SIAM J. SCI. COMPUT.
Vol. 42, No. 3, pp. A1809–A1835




# IMPLEMENTING A SMOOTH EXACT PENALTY FUNCTION FOR EQUALITY-CONSTRAINED NONLINEAR OPTIMIZATION*

RON ESTRIN†, MICHAEL P. FRIEDLANDER‡, DOMINIQUE ORBAN§, AND MICHAEL A. SAUNDERS¶

*Dedicated to Roger Fletcher*

**Abstract.** We develop a general equality-constrained nonlinear optimization algorithm based on a smooth penalty function proposed by Fletcher in 1970. Although it was historically considered to be computationally prohibitive in practice, we demonstrate that the computational kernels required are no more expensive than other widely accepted methods for nonlinear optimization. The main kernel required to evaluate the penalty function and its derivatives is solving a structured linear system. We show how to solve this system efficiently by storing a single factorization at each iteration when the matrices are available explicitly. We further show how to adapt the penalty function to the class of factorization-free algorithms by solving the linear system iteratively. The penalty function therefore has promise when the linear system can be solved efficiently, e.g., for PDE-constrained optimization problems where efficient preconditioners exist. We discuss extensions including handling simple constraints explicitly, regularizing the penalty function, and inexact evaluation of the penalty function and its gradients. We demonstrate the merits of the approach and its various features on some nonlinear programs from a standard test set, and some PDE-constrained optimization problems.

**Key words.** nonlinear programming, exact penalty, smooth penalty, factorization-free, iterative methods, trust-region

**AMS subject classifications.** 90C30, 65K05, 90C06, 90C46

**DOI.** 10.1137/19M1238265

## 1. Introduction.

We consider a penalty function approach for solving general equality-constrained nonlinear optimization problems

$$\text{(NP)} \qquad \underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; f(x) \;\; \text{subject to} \;\; c(x) = 0 \;:\; y,$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^m$ are smooth functions ($m \le n$), and $y \in \mathbb{R}^m$ is the vector of Lagrange multipliers. A smooth exact penalty function $\phi_\sigma$ is used to eliminate the constraints $c(x) = 0$. The penalty function is the Lagrangian $L(x, y) = f(x) - c(x)^T y$ evaluated at $y = y_\sigma(x)$ (defined in (2.2a)) treated as a function of $x$ depending on a parameter $\sigma > 0$. Hence, the penalty function depends only on the primal variables $x$. It was first proposed by Fletcher (1970) for (NP). A long-held view is that Fletcher's penalty function is not practical because it is costly to compute

*Submitted to the journal's Methods and Algorithms for Scientific Computing section January 11, 2019; accepted for publication (in revised form) March 20, 2020; published electronically June 25, 2020.

https://doi.org/10.1137/19M1238265

**Funding:** The work of the second author was partially supported by the Office of Naval Research through grant N00014-17-1-2009. The work of the third author was supported by NSERC Discovery Grant 299010-04. The work of the fourth author was partially supported by the National Institute of General Medical Sciences of the National Institutes of Health through grant U01GM102098.

†Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305-4042 (restrin@stanford.edu).

‡Department of Computer Science, University of British Columbia, Vancouver V6T 1Z4, BC, Canada (mpf@cs.ubc.ca).

§GERAD and Department of Mathematics and Industrial Engineering, École Polytechnique, Montréal H3C 3A7, QC, Canada (dominique.orban@gerad.ca).

¶Systems Optimization Laboratory, Department of Management Science and Engineering, Stanford University, Stanford, CA 94305-4121 (saunders@stanford.edu).

(Bertsekas, 1975; Conn et al., 2000; Nocedal and Wright, 2006). In particular, Nocedal and Wright (2006, p. 436) warn that "although this merit function has some interesting theoretical properties, it has practical limitations. . . ." Our aim is to challenge that notion and to demonstrate that the computational kernels are no more expensive than other widely accepted methods for nonlinear optimization, such as sequential quadratic programming.

