 4.1 and 4.2 and Theorem 1, (23) holds. First, from (29a)

$$f(x_{k,0}) + h_{\tau_k}(c(x_{k,0})) - f_{\text{low}} \leq \kappa_h \tau_k + \kappa_f. \tag{31}$$

The form of $\sigma_{k,0}$ in Line 4 of Algorithm 1, Step Assumption 4.2 and (29b) give

$$\sigma_{\max,k} = \max(\sigma_{k,0}, \alpha_3 \kappa_{m,k}) \leq \max(\beta_3 \tau_k, \beta_4, \alpha_3(\kappa_{m_1} \tau_k + \kappa_{m_2})) \leq \alpha_4 \tau_k + \alpha_5, \tag{32}$$

so that

$$\frac{\sigma_{\max,k}}{\sigma_{k,0}} \leq \frac{\alpha_4 \tau_k + \alpha_5}{\max(\beta_3 \tau_k, \beta_4)} \leq \frac{\alpha_4 \tau_k + \alpha_5}{\beta_3 \tau_k} \leq \frac{\alpha_4 \tau_0 + \alpha_5}{\beta_3 \tau_0}, \tag{33}$$

because $\tau \mapsto \frac{\alpha_4 \tau + \alpha_5}{\beta_3 \tau}$ is decreasing on $\mathbb{R}^+$, and $\tau_k \geq \tau_0$. Combining (31), (32) and (33) into (23) gives

$$\begin{aligned} j_k(\epsilon_k) &\leq \frac{\alpha_1(\kappa_h \tau_k + \kappa_f)(\alpha_4 \tau_k + \alpha_5)}{\epsilon_k^2} + \alpha_2 \left| \log\left(\frac{\alpha_4 \tau_0 + \alpha_5}{\beta_3 \tau_0} \right) \right| \\ &\leq \frac{\alpha_1(\kappa_h \bar{\tau} + \kappa_f)(\alpha_4 \bar{\tau} + \alpha_5)}{\epsilon^2} + \alpha_2 \left| \log\left(\frac{\alpha_4 \tau_0 + \alpha_5}{\beta_3 \tau_0} \right) \right|. \end{aligned}$$

Thus, we obtain the desired bound on $\sum_{k=0}^{k(\epsilon)-1} j_k(\epsilon_k)$.

We now turn to the second part of the theorem. First note that whenever τ_k is increased, Line 6 of Algorithm 1 implies

$$\sigma_{k,j}^{1/2} \xi(x_k; \sigma_{k,j}, \tau_k)^{1/2} \leq \epsilon_k < \theta^{1/2}(x_k) \quad \text{with} \quad j = j_k(\epsilon_k). \tag{34}$$

Using the notation of Lemmas 2 and 3, if we assume that $\sigma_{k,j} \geq \sqrt{2}\epsilon_k$, then (34) and Lemma 2 imply

$$\sigma_{k,j}^{1/2} \xi(x_k; \sigma_{k,j}, \tau_k)^{1/2} \geq \tfrac{1}{\sqrt{2}} \xi_{\text{TR}}(x_k, \tau_k). \tag{35}$$

From Algorithm 1, $\sigma_{k,j} \geq \beta_4 > 0$ for all k and j. Because $\epsilon_k \leq \beta_4/\sqrt{2}$ for all sufficiently large k, we may assume without loss of generality that (35) holds. Furthermore, for any $s_{k,\text{TR}} \in \operatorname{argmin}_{\|s\| \leq 1} \psi_1(s; x_k)$,

$$\begin{aligned} \xi_{\text{TR}}(x_k, \tau_k) &= f(x_k) + h_{\tau_k}(x_k) - \min_{\|s\| \leq 1} (\varphi + \psi_{\tau_k})(s; x_k) && (36) \\ &\geq f(x_k) + h_{\tau_k}(x_k) - (\varphi + \psi_{\tau_k})(s_{k,\text{TR}}; x_k) && (37) \\ &= \tau_k (h_1(x_k) - \min_{\|s\| \leq 1} \psi_1(s; x)) - \nabla f(x_k)^T s_{k,\text{TR}} && (38) \\ &= \tau_k \theta_{\text{TR}}(x_k) - \nabla f(x_k)^T s_{k,\text{TR}} && (39) \\ &\geq \tau_k \theta_{\text{TR}}(x_k) - \|\nabla f(x_k)\|, && (40) \end{aligned}$$

where we used the definition of the models (7) on the third line, the definition of θ_{TR} (25b) on the fourth and the Cauchy-Schwarz inequality on the last line combined with the fact that $\|s_{k,\text{TR}}\| \leq 1$. Substituting (40) into (35) then gives

$$\sigma_{k,j}^{1/2}\xi(x_k;\sigma_{k,j},\tau_k)^{1/2} \geq \tfrac{1}{\sqrt{2}}\left(\tau_k\theta_{\text{TR}}(x_k) - \|\nabla f(x_k)\|\right). \tag{41}$$

At this point, we consider two cases. Assume first that $\theta(x_k)^{1/2} \leq \frac{1}{\sqrt{2}}$. Lemma 3 and (41) combine with (34) to give

$$\tfrac{1}{\sqrt{2}}\left(\tau_k\theta(x_k) - \|\nabla f(x_k)\|\right) \leq \sigma_{k,j}^{1/2}\xi(x_k;\sigma_{k,j},\tau_k)^{1/2} \leq \theta(x_k)^{1/2} \leq \tfrac{1}{\sqrt{2}}(\theta(x_k)+1), \tag{42}$$

whenever τ_k is increased, where we used the fact that $\sqrt{t} \leq \frac{t+1}{2} \leq \frac{t+1}{\sqrt{2}}$ for any $t > 0$ in the last inequality. Thus, (42) implies

$$\|\nabla f(x_k)\| \geq (\tau_k - 1)\theta(x_k) - 1. \tag{43}$$

Because $\tau_k > 1$ for all large enough k, and $\|\nabla f(x_k)\| \leq \kappa_g$, (43) yields $\theta(x_k) \leq (\kappa_g + 1)/(\tau_k - 1)$ whenever τ_k is increased. Therefore, $\theta^{1/2}(x_k) \leq \epsilon_k$ whenever

$$\tau_k \geq \frac{\kappa_g + 1}{\epsilon_k^2} + 1. \tag{44}$$

Assume instead that $\theta^{1/2}(x_k) > \frac{1}{\sqrt{2}}$. Lemma 3 and (41) imply

$$\sigma_{k,j}^{1/2}\xi(x_k;\sigma_{k,j},\tau_k)^{1/2} \geq \tfrac{1}{\sqrt{2}}\left(\tfrac{1}{\sqrt{2}}\tau_k\theta(x_k)^{1/2} - \|\nabla f(x_k)\|\right). \tag{45}$$

Using the same approach that led from (42) to (44), with the difference that we do not require the final inequality in (42), we find in this case

$$\tau_k \geq \frac{\sqrt{2}\kappa_g}{\epsilon_k} + 2. \tag{46}$$

Combining (44) with (46) gives $\theta(x_k)^{1/2} \leq \epsilon_k$ whenever

$$\tau_k \geq \max\left(\frac{\kappa_g + 1}{\epsilon_k^2} + 1,\ \frac{\sqrt{2}\kappa_g}{\epsilon_k} + 2\right). \tag{47}$$

For small enough values of ϵ_k, i.e., for all large enough k, $\frac{\kappa_g+1}{\epsilon_k^2} + 1 > \frac{\sqrt{2}\kappa_g}{\epsilon_k} + 2$, and we may assume without loss of generality that $\theta(x_k)^{1/2} \leq \epsilon_k$ whenever (44) holds.

