



**Titre:** Nonsmooth exact penalty methods for equality-constrained optimization: complexity and implementation  
Title:

**Auteurs:** Youssef Diouane, Maxence Gollier, & Dominique Orban  
Authors:

**Date:** 2026

**Type:** Article de revue / Article

**Référence:** Diouane, Y., Gollier, M., & Orban, D. (2026). Nonsmooth exact penalty methods for equality-constrained optimization: complexity and implementation. SIAM Journal on Optimization, 36(2), 626-650. <https://doi.org/10.1137/24m1705974>  
Citation:

 **Document en libre accès dans PolyPublie**  
Open Access document in PolyPublie

**URL de PolyPublie:** <https://publications.polymtl.ca/76024/>  
PolyPublie URL:

**Version:** Version finale avant publication / Accepted version  
Révisé par les pairs / Refereed

**Conditions d'utilisation:** Creative Commons Attribution 4.0 International (CC BY)  
Terms of Use:

 **Document publié chez l'éditeur officiel**  
Document issued by the official publisher

**Titre de la revue:** SIAM Journal on Optimization (vol. 36, no. 2)  
Journal Title:

**Maison d'édition:** Society for Industrial and Applied Mathematics  
Publisher:

**URL officiel:** <https://doi.org/10.1137/24m1705974>  
Official URL:

**Mention légale:**  
Legal notice:

# NONSMOOTH EXACT PENALTY METHODS FOR EQUALITY-CONSTRAINED OPTIMIZATION: COMPLEXITY AND IMPLEMENTATION

YOUSSEF DIOUANE\*, MAXENCE GOLLIER†, AND DOMINIQUE ORBAN‡

**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. Recent advances in proximal methods have led to increasingly efficient solvers for nonsmooth optimization. We study a general exact penalty algorithm and use it to show that the exact  $\ell_2$ -penalty method for equality-constrained optimization can, in fact, be implemented efficiently by solving the penalized problem using 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. While the Lipschitz continuity of the objective gradient is not required for convergence in view of recent works, it is used in our analysis to derive the complexity bound. 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 superior to the augmented Lagrangian in terms of robustness and efficiency, and is competitive with the SQP method in terms of robustness, though the latter retains a slight edge in terms of number of problem function evaluations.

**Key words.** Equality-constrained optimization, penalty methods, proximal methods.

**AMS subject classifications.** 90C06, 90C30, 90C53.

**1. Introduction.** We consider the problem

$$(1.1) \quad \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) \quad \text{subject to} \quad c(x) = 0,$$

where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  and  $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$  are  $\mathcal{C}^1$  and both may be nonconvex. We solve (1.1) by solving a sequence of unconstrained, nonsmooth penalized problems

$$(1.2) \quad \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + \tau h(c(x)),$$

where  $h = \|\cdot\|$  denotes a norm on  $\mathbb{R}^n$ , and  $\tau > 0$  is the penalty parameter. This approach was first proposed by Pietrzykowski [46], who used the  $\ell_1$ -norm. Penalty approaches such as (1.2) are attractive because, under standard assumptions, for all sufficiently large and finite  $\tau$ , solutions of (1.1) are solutions of (1.2). However, the nonsmoothness of  $h$  caused them to fall out of favor in practice, and other methods, such as the augmented-Lagrangian method [34, 47] were preferred.

---

\*GERAD and Department of Mathematics and Industrial Engineering, Polytechnique Montréal. E-mail: [youssef.diouane@polymtl.ca](mailto:youssef.diouane@polymtl.ca). Research supported by an NSERC Discovery grant.

†GERAD and Department of Mathematics and Industrial Engineering, Polytechnique Montréal. E-mail: [maxence-2.gollier@polymtl.ca](mailto:maxence-2.gollier@polymtl.ca).

‡GERAD and Department of Mathematics and Industrial Engineering, Polytechnique Montréal. E-mail: [dominique.orban@gerad.ca](mailto:dominique.orban@gerad.ca). Research supported by an NSERC Discovery grant.

Independently, attention was given recently to nonsmooth regularized problems

$$(1.3) \quad \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) + h(x),$$

where  $f$  is as before, and  $h$  is proper and lower semi-continuous (lsc), and may be nonconvex. In particular, proximal methods [29, 39] 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 Mangasarian-Fromowitz constraint qualification and Lipschitz continuity of the objective gradient and constraint Jacobian. While the Lipschitz continuity of the objective gradient is not required for convergence in view of [23], it is used in our analysis to derive the complexity bound. 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 [16, §3]. Although both bounds are correct, we justify the difference by arguing that our feasibility measure is properly scaled. We use our general analysis to implement the  $\ell_2$ -norm penalty method, though  $\ell_1$  and  $\ell_\infty$  variants may also be implemented efficiently, building upon the same framework. 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 superior to the augmented Lagrangian in terms of robustness and efficiency, and is competitive with the SQP method in terms of robustness, though the latter retains a slight edge 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 3](#) we introduce the models of  $f$  and  $c$  in (1.1) that are used to compute steps in our algorithm. In [Section 4](#), we present our algorithm and derive complexity bounds. In [Section 5](#) we derive procedures to evaluate 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 \[46\]](#) first reported the advantage of using a nonsmooth penalty function. His analysis only covered the  $\ell_1$ -penalty but subsequently, [Charalambous \[17\]](#), [Han and Mangasarian \[33\]](#), [Coleman and Conn \[19\]](#), [Bazaraa and Goode \[7\]](#), and [Huang and Ng \[35\]](#) strengthened his results.

[Cartis et al. \[16\]](#) 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 [16] 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. Indeed, we show that solving the subproblem of the exact penalty method can be done by solving a trust-region subproblem for which algorithms are well-known for certain choices of norms. By contrast, the algorithms in [16, 32] require to solve a nonsmooth subproblem

with a trust-region constraint, which is difficult in practice, and those authors provide no indication as to how that problem may be solved. Additionally, our convergence analysis holds under weaker assumptions.

Several authors reformulate (1.2) as a smooth inequality-constrained problem. For instance, (1.2) can be written equivalently

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

when  $h(x) = \|x\|_1$ , 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. A similar transformation exists for  $h(x) = \|x\|_\infty$ , or any other polyhedral norm.

[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 \[47\]](#), 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 [20, 21], MINOS [41, 42], and ALGENCAN [1, 10]. In our numerical experiments, we use the recent implementation of [dos Santos and Siqueira \[25\]](#), named Percival. In the context of exact penalty methods, [Kuhlmann and Büskens \[36\]](#) proposed an augmented-Lagrangian-like approach in which the quadratic penalty is replaced by an exact  $\ell_2$ -penalty. Our approach differs in that we add the penalty term directly to the objective of (1.1). Moreover, while both approaches lead—via the exact penalty term in (1.2)—to optimality conditions involving a trust-region-like term, they ignore this term when computing steps, using it only to update penalty parameters, which is in contrast with our approach. Additionally, their approach consists in applying Newton’s method to the first-order optimality condition of the penalized problem, which differs from the proximal-gradient method that we use. Their algorithm has been implemented within the solver WORHP [14].

The proximal-gradient (PG) method [29, 39] aims to solve problems of the form (1.3) 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 \[45\]](#) review PG in the convex case with an insightful chapter on their interpretation. [Lee et al. \[38\]](#) 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 \[53\]](#) proposes a general framework for accelerated PG. Proximal algorithms are related to augmented-Lagrangian methods [51, 52]. 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  $h \circ c$  in (1.2) 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  $x$ . We say a function  $f$  is  $\mathcal{C}^1$  if it is continuously differentiable. For any matrix  $A$ , its Moore-Penrose pseudo-inverse is  $A^\dagger$ . For  $p \geq 1$ , the vector  $\ell_p$ -norm is  $\|\cdot\|_p$ .  $\|\cdot\|$  is any norm on  $\mathbb{R}^n$  and  $\|\cdot\|_*$  is its dual norm, defined as  $\|y\|_* := \sup_{\|x\| \leq 1} x^T y$ . For matrices,  $\|\cdot\|_p$  represents the operator norm with respect to the vector  $\ell_p$ -norm. Similarly,  $\mathbb{B}_p$  represents the unit ball, centered at the origin in the  $\ell_p$ -norm. 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  $\mathcal{S} \subseteq \mathbb{R}^n$ ,  $\chi(\cdot | \mathcal{S})$  is the *indicator* function, namely, for any  $x \in \mathbb{R}^n$ ,  $\chi(x | \mathcal{S}) = 0$  if  $x \in \mathcal{S}$  and  $+\infty$  otherwise. If  $g, h$  are two positive functions of  $\epsilon > 0$ , the notation  $g = \mathcal{O}(h)$  means that there is  $C > 0$  such that  $\limsup_{\epsilon \rightarrow 0} g(\epsilon)/h(\epsilon) \leq C$ . The notation  $g = o(h)$  means that  $\lim_{\epsilon \rightarrow 0} g(\epsilon)/h(\epsilon) = 0$ .

**2. Background.** The Mangasarian-Fromowitz constraint qualification (MFCQ) holds at  $x \in \mathbb{R}^n$  for (1.1) if  $J(x)$  has full row rank.

The element  $\bar{x} \in \mathbb{R}^n$  is a strict minimum of (1.1) 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 norm penalty relies on existence of Lagrange multipliers, and means the following.

PROPOSITION 2.1 (33, Theorem 4.4). *If  $\bar{x}$  is a strict minimum of (1.1) where the MFCQ holds, for every  $\tau \geq \|\bar{y}\|_*$ ,  $\bar{x}$  is a local minimum of (1.2), where  $\bar{y}$  is the unique vector of Lagrange multipliers at  $\bar{x}$ .*

Consider  $\phi : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$  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  $\phi$  at  $\bar{x}$ , and we write  $v \in \hat{\partial}\phi(\bar{x})$ , if  $\phi(x) \geq \phi(\bar{x}) + v^T(x - \bar{x}) + o(\|x - \bar{x}\|_2)$ . The set of regular subgradients is called the Fréchet subdifferential. We say that  $v$  is a general subgradient of  $\phi$  at  $\bar{x}$ , and we write  $v \in \partial\phi(\bar{x})$ , if there are  $\{x_k\}$  and  $\{v_k\}$  such that  $\{x_k\} \rightarrow \bar{x}$ ,  $\{\phi(x_k)\} \rightarrow \phi(\bar{x})$ ,  $v_k \in \hat{\partial}\phi(x_k)$  for all  $k$  and  $\{v_k\} \rightarrow v$ . The set of general subgradients is called the limiting subdifferential [49, Definition 8.3].

We call  $h : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$  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  $\liminf_{x \rightarrow \bar{x}} h(x) = h(\bar{x})$ .