The penalty function is *exact* because local minimizers of (NP) are minimizers of the penalty function for all values of $\sigma$ larger than a finite threshold $\sigma^*$. The main computational kernel for evaluating the penalty function and its derivatives is the solution of a certain saddle-point system; see (4.8). If the system matrix is available explicitly, we show how to factorize it once and reuse the factors to evaluate the penalty function and its derivatives. We also adapt the penalty function for *factorization-free* optimization by solving the linear system iteratively. This makes the penalty function particularly applicable for certain problem classes, such as PDE-constrained optimization problems when good preconditioners exist; see section 9.

**1.1. Related work on penalty functions.** Penalty functions have long been used to solve constrained problems by transforming them into unconstrained problems that penalize violations of feasibility. We provide a brief overview of common penalty methods and their relation to Fletcher's penalty $\phi_\sigma(x)$. More detail is given by di Pillo and Grippo (1984), Conn et al. (2000), and Nocedal and Wright (2006).

The simplest example is the quadratic penalty function (Courant, 1943), which removes the nonlinear constraints by adding $\frac{1}{2}\rho\|c(x)\|^2$ to the objective (for some $\rho > 0$). There are two main drawbacks: a sequence of optimization subproblems must be solved with increasing $\rho$, and a feasible solution is obtained only when $\rho \to \infty$. As $\rho$ increases, the subproblems become increasingly difficult to solve.

An alternative to smooth nonexact penalty functions is an exact nonsmooth function such as the 1-norm penalty $\rho\|c(x)\|_1$ (Pietrzykowski, 1969; Fletcher, 1985). However, only nonsmooth optimization methods apply, which typically exhibit slower convergence. Maratos (1978) further noted that nonsmooth merit functions may reject steps and prevent quadratic convergence.

Another distinct approach is the class of augmented Lagrangian methods, independently introduced by Hestenes (1969) and Powell (1969). These minimize a sequence of augmented Lagrangians, $L_{\rho_k}(x, y_k) = L(x, y_k) + \frac{1}{2}\rho_k\|c(x)\|^2$. When $y_k$ is optimal, $L_{\rho_k}(x, y_k)$ is exact for sufficiently large $\rho_k$, thus avoiding the stability issues of the quadratic penalty. However, a sequence of subproblems must be solved to drive $y_k$ to optimality.

Although these penalty functions have often been successful in practice, in light of their drawbacks, a class of smooth exact penalty functions has been explored (di Pillo and Grippo, 1984; Zavala and Anitescu, 2014). With smooth exact penalty functions, constrained optimization problems such as (NP) can be replaced by a single smooth unconstrained optimization problem (provided the penalty parameter is sufficiently large). Approximate second-order methods can be applied to obtain at least superlinear local convergence. These methods are variations of minimizing the augmented Lagrangian, where either the multipliers are parametrized in terms of $x$, or they are kept independent and the gradient of the Lagrangian is penalized. The price for smoothness (as we find for $\phi_\sigma$) is that a derivative of the penalty function requires a higher-order derivative from the original problem data. That is, evaluating $\phi_\sigma$ requires $\nabla f$ and $\nabla c$; $\nabla\phi_\sigma$ requires $\nabla^2 f$ and $\nabla^2 c_i$; and so on. The third derivative terms are typically discarded during computation, but it can be shown that

superlinear convergence is retained (Fletcher, 1973, Theorem 2).

Fletcher (1970) introduced the class of smooth exact penalty functions from which $\phi_\sigma$ originates. Extensions and variations of this class have been explored by several authors, whose contributions are described by Conn et al. (2000, section 14.6). However, Fletcher (1970) envisioned his method being applied to small problems and assumed "the matrices in the problem are nonsparse." Further, most developments surrounding this method focused on linesearch schemes that require computing an explicit Hessian approximation and using it to compute a Newton direction. One of our goals is to show how to adapt the method to large-scale problems by taking advantage of computational advances made since Fletcher's proposal. Improved sparse matrix factorizations and iterative methods for solving linear systems, and modern Newton-CG trust-region methods (Ph. L. Toint, 1981; Steihaug, 1983), play a key role in the efficient implementation of his penalty function. We also show how regularization can be used to accommodate certain constraint degeneracy, and we explain how to take advantage of inexact evaluations of functions and gradients.