Overall, (44) implies $\theta^{1/2}(x_k) \leq \epsilon$ whenever

$$\tau_k \geq \frac{\kappa_g + 1}{\epsilon^2} + 1. \tag{48}$$

The value on the right-hand side of (48) is the counterpart of $\bar{\tau}$ from the first part of the proof, with the crucial difference that it depends on ϵ. Still, we may use the value of the right-hand side of (48) in place of $\bar{\tau}$ in the first part of the proof. □

The existence of κ_h and κ_f in (29a) is guaranteed by the fact that f, h and c are continuous functions.

Combining Proposition 1 with Theorem 2, we see that the assumptions of the first case of Theorem 2 hold under the MFCQ. Therefore, the bound in the first case of Theorem 2 is the one we expect to

observe the most in practical settings, while the bound in the second case should only happen if the problem is degenerate, i.e., does not admit Lagrange multipliers.

Comparing the bounds in Theorem 2 with those of Cartis et al. [15, Theorem 3.2], we see two differences: an additional $\log(\epsilon/\epsilon_0)$ term in our first case, and we found an $\mathcal{O}(\epsilon^{-8})$ instead of their $\mathcal{O}(\epsilon^{-5})$ complexity.

The additional $\log(\epsilon/\epsilon_0)$ comes from the fact that we used an increasing accuracy strategy in Algorithm 1, as we do not need to compute high-accuracy solutions when far from feasibility. However, if one chooses to enforce constant accuracy throughout, then $\epsilon_0 = \epsilon$, which implies $\log(\epsilon/\epsilon_0) = 0$.

Regarding the difference in the second case, notice first that our measure $\theta^{1/2}$ does not scale linearly with Cartis et al.'s θ_{TR} but θ does. We can easily find examples for which $\theta(x) = \theta_{\text{TR}}(x)$. Consider $c : \mathbb{R} \to \mathbb{R}$, $c(x) = x$. For $x \in \mathbb{R}$,

$$\operatorname*{argmin}_{|s|\leq 1} |x + s| = \operatorname*{argmin}_{s} \tfrac{1}{2}s^2 + |x + s| = \begin{cases} -1 & \text{if } x > 1, \\ -x & \text{if } |x| \leq 1, \\ 1 & \text{if } x < -1, \end{cases}$$

which implies from (19) and (25b) that $\theta(x) = \theta_{\text{TR}}(x) = \min(|x|, 1)$. Now, we can see that the issue really comes from Lemma 3; if instead of the inequality in Lemma 3 we had an inequality of the type $\theta_{\text{TR}} > \kappa\theta^{1/2}$ for some constant $\kappa > 0$, then using the same approach that led to (46) we would have found an upper bound on τ_k in $\mathcal{O}(\epsilon^{-1})$ instead of $\mathcal{O}(\epsilon^{-2})$ which would have then led to the $\mathcal{O}(\epsilon^{-5})$ complexity.

In addition, any $s^* \in \text{prox}_{1\psi_1}(0) = \arg\min_s \frac{1}{2}\|s\|^2 + \psi_1(s; x)$ is such that $0 \in s^* + \partial\psi_1(s^*; x)$, which is equivalent to

$$-s^* \in \partial\psi_1(s^*; x). \tag{49}$$

For any $g_{s^*,x} \in \partial\psi_1(s^*; x)$, convexity of $\psi_1(\cdot; x)$ and [35, Corollary D.2.1.3] imply

$$\psi_1(s^*; x) = \psi_1(0; x) + g_{s^*,x}^T s^* + o(\|s^*\|)$$

The choice $g_{s^*,x} = -s^*$, (49) and (19) combine to give $\theta(x) = \|s^*\|^2 + o(\|s^*\|)$. Hence, θ varies like the squared norm of a subgradient, which justifies taking the square root of θ to define an optimality measure. In addition, Lemma 3 implies $\theta_{\text{TR}} \geq \theta$ as $\theta \to 0$. The above leads us to believe that the $\mathcal{O}(\epsilon^{-8})$ is the correct complexity for exact penalty algorithms when the penalty parameter is unbounded.

Finally, our assumptions are weaker than those of Cartis et al. [15] as we only require boundedness of $\{\nabla f(x_k)\}$, whereas they require boundedness of $\{x_k\}$.

A quasi-Newton variant

Instead of using R2 as solver for (2) in Algorithm 1, we explore variant R2N recently proposed in [23] that uses or approximates second order information on f, and show that it preserves the convergence and complexity properties. As it turns out, using second-order models in the subproblem does not improve the complexity bounds of the outer loop, but we still expect improvements in practice as more information is given to enrich the model. At inner iteration j of outer iteration k, instead of (12a), we compute $s_{k,j}$ as an approximate solution of

$$\operatorname*{minimize}_{s} \; m_Q(s; x_{k,j}, B_{k,j}, \sigma_{k,j}), \tag{50}$$

where $m_Q(s; x, B, \sigma) \coloneqq \varphi_Q(s; x, B) + \psi_{\tau_k}(s; x) + \frac{1}{2}\sigma\|s\|^2$, and $\varphi_Q(s; x, B) \coloneqq \varphi(s; x) + \frac{1}{2}s^T Bs$, with φ defined in (8a), $B = B^T \in \mathbb{R}^{n\times n}$. In particular, B may be a quasi-Newton approximation of $\nabla^2 f(x)$, or $\nabla^2 f(x)$ if it exists. Diouane et al. [23] do not require B to be positive (semi-)definite. However, it

is not difficult to see that (50) is unbounded below along any negative curvature direction of $B + \sigma I$. Hence, we always adjust σ so that $\varphi_Q(s; x, B) + \frac{1}{2}\sigma\|s\|^2$ is convex. Clearly, Model Assumption 3.1 continues to hold for φ_Q.

Although the analysis of [23] does not depend on it, we make the following simplifying assumption.