If  $\phi$  is  $\mathcal{C}^1$  over a neighborhood of  $x$ ,  $\partial\phi(x) = \{\nabla\phi(x)\}$  [49, §8.8]. In what follows, we rely on the following criticality property.

PROPOSITION 2.2 (49, Theorem 10.1). *If  $\phi : \mathbb{R}^n \rightarrow \bar{\mathbb{R}}$  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 (1.1) if there is  $\bar{y} \in \mathbb{R}^m$  such that  $\nabla f(\bar{x}) = J(\bar{x})^T \bar{y}$  and  $c(\bar{x}) = 0$ . Moreover, if there exists  $\epsilon_1, \epsilon_2 > 0$  such that there is a norm for which  $\|\nabla f(\bar{x}) - J(\bar{x})^T \bar{y}\| \leq \epsilon_1$  and  $\|c(\bar{x})\| \leq \epsilon_2$ , then we call  $\bar{x}$  an  $(\epsilon_1, \epsilon_2)$ -approximate KKT point of (1.1). Since all norms are equivalent on  $\mathbb{R}^n$ , this definition of an approximate KKT point is independent of the specific norm chosen.

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

$$(2.1) \quad \underset{\nu h}{\text{prox}}(x) := \underset{u}{\text{argmin}} \frac{1}{2}\nu^{-1}\|u - x\|_2^2 + h(u).$$

PROPOSITION 2.3 (49, Theorem 1.25). *Let  $h : \mathbb{R}^n \rightarrow \overline{\mathbb{R}}$  be proper lsc and bounded below. For all  $\nu > 0$  and all  $x \in \mathbb{R}^n$ ,  $\text{prox}_{\nu h}(x)$  is nonempty and compact.*

The proximal mapping of  $\nu h$  is nonempty under more general conditions than stated in Proposition 2.3, e.g., prox-boundedness [49, Definition 1.23], but boundedness is sufficient for our purposes.

We now intentionally use the notation  $\varphi$  and  $\psi$  instead of  $f$  and  $h$  for reasons that become clear in subsequent sections. Inspired by (1.2), we consider the generic nonsmooth problem

$$(2.2) \quad \underset{s}{\text{minimize}} \quad \varphi(s) + \psi(s),$$

where  $\varphi$  is  $\mathcal{C}^1$ , and  $\psi$  is proper, lsc, and bounded below. A popular method to solve (2.2) is the *proximal-gradient* (PG) method [29, 39]. The PG iteration is

$$(2.3) \quad s_{i+1} \in \underset{\nu_i \psi}{\text{prox}}(s_i - \nu_i \nabla \varphi(s_i)), \quad i \geq 0,$$

where  $\nu_i > 0$  is a step length. Descent is guaranteed when  $\nabla \varphi$  is Lipschitz-continuous and  $\nu_i$  is chosen appropriately. Recall that  $\phi : \mathbb{R}^{n_1} \rightarrow \mathbb{R}^{n_2}$  is Lipschitz-continuous with Lipschitz constant  $L > 0$  whenever  $\|\phi(x) - \phi(y)\|_2 \leq L\|x - y\|_2$  for all  $x, y \in \mathbb{R}^{n_1}$ .

PROPOSITION 2.4 (12, Lemma 2). *Let  $\varphi$  be  $\mathcal{C}^1$ ,  $\nabla \varphi$  be Lipschitz-continuous with Lipschitz constant  $L \geq 0$ , and  $\psi$  be proper, lsc and bounded below. For any  $0 < \nu < L^{-1}$  and  $s_0 \in \mathbb{R}^n$  where  $\psi$  is finite, (2.3) is such that  $(\varphi + \psi)(s_{i+1}) \leq (\varphi + \psi)(s_i) - \frac{1}{2}(\nu^{-1} - L)\|s_{i+1} - s_i\|_2^2$  for all  $i \geq 0$ . If  $L = 0$ , such as arises when  $\varphi$  is linear, the above is taken to mean  $L^{-1} = +\infty$ .*

**3. Models.** For given  $x \in \mathbb{R}^n$ , consider models

$$(3.1a) \quad \varphi(s; x) \approx f(x + s),$$

$$(3.1b) \quad \psi(s; x) \approx h(c(x + s)) = \|c(x + s)\|,$$

$$(3.1c) \quad m(s; x) := \varphi(s; x) + \psi(s; x),$$

where  $\approx$  means that the left-hand side is an approximation of the right-hand side. The type of approximation expected of (3.1) 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  $x \in \mathbb{R}^n$ ,  $\psi$  is proper lsc, and satisfies  $\psi(0; x) = h(c(x))$  and  $\partial_s \psi(0; x) = \partial(h \circ c)(x)$ .*

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

We first focus our attention on the first order models

$$(3.2a) \quad \varphi(s; x) := f(x) + \nabla f(x)^T s,$$

$$(3.2b) \quad \psi(s; x) := h(c(x) + J(x)s) = \|c(x) + J(x)s\|.$$

For future reference, define

$$(3.3) \quad g_x : \mathbb{R}^n \rightarrow \mathbb{R}^m, \quad g_x(s) := c(x) + J(x)s \quad (x \in \mathbb{R}^n),$$

From [?, Section D.3],

$$(3.4) \quad \partial h(z) = \{\|y\|_* \leq 1 \mid y^T z = \|z\|\}.$$

It follows from (3.4) that for any  $y \in \partial h(z)$ ,

$$(3.5) \quad \|y\|_* < 1 \implies z = 0,$$

because, from Hölder's inequality and (3.4), we have  $\|z\| = y^T z \leq \|y\|_* \|z\|$ . The following lemma gives the subdifferential of  $\psi$ , which will be useful to derive a stationarity measure for (1.2).

LEMMA 3.1. *Let  $x \in \mathbb{R}^n$ . Then, for all  $s \in \mathbb{R}^n$ ,*

$$(3.6) \quad \partial_s \psi(s; x) = J(x)^T \partial h(c(x)) + J(x)s$$

where  $\psi$  is defined in (3.2b) and  $\partial h$  is given in (3.4).

*Proof.* Let  $g_x$  be as in (3.3). Since  $\text{dom } h = \mathbb{R}^m$  and  $c(x) \in \text{Range}(g_x) \subseteq \mathbb{R}^m$ , the result follows from [50, Theorem 23.9] and (3.2b).  $\square$

Since  $h$  is bounded below, lsc and convex, [49, Theorem 10.6] yields  $\partial(h \circ c)(x) = J(x)^T \partial h(c(x))$ , which, by Lemma 3.1, is equal to  $\partial_s \psi(0; x)$ . Thus, (3.2) satisfy Model Assumption 3.1.

**4. Algorithm and convergence analysis.** At iteration  $k \in \mathbb{N}$ , we solve (1.2) exactly for fixed  $\tau_k > 0$ . The process of updating  $\tau_k$  is the *outer* iteration. Solving (1.2) with  $\tau = \tau_k$  is the  $k$ -th set of *inner* iterations with inner iterates  $x_{k,j}$ .

We solve (1.2) with method R2 [3, Algorithm 6.1]; a quadratic regularization method that uses models (3.2) and may be viewed as an adaptive variant of PG that does not require knowledge of the Lipschitz constant of the gradient of  $f$  (or even its existence for convergence). At inner iteration  $j$ , we compute

$$(4.1a) \quad \{s_{k,j,\text{cp}}\} = \underset{s}{\text{argmin}} \varphi(s; x_{k,j}) + \tau_k \psi(s; x_{k,j}) + \frac{1}{2} \sigma_{k,j} \|s\|_2^2$$

$$(4.1b) \quad = \underset{\sigma_{k,j}^{-1} \tau_k \psi(\cdot; x_{k,j})}{\text{prox}} \left( -\sigma_{k,j}^{-1} \nabla \varphi(0; x_{k,j}) \right),$$

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  $\varphi(s; x_{k,j}) + \tau_k \psi(s; x_{k,j})$  with step length  $\sigma_{k,j}^{-1}$  initialized with  $s_{k,j,0} = 0$  [3, §3.2]. Convexity of  $\varphi(\cdot; x_{k,j})$  and  $\tau_k \psi(\cdot; x_{k,j})$  ensures that the proximal operator in (4.1) is single-valued. 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  $x_{k,j+1} := x_{k,j} + s_{k,j,\text{cp}}$  or  $x_{k,j+1} := x_{k,j}$  depending on whether  $s_{k,j,\text{cp}}$  results in sufficient decrease in the objective of (1.2) or not. The complete algorithm is given in Appendix A; we refer the reader to [2, 3] for details. In particular, the quantity

$$(4.2) \quad \xi(x_{k,j}; \sigma_{k,j}, \tau_k) := f(x_{k,j}) + \tau_k h(c(x_{k,j})) - (\varphi + \tau_k \psi)(s_{k,j,\text{cp}}; x_{k,j})$$

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

$$(4.3) \quad \sigma_{k,j}^{1/2} \xi(x_{k,j}; \sigma_{k,j}, \tau_k)^{1/2}.$$

PROPOSITION 4.1. *Let  $\sigma > 0$ ,  $\tau > 0$ ,  $\kappa_c > 0$ ,  $\epsilon > 0$  and  $\bar{x} \in \mathbb{R}^n$ . Assume that  $\sigma^{1/2} \xi(\bar{x}; \sigma, \tau)^{1/2} \leq \epsilon$ , and let  $\{s_{\text{cp}}\} = \underset{\sigma^{-1} \tau \psi(\cdot; \bar{x})}{\text{prox}} (-\sigma^{-1} \nabla \varphi(0; \bar{x}))$ . Then, there is  $\bar{y} \in \tau \partial h(c(x)) + J(x)s_{\text{cp}}$  such that  $\|\nabla f(\bar{x}) + J(\bar{x})^T \bar{y}\|_2 \leq \sqrt{2}\epsilon$ . Moreover, if  $\|c(\bar{x})\|_2 \leq \kappa_c \epsilon$ ,  $\bar{x}$  is an  $(\sqrt{2}\epsilon, \kappa_c \epsilon)$ -approximate KKT point of (1.1).*

*Proof.* By definition of  $s_{\text{cp}}$  and (4.1),

$$(4.4) \quad \{s_{\text{cp}}\} = \underset{s}{\operatorname{argmin}} \varphi(s; \bar{x}) + \tau\psi(s; \bar{x}) + \frac{1}{2}\sigma\|s\|_2^2.$$

On the other hand, Proposition 2.4 and (4.2) imply

$$(4.5) \quad \epsilon > \sigma^{1/2}\xi(\bar{x}, \sigma, \tau)^{1/2} \geq \frac{\sigma\|s_{\text{cp}}\|_2}{\sqrt{2}}.$$