**1.2. Outline.** We introduce the penalty function in section 2 and describe its properties and derivatives in section 3. In section 4 we discuss options for efficiently evaluating the penalty function and its derivatives. We then discuss modifications of the penalty function in sections 5–6 to take advantage of linear constraints and to regularize the penalty function if the constraint Jacobian is rank-deficient. In some applications, it may be necessary to solve large linear systems inexactly, and we show in section 7 how the resulting imprecision can be accommodated. Other practical matters are described in section 8. We apply the penalty function to several optimization problems in section 9 and conclude with future research directions in section 10.

**2. The penalty function for equality constraints.** For (NP), Fletcher's penalty function is

$$\phi_\sigma(x) := f(x) - c(x)^T y_\sigma(x), \tag{2.1}$$

where $y_\sigma(x)$ are Lagrange multiplier estimates defined with other items as

$$y_\sigma(x) := \arg\min_y \left\{ \tfrac{1}{2}\|A(x)y - g(x)\|_2^2 + \sigma c(x)^T y \right\}, \qquad g(x) := \nabla f(x), \tag{2.2a}$$

$$A(x) := \nabla c(x) = \begin{bmatrix} g_1(x) & \dots & g_m(x) \end{bmatrix}, \qquad g_i(x) := \nabla c_i(x), \tag{2.2b}$$

$$Y_\sigma(x) := \nabla y_\sigma(x). \tag{2.2c}$$

Note that $A$ and $Y_\sigma$ are $n$-by-$m$ matrices. The form of $y_\sigma(x)$ is reminiscent of the variable-projection algorithm of Golub and Pereyra (1973) for separable nonlinear least-squares problems.

We assume that (NP) satisfies some variation of the following conditions:

- (A1) $f$ and $c$ are twice continuously differentiable and either
  - (A1a) have Lipschitz second-derivatives or
  - (A1b) are three-times continuously differentiable.
- (A2) The linear independence constraint qualification (LICQ) is satisfied at
  - (A2a) stationary points of (NP) or
  - (A2b) all $n$-vectors $x$.

(LICQ is satisfied at a point $x$ if the vectors $\{\nabla c_i(x)\}_{i=1}^m$ are linearly independent.)

- (A3) The problem is feasible; i.e., there exists $x$ such that $c(x) = 0$.

Assumption (A1b) ensures that $\phi_\sigma$ has two continuous derivatives and is typical for smooth exact penalty functions (Bertsekas, 1982, Proposition 4.16). However, we use at most two derivatives of $f$ and $c$ throughout. We typically assume (A1b) to simplify the discussion, but this assumption can often be weakened to (A1a). We also initially assume that (NP) satisfies (A2b) so that $Y_\sigma(x)$ and $y_\sigma(x)$ are uniquely defined. We relax this assumption to (A2a) in section 6.

**2.1. Notation.** Denote $x^*$ as a local minimizer of (NP), with corresponding dual solution $y^*$. Let $H(x) = \nabla^2 f(x)$, $H_i(x) = \nabla^2 c_i(x)$, and define

$$g_L(x,y) := g(x) - A(x)y, \qquad g_\sigma(x) := g_L(x, y_\sigma(x)), \tag{2.3a}$$

$$H_L(x,y) := H(x) - \sum_{i=1}^m y_i H_i(x), \qquad H_\sigma(x) := H_L(x, y_\sigma(x)) \tag{2.3b}$$

as the gradient and Hessian of $L(x,y)$ evaluated at $x$ and $y$, or evaluated at $y_\sigma(x)$. We also define the matrix operators