Step Assumption 4.3. There is a constant κ_B such that for each iteration k and j of Algorithm 1 that uses R2N, $B_{k,j} = B_{k,j}^T$ and $\|B_{k,j}\| \leq \kappa_B$.

R2N use the same $\xi(x_{k,j}; \nu_{k,j}^{-1}, \tau_k)$ as in (13) as a stationarity measure and uses (14) as a stopping criterion as well, with the difference that $\nu_{k,j}^{-1} := \theta/(\sigma_{k,j} + \|B_{k,j}\|)$, where $\theta \in (0, 1)$ is a parameter. An analogue of Theorem 2 holds for Algorithm 1 with R2N as subsolver. If we denote $\tilde{\sigma}_{k,j} := {\nu_{k,j}}^{-1}$, Step Assumption 4.1 yields

$$\sigma_{k,j}\theta^{-1} \leq \tilde{\sigma}_{k,j} \leq \sigma_{k,j}\theta^{-1} + \kappa_B\theta^{-1}.$$

With these inequalities, we find $\tilde{\sigma}_{\max,k} = \theta^{-1}\sigma_{\max,k} + \theta^{-1}\kappa_B$ which acts as the $\sigma_{\max,k}$ defined previously. With that, we can establish Theorem 2 with $\xi(x_{k,j}; \nu_{k,j}^{-1}, \tau_k)$ as an inner stationarity measure.

The updated complexity bounds rest upon the following result.

Theorem 3 (23, Theorems 6.4 and 6.5)**.** *Let Model Assumption 3.1 and Step Assumptions 4.1 and 4.3 hold and f be bounded below by f_{low}. Let $0 < \eta_1 \leq \eta_2 < 1$ and $0 < \gamma_3 \leq 1 < \gamma_2$ be the parameters of R2N. Then,*

$$j_k(\epsilon_k) \leq (2\kappa_B(1 + \sigma_{\max,k}) + \sigma_{\max,k})\left(1 + |\log_{\gamma_1}(\gamma_3)|\right)\frac{\Delta(f+h)}{\eta_1\alpha\epsilon_k^2} + \log_{\gamma_1}\left(\frac{\sigma_{\max,k}}{\sigma_{k,0}}\right),$$

where $\Delta(f + h) := f(x_{k,0}) + h_{\tau_k}(c(x_{k,0})) - f_{\text{low}}$.

Theorem 4. *Assume that Step Assumption 4.2 and the assumptions of Theorem 3 hold. Assume that there is $\bar{\tau} \geq 0$, independent of ϵ, such that $\theta(x_{k+1})^{1/2} < \epsilon$ whenever $\tau_k \geq \bar{\tau}$. Then Algorithm 1 with stopping criterion (20) using R2N to solve (2) terminates either with an approximate KKT point of (1) or with an infeasible critical point of the feasibility measure (19) in at most*

$$\left(\left\lceil\frac{\bar{\tau} - \tau_0}{\beta_1}\right\rceil + \left\lceil\log_{\beta_2}(\epsilon/\epsilon_0)\right\rceil\right)\left(\frac{\alpha_6}{\alpha\epsilon^2} + \alpha_2\left|\log\left(\frac{\alpha_4\tau_0 + \alpha_5}{\beta_3\tau_0}\right)\right|\right)$$

inner iterations, which is an overall complexity of $\mathcal{O}((\bar{\tau} + |\log(\epsilon/\epsilon_0)|)\,\bar{\tau}^2\epsilon^{-2})$, where α_1, α_2, α_3 are as in Theorem 1, α_4, α_5, κ_h, κ_f are as in Theorem 2, κ_{m_1} and κ_{m_2} are as in Step Assumption 4.2, κ_B is as in Step Assumption 4.3, and

$$\alpha_6 := \alpha_1(\kappa_h\bar{\tau} + \kappa_f)((\alpha_4\bar{\tau} + \alpha_5)(1 + 2\kappa_B) + 2\kappa_B).$$

Alternatively, assume τ_k grows unbounded with k. Assume further that there is $\kappa_g \geq 0$ such that $\|\nabla f(x_k)\| \leq \kappa_g$ for all k. Then, Algorithm 1 with stopping criterion (20) using R2N to solve (2) terminates either with an approximate KKT point of (1) or with an infeasible critical point of the feasibility measure (19) in as many iterations as in the first case, but replacing $\bar{\tau}$ with $(\kappa_g + 1)\epsilon^{-2} + 1$, which is an overall complexity of $\mathcal{O}(\epsilon^{-8})$.

The proof of Theorem 4 is almost identical to that of Theorem 2. The only difference is that the bound (33) becomes $\tilde{\sigma}_{\max,k}/\tilde{\sigma}_{k,0} \leq (\sigma_{\max,k} + \kappa_B)/\sigma_{k,0}$.

5 Proximal operators

We now give a closed form solution for the computation of $s_{k,j,\text{cp}}$ in (12a). We start with general results and then specialize them to our case. In addition, when R2N is used, we provide a closed form solution for the computation of $s_{k,j}$ in (50). Thus, resorting to an iterative solver for nonsmooth problems to compute $s_{k,j}$ is not necessary.

Theorem 5. *Let $Q \in \mathbb{R}^{n\times n}$ be symmetric positive definite, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $d \in \mathbb{R}^n$, $\tau > 0$ and $p \in \mathbb{N}$. Let q be such that $1/p + 1/q = 1$. The unique solution of*

$$\underset{u\in\mathbb{R}^n}{\text{minimize}} \ \tfrac{1}{2}u^\top Qu - d^\top u + \tau\|Au+b\|_p, \tag{51}$$

is

$$u^* = Q^{-1}(d + A^T y^*) \tag{52}$$

where y^* is a solution of

$$\underset{y\in\mathbb{R}^m}{\text{maximize}} \ -\tfrac{1}{2}(d+A^Ty)^TQ^{-1}(d+A^Ty) - b^Ty \quad \text{subject to } \|y\|_q \le \tau. \tag{53}$$

If A has full row rank, y^* is unique.