Convexity, the first-order conditions of (4.4) and Proposition 2.2 yield

$$(4.6) \quad 0 \in \nabla\varphi(s_{\text{cp}}; \bar{x}) + \tau\partial_s\psi(s_{\text{cp}}; \bar{x}) + \sigma s_{\text{cp}}.$$

Thus, by (3.2a), Lemma 3.1 and injecting (4.6) into (4.5), there is  $y \in \partial h(c(x) + J(x)s_{\text{cp}})$  such that  $\|\nabla f(\bar{x}) + \tau J(\bar{x})^T y\|_2 \leq \sqrt{2}\epsilon$ , and the result holds with  $\bar{y} := \tau y$ . If additionally,  $\|c(\bar{x})\|_2 \leq \kappa_c \epsilon$ ,  $\bar{x}$  is a  $(\sqrt{2}\epsilon, \kappa_c \epsilon)$ -approximate KKT point of (1.1).  $\square$

In view of Proposition 4.1, the aim of the outer iterations is to find a feasible point for (1.1). We focus on the feasibility problem

$$(4.7) \quad \underset{x \in \mathbb{R}^n}{\operatorname{minimize}} \quad 0 \quad \text{subject to} \quad c(x) = 0.$$

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

$$(4.8) \quad \theta(x) := h(c(x)) - \psi(s^*; x), \quad \{s^*\} = \underset{s}{\operatorname{prox}}(0) = \underset{s}{\operatorname{argmin}} \psi(s; x) + \frac{1}{2}\|s\|_2^2.$$

Note that  $\theta(x)$  is  $\xi(x; 1, 1)$  in the case where  $f = 0$ . In particular, Proposition 4.1 applies, so that for any  $\epsilon > 0$ ,  $\theta(x)^{1/2} \leq \epsilon$  implies  $\|J(x)^T y\|_2 \leq \sqrt{2}\epsilon$  for some  $y \in \partial h(c(x) + J(x)s^*)$ . From (3.4)  $\|y\|_* \leq 1$  and, in particular, if  $\|y\|_* < 1$ , (3.5) implies  $c(x) + J(x)s^* = 0$ . Hence, (4.8) implies  $h(c(x)) \leq \epsilon^2$ .

By analogy with (4.3), we define our feasibility measure as  $\theta(x)^{1/2}$ . Algorithm 4.1 summarizes the outer iteration.

---

**Algorithm 4.1** 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  $\beta_3, \beta_4 > 0$ .
- 3: **for**  $k = 0, 1, \dots$  **do**
- 4:     Compute an approximate solution of (1.2) with  $\tau := \tau_k$  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  $x_{k+1} := x_{k,j}$ .
  - 6:     If  $\theta^{1/2}(x_{k+1}) > \epsilon_k$ , choose  $\tau_{k+1} \geq \tau_k + \beta_1$  and set  $\epsilon_{k+1} := \epsilon_k$ . Otherwise, set  $\tau_{k+1} := \tau_k$  and  $\epsilon_{k+1} := \beta_2\epsilon_k$ .
  - 7: **end for**
- 

Algorithm 4.1 is similar to [16, 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.

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 4.1, there is  $\kappa_{m,k} > 0$  such that for each corresponding inner iteration  $j$ ,  $s_{k,j,\text{cp}}$  generated according to (4.1) satisfies*

$$(4.9) \quad |f(x_{k,j} + s_{k,j,\text{cp}}) + \tau_k h(c(x_{k,j} + s_{k,j,\text{cp}})) - (\varphi + \tau_k \psi)(s_{k,j,\text{cp}}; x_{k,j})| \leq \kappa_{m,k} \|s_{k,j,\text{cp}}\|_2^2.$$

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 4.1,  $\kappa_{m,k} \leq \kappa_{m_1} \tau_k + \kappa_{m_2}$ , 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. Indeed, if  $\nabla f$  and  $J$  have Lipschitz constants  $L_g$  and  $L_J$ , respectively, then for any  $x, s \in \mathbb{R}^n$  and  $\tau > 0$ ,  $|f(x+s) + \tau h(c(x+s)) - \varphi(s; x) - \tau \psi(s; x)| \leq \frac{1}{2} L_g \|s\|_2^2 + \tau \frac{1}{2} L_J \|s\|_2^2$ , where we used the triangle inequality, the descent lemma for smooth functions [12, Lemma 1] on  $f$  and  $c$ , and the fact that  $\tau h$  is Lipschitz-continuous with constant  $\tau$ . Given the recent developments by Diouane et al. [23], Lipschitz continuity of  $\nabla f$  is not necessary to establish convergence of Algorithm 4.1. However, it remains essential to derive complexity bounds, which is the focus of the present analysis. Informally, Step Assumption 4.1 bounds the relative error of the models at each outer iteration, while Step Assumption 4.2 states that the model quality is expected to deteriorate at most linearly with the penalty parameter.

For a tolerance  $\epsilon > 0$ , we wish to find the number of inner and outer iterations of Algorithm 4.1 until

$$(4.10) \quad \theta(x_k)^{1/2} \leq \epsilon.$$

We use the following notation in the complexity analysis:

$$(4.11) \quad j_k(\epsilon_k) := \min\{j \in \mathbb{N} \mid \sigma_{k,j}^{1/2} \xi(x_{k,j}; \sigma_{k,j}, \tau_k)^{1/2} \leq \epsilon_k\} \quad (k \in \mathbb{N}),$$

$$(4.12) \quad k(\epsilon) := \min\{k \in \mathbb{N} \mid \theta(x_k)^{1/2} \leq \epsilon\}.$$

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 4.2 (3, Theorem 6.4 and 2). *Let Model Assumption 3.1 and Step Assumption 4.1 hold, and  $f$  be bounded below by  $f_{\text{low}}$ . Let  $0 < \eta_1 \leq \eta_2 < 1$  and  $0 < \gamma_3 \leq 1 < \gamma_2$  be the parameters of Algorithm A.1. Then,*

$$(4.13) \quad j_k(\epsilon_k) \leq \alpha_1 \sigma_{\max,k} \frac{f(x_{k,0}) + \tau_k h(c(x_{k,0})) - f_{\text{low}}}{\epsilon_k^2} + \alpha_2 \left| \log \left( \frac{\sigma_{\max,k}}{\sigma_{k,0}} \right) \right|,$$

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

$$(4.14) \quad \alpha_1 := (1 + |\log_{\gamma_1}(\gamma_3)|) > 0, \quad \alpha_2 := |1/\log(\gamma_1)| > 0, \quad \alpha_3 := 2\gamma_2/(1 - \eta_2) > 2.$$

Because our approach is similar to that of [16], the next lemmas give estimates between our measures, based on  $\xi$  and  $\theta$ , and a generalization of theirs. For all  $x \in \mathbb{R}^n$  and  $\tau > 0$ , the generalization of their measures [16, Equations (3.1) and (3.10)] to any exact norm penalty are

$$(4.15a) \quad \xi_{\text{TR}}(x, \tau) := f(x) + \tau h(c(x)) - \min_{\|s\|_2 \leq 1} (\varphi + \tau \psi)(s; x),$$

$$(4.15b) \quad \theta_{\text{TR}}(x) := h(c(x)) - \min_{\|s\|_2 \leq 1} \psi(s; x).$$

Their measures use the  $\ell_2$ -norm for  $\psi$  and  $h$  because their analysis is limited to the  $\ell_2$ -norm penalty.

LEMMA 4.3. For any  $x \in \mathbb{R}^n$ ,  $\tau > 0$  and  $\sigma > 0$ ,

$$(4.16) \quad \xi(x; \sigma, \tau) \geq \frac{1}{2} \min(1, \sigma^{-1} \xi_{\text{TR}}(x, \tau)) \xi_{\text{TR}}(x, \tau).$$

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

$$(4.17) \quad \sigma^{1/2} \xi(x; \sigma, \tau)^{1/2} \geq \frac{1}{\sqrt{2}} \xi_{\text{TR}}(x, \tau).$$

*Proof.* Since  $f$  and  $c$  are  $\mathcal{C}^1$ , and  $\tau h$  is convex and globally Lipschitz continuous, [16, Lemma 2.5] gives  $\xi(x, \sigma, \tau) - \frac{1}{2} \sigma \|s\|_2^2 \geq \frac{1}{2} \min(1, \sigma^{-1} \xi_{\text{TR}}(x, \tau)) \xi_{\text{TR}}(x, \tau)$ . Since  $\xi(x, \sigma, \tau) \geq \xi(x, \sigma, \tau) - \frac{1}{2} \sigma \|s\|_2^2$ , (4.16) 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  $\sigma^{-1} \xi_{\text{TR}}(x, \tau) \leq 1$ , as that will imply (4.17) by way of (4.16). Assume by contradiction that  $\sigma^{-1} \xi_{\text{TR}}(x, \tau) > 1$ . Our assumption and (4.16) imply  $\epsilon^2 \sigma^{-1} \geq \xi(x; \sigma, \tau) \geq \frac{1}{2} \xi_{\text{TR}}(x, \tau)$ . We multiply both sides by  $\sigma^{-1}$ , and obtain  $\epsilon^2 \sigma^{-2} \geq \frac{1}{2} \sigma^{-1} \xi_{\text{TR}}(x, \tau) > \frac{1}{2}$ , which contradicts our assumption that  $\sigma \geq \sqrt{2}\epsilon$ .  $\square$

LEMMA 4.4. For any  $x \in \mathbb{R}^n$ ,  $\theta_{\text{TR}}(x) \geq \min(\frac{1}{\sqrt{2}}, \theta(x)^{1/2}) \theta(x)^{1/2}$ .

*Proof.* For  $\{s^*\} = \text{prox}_{\psi}(0) = \text{argmin}_s \psi(s; x) + \frac{1}{2} \|s\|_2^2$ , Proposition 2.4 implies

$$(4.18) \quad \theta(x) = \psi(0; x) - \psi(s^*; x) \geq \frac{1}{2} \|s^*\|_2^2.$$

Assume first that  $\theta(x) \leq \frac{1}{2}$ . Then, (4.18) implies  $\|s^*\|_2 \leq 1$ . Hence,

$$\min_{\|s\|_2 \leq 1} h(c(x) + J(x)s) = \min_{\|s\|_2 \leq 1} \psi(s; x) \leq \psi(s^*; x),$$

which, together with (4.8) and (4.15b), implies that  $\theta_{\text{TR}}(x) \geq \theta(x)$ .