$$S(x,v) := \nabla_x [A(x)^T v] = \nabla_x \begin{bmatrix} g_1(x)^T v \\ \vdots \\ g_m(x)^T v \end{bmatrix} = \begin{bmatrix} v^T H_1(x) \\ \vdots \\ v^T H_m(x) \end{bmatrix},$$

$$T(x,w) := \nabla_x [A(x)w] = \nabla_x \left[ \sum_{i=1}^m w_i g_i(x) \right] = \sum_{i=1}^m w_i H_i(x),$$

where $v \in \mathbb{R}^n$, $w \in \mathbb{R}^m$, and $T(x,w)$ is a symmetric matrix. The operation of multiplying the adjoint of $S$ with a vector $w$ is described by

$$S(x,v)^T w = \left[ \sum_{i=1}^m w_i H_i(x) \right] v = T(x,w)v = T(x,w)^T v\,.$$

If $A(x)$ has full rank $m$, the operators

$$P(x) := A(x)\big(A(x)^T A(x)\big)^{-1} A(x)^T \quad \text{and} \quad \bar{P}(x) := I - P(x)$$

define, respectively, orthogonal projectors onto range$(A(x))$ and its complement. More generally, for a matrix $M$, respectively define $P_M$ and $\bar{P}_M$ as the orthogonal projectors onto range$(M)$ and ker$(M)$. We define $M^\dagger$ as the Moore–Penrose pseudoinverse, where $M^\dagger = (M^T M)^{-1} M^T$ if $M$ has full column rank.

Let $\lambda_{\min}(M)$ denote the smallest eigenvalue of a square matrix $M$, and let $\sigma_{\min}(M)$ denote the smallest singular value for a general matrix $M$. Unless otherwise indicated, $\|\cdot\|$ is the 2-norm for vectors and matrices. For $M$ positive definite, $\|u\|_M^2 = u^T M u$ is the energy norm. Define $\mathbb{1}$ as the vector of all ones.

**3. Properties of the penalty function.** We show how the penalty function $\phi_\sigma(x)$ naturally expresses the optimality conditions of (NP). We also give explicit expressions for the threshold value of the penalty parameter $\sigma$.

**3.1. Derivatives of the penalty function.** The gradient and Hessian of $\phi_\sigma$ may be written as

$$\nabla \phi_\sigma(x) = g_\sigma(x) - Y_\sigma(x) c(x), \tag{3.1a}$$

$$\nabla^2 \phi_\sigma(x) = H_\sigma(x) - A(x) Y_\sigma(x)^T - Y_\sigma(x) A(x)^T - \nabla_x \left[ Y_\sigma(x) c \right], \tag{3.1b}$$

where the last term $\nabla_x[Y_\sigma(x)c]$ purposely drops the argument on $c$ to emphasize that this gradient is made on the product $Y_\sigma(x)c$ with $c := c(x)$ held fixed. This term involves third derivatives of $f$ and $c$, and as we shall see, it is convenient and computationally efficient to ignore it. We leave it unexpanded.

**3.2. Optimality conditions.** The penalty function $\phi_\sigma$ is closely related to the Lagrangian $L(x, y)$ associated with (NP). To make this connection clear, we define the Karush–Kuhn–Tucker (KKT) optimality conditions for (NP) in terms of formulas related to $\phi_\sigma$ and its derivatives. From the definition of $\phi_\sigma$ and $y_\sigma$ and the derivatives (3.1), the following definitions are equivalent to the KKT conditions.

DEFINITION 1 (first-order KKT point). *A point $x^*$ is a first-order KKT point of* (NP) *if for any $\sigma \geq 0$ the following hold:*

$$c(x^*) = 0, \tag{3.2a}$$

$$\nabla\phi_\sigma(x^*) = 0. \tag{3.2b}$$

*The Lagrange multipliers associated with $x^*$ are $y^* := y_\sigma(x^*)$.*

DEFINITION 2 (second-order KKT point). *The first-order KKT point $x^*$ satisfies the second-order necessary KKT condition for* (NP) *if for any $\sigma \geq 0$,*

$$p^T\nabla^2\phi_\sigma(x^*)p \geq 0 \quad \textit{for all } p \textit{ such that } A(x^*)^Tp = 0,$$