Proof. As (51) is strictly convex, it has a unique solution u^*. It may be written

$$\underset{u\in\mathbb{R}^n \ z\in\mathbb{R}^m}{\text{minimize}} \ \tfrac{1}{2}u^TQu - d^Tu + \tau\|z\|_p \quad \text{subject to } \ z = Au + b. \tag{54}$$

The Lagrangian of (54) is $\mathcal{L}(u,z;y) = \left[\frac{1}{2}u^TQu - d^Tu - (A^Ty)^Tu\right] + \left[\tau\|z\|_p + y^Tz\right] - b^Ty$, and is separable with respect to u and z. The minimizer of the terms in u is $u^* = Q^{-1}(d + A^Ty)$, and corresponds to an optimal value of $-\frac{1}{2}(d+A^Ty)^TQ^{-1}(d+A^Ty)$. Regarding the terms in z,

$$\min_z \left[\tau\|z\|_p + y^Tz\right] = -\max_z \left[(-y)^Tz - \tau\|z\|_p\right] = -\zeta^*(-y),$$

where $\zeta : w \mapsto \tau\|w\|_p$ and ζ^* is its Fenchel conjugate. Using [13, Example 3.26], $\zeta^*(w) = \tau\chi(\frac{w}{\tau} \mid \mathbb{B}_q)$. We conclude that

$$\min_z \left[\tau\|z\|_p + y^Tz\right] = \begin{cases} 0 & \|y\|_q \le \tau, \\ -\infty & \|y\|_q > \tau. \end{cases}$$

The above gives us (53) as the dual of (54). Since $(u,z) = (0,b)$ is feasible for (54), the relaxed Slater condition, hence strong duality, holds for (54) and (53). If A has full row rank, (53) is strictly concave, and therefore has a unique solution. □

When $p = 2$, a closed-form solution emerges for y^* in Theorem 5.

Theorem 6. *Under the assumptions of Theorem 5, in the case where $p = 2$, a solution of (53) is given by*

$$y^* = \begin{cases} -(AQ^{-1}A^T)^\dagger(AQ^{-1}d+b) & \text{if } \|(AQ^{-1}A^T)^\dagger(AQ^{-1}d+b)\|_2 \le \tau \\ & \text{and } AQ^{-1}d+b \in \text{Range}(AQ^{-1}A^T) \\ -(AQ^{-1}A^T+\alpha^*I)^{-1}(AQ^{-1}d+b) & \text{otherwise,} \end{cases}$$

where α^* is the unique positive root of the strictly decreasing function

$$g(\alpha) = \|(AQ^{-1}A^T + \alpha I)^{-1}(AQ^{-1}d+b)\|_2^2 - \tau^2.$$

Proof. In the case where $p = 2$, $q = 2$ and we can rewrite (53) as

$$\underset{y\in\mathbb{R}^m}{\text{minimize}} \ \tfrac{1}{2}(d+A^Ty)^TQ^{-1}(d+A^Ty) + b^Ty \quad \text{subject to } \|y\|_2^2 \le \tau^2. \tag{55}$$

Whether A has full row rank or not, (55) is convex and satisfies Slater's condition. Thus, by the KKT conditions, y solves (55) if and only if there is α^* such that

$$(AQ^{-1}A^T + \alpha^*I)y + AQ^{-1}d + b = 0, \quad \text{and} \quad 0 \le \alpha^* \perp (\tau - \|y\|_2) \ge 0. \tag{56}$$

There are now two cases. In the first case, $\alpha^* = 0$. Then by (56), $(AQ^{-1}A^T)y = -(AQ^{-1}d + b)$. Thus, if $AQ^{-1}d + b \in \text{Range}(AQ^{-1}A^T)$, by primal feasibility, the pseudo-inverse is the solution whenever $\|(AQ^{-1}A^\top)^\dagger(AQ^{-1}d + b)\|_2 \leq \tau$. In the second case, $\alpha^* > 0$ and (56) yields $\|y\|_2^2 = \tau^2$ and $y = -(AQ^{-1}A^T + \alpha^* I)^{-1}(AQ^{-1}d+b)$, because $AQ^{-1}A^\top + \alpha^* I$ is nonsingular, and where $g(\alpha^*) = 0$. □

When A has full row rank, the solution given by Theorem 6 is unique. When A is rank deficient in Theorem 6 and $\alpha^* = 0$, the pseudo-inverse yields the minimum Euclidean norm solution of (56). If the latter does not have norm at most $\nu\tau$, no other solution does, and we must be in the case $\alpha^* > 0$. The next result specializes Theorem 6 to the evaluations of the proximal operator for $\psi_\tau(\cdot; x)$.

Corollary 1. *Let $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, and $\tau > 0$. Define $\eta_\tau : u \mapsto \tau\|Au + b\|_2$. For $\nu > 0$,*

$$\operatorname*{prox}_{\nu\eta_\tau}(w) \ni \begin{cases} w - A^T(AA^T)^\dagger(Aw + b) & \text{if } \|(AA^T)^\dagger(Aw + b)\|_2 \leq \tau \\ & \text{and } Aw + b \in \text{Range}(AA^T) \\ w - A^T(AA^T + \alpha^* I)^{-1}(Aw + b) & \text{otherwise,} \end{cases} \tag{57}$$

where α^* is the unique positive root of the strictly decreasing function

$$g(\alpha) = \|(AA^T + \alpha I)^{-1}(Aw + b)\|_2^2 - \nu^2\tau^2.$$

Proof. By (4), $\operatorname{prox}_{\nu\eta_\tau}(w) = \operatorname{argmin}_{u\in\mathbb{R}^n} \frac{1}{2}u^T I u - w^T u + \nu\tau\|Au + b\|_2$. Replacing Q with I_n, d with w and τ with $\nu\tau$ in Theorems 5 and 6 concludes. □

In the context of R2 and R2N, we will apply Corollary 1 with $\tau = \tau_k$, $\eta_\tau := \psi_{\tau_k}(\cdot; x_k)$, $A = J(x_k)$, $b = c(x_k)$, and $\nu = \sigma_k^{-1}$.

The same methodology allows us to derive a closed-form solution of (50).

Corollary 2. *Let $A \in \mathbb{R}^{m\times n}$ have full row rank, let $b \in \mathbb{R}^m$, $x \in \mathbb{R}^n$ and $\tau > 0$. Let B be an $n \times n$ symmetric matrix such that $Q := \nu B + I$ is positive definite, and $\eta_\tau : u \to \tau\|Au + b\|_2$. For $\nu > 0$, define*

$$\operatorname*{prox}_{\nu\eta_\tau}(w, B) \coloneqq \operatorname*{argmin}_{u\in\mathbb{R}^n} \ \tfrac{1}{2}\nu^{-1}\|u - w\|^2 + \psi_\tau(u; x) + \tfrac{1}{2}u^T Bu. \tag{58}$$

Let $v \coloneqq AQ^{-1}w + b$. Then,

$$\operatorname*{prox}_{\nu\eta_\tau}(w, B) \ni \begin{cases} Q^{-1}\left(w - A^T(AQ^{-1}A^T)^\dagger v\right) & \text{if } \|(AQ^{-1}A^T)^\dagger v\|_2 \leq \tau \\ & \text{and } v \in \text{Range}(AQ^{-1}A^T) \\ Q^{-1}\left(w - A^T(AQ^{-1}A^T + \alpha^* I)^{-1} v\right) & \text{otherwise,} \end{cases}$$

where α^* is the unique positive root of the strictly decreasing function

$$g(\alpha) = \|(AQ^{-1}A^T + \alpha I)^{-1}(AQ^{-1}w + b)\|_2^2 - \nu^2\tau^2.$$