Assume next that  $\theta(x) > \frac{1}{2}$ . Note that (4.18) may be written  $\|\frac{1}{\sqrt{2}} \theta(x)^{-1/2} s^*\|_2 \leq 1$ . Since  $s \rightarrow h(c(x) + J(x)s)$  is convex and  $\frac{1}{\sqrt{2}} \theta(x)^{-1/2} < 1$ ,

$$\begin{aligned} \min_{\|s\|_2 \leq 1} h(c(x) + J(x)s) &\leq h(c(x) + \frac{1}{\sqrt{2}} \theta(x)^{-1/2} J(x)s^*) \\ &\leq \left(1 - \frac{1}{\sqrt{2}} \theta(x)^{-1/2}\right) h(c(x)) + \frac{1}{\sqrt{2}} \theta(x)^{-1/2} h(c(x) + J(x)s^*). \end{aligned}$$

We then have from (4.8) and (4.15b),  $\theta_{\text{TR}}(x) \geq \frac{1}{\sqrt{2}} \theta(x)^{-1/2} \theta(x) = \frac{1}{\sqrt{2}} \theta(x)^{1/2}$ .  $\square$

THEOREM 4.5. Assume that Step Assumption 4.2 holds and that  $f$  is bounded below by  $f_{\text{low}}$ . Assume that there is  $\bar{\tau} \geq 0$ , independent of  $\epsilon$ , such that  $\theta(x_{k+1})^{1/2} < \epsilon$  whenever  $\tau_k \geq \bar{\tau}$ . Then Algorithm 4.1 with stopping criterion (4.10) terminates either with an approximate KKT point of (1.1) or with an infeasible critical point of the feasibility measure (4.8) 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  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha_3$  are defined in (4.14),  $\kappa_{m_1}$  and  $\kappa_{m_2}$  are defined in Step Assumption 4.2,

and  $\kappa_f$ ,  $\kappa_h$  and  $\alpha_4$  are defined as

$$(4.19a) \quad \kappa_h := \max_{k < k(\epsilon)} h(c(x_{k,0})) > 0, \quad \kappa_f := \max_{k < k(\epsilon)} f(x_{k,0}) - f_{\text{low}} \geq 0,$$

$$(4.19b) \quad \alpha_4 := \max(\beta_3, \alpha_3 \kappa_{m_1}) > 0, \quad \alpha_5 := \max(\beta_4, \alpha_3 \kappa_{m_2}) > 0.$$

Alternatively, assume  $\tau_k$  grows unbounded with  $k$ . Assume further that there is  $\kappa_g \geq 0$  such that  $\|\nabla f(x_k)\|_2 \leq \kappa_g$  for all  $k$ . Then, [Algorithm 4.1](#) with stopping criterion [\(4.10\)](#) terminates either with an approximate KKT point of [\(1.1\)](#) or with an infeasible critical point of the feasibility measure [\(4.8\)](#) 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 4.1](#), there are at most  $\lceil \log(\epsilon/\epsilon_0)/\log(\beta_2) \rceil = \lceil \log_{\beta_2}(\epsilon/\epsilon_0) \rceil$  outer iterations where  $\tau_k$  is not increased until  $\theta(x_k)^{1/2}$  attains or drops below  $\epsilon$ . Similarly, there are at most  $\lceil (\bar{\tau} - \tau_0)/\beta_1 \rceil$  outer iterations where  $\tau_k$  is increased until it reaches or attains  $\bar{\tau}$ . Hence,  $\tau_k$  attains or exceeds  $\bar{\tau}$  in

$$(4.20) \quad k(\epsilon) \leq \left\lceil \frac{\bar{\tau} - \tau_0}{\beta_1} \right\rceil + \lceil \log_{\beta_2}(\epsilon/\epsilon_0) \rceil$$

outer iterations. We now find an estimate on  $j_k(\epsilon_k)$ . By [Step Assumptions 4.1](#) and [4.2](#) and [Theorem 4.2](#), [\(4.13\)](#) holds. First, from [\(4.19a\)](#)

$$(4.21) \quad f(x_{k,0}) + \tau_k h(c(x_{k,0})) - f_{\text{low}} \leq \kappa_h \tau_k + \kappa_f.$$

The form of  $\sigma_{k,0}$  in [Line 4](#) of [Algorithm 4.1](#), [Step Assumption 4.2](#), [\(4.19b\)](#) and the definition of  $\sigma_{\max,k}$  in [Theorem 4.2](#) give

$$(4.22) \quad \sigma_{\max,k} = \max(\sigma_{k,0}, \alpha_3 \kappa_{m,k}) \leq \max(\beta_3 \tau_k, \beta_4, \alpha_3(\kappa_{m_1} \tau_k + \kappa_{m_2})) \leq \alpha_4 \tau_k + \alpha_5,$$

so that

$$(4.23) \quad \frac{\sigma_{\max,k}}{\sigma_{k,0}} \leq \frac{\alpha_4 \tau_k + \alpha_5}{\max(\beta_3 \tau_k, \beta_4)} \leq \frac{\alpha_4 \tau_k + \alpha_5}{\beta_3 \tau_k} \leq \frac{\alpha_4 \tau_0 + \alpha_5}{\beta_3 \tau_0},$$

because  $\tau \mapsto \frac{\alpha_4 \tau + \alpha_5}{\beta_3 \tau}$  is decreasing on  $\mathbb{R}^+$ , and  $\tau_k \geq \tau_0$ . Combining [\(4.21\)](#), [\(4.22\)](#) and [\(4.23\)](#) into [\(4.13\)](#) 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  $\tau_k$  is increased, [Line 6](#) of [Algorithm 4.1](#) implies

$$(4.24) \quad \sigma_{k,j}^{1/2} \xi(x_k; \sigma_{k,j}, \tau_k)^{1/2} \leq \epsilon_k < \theta^{1/2}(x_k) \quad \text{with } j = j_k(\epsilon_k).$$

Using the notation of [Lemmas 4.3](#) and [4.4](#), if we assume that  $\sigma_{k,j} \geq \sqrt{2}\epsilon_k$ , then [\(4.24\)](#) and [Lemma 4.3](#) imply

$$(4.25) \quad \sigma_{k,j}^{1/2} \xi(x_k; \sigma_{k,j}, \tau_k)^{1/2} \geq \frac{1}{\sqrt{2}} \xi_{\text{TR}}(x_k, \tau_k).$$

From [Algorithm 4.1](#),  $\sigma_{k,j} \geq \beta_4 > 0$  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 [\(4.25\)](#) holds. Furthermore, for any  $s_{k,\text{TR}} \in \operatorname{argmin}_{\|s\|_2 \leq 1} \psi(s; x_k)$ ,

$$(4.26) \quad \xi_{\text{TR}}(x_k, \tau_k) = f(x_k) + \tau_k h(x_k) - \min_{\|s\|_2 \leq 1} (\varphi + \tau_k \psi)(s; x_k)$$

$$(4.27) \quad \geq f(x_k) + \tau_k h(x_k) - (\varphi + \tau_k \psi)(s_{k,\text{TR}}; x_k)$$

$$(4.28) \quad = \tau_k (h(x_k) - \min_{\|s\|_2 \leq 1} \psi(s; x_k)) - \nabla f(x_k)^T s_{k,\text{TR}}$$

$$(4.29) \quad = \tau_k \theta_{\text{TR}}(x_k) - \nabla f(x_k)^T s_{k,\text{TR}}$$

$$(4.30) \quad \geq \tau_k \theta_{\text{TR}}(x_k) - \|\nabla f(x_k)\|_2,$$

where we used the definition of the models [\(3.1\)](#) on the third line, the definition of  $\theta_{\text{TR}}$  [\(4.15b\)](#) on the fourth and the Cauchy-Schwarz inequality on the last line combined with the fact that  $\|s_{k,\text{TR}}\|_2 \leq 1$ . Substituting [\(4.30\)](#) into [\(4.25\)](#) then gives

$$(4.31) \quad \sigma_{k,j}^{1/2} \xi(x_k; \sigma_{k,j}, \tau_k)^{1/2} \geq \frac{1}{\sqrt{2}} (\tau_k \theta_{\text{TR}}(x_k) - \|\nabla f(x_k)\|).$$

At this point, we consider two cases. Assume first that  $\theta(x_k)^{1/2} \leq \frac{1}{\sqrt{2}}$ . [Lemma 4.4](#) and [\(4.31\)](#) combine with [\(4.24\)](#) to give

$$(4.32) \quad \frac{1}{\sqrt{2}} (\tau_k \theta(x_k) - \|\nabla f(x_k)\|_2) \leq \sigma_{k,j}^{1/2} \xi(x_k; \sigma_{k,j}, \tau_k)^{1/2} \leq \theta(x_k)^{1/2} \leq \frac{1}{\sqrt{2}} (\theta(x_k) + 1),$$

whenever  $\tau_k$  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, [\(4.32\)](#) implies

$$(4.33) \quad \|\nabla f(x_k)\|_2 \geq (\tau_k - 1)\theta(x_k) - 1.$$

Because  $\tau_k > 1$  for all large enough  $k$ , and  $\|\nabla f(x_k)\|_2 \leq \kappa_g$ , [\(4.33\)](#) yields  $\theta(x_k) \leq (\kappa_g + 1)/(\tau_k - 1)$  whenever  $\tau_k$  is increased. Therefore,  $\theta^{1/2}(x_k) \leq \epsilon_k$  whenever

$$(4.34) \quad \tau_k \geq \frac{\kappa_g + 1}{\epsilon_k} + 1.$$

Assume instead that  $\theta^{1/2}(x_k) > \frac{1}{\sqrt{2}}$ . [Lemma 4.4](#) and [\(4.31\)](#) imply

$$(4.35) \quad \sigma_{k,j}^{1/2} \xi(x_k; \sigma_{k,j}, \tau_k)^{1/2} \geq \frac{1}{\sqrt{2}} \left( \frac{1}{\sqrt{2}} \tau_k \theta(x_k)^{1/2} - \|\nabla f(x_k)\|_2 \right).$$

Using the same approach that led from [\(4.32\)](#) to [\(4.34\)](#), with the difference that we do not require the final inequality in [\(4.32\)](#), we find in this case

$$(4.36) \quad \tau_k \geq \frac{\sqrt{2}\kappa_g}{\epsilon_k} + 2.$$

Combining [\(4.34\)](#) with [\(4.36\)](#) gives  $\theta(x_k)^{1/2} \leq \epsilon_k$  whenever

$$(4.37) \quad \tau_k \geq \max \left( \frac{\kappa_g + 1}{\epsilon_k} + 1, \frac{\sqrt{2}\kappa_g}{\epsilon_k} + 2 \right).$$

For small enough values of  $\epsilon_k$ , i.e., for all large enough  $k$ ,  $\frac{\kappa_g+1}{\epsilon_k} + 1 > \frac{\sqrt{2}\kappa_g}{\epsilon_k} + 2$ , and we may assume without loss of generality that  $\theta(x_k)^{1/2} \leq \epsilon_k$  whenever (4.34) holds.