*i.e., $\bar{P}(x^*)\nabla^2\phi_\sigma(x^*)\bar{P}(x^*) \succeq 0$. The condition is sufficient if the inequality is strict.*

The second-order KKT condition says that at $x^*$, $\phi_\sigma$ has nonnegative curvature along directions in the tangent space of the constraints. However, at $x^*$, increasing $\sigma$ will increase curvature along the normal cone of the feasible set. We derive a threshold value for $\sigma$ that causes $\phi_\sigma$ to have nonnegative curvature at a second-order KKT point $x^*$, as well as a condition on $\sigma$ that ensures stationary points of $\phi_\sigma$ are primal feasible. For a given first- or second-order KKT pair $(x^*, y^*)$ of (NP), we define

$$\sigma^* := \tfrac{1}{2}\lambda^+_{\max}\left(P(x^*)H_{\scriptscriptstyle L}(x^*, y^*)P(x^*)\right), \tag{3.3}$$

where $\lambda^+_{\max}(\cdot) = \max\{\lambda_{\max}(\cdot), 0\}$.

LEMMA 3. *If $c(x) \in \text{range}(A(x)^T)$, then $y_\sigma(x)$ satisfies*

$$A(x)^TA(x)y_\sigma(x) = A(x)^Tg(x) - \sigma c(x). \tag{3.4}$$

*Further, if $A(x)$ has full rank, then*

$$A(x)^TA(x)Y_\sigma(x)^T = A(x)^T[H_\sigma(x) - \sigma I] + S(x, g_\sigma(x)). \tag{3.5}$$

*Proof.* For any $x$, the necessary and sufficient optimality conditions for (2.2a) give (3.4). By differentiating both sides of (3.4), we obtain

$$S(x, A(x)y_\sigma(x)) + A(x)^T\left[T(x, y_\sigma(x)) + A(x)Y_\sigma(x)^T\right] = S(x, g(x)) + A(x)^T[H(x) - \sigma I].$$

From definitions (2.3), we obtain (3.5). ∎

THEOREM 4 (threshold penalty value). *Suppose $\nabla\phi_\sigma(\bar{x}) = 0$ for some $\bar{x}$, and let $x_1^*$ and $x_2^*$ be a first-order and a second-order necessary KKT point, respectively, for* (NP). *Let $\sigma^*$ be defined as in* (3.3). *Then*

$$\text{(3.6a)} \qquad \sigma > \|A(\bar{x})^T Y_\sigma(\bar{x})\| \implies g(\bar{x}) = A(\bar{x}) y_\sigma(\bar{x}), \quad c(\bar{x}) = 0;$$

$$\text{(3.6b)} \qquad \sigma \geq \|A(x_1^*) Y_\sigma(x_1^*)^T\| \implies \sigma \geq \sigma^*;$$

$$\text{(3.6c)} \qquad \nabla^2\phi_\sigma(x_2^*) \succeq 0 \iff \sigma \geq \bar{\sigma} := \tfrac{1}{2}\lambda_{\max}\left(P(x_2^*) H_L(x_2^*, y^*) P(x_2^*)\right).$$

*If $x_2^*$ is second-order sufficient, then the inequalities in* (3.6c) *hold strictly. Observe that $\sigma^* = \max\{\bar{\sigma}, 0\}$ and that $\bar{\sigma}$ could be negative.*

*Proof.* We prove (3.6a), (3.6c), and (3.6b) in order.