Proof. By (58), $\operatorname{prox}_{\nu\psi_\tau(\cdot;x)}(w, B) = \operatorname{argmin}_{u\in\mathbb{R}^n} \frac{1}{2}u^T Q u - w^T u + \nu\tau\|Au + b\|_2$. Replacing Q with $\nu B + I$, d with w and τ with $\nu\tau$ in Theorems 5 and 6 concludes. □

From Theorem 5, we see that solving (12a) requires solving the trust-region problem (53) in $\mathbb{R}^m$, or, equivalently, (57). An efficient procedure to solve such problems in the ℓ_2 norm case was proposed by Moré and Sorensen [39]. We follow their approach. In the light of Corollary 1 and using the same notation, it should be clear that any efficient procedure to solve (53) should try to find the root of $g(\alpha)$

in as few iterations as possible because each evaluation of g requires solving a linear system. The idea in [39] is to solve the equivalent *secular* equation

$$g(\alpha) = 0 \quad \iff \quad \frac{1}{\|(AA^T + \alpha I)^{-1}(Ax + b)\|_2} = \frac{1}{\nu\tau}, \tag{59}$$

which has preferable numerical properties. They use Newton's method to solve (59). We summarize the procedure as Algorithm 2 with the help of the following result.

Lemma 4 (21, Lemma 7.3.1). *Define* $\phi(\alpha) \coloneqq \dfrac{1}{\|s(\alpha)\|_2} - \dfrac{1}{\nu\tau}$, *where*

$$s(\alpha) \coloneqq \begin{cases} -(AA^T)^{\dagger}(Ax + b) & \text{if } \alpha = 0, \\ -(AA^T + \alpha I)^{-1}(Ax + b) & \text{if } \alpha > 0. \end{cases}$$

Then, ϕ *is strictly increasing and concave, and for all* $\alpha > 0$,

$$\phi'(\alpha) = -\frac{s(\alpha)^T \nabla_\alpha s(\alpha)}{\|s(\alpha)\|_2^3}, \quad \nabla s(\alpha) = -(AA^T + \alpha I)^{-1} s(\alpha).$$

If A *has full row rank then the above holds for* $\alpha = 0$ *as well.*

Algorithm 2 Proximal operator evaluation.

1: Compute $s(0)$ as given by Lemma 4.
2: If $\|s(0)\| \leq \nu\tau$ and $AA^T s(0) - (Ax + b) = 0$, return $x + A^T s(0)$
3: Choose $\theta \in (0, 1)$. Set $k = 0$.
4: If A has full row rank, set $\alpha_0 \coloneqq 0$. Otherwise, set $\alpha_0 \coloneqq \sqrt{\epsilon_M}$ and compute $s(\alpha_0)$.
5: **while** $\|s(\alpha_k)\| \neq \nu\tau$ **do**
6: Compute α_{k+1} as α^+ given by (60).
7: If $\alpha^+ \leq 0$, reset $\alpha_{k+1} \coloneqq \theta\alpha_k$.
8: Compute $s(\alpha_{k+1})$.
9: Let $k \coloneqq k + 1$.
10: **end while**
11: **return** $x + A^T s(\alpha_k)$.

Instead of computing the Cholesky factorization of $AA^T + \alpha I$, we compute the QR factorization of $A_\alpha^T \coloneqq \begin{pmatrix} A & \sqrt{\alpha} I_m \end{pmatrix}^T$. The R factor coincides with the Cholesky factor. The update of Newton's method can be found using Lemma 4:

$$\alpha^+ = \alpha_k - \frac{\phi(\alpha_k)}{\phi'(\alpha_k)} = \alpha_k - \frac{(\nu\tau - \|s(\alpha_k)\|)\,\|s(\alpha_k)\|^2}{\nu\tau s(\alpha_k)^T \nabla s(\alpha_k)}. \tag{60}$$

Algorithm 2 obtains $q_k = s(\alpha_k)$ by solving $R^T R q_k = -(Ax+b)$. Instead of performing another solve with R and R^T to obtain $\nabla s(\alpha_k)$, we observe that $s(\alpha_k)^T \nabla s(\alpha_k) = -q_k^T R^{-1} R^{-T} q_k = -\|R^{-T} q_k\|^2 = -\|p_k\|^2$, so only one extra solve with R^T is necessary.

When evaluating $s(0)$ on Line 1, we check whether A is rank deficient or not by looking for zeros on the diagonal of R. If so, we compute the least-norm solution of the underdetermined linear system. [21, Lemma 7.3.2] indicates that convergence is assured once $\alpha \in \mathcal{L} \coloneqq (\max(0, -\lambda_1), \alpha^*]$, where α^* solves (59) and where λ_1 is the smallest eigenvalue of AA^T, for all Newton iterates remain in $\mathcal{L}$ and converge to α^*. If $\alpha = 0$ does not yield the solution, we initialize Newton's method from an α_0 chosen so that $\phi'(\alpha_0)$ can be computed. If ever $\alpha_k > \alpha^*$, convergence is not assured. However, [21, Lemma 7.3.3] indicates that, due to the concavity of ϕ, the next iterate will either be in $\mathcal{L}$ or non-positive. Line 7 safeguards against $\alpha < 0$ by restarting the Newton iterations from a positive value smaller than α_k in hopes of eventually landing in $\mathcal{L}$.

We now turn to the solution of (50). For any $\alpha \geq 0$, the element $u \in \operatorname{prox}_{\nu\eta_\tau}(w, B)$ given by Corollary 2 can equivalently be obtained from

$$\begin{bmatrix} -Q & A^T \\ A & \alpha I \end{bmatrix} \begin{bmatrix} u \\ y \end{bmatrix} = \begin{bmatrix} -w \\ -b \end{bmatrix}. \tag{61}$$

Because B will typically be a limited-memory quasi-Newton approximation in our implementation, materializing, let alone factorizing, (61) would be inefficient. Thus, we use MINRES-QLP [17]. Even though only an approximate minimizer of (50) is required, we leave the MINRES-QLP to their default values, and compute an accurate solution. If $\alpha = 0$ and A is rank deficient, MINRES-QLP is guaranteed to return the mininimum-norm solution of (61), but not necessarily $y = -(AQ^{-1}A^T)^\dagger v$ required by Corollary 2. Computing such y with MINRES-QLP can be done by solving a larger symmetric saddle-point system that represents the optimality conditions of the least-norm problem that defines y. Our preliminary implementation simply searches for $\alpha \geq \sqrt{\varepsilon_M}$, where ε_M is the machine epsilon. We will study and report on alternative computations of y in follow-up research.