Overall, (4.34) implies  $\theta^{1/2}(x_k) \leq \epsilon$  whenever

$$(4.38) \quad \tau_k \geq \frac{\kappa_g + 1}{\epsilon^2} + 1.$$

The value on the right-hand side of (4.38) is the counterpart of  $\bar{\tau}$  from the first part of the proof, with the crucial difference that it depends on  $\epsilon$ . Still, we may use the value of the right-hand side of (4.38) in place of  $\bar{\tau}$  in the first part of the proof.  $\square$

The existence of  $\kappa_h$  and  $\kappa_f$  in (4.19a) is guaranteed by the fact that  $f$ ,  $h$  and  $c$  are continuous functions.

Combining Proposition 2.1 with Theorem 4.5, we see that the assumptions of the first case of Theorem 4.5 hold under the MFCQ. Therefore, the bound in the first case of Theorem 4.5 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 4.5 with those of Cartis et al. [16, 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 4.1, as we do not need to compute high-accuracy solutions when far from feasibility. However, if one chooses to enforce constant accuracy throughout, as is done in [16], 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  $\theta_{\text{TR}}$  but  $\theta$  does. We can easily find examples for which  $\theta(x) = \theta_{\text{TR}}(x)$ . Consider  $c : \mathbb{R} \rightarrow \mathbb{R}$ ,  $c(x) = x$ . For  $x \in \mathbb{R}$ , from (3.2b),  $\psi(s; x) = |x + s|$  and

$$\{s^*\} = \underset{|s| \leq 1}{\operatorname{argmin}} \psi(s; x) = \underset{1\psi}{\operatorname{prox}}(0) = \begin{cases} \{-1\} & \text{if } x > 1, \\ \{-x\} & \text{if } |x| \leq 1, \\ \{1\} & \text{if } x < -1, \end{cases}$$

which implies from (4.8) and (4.15b) that

$$\begin{aligned} \theta(x) &= |x| - |x + s^*| = \min(|x|, 1), \\ \theta_{\text{TR}}(x) &= |x| - |x + s^*| = \min(|x|, 1). \end{aligned}$$

Now, we can see that the issue really comes from Lemma 4.4; if instead of the inequality in Lemma 4.4 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 (4.36), we would have found an upper bound on  $\tau_k$  in  $\mathcal{O}(\epsilon^{-1})$  instead of  $\mathcal{O}(\epsilon^{-2})$  which would have then led to the  $\mathcal{O}(\epsilon^{-5})$  complexity.

Furthermore, if we consider now  $c : \mathbb{R} \rightarrow \mathbb{R}$ ,  $c(x) = x^2/8$ . For  $x \in \mathbb{R}$ , from (3.2b),

$\psi(s; x) = \frac{1}{4}|\frac{1}{2}x^2 + xs|$  and

$$\{s^*\} = \underset{1\psi}{\text{prox}}(0) = \underset{s}{\text{argmin}} \frac{1}{2}s^2 + \psi(s; x) = \{-\frac{1}{2}x\}$$

$$\{s_{\text{TR}}^*\} = \underset{|s| \leq 1}{\text{argmin}} \psi(s; x) = \begin{cases} \{-1\} & \text{if } x > 2, \\ \{-\frac{x}{2}\} & \text{if } |x| \leq 2, \\ \{1\} & \text{if } x < -2. \end{cases}$$

Then, from (4.8) and (4.15b),

$$\begin{aligned} \theta(x) &= \frac{1}{8}|x^2| - \frac{1}{4}|\frac{1}{2}x^2 + xs^*| = \frac{1}{8}x^2. \\ \theta_{\text{TR}}(x) &= \frac{1}{8}|x^2| - \frac{1}{4}|\frac{1}{2}x^2 + xs_{\text{TR}}^*| = \frac{1}{4} \min(|x|, \frac{1}{2}x^2). \end{aligned}$$

Therefore in this case, for all  $x \in \mathbb{R}$ ,  $\theta_{\text{TR}}(x) = \min(\frac{1}{\sqrt{2}}, \theta(x)^{1/2})\theta(x)^{1/2}$  so that the bound in Lemma 4.4 is tight.

For  $\{s^*\} = \underset{1\psi}{\text{prox}}(0) = \underset{s}{\text{argmin}} \frac{1}{2}\|s\|_2^2 + \psi(s; x)$ ,

$$(4.39) \quad -s^* \in \partial\psi(s^*; x).$$

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

$$\psi(s^*; x) = \psi(0; x) + g_{s^*,x}^T s^* + o(\|s^*\|_2).$$

The choice  $g_{s^*,x} = -s^*$ , (4.8) and (4.39) combine to give  $\theta(x) = \|s^*\|_2^2 + o(\|s^*\|_2)$ . Hence,  $\theta$  varies like the squared norm of a subgradient, which justifies taking the square root of  $\theta$  to define a stationarity measure. In addition, Lemma 4.4 implies  $\theta_{\text{TR}} \geq \theta$  as  $\theta \rightarrow 0$ . The above leads us to believe that the  $\mathcal{O}(\epsilon^{-8})$  is a more realistic worst-case complexity bound for exact penalty algorithms when the penalty parameter is unbounded, though neither bound has been proven to be tight.

Finally, the assumptions of Theorem 4.5 are weaker than those of Cartis et al. [16] as we only require boundedness of  $\{\nabla f(x_k)\}$ , whereas they require boundedness of  $\{x_k\}$ .

**A quasi-Newton variant.** Instead of using [3, Algorithm 6.1] as solver for (1.2) in Algorithm 4.1, we explore variant R2N recently proposed in [23] that uses or approximates second-order information on  $f$ , and show that it preserves 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 (4.1a), we compute  $s_{k,j}$  as an approximate solution of

$$(4.40) \quad \underset{s}{\text{minimize}} \quad m_Q(s; x_{k,j}, B_{k,j}) + \frac{1}{2}\sigma_{k,j}\|s\|_2^2,$$

where  $m_Q(s; x, B) := \varphi_Q(s; x, B) + \tau_k\psi(s; x)$ , and  $\varphi_Q(s; x, B) := \varphi(s; x) + \frac{1}{2}s^T B s$ , with  $\varphi$  defined in (3.2a),  $B = B^T \in \mathbb{R}^{n \times n}$ . 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 (4.40) is unbounded below along any negative curvature direction of  $B + \sigma I$ . Hence, we always adjust  $\sigma$  so that  $\varphi_Q(s; x, B) + \frac{1}{2}\sigma\|s\|_2^2$  is convex. Clearly, Model Assumption 3.1 continues to hold for  $\varphi_Q$ .

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

STEP ASSUMPTION 4.3. *There is a constant  $\kappa_B$  such that for each iteration  $k$  and  $j$  of [Algorithm 4.1](#) that uses R2N,  $\|B_{k,j}\|_2 \leq \kappa_B$ .*

R2N uses  $\xi(x_{k,j}; \nu_{k,j}^{-1}, \tau_k)$ , defined in (4.2), as a stationarity measure, where  $\nu_{k,j}^{-1} := \theta_1 / (\sigma_{k,j} + \|B_{k,j}\|_2)$ , and where  $\theta_1 \in (0, 1)$  is a parameter.

An analogue of [Theorem 4.5](#) holds for [Algorithm 4.1](#) with R2N as subsolver. If we denote  $\tilde{\sigma}_{k,j} := \nu_{k,j}^{-1}$ , [Step Assumption 4.1](#) yields

$$\sigma_{k,j} \theta_1^{-1} \leq \tilde{\sigma}_{k,j} \leq \sigma_{k,j} \theta_1^{-1} + \kappa_B \theta_1^{-1}.$$

With these inequalities, we find  $\tilde{\sigma}_{\max,k} = \theta_1^{-1} \sigma_{\max,k} + \theta_1^{-1} \kappa_B$  which acts as the  $\sigma_{\max,k}$  defined previously. With that, we can establish [Theorem 4.5](#) with  $\xi(x_{k,j}; \nu_{k,j}^{-1}, \tau_k)$  as inner stationarity measure.

The updated complexity bounds rest upon the following result.

THEOREM 4.6 ([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  $f_{\text{low}}$ . Let  $0 < \eta_1 \leq \eta_2 < 1$  and  $0 < \gamma_3 \leq 1 < \gamma_2$  be the parameters of [Algorithm A.1](#). Then,*

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

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

THEOREM 4.7. *Assume that [Step Assumption 4.2](#) and the assumptions of [Theorem 4.6](#) hold. Assume that there is  $\bar{\tau} \geq 0$ , independent of  $\epsilon$ , such that  $\theta(x_{k+1})^{1/2} < \epsilon$  whenever  $\tau_k \geq \bar{\tau}$ . Then [Algorithm 4.1](#) with stopping criterion (4.10) using [Algorithm A.1](#) to solve (1.2) terminates either with an approximate KKT point of (1.1) or with an infeasible critical point of the feasibility measure (4.8) in at most*

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

inner iterations, which is an overall complexity of  $\mathcal{O}((\bar{\tau} + |\log(\epsilon/\epsilon_0)|) \bar{\tau}^2 \epsilon^{-2})$ , where  $\alpha_1, \alpha_2, \alpha_3$  are as in [Theorem 4.2](#),  $\alpha_4, \alpha_5, \kappa_h, \kappa_f$  are as in [Theorem 4.5](#),  $\kappa_{m_1}$  and  $\kappa_{m_2}$  are as in [Step Assumption 4.2](#),  $\kappa_B$  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  $\tau_k$  grows unbounded with  $k$ . Assume further that there is  $\kappa_g \geq 0$  such that  $\|\nabla f(x_k)\|_2 \leq \kappa_g$  for all  $k$ . Then, [Algorithm 4.1](#) with stopping criterion (4.10) using R2N to solve (1.2) terminates either with an approximate KKT point of (1.1) or with an infeasible critical point of the feasibility measure (4.8) 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 4.7](#) is almost identical to that of [Theorem 4.5](#). The only difference is that the bound (4.23) becomes  $\tilde{\sigma}_{\max,k}/\tilde{\sigma}_{k,0} \leq (\sigma_{\max,k} + \kappa_B)/\sigma_{k,0}$ .

**5. Proximal operators.** We now provide an explicit characterization of the solution  $s_{k,j,\text{cp}}$  in (4.1a). We begin with general results and then specialize them to our setting. Moreover, when [Algorithm A.1](#) is used, we derive an efficient procedure for computing  $s_{k,j}$  in (4.40) that avoids the need for an iterative solver for nonsmooth problems. Although the computation may involve solving a scalar equation, we believe it remains more efficient than general-purpose proximal evaluations.