*Proof of* (3.6a). The condition $\nabla\phi_\sigma(\bar{x}) = 0$ implies that $y_\sigma(\bar{x})$ is well defined and $c(\bar{x}) \in \text{range}(A(\bar{x})^T)$, so that

$$g(\bar{x}) = A(\bar{x}) y_\sigma(\bar{x}) + Y_\sigma(\bar{x}) c(\bar{x}).$$

Substituting (3.4) evaluated at $\bar{x}$ into this equation yields, after simplifying,

$$A(\bar{x})^T Y_\sigma(\bar{x}) c(\bar{x}) = \sigma c(\bar{x}).$$

Taking norms of both sides and using submultiplicativity gives the inequality $\sigma\|c(\bar{x})\| \leq \|A(\bar{x})^T Y_\sigma(\bar{x})\| \, \|c(\bar{x})\|$, which immediately implies that $c(\bar{x}) = 0$. The condition $\nabla\phi_\sigma(\bar{x}) = 0$ then becomes $g_\sigma(\bar{x}) = 0$.

*Proof of* (3.6c). Because $x_2^*$ satisfies (3.2), we have $g_\sigma(x_2^*) = 0$ and $y^* = y_\sigma(x_2^*)$, independently of $\sigma$. It follows from (3.5), $H_L(x_2^*, y^*) = H_\sigma(x_2^*)$, $S(x_2^*, g_\sigma(x_2^*)) = 0$, and the definition of the projector $P = P(x_2^*)$ that

$$\begin{aligned} A(x_2^*) Y_\sigma(x_2^*)^T &= A(x_2^*) \left(A(x_2^*)^T A(x_2^*)\right)^{-1} A(x_2^*)^T [H_\sigma(x_2^*) - \sigma I] \\ &= P(H_L(x_2^*, y^*) - \sigma I). \end{aligned} \tag{3.7}$$

We substitute this equation into (3.1b) and use the relation $P + \bar{P} = I$ to obtain

$$\begin{aligned} \nabla^2\phi_\sigma(x_2^*) &= H_L(x_2^*, y^*) - P H_L(x_2^*, y^*) - H_L(x_2^*, y^*) P + 2\sigma P \\ &= \bar{P} H_L(x_2^*, y^*) \bar{P} - P H_L(x_2^*, y^*) P + 2\sigma P. \end{aligned}$$

Note that $\bar{P} H_L(x_2^*, y^*) \bar{P} \succeq 0$ because $x_2^*$ is a second-order KKT point. As $P$ and $\bar{P}$ have orthogonal ranges, $\sigma$ must be large enough to ensure $2\sigma P - P H_L(x_2^*, y^*) P \succeq 0$, which is equivalent to $\sigma \geq \bar{\sigma}$.

*Proof of* (3.6b). With $x_1^*$ in (3.7), $y^* = y_\sigma(x_1^*)$, and the properties of $P$, we have

$$\begin{aligned} \sigma \geq \|A(x_1^*) Y_\sigma(x_1^*)^T\| &= \|P(H_L(x_1^*, y^*) - \sigma I)\| \\ &\geq \|P(H_L(x_1^*, y^*) - \sigma I) P\| \\ &\geq \|P H_L(x_1^*, y^*) P\| - \sigma\|P\| \geq 2\sigma^* - \sigma. \end{aligned}$$

Thus, $\sigma \geq \sigma^*$ as required. ∎

According to (3.6c), if $x^*$ is a second-order KKT point, there exists a threshold value $\bar{\sigma}$ such that $\phi_\sigma$ has nonnegative curvature for $\sigma \geq \bar{\sigma}$. As penalty parameters are typically nonnegative, we treat $\sigma^* = \max\{\bar{\sigma}, 0\}$ as the threshold. Unfortunately, as for many exact penalty functions, Theorem 4 allows the possibility of stationary points of $\phi_\sigma(x)$ that are not feasible points of (NP); for an example, see Appendix A.1. However, we rarely encounter this in practice with feasible problems, and minimizers of $\phi_\sigma(x)$ usually correspond to feasible (and therefore optimal) points of (NP).