In order to solve $g(\alpha) = 0$ with g as in Corollary 2, we solve the *secular* equation

$$\phi_Q(\alpha) = 0, \qquad \phi_Q(\alpha) \coloneqq \frac{1}{\|(AQ^{-1}A^T + \alpha I)^{-1}(AQ^{-1}x + b)\|_2} - \frac{1}{\nu\tau}.$$

If $y(\alpha)$ solves (61),

$$\frac{\phi_Q(\alpha)}{\phi_Q'(\alpha)} = \frac{\|y(\alpha)\|^2}{y(\alpha)^T(AQ^{-1}A^T + \alpha I)^{-1}y(\alpha)}\left(1 - \frac{\|y(\alpha)\|}{\nu\tau}\right). \tag{62}$$

Computing the denominator above requires $w \coloneqq (AQ^{-1}A^T + \alpha I)^{-1}y(\alpha)$ found with MINRES-QLP via

$$\begin{bmatrix} -Q & A^T \\ A & \alpha I \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ y(\alpha) \end{bmatrix}. \tag{63}$$

6 Implementation and experiments

We implemented all algorithms above in Julia 1.10 [9]. Algorithm 1, R2 and R2N are part of the RegularizedOptimization.jl package [5]; Algorithm 2 is in ShiftedProximalOperators.jl [6].

We compare Algorithm 1 against the augmented-Lagrangian solver Percival [4, 25] and the IPOPT [43, 52] solver on equality-constrained problems from the CUTEst collection [30, 42] with fewer constraints than variables, and with at most 300 variables. This results in a set of 50 problems. Each problem is given a limit of 5 minutes of CPU time. In what follows, we do not provide CPU time comparisons because our implementation is still preliminary, and memory usage is not optimal.

Percival and IPOPT are second-order methods while Algorithm 1 with R2 is a first-order method. For the comparison to be fair, in the problems that we expose to Percival and IPOPT, we replace the Hessian of the objective with a multiple of the identity, $\nabla^2 f(x) \approx \sigma_f(x) I$, and replace the constraint Hessians with the zero matrix, $\nabla^2 c_i(x) \approx 0$ for $1 \leq i \leq m$. Because R2 updates $\sigma_{k,j}$ at each iteration, we update $\sigma_f(x)$ as in the spectral gradient method [11], i.e., $\sigma_f(x_{k,j}) \coloneqq s_{k,j}^T y_{k,j} / s_{k,j}^T s_{k,j}$, where $s_{k,j} \coloneqq x_{k,j} - x_{k,j-1}$ and $y_{k,j} \coloneqq \nabla f(x_{k,j}) - \nabla f(x_{k,j-1})$.

The optimality measures in (13) and (19), though appropriate in our context, are inconvenient to compare with solvers for smooth optimization, as the latter use criteria based on the KKT conditions for (1). Therefore, our implementation of Algorithm 1 stops as soon as

$$\|\nabla f(x_k) + J(x_k)^T y_k^{\text{LS}}\| \leq \epsilon \quad \text{and} \quad \|c(x_k)\| \leq \epsilon,$$

where y_k^{LS} is computed as the minimum least-squares solution of $\nabla f(x_k) + J(x_k)^T y = 0$.

Percival and IPOPT use all default parameters. In these preliminary small-scale tests, all solvers use an absolute stopping test with $\epsilon = 10^{-3}$. In Algorithm 1, we used $\beta_1 = \tau_0 = 500$, $\epsilon_0 = 10^{-2}$, $\beta_2 = 0.1$, $\beta_3 = 10^{-2}$, $\beta_4 = \varepsilon_M \approx 2.2 \times 10^{-16}$. In Algorithm 2, we used QRMumps.jl [14] for the QR factorizations and set $\theta = 0.8$. The stopping criterion for Newton's method is $|\|y_k\| - \nu\tau| < \varepsilon_M^{0.75}$. We approximate the least-norm solution of $AA^Tx = b$ with the solution of $(A + \sqrt{\epsilon_M}I)x = b$. Finally, we impose the lower bound $\alpha_k \geq \epsilon_M^{0.75}$ in order to instabilities when A is rank deficient and α^* appears to be zero. More sophisticated strategies will be required in later versions of our implementation. We set the maximal number of iterations for both Algorithm 1 and Algorithm 2 to 10^4. All tests are performed in double precision.

We also compare Algorithm 1 combined with R2N to IPOPT and Percival. All parameters are the same as before, and we experiment with both LBFGS and LSR1 Hessian approximations with memory 5. The MINRES-QLP tolerance is set to ε_M and its maximum number of iterations to 10^4 to evaluate (58).

We report our results in the form of Dolan and Moré [24] performance profiles in Figure 1 in terms of number of objective, gradient and constraint evaluations. When using R2N, we only report results with LBFGS; results with LSR1 are nearly identical. In the left column of Figure 1, we see that our implementation is competitive with first-order Percival and IPOPT on all three metrics. Because Percival never evaluates the Jacobian but only Jacobian-vector products, we do not provide profiles in terms of Jacobian evaluations. The profiles suggest that the performance of Algorithm 1