**THEOREM 5.1.** *Let  $Q \in \mathbb{R}^{n \times n}$  be symmetric positive definite,  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $d \in \mathbb{R}^n$  and  $\tau > 0$ . The unique solution of*

$$(5.1) \quad \underset{u \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2} u^\top Q u - d^\top u + \tau \|Au + b\|,$$

is

$$(5.2) \quad u^* = Q^{-1}(d + A^T y^*)$$

where  $y^*$  is a solution of

$$(5.3) \quad \underset{y \in \mathbb{R}^m}{\text{maximize}} \quad -\frac{1}{2}(d + A^T y)^\top Q^{-1}(d + A^T y) - b^\top y \quad \text{subject to} \quad \|y\|_* \leq \tau.$$

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

*Proof.* As (5.1) is strictly convex, it has a unique solution  $u^*$ . It may be written

$$(5.4) \quad \underset{u \in \mathbb{R}^n, z \in \mathbb{R}^m}{\text{minimize}} \quad \frac{1}{2} u^\top Q u - d^\top u + \tau \|z\| \quad \text{subject to} \quad z = Au + b.$$

The Lagrangian of (5.4) is  $\mathcal{L}(u, z; y) = \left[ \frac{1}{2} u^\top Q u - d^\top u - (A^T y)^\top u \right] + \left[ \tau \|z\| + y^\top z \right] - b^\top y$ , and is separable with respect to  $u$  and  $z$ . The minimizer of the terms in  $u$  is  $u^* = Q^{-1}(d + A^T y)$ , and corresponds to an optimal value of  $-\frac{1}{2}(d + A^T y)^\top Q^{-1}(d + A^T y)$ . Regarding the terms in  $z$ ,

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

where  $(\tau h)^*$  is the Fenchel conjugate of  $\tau h$ . Using [13, Example 3.26],  $(\tau h)^*(w) = \tau \chi(w/\tau \mid \mathbb{B}_*)$ . We conclude that

$$\min_z \left[ \tau \|z\| + y^\top z \right] = \begin{cases} 0 & \|y\|_* \leq \tau, \\ -\infty & \|y\|_* > \tau. \end{cases}$$

The above gives us (5.3) as the dual of (5.4). Since  $(u, z) = (0, b)$  is feasible for (5.4), the relaxed Slater condition, hence strong duality, holds for (5.4) and (5.3). If  $A$  has full row rank, (5.3) is strictly concave, and therefore has a unique solution.  $\square$

In view of [Theorem 5.1](#), computing a step under a general norm penalty reduces to solving a trust-region subproblem in the dual norm. When  $h = \|\cdot\|_2$ , this subproblem can be solved efficiently, as it has been extensively studied over the past decades [22, §7]. For this reason, we restrict our attention to this special case for the remainder of the paper.

**THEOREM 5.2.** *Under the assumptions of [Theorem 5.1](#), in the case where  $\|\cdot\| = \|\cdot\|_2$ , a solution of (5.3) is given by*

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

where  $\alpha^*$  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  $p = 2$ , we can rewrite (5.3) as

$$(5.5) \quad \underset{y \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{2}(d + A^T y)^T Q^{-1}(d + A^T y) + b^T y \quad \text{subject to} \quad \|y\|_2^2 \leq \tau^2.$$

Whether  $A$  has full row rank or not, (5.5) is convex and satisfies Slater's condition. Thus, by the KKT conditions,  $y$  solves (5.5) if and only if there is  $\alpha^*$  such that

$$(5.6) \quad (AQ^{-1}A^T + \alpha^*I)y + AQ^{-1}d + b = 0, \quad \text{and} \quad 0 \leq \alpha^* \perp (\tau - \|y\|_2) \geq 0.$$

There are now two cases. In the first case,  $\alpha^* = 0$ . Then by (5.6),  $(AQ^{-1}A^T)y = -(AQ^{-1}d + b)$ . Thus, if  $AQ^{-1}d + b \in \text{Range}(AQ^{-1}A^T)$ , by primal feasibility, the pseudo-inverse is the solution whenever  $\|(AQ^{-1}A^T)^\dagger(AQ^{-1}d + b)\|_2 \leq \tau$ . In the second case,  $\alpha^* > 0$  and (5.6) yields  $\|y\|_2^2 = \tau^2$  and  $y = -(AQ^{-1}A^T + \alpha^*I)^{-1}(AQ^{-1}d + b)$ , because  $AQ^{-1}A^T + \alpha^*I$  is nonsingular, and where  $g(\alpha^*) = 0$ .  $\square$

When  $A$  has full row rank, the solution given by Theorem 5.2 is unique. When  $A$  is rank deficient in Theorem 5.2 and  $\alpha^* = 0$ , the pseudo-inverse yields the minimum Euclidean norm solution of the linear system in (5.6). If the latter does not have norm at most  $\tau$ , no other solution does, and we must be in the case  $\alpha^* > 0$ . The next result specializes Theorem 5.2 to the evaluations of the proximal operator for  $\tau\psi(\cdot; x)$ .

**COROLLARY 5.3.** *Let  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ , and  $\tau > 0$ . Let  $\eta : u \rightarrow \tau\|Au + b\|_2$ . For  $\nu > 0$ ,*

$$(5.7) \quad \underset{\nu\eta}{\text{prox}}(w) \ni \begin{cases} w - A^T(AA^T)^\dagger(Aw + b) & \text{if } \|(AA^T)^\dagger(Aw + b)\|_2 \leq \nu\tau \\ & \text{and } Aw + b \in \text{Range}(AA^T) \\ w - A^T(AA^T + \alpha^*I)^{-1}(Aw + b) & \text{otherwise,} \end{cases}$$

where  $\alpha^*$  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 (2.1),  $\text{prox}_{\nu\eta}(w) = \underset{u \in \mathbb{R}^n}{\text{argmin}} \frac{1}{2}u^T Iu - w^T u + \nu\tau\|Au + b\|_2$ . Replacing  $Q$  with  $I_n$ ,  $d$  with  $w$  and  $\tau$  with  $\nu\tau$  in Theorems 5.1 and 5.2 concludes.  $\square$

In the context of Algorithm A.1, we will apply Corollary 5.3 with  $\tau = \tau_k$ ,  $\eta := \tau_k\psi(\cdot; x_k)$ ,  $A = J(x_k)$ ,  $b = c(x_k)$ , and either  $\nu = \sigma_k^{-1}$  or  $\nu = \frac{\theta_1}{\|B_k\|_2 + \sigma_k}$  depending on whether we use R2 or R2N respectively.

The same methodology allows us to derive an explicit characterization of solutions of (4.40).

**COROLLARY 5.4.** *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  $\eta : u \rightarrow \tau\|Au + b\|_2$ . For  $\nu > 0$ , define*

$$(5.8) \quad \underset{\nu\eta}{\text{prox}}(w, B) := \underset{u \in \mathbb{R}^n}{\text{argmin}} \frac{1}{2}\nu^{-1}\|u - w\|_2^2 + \eta(u) + \frac{1}{2}u^T Bu.$$

Let  $v := AQ^{-1}w + b$ . Then, the only element in  $\text{prox}_{\nu\eta}(w, B)$  is

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

where  $\alpha^*$  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 (5.8),  $\text{prox}_{\nu\eta}(w, B) = \text{argmin}_{u \in \mathbb{R}^n} \frac{1}{2}u^T Qu - w^T u + \nu\tau\|Au + b\|_2$ . Replacing  $Q$  with  $\nu B + I$ ,  $d$  with  $w$  and  $\tau$  with  $\nu\tau$  in Theorems 5.1 and 5.2 concludes.  $\square$

An efficient procedure to solve the  $\ell_2$ -norm trust-region subproblem was proposed by Moré and Sorensen [40]. We follow their approach. In the light of Corollary 5.3 and using the same notation, it should be clear that any efficient procedure to solve (5.3) 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 [40] is to solve the equivalent secular equation

$$(5.9) \quad g(\alpha) = 0 \iff \frac{1}{\|(AA^T + \alpha I)^{-1}(Ax + b)\|_2} = \frac{1}{\nu\tau},$$

which has preferable numerical properties. They use Newton's method to solve (5.9). We summarize the procedure as Algorithm 5.1 with the help of the following result.

LEMMA 5.5 (22, Lemma 7.3.1). Define  $\phi(\alpha) := \frac{1}{\|s(\alpha)\|_2} - \frac{1}{\nu\tau}$ , where

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

Then,  $\phi$  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 5.1** Proximal operator evaluation.

---

- 1: Compute  $s(0)$  as given by Lemma 5.5.
  - 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  $\alpha_0 := 0$ . Otherwise, set  $\alpha_0 := \sqrt{\epsilon_M}$  and compute  $s(\alpha_0)$ .
  - 5: **while**  $\|s(\alpha_k)\| \neq \nu\tau$  **do**
  - 6:   Compute  $\alpha_{k+1}$  as  $\alpha^+$  given by (5.10).
  - 7:   If  $\alpha^+ \leq 0$ , reset  $\alpha_{k+1} := \theta\alpha_k$ .
  - 8:   Compute  $s(\alpha_{k+1})$ .
  - 9:   Let  $k := k + 1$ .
  - 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  $A_\alpha^T := (A \ \sqrt{\alpha}I_m)^T$ . The  $R$  factor coincides with the Cholesky factor. The update of Newton's method can be found using Lemma 5.5:

$$(5.10) \quad \alpha^+ = \alpha_k - \frac{\phi(\alpha_k)}{\phi'(\alpha_k)} = \alpha_k - \frac{(\nu\tau - \|s(\alpha_k)\|) \|s(\alpha_k)\|^2}{\nu\tau s(\alpha_k)^T \nabla s(\alpha_k)}.$$

**Algorithm 5.1** 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. [22, Lemma 7.3.2] indicates that convergence is assured once  $\alpha \in \mathcal{L} := (\max(0, -\lambda_1), \alpha^*]$ , where  $\alpha^*$  solves (5.9) and where  $\lambda_1$  is the smallest eigenvalue of  $AA^T$ , for all Newton iterates remain in  $\mathcal{L}$  and converge to  $\alpha^*$ . If  $\alpha = 0$  does not yield the solution, we initialize Newton's method from an  $\alpha_0$  chosen so that  $\phi'(\alpha_0)$  can be computed. If ever  $\alpha_k > \alpha^*$ , convergence is not assured. However, [22, Lemma 7.3.3] indicates that, due to the concavity of  $\phi$ , 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  $\alpha_k$  in hopes of eventually landing in  $\mathcal{L}$ .