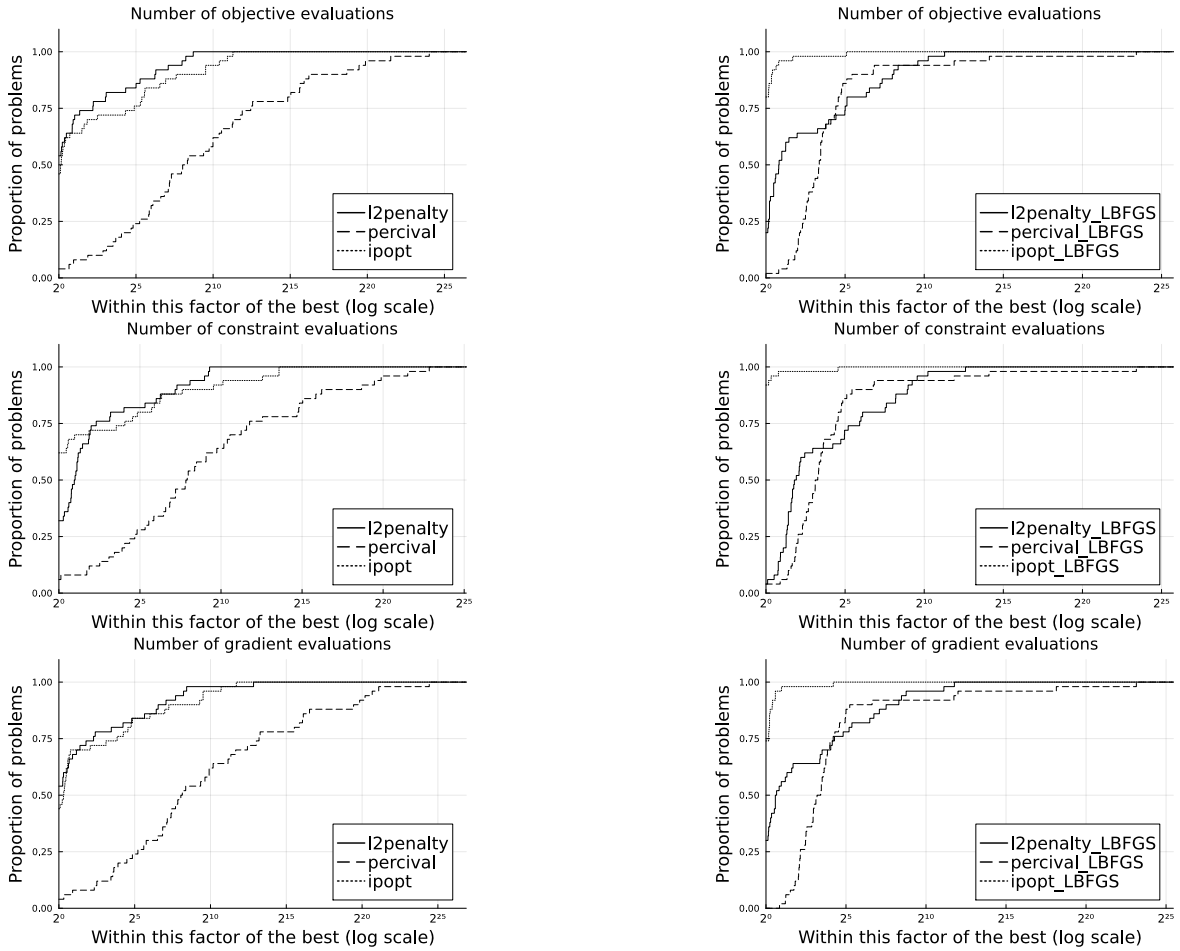

Figure 1: Left: Algorithm 1 with the R2 subsolver against Percival and IPOPT with a spectral quasi-Newton approximation. Right: Algorithm 1 with the R2N subsolver against Percival and IPOPT with a LFGS quasi-Newton approximation.

is very promising for problems where, for some reason, constraint Hessians cannot be evaluated or approximated. In the right column of Figure 1, we see that Algorithm 1 is very on par with Percival, while IPOPT is ahead of both. Whether we use R2 or R2N, the robustness of Algorithm 1 is on par with the other two solvers. We think that these results are strongly in favor of considering exact penalty approaches as an alternative to augmented-Lagrangian approaches.

7 Discussion and future work

Algorithm 1 is, to the best of our knowledge, the first practical implementation of the exact ℓ_2-penalty method. On small-scale problems, our preliminary implementation is on par with an augmented Lagrangian method. It is also competitive with IPOPT when spectral Hessian approximations are used in the latter in terms of efficiency and robustness. It remains competitive in terms of robustness when using limited-memory quasi-Newton Hessian approximations. Improvements in terms of efficiency are the subject of active research.

Among possible improvements, the solution of the secular equations requires attention in the presence of rank-deficient Jacobians when the solution is at, or near, the origin. In Algorithm 2, one idea is to use the method of Golub and Riley [22] to compute the least-norm solutions of rank-deficient overdetermined problems. The latter requires a single QR factorization and may be viewed as a form of iterative refinement. A possible improvement in terms of CPU time is to solve the secular equation (59) in reduced-precision arithmetic. Similarly, MINRES-QLP does not provide the required least-norm solution in Corollary 2. When it appears that the solution is $\alpha^* = 0$ and the Jacobian is rank deficient, a different saddle-point system can be solved that provides the required solution.

The framework of Section 4 is general and can be extended to other norms. In particular, Theorem 5 suggests that we could implement the exact ℓ_1-penalty method originally proposed by Pietrzykowski [45]. Indeed, evaluating the proximal operator amounts to solving an convex bound-constrained subproblem for which there are efficient polynomial time algorithms [21, Chapter 7.8].

A trust-region variant of Algorithm 1 based on [3, Algorithm 3.1] instead of R2 would enjoy the same asymptotic complexity bounds, although evaluation of the proximal operators would be significantly harder.

In future research, we plan to extend our analysis to constrained problems where the objective is the sum of a smooth function f and a nonsmooth regularizer h:

$$\min_x f(x) + h(x) \text{ subject to } c(x) = 0.$$

References

[1] R. Andreani, E. G. Birgin, J. M. Martínez, and M. L. Schuverdt. On augmented Lagrangian methods with general lower-level constraints. SIAM J. Optim., 18(4):1286–1309, 2008.

[2] A. Aravkin, R. Baraldi, G. Leconte, and D. Orban. Corrigendum: A proximal quasi-Newton trust-region method for nonsmooth regularized optimization. Cahier du GERAD G–2021–12–SM, GERAD, Montréal, QC, Canada, 2024.

[3] A. Y. Aravkin, R. Baraldi, and D. Orban. A proximal quasi-Newton trust-region method for nonsmooth regularized optimization. SIAM J. Optim., 32:900–929, 2022.

[4] S. Arreckx, A. B. Lambe, J. R. R. A. Martins, and D. Orban. A matrix-free augmented Lagrangian algorithm with application to large-scale structural design optimization. Optim. Eng., 17(2):359–384, June 2016.

[5] R. Baraldi and D. Orban. RegularizedOptimization.jl: Algorithms for regularized optimization. `https://github.com/JuliaSmoothOptimizers/RegularizedOptimization.jl`, 2022.

[6] R. Baraldi and D. Orban. ShiftedProximalOperators.jl: Proximal operators for regularized optimization. `https://github.com/JuliaSmoothOptimizers/ShiftedProximalOperators.jl`, 2022.

[7] M. S. Bazaraa and J. J. Goode. Sufficient conditions for a globally exact penalty function without convexity. In M. Guignard, editor, Optimality and Stability in Mathematical Programming, volume 19 of Mathematical Programming Studies, pages 1–15. Springer Verlag, Heidelberg, Berlin, New York, 1982.

[8] D. P. Bertsekas. Constrained Optimization and Lagrange Multiplier Methods. Academic Press, New York, 1982.

[9] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. SIAM Rev., 59(1):65–98, 2017.

[10] E. Birgin and J. M. Martinez. Practical Augmented Lagrangian Methods for Constrained Optimization. Number 10 in Fundamentals of Algorithms. SIAM, Philadelphia, USA, 2014.

[11] E. G. Birgin, J. M. Martínez, and M. Raydan. Spectral projected gradient methods: Review and perspectives. 60(3):1–21, 2014.

[12] J. Bolte, S. Sabach, and M. Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Math. Program., 146:459–494, 2013.

[13] S. Boyd and L. Vandenberghe. Convex Optimization. Cambridge University Press, 2004.

[14] A. Buttari and A. Montoison. QRMumps.jl: A Julia interface to qr_mumps. https://github.com/JuliaSmoothOptimizers/QRMumps.jl, April 2021.

[15] C. Cartis, N. I. M. Gould, and Ph. L. Toint. On the evaluation complexity of composite function minimization with applications to nonconvex nonlinear programming. SIAM J. Optim., 21:1721–1739, 2011.

[16] C. Charalambous. A lower bound for the controlling parameters of the exact penalty functions. Math. Program., 15:278–290, 1978.

[17] S.-C. T. Choi, C. C. Paige, and M. A. Saunders. MINRES-QLP: A Krylov subspace method for indefinite or singular symmetric systems. SIAM J. Sci. Comput., 33(4):1810–1836, 2011.

[18] T. F. Coleman and A. R. Conn. Second-order conditions for an exact penalty function. Math. Program., 19:178–185, 1980.

[19] A. R. Conn, N. I. M. Gould, and Ph. Toint. A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds. SIAM J. Numer. Anal., 28(2):545–572, 1991.

[20] A. R. Conn, N. I. M. Gould, and Ph. Toint. LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A). Springer Verlag, Heidelberg, Berlin, New York, 1992.

[21] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Trust-Region Methods. Number 1 in MPS-SIAM Series on Optimization. SIAM, Philadelphia, USA, 2000.

[22] A. Dax and L. Eldén. Approximating minimum norm solutions of rank-deficient least squares problems. Numer. Linear Algebra Appl., 5(2):79–99, 1998.

[23] Y. Diouane, M. L. Habiboullah, and D. Orban. A proximal modified quasi-Newton method for nonmooth regularized optimization. Cahier du GERAD G–2024–64, GERAD, Montréal, QC, Canada, 2024.

[24] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. Math. Program., 91:201–213, 2002.

[25] E. A. dos Santos and A. S. Siqueira. Percival.jl: an augmented Lagrangian method. `https://github.com/JuliaSmoothOptimizers/Percival.jl`, 2020.

[26] R. Estrin, M. P. Friedlander, D. Orban, and M. A. Saunders. Implementing a smooth exact penalty function for equality-constrained nonlinear optimization. SIAM J. Sci. Comput., 42:A1809–A1835, 2020.

[27] R. Estrin, M. P. Friedlander, D. Orban, and M. A. Saunders. Implementing a smooth exact penalty function for general constrained nonlinear optimization. SIAM J. Sci. Comput., 42:A1836–A1859, 2020.

[28] R. Fletcher. A class of methods for nonlinear programming with termination and convergence properties. Integer and Nonlinear Programming, pages 157–175, 1970.

[29] M. Fukushima and H. Mine. A generalized proximal point algorithm for certain non-convex minimization problems. 12:989–1000, 1981.

[30] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEst: a constrained and unconstrained testing environment with safe threads. Comput. Optim. Appl., 60:545–557, 2015.

[31] N. I. M. Gould, D. Orban, and Ph. L. Toint. An interior-point ℓ_1-penalty method for nonlinear optimization. In M. Al-Baali, L. Grandinetti, and A. Purnama, editors, Recent Developments in Numerical Analysis and Optimization, volume 134 of Proceedings in Mathematics and Statistics, pages 117–150, Switzerland, 2015. Springer. special issue of NAOIII, Muscat, Oman, 2014.

[32] G. N. Grapiglia, J. Yuan, and Y.-X. Yuan. On the convergence and worst-case complexity of trust-region and regularization methods for unconstrained optimization. Math. Program., 152:491–520, 2015.

[33] S. P. Han and O. L. Mangasarian. Exact penalty functions in nonlinear programming. Math. Program., 17:251–269, 1979.

[34] M. R. Hestenes. Multiplier and gradient methods. J. Optim. Theory and Applics., 4(5):303–320, 1969.

[35] J.-B. Hiriart-Urruty and C. Lemaréchal. Fundamentals of Convex Analysis. Springer Verlag, Heidelberg, Berlin, New York, 2001.

[36] L. R. Huang and K. F. Ng. Second-order necessary and sufficient conditions in nonsmooth optimization. Math. Program., 66:379–402, 1994.

[37] J. D. Lee, Y. Sun, and M. A. Saunders. Proximal Newton-type methods for minimizing composite functions. SIAM J. Optim., 24(3):1420–1443, 2014.

[38] P. L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. SIAM J. Numer. Anal., 16:964–979, 1979.

[39] J. J. Moré and D. C. Sorensen. Computing a trust region step. SIAM J. Sci. and Statist. Comput., 4: 553–572, 1983.

[40] B. A. Murtagh and M. A. Saunders. Large-scale linearly constrained optimization. Math. Program., 14: 41–72, 1978.

[41] B. A. Murtagh and M. A. Saunders. A projected lagrangian algorithm and its implementation for sparse nonlinear constraints. In A. G. Buckley and J.-L. Goffin, editors, Algorithms for Constrained Minimization of Smooth Nonlinear Functions, volume 16 of Mathematical Programming Studies, pages 84–117. Springer Verlag, Heidelberg, Berlin, New York, 1982.

[42] D. Orban, A. Soares Siqueira, and contributors. CUTEst.jl: Julia's cutest interface. https://github.com/JuliaSmoothOptimizers/CUTEst.jl, 2023.

[43] D. Orban, A. Soares Siqueira, and contributors. NLPModelsIpopt.jl: A thin ipopt wrapper for nlpmodels. https://github.com/JuliaSmoothOptimizers/NLPModelsIpopt.jl, 2023.

[44] N. Parikh and S. Boyd. Proximal algorithms. Foundations and Trends in Optimization, 1(3):127–239, 2014.

[45] T. Pietrzykowski. An exact potential method for constrained maxima. SIAM J. Numer. Anal., 6(2): 299–304, 1969.

[46] M. J. D. Powell. A method for nonlinear constraints in minimization problems. In R. Fletcher, editor, Optimization, pages 283–298. Academic Press, London, 1969. http://www.ams.org/mathscinet-getitem?mr=42:7284.

[47] R. Rockafellar and R. J.-B. Wets. Variational Analysis. Springer Verlag, 1998.

[48] R. T. Rockafellar. Convex Analysis. Princeton University Press, 1970.

[49] R. T. Rockafellar. A dual approach to solving nonlinear programming problems by unconstrained optimization. Math. Program., 3:354–373, 1973.

[50] R. T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. Math. Oper. Res., 1(2):97–116, 1973.

[51] P. Tseng. A modified forward-backward splitting method for maximal monotone mappings. SIAM J. Optim., 38(2):431–446, 2000.

[52] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program., 106:25–57, 2006.