We now turn to the solution of (4.40). For any  $\alpha \geq 0$ , the element  $u \in \text{prox}_{\nu\eta_\tau}(w, B)$  given by **Corollary 5.4** can equivalently be obtained from

$$(5.11) \quad \begin{bmatrix} -Q & A^T \\ A & \alpha I \end{bmatrix} \begin{bmatrix} u \\ y \end{bmatrix} = \begin{bmatrix} -w \\ -b \end{bmatrix} \quad \text{where } Q = \nu B + I.$$

Because  $B$  will typically be a limited-memory quasi-Newton approximation in our implementation, materializing, let alone factorizing, (5.11) would be inefficient. Thus, we use MINRES-QLP [18]. Even though only an approximate minimizer of (4.40) is required, we compute an accurate solution of (5.11) by setting both the absolute and relative tolerances of MINRES-QLP to  $\epsilon_M^{0.7}$  where  $\epsilon_M$  is the machine epsilon. If  $\alpha = 0$  and  $A$  is rank deficient, MINRES-QLP is guaranteed to return the minimum-norm solution of (5.11), but not necessarily  $y = -(AQ^{-1}A^T)^\dagger v$  required by **Corollary 5.4**. 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  $\alpha \geq \sqrt{\epsilon_M}$ , where  $\epsilon_M$  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 5.4**, we solve the *secular* equation

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

If  $y(\alpha)$  solves (5.11),

$$(5.12) \quad \frac{\phi_Q(\alpha)}{\phi'_Q(\alpha)} = \frac{\|y(\alpha)\|^2}{y(\alpha)^T (AQ^{-1}A^T + \alpha I)^{-1} y(\alpha)} \left(1 - \frac{\|y(\alpha)\|}{\nu\tau}\right).$$

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

$$(5.13) \quad \begin{bmatrix} -Q & A^T \\ A & \alpha I \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ y(\alpha) \end{bmatrix}.$$

**6. Implementation and experiments.** We implemented all algorithms above in Julia 1.11 [9]. **Algorithms 4.1** and **A.1** are part of the **RegularizedOptimization.jl** package [5]; **Algorithm 5.1** is in **ShiftedProximalOperators.jl** [6].

We compare [Algorithm 4.1](#) against the augmented-Lagrangian solver Percival [4, 25] and the IPOPT [44] solver—which reduces to an SQP method for (1.1)—on equality-constrained problems from the CUTEst collection [30, 43] 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 4.1](#) 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.,  $\sigma_f(x_{k,j}) := s_{k,j}^T y_{k,j} / s_{k,j}^T s_{k,j}$ , where  $s_{k,j} := x_{k,j} - x_{k,j-1}$  and  $y_{k,j} := \nabla f(x_{k,j}) - \nabla f(x_{k,j-1})$ .

The optimality measures in (4.2) and (4.8), 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 (1.1). Therefore, our implementation of [Algorithm 4.1](#) stops as soon as

$$(6.1) \quad \|\nabla f(x_k) + J(x_k)^T y_k\|_\infty \leq \epsilon \quad \text{and} \quad \|c(x_k)\|_\infty \leq \epsilon,$$

where  $y_k$  is the solution of (5.1) in the case where  $Q = \sigma_k I$ .

Percival is modified so that it also stops when it finds points that satisfy (6.1), which is the criterion used in IPOPT. We disabled IPOPT’s automatic scaling of the problem statement, by setting `nlp_scaling_method` to “none”, and the scaling of the dual feasibility with respect to the dual variables by setting `s_max` to the maximum of the Float64 number system. In these preliminary small-scale tests, all solvers use an absolute stopping test with  $\epsilon = 10^{-3}$ . This is done in IPOPT by setting the parameter `acceptable_iter` to 0. In [Algorithm 4.1](#), we used  $\beta_1 = \tau_0 = \sqrt{nm}$ ,  $\epsilon_0 = 10^{-2}$ ,  $\beta_2 = 0.1$ ,  $\beta_3 = 10^{-2}$ ,  $\beta_4 = \epsilon_M \approx 2.2 \times 10^{-16}$ , where  $n$  and  $m$  are respectively the number of variables and constraints of the problem. In [Algorithm 5.1](#), we used `QRMumps.jl` [15] for the QR factorizations and set  $\theta = 0.8$ . The stopping criterion for [Algorithm 5.1](#) is  $\|s(\alpha_k)\|_2 - \nu\tau < \epsilon_M^{0.3}$ . We approximate the least-norm solution of  $AA^T x = b$  using the Golub-Riley iteration [?] with a regularization parameter of  $\epsilon_M^{0.9}$ . The latter requires a single QR factorization and may be viewed as a form of iterative refinement. More sophisticated strategies will be required in later versions of our implementation. We set the maximal number of iterations for [Algorithm 5.1](#) to 10. All tests are performed in double precision.

We also compare [Algorithm 4.1](#) 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 6. At each successful iteration, the difference of gradients of the Lagrangian of (1.1),

$$\nabla \mathcal{L}(x_k, y_k^{LS}) - \nabla \mathcal{L}(x_{k-1}, y_k^{LS}),$$

is used to update  $B_k$ , where  $y_k^{LS}$  is the minimum least-squares solution of  $\nabla f(x_k) + J(x_k)^T y = 0$ . When we use an LBFGS approximation, our implementation uses damping [48, Section 5] with damping parameter 0.1. The MINRES-QLP tolerance is set to  $\epsilon_M$  and its maximum number of iterations to  $10^4$  to evaluate (5.8).

We report our results in the form of [Dolan and Moré](#) [24] performance profiles in [Figure 6.1](#) in terms of number of objective, gradient and constraint evaluations. Recall

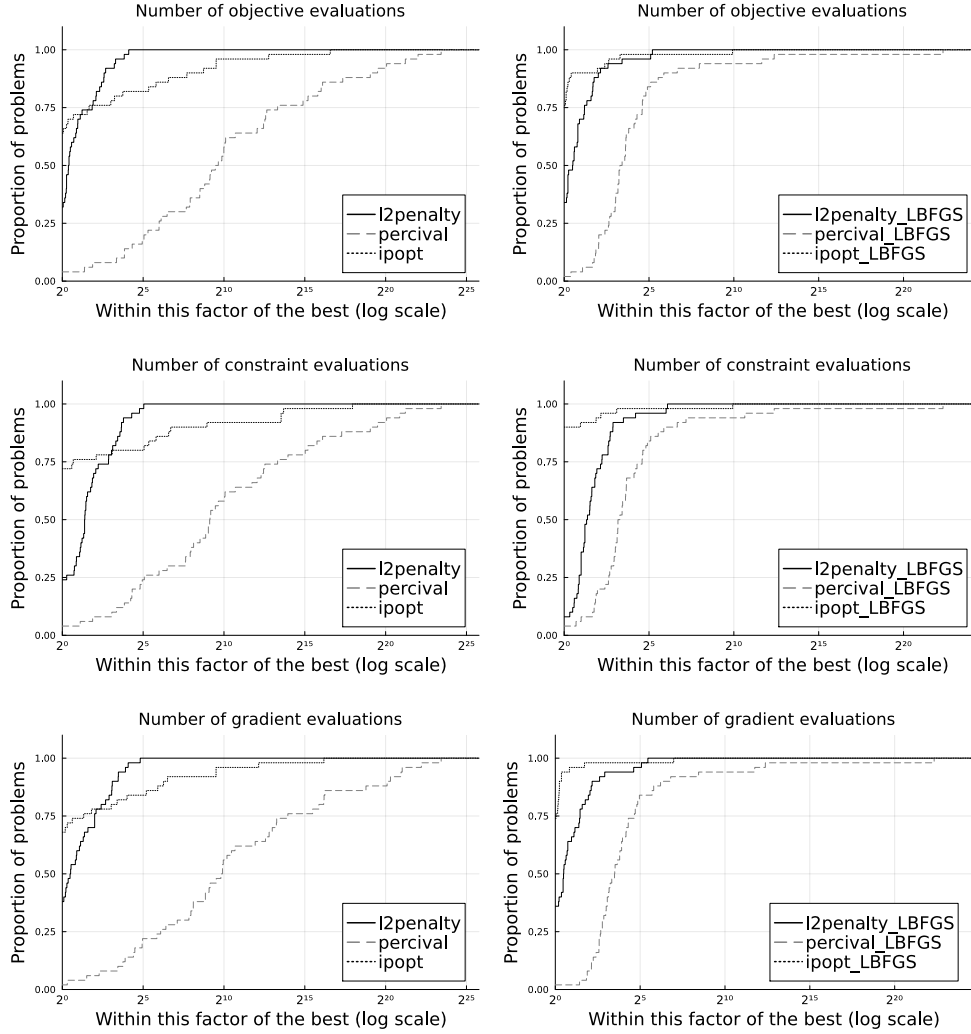


FIG. 6.1. Left: *Algorithm 4.1* with the R2 subsolver against Percival and IPOPT with a spectral quasi-Newton approximation. Right: *Algorithm 4.1* with the R2N subsolver against Percival and IPOPT with a LBFGS quasi-Newton approximation.

that a performance profile [24] is defined as follows. For a set  $P$  of problems and a set  $S$  of solvers, we compute a performance metric  $t_{p,s}$  for each  $s \in S$  and  $p \in P$ . We then compute the ratio  $r_{p,s} := t_{p,s} / \min_s t_{p,s}$  for each  $p \in P$  and  $s \in S$ . If  $n_p$  is the number of problems in  $P$ , the performance profile for solver  $s$  is

$$\pi_s(\omega) := \frac{1}{n_p} \sum_{p \in P} \mathbf{1}_{r_{p,s} \leq \omega},$$

where, for  $\omega \geq 0$ ,  $\mathbf{1}_{r_{p,s} \leq \omega} = 1$  if  $r_{p,s} \leq \omega$ , and  $\mathbf{1}_{r_{p,s} \leq \omega} = 0$  otherwise.

When using R2N, we only report results with LBFGS; results with LSR1 are nearly identical. In the left column of Figure 6.1, 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 4.1](#) is very promising for problems where, for some reason, constraint Hessians cannot be evaluated or approximated. In the right column of [Figure 6.1](#), we see that [Algorithm 4.1](#) is ahead of Percival, while IPOPT is competitive with our implementation in terms of objective and gradient evaluations. Whether we use R2 or R2N, the robustness of [Algorithm 4.1](#) is on par with the other two solvers. From the 50 problems considered in our experiment, two of them are infeasible, namely SSINE and VANDANIUMS. Both were successfully detected as infeasible by our algorithm when using R2, whereas IPOPT only found VANDANIUMS and Percival found none. When using R2N, only VANDANIUMS was detected by both IPOPT and [Algorithm 4.1](#), and IPOPT reported a false positive with DIXCHLNG, Percival did not find any in this case either. 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 4.1](#) is, to the best of our knowledge, the first practical implementation of the exact  $\ell_2$ -penalty method. On small-scale problems, our preliminary implementation is ahead of an augmented Lagrangian method. It is also ahead of 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. A possible improvement in terms of CPU time is to solve [\(5.9\)](#) in reduced-precision arithmetic. Similarly, MINRES-QLP does not provide the required least-norm solution in [Corollary 5.4](#). 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.

Our framework is general and allows us to implement the exact penalty method in any other norm as long as the resulting trust-region subproblem can be solved efficiently. In particular, [Theorem 5.1](#) suggests that we could implement the exact  $\ell_1$ -penalty method originally proposed by [Pietrzykowski \[46\]](#). Indeed, evaluating the proximal operator amounts to solving a convex bound-constrained subproblem for which there are efficient polynomial time algorithms [[22](#), Chapter 7.8].

A trust-region variant of [Algorithm 4.1](#) 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  $h$ :

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

**Acknowledgements.** We would like to thank Tangi Migot and Alexis Montoisson for their kind assistance with the numerical implementation of our algorithm.

#### Appendix A. Inner algorithms.

[Algorithm A.1](#) states [[23](#), Algorithm 4.1] for completeness; [[3](#), Algorithm 6.1] is a special case of [Algorithm A.1](#) with  $B_j = 0$  and  $\alpha = 1$ .

Since  $h$  is bounded below, finding an  $x_0$  where  $h$  is finite in [Algorithm A.1](#) is always trivial. Although neither [Aravkin et al. \[3\]](#) nor [Diouane et al. \[23\]](#) mention a

---

**Algorithm A.1** R2N: A proximal modified Quasi-Newton method, [23, Algorithm 4.1]

---

**Require:** The value  $\tau := \tau_k > 0$  of the penalty parameter is given by the outer iteration.

- 1: Choose constants  $\sigma_{\min} > 0$ ,  $0 < \eta_1 \leq \eta_2 < 1$  and  $0 < \gamma_3 \leq 1 < \gamma_1 \leq \gamma_2$ ,  $0 < \alpha < 1$  and  $\beta > 1$ .
- 2: Choose  $x_0 \in \mathbb{R}^n$  where  $h$  is finite,  $\sigma_0 \geq \sigma_{\min}$ , compute  $f(x_0) + \tau h(x_0)$ .
- 3: **for**  $j = 0, 1, \dots$  **do**
- 4:   Choose  $B_j = B_j^T \in \mathbb{R}^{n \times n}$ .
- 5:   Set  $\nu_j = \alpha / (\|B_j\|_2 + \sigma_j)$ .
- 6:   Compute  $s_{j,\text{cp}} \in \text{prox}_{\nu_j \tau \psi(\cdot; x_j)}(-\nu_j \nabla f(x_j))$ .
- 7:   Compute an approximate minimizer  $s_j$  of  $m_Q(s; x_j, B_j) + \frac{1}{2} \sigma_j \|s\|_2^2$  such that

$$m_Q(s_j; x_j, B_j) + \frac{1}{2} \sigma_j \|s_j\|_2^2 \leq m_Q(s_{j,\text{cp}}; x_j, B_j) + \frac{1}{2} \sigma_j \|s_{j,\text{cp}}\|_2^2.$$

- 8:   If  $\|s_j\|_2 > \beta \|s_{j,\text{cp}}\|_2$ , reset  $s_j = s_{j,\text{cp}}$ .
- 9:   Compute the ratio

$$\rho_j := \frac{f(x_j) + \tau h(x_j) - (f(x_j + s_j) + \tau h(x_j + s_j))}{\varphi_Q(0; x_j, B_j) + \tau \psi(0; x_j) - (\varphi_Q(s_j; x_j, B_j) + \tau \psi(s_j; x_j))}.$$

- 10:   If  $\rho_j \geq \eta_1$ , set  $x_{j+1} = x_j + s_j$ . Otherwise, set  $x_{j+1} = x_j$ .
- 11:   Update the regularization parameter according to

$$\sigma_{j+1} \in \begin{cases} [\gamma_3 \sigma_j, \sigma_j] & \text{if } \rho_j \geq \eta_2, \\ [\sigma_j, \gamma_1 \sigma_j] & \text{if } \eta_1 \leq \rho_j < \eta_2, \\ [\gamma_1 \sigma_j, \gamma_2 \sigma_j] & \text{if } \rho_j < \eta_1. \end{cases}$$

- 12:   Set  $\sigma_{j+1} \leftarrow \max(\sigma_{j+1}, \sigma_{\min})$ .
  - 13: **end for**
- 

minimum regularization parameter  $\sigma_{\min}$ , it does not change the complexity of their algorithms. For example, building upon [3], [Leconte and Orban \[37\]](#) use a maximum trust-region parameter  $\Delta_{\max}$  that is analogous to  $\sigma_{\min}$  in our context and which does not change the complexity of the algorithm. In view of [Theorems 5.1 and 5.2](#),  $B_j + \sigma_j I$  should be positive definite when computing a step. We argue however that if this is not the case then [Line 8](#) will be triggered, causing the algorithm to retreat to the Cauchy point

**LEMMA A.1.** *Let  $d \in \mathbb{R}^n$ ,  $B = B^T \in \mathbb{R}^{n \times n}$ ,  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $\tau > 0$ , and  $\sigma > 0$ . If  $B + \sigma I$  is not positive semi-definite,*

$$(A.1) \quad m(s) := d^T s + \frac{1}{2} s^T B s + \tau \|A s + b\|_2 + \frac{1}{2} \sigma \|s\|_2^2$$

*is such that  $\lim_{t \rightarrow \infty} \min_{\|s\| \leq t} m(s) = -\infty$ .*

*Proof.* Assume that  $B + \sigma I$  is not positive semi-definite. Then, there is a vector  $\bar{s} \in \mathbb{R}^n$  with  $\|\bar{s}\|_2 = 1$  such that  $\bar{s}^T (B + \sigma I) \bar{s} < 0$ . This implies that

$$\lim_{t \rightarrow \infty} m(t\bar{s}) \leq \lim_{t \rightarrow \infty} \frac{1}{2} t^2 \bar{s}^T (B + \sigma I) \bar{s} + t(d^T \bar{s} + \tau \|A \bar{s}\|_2) + \tau \|b\|_2 = -\infty. \quad \square$$

From [Lemma A.1](#), whenever  $B_j + \sigma_j I$  is not positive definite, the minimizer of  $m_Q(\cdot; x_j, B_j) + \frac{1}{2}\sigma_j \|\cdot\|_2^2$  is arbitrarily large so that [Line 8](#) is activated for any  $\beta > 0$ .

## 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. Arreco, 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. Martínez. [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] C. Büskens and D. Wassel. [The ESA NLP solver WORHP](#). In G. Fasano and J. D. Pintér, editors, *Modeling and Optimization in Space Engineering*, volume 73 of *Springer Optimization and Its Applications*, pages 85–110. Springer Verlag, Heidelberg, Berlin, New York, 2013.
- [15] A. Buttari and A. Montoison. [QRmumps.jl: A Julia interface to qr\\_mumps](#). <https://github.com/JuliaSmoothOptimizers/QRmumps.jl>, April 2021.
- [16] 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.
- [17] C. Charalambous. [A lower bound for the controlling parameters of the exact penalty functions](#). *Math. Program.*, 15:278–290, 1978.
- [18] 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.
- [19] T. F. Coleman and A. R. Conn. [Second-order conditions for an exact penalty function](#). *Math. Program.*, 19:178–185, 1980.
- [20] 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.
- [21] 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.
- [22] 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.
- [23] Y. Diouane, M. L. Habiboullah, and D. Orban. [A proximal modified quasi-Newton method for nonsmooth regularized optimization](#). Cahier du GERAD G-2024-64, GERAD, Montréal, QC, Canada, 2024. To appear in *SIAM J. Optim.*
- [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>, 2024.

- [//github.com/JuliaSmoothOptimizers/Percival.jl](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  $\ell_1$ -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] L. R. Huang and K. F. Ng. **Second-order necessary and sufficient conditions in nonsmooth optimization**. *Math. Program.*, 66:379–402, 1994.
- [36] R. Kuhlmann and C. Büskens. **A primal-dual augmented Lagrangian penalty-interior-point filter line search algorithm**. *Math. Oper. Res.*, 87(3):451–483, 2014.
- [37] G. Leconte and D. Orban. **Complexity of trust-region methods with potentially unbounded hessian approximations for smooth and nonsmooth optimization**. Cahier du GERAD G-2023-65, GERAD, Montréal, QC, Canada, 2025. To appear in *Math. Program.*
- [38] 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.
- [39] P. L. Lions and B. Mercier. **Splitting algorithms for the sum of two nonlinear operators**. *SIAM J. Numer. Anal.*, 16:964–979, 1979.
- [40] J. J. Moré and D. C. Sorensen. **Computing a trust region step**. *SIAM J. Sci. and Statist. Comput.*, 4:553–572, 1983.
- [41] B. A. Murtagh and M. A. Saunders. **Large-scale linearly constrained optimization**. *Math. Program.*, 14:41–72, 1978.
- [42] 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.
- [43] D. Orban, A. Soares Siqueira, and contributors. **CUTEst.jl: Julia’s CUTEst interface**. <https://github.com/JuliaSmoothOptimizers/CUTEst.jl>, 2023.
- [44] D. Orban, A. Soares Siqueira, and contributors. **NLPModelsIpopt.jl: A thin IPOPT wrapper for NLPModels**. <https://github.com/JuliaSmoothOptimizers/NLPModelsIpopt.jl>, 2023.
- [45] N. Parikh and S. Boyd. **Proximal algorithms**. *Foundations and Trends in Optimization*, 1(3): 127–239, 2014.
- [46] T. Pietrzykowski. **An exact potential method for constrained maxima**. *SIAM J. Numer. Anal.*, 6(2):299–304, 1969.
- [47] 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>.
- [48] M. J. D. Powell. **Algorithms for nonlinear constraints that use lagrangian functions**. *Math. Program.*, 14:224–248, 1978.
- [49] R. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer Verlag, 1998.
- [50] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.
- [51] R. T. Rockafellar. **A dual approach to solving nonlinear programming problems by unconstrained optimization**. *Math. Program.*, 3:354–373, 1973.
- [52] R. T. Rockafellar. **Augmented Lagrangians and applications of the proximal point algorithm in convex programming**. *Math. Oper. Res.*, 1(2):97–116, 1973.

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