**3.3. Additional quadratic penalty.** In light of Theorem 4, it is somewhat unsatisfying that local minimizers of $\phi_\sigma(x)$ might not be local minimizers of (NP). We may add a quadratic penalty term to promote feasibility, and under mild conditions ensure that minimizers of $\phi_\sigma$ are KKT points of (NP). Like Fletcher (1970), we define

$$\phi_{\sigma,\rho}(x) := \phi_\sigma(x) + \tfrac{1}{2}\rho\|c(x)\|^2 = f(x) - [y_\sigma(x) - \tfrac{1}{2}\rho c(x)]^T c(x). \tag{3.8}$$

The multiplier estimates are now shifted by the constraint violation, similar to an augmented Lagrangian. All expressions for the derivatives follow as before with an additional term from the quadratic penalty.

THEOREM 5 (threshold penalty value for quadratic penalty). *Let $\mathcal{S} \subset \mathbb{R}^n$ be a compact set, and suppose that $\sigma_{\min}(A(x)) \geq \lambda > 0$ for all $x \in \mathcal{S}$. Then for any $\sigma \geq 0$ there exists $\rho^*(\sigma) > 0$ such that for all $\rho > \rho^*(\sigma)$, if $\nabla\phi_{\sigma,\rho}(\bar{x}) = 0$ and $\bar{x} \in \mathcal{S}$, then $\bar{x}$ is a first-order KKT point for* (NP).

*Proof.* The condition $\nabla\phi_{\sigma,\rho}(\bar{x}) = 0$ implies that

$$g(\bar{x}) - A(\bar{x})y_\sigma(\bar{x}) - Y_\sigma(\bar{x})c(\bar{x}) = \rho A(\bar{x})c(\bar{x}).$$

We premultiply with $A(\bar{x})^T$ and use (3.4) to obtain

$$\left(\sigma I - A(\bar{x})^T Y_\sigma(\bar{x})\right) c(\bar{x}) = \rho A(\bar{x})^T A(\bar{x})c(\bar{x}). \tag{3.9}$$

The left-hand side of (3.9) is a continuous matrix function with finite supremum $R(\sigma) := \sup_{x\in\mathcal{S}} \|\sigma I - A(x)^T Y_\sigma(x)\|$ defined over the compact set $\mathcal{S}$. We now define $\rho^*(\sigma) := R(\sigma)/\lambda^2$, so that for $\rho > \rho^*(\sigma)$,

$$\begin{aligned} R(\sigma)\|c(\bar{x})\| &\geq \|\sigma I - A(\bar{x})^T Y_\sigma(\bar{x})\| \cdot \|c(\bar{x})\| \\ &\geq \|\left(\sigma I - A(\bar{x})^T Y_\sigma(\bar{x})\right) c(\bar{x})\| \\ &= \rho\|A(\bar{x})^T A(\bar{x})c(\bar{x})\| \geq \rho\lambda^2\|c(\bar{x})\|. \end{aligned}$$

The above inequality only holds when $c(\bar{x}) = 0$ because $\rho\lambda^2 > R(\sigma)$, so $\bar{x}$ is feasible for (NP). Because $c(\bar{x}) = 0$ and $\nabla\phi_\sigma(\bar{x}) = \nabla\phi_{\sigma,\rho}(\bar{x}) = 0$, $\bar{x}$ is a first-order KKT point. ∎

We briefly consider the case $\sigma = 0$ and $\rho > 0$. The threshold value to ensure positive semidefiniteness of $\nabla^2\phi_{\sigma,\rho}$ at a second-order KKT pair $(x^*, y^*)$ to (NP) is

$$\rho^* = \lambda^+_{\max}\left(A(x^*)^\dagger H_L(x^*, y^*)\left(A(x^*)^\dagger\right)^T\right).$$

Fletcher (1970) gives an almost identical (but slightly looser) expression for $\rho^*$. This threshold parameter is more difficult to interpret in terms of the problem data compared to $\sigma^*$ due to the pseudoinverse. We give a theorem analogous to Theorem 4.

THEOREM 6. *Suppose $\sigma = 0$ and $\rho \geq 0$. Let $\nabla\phi_{\sigma,\rho}(\bar{x}) = 0$ for some $\bar{x}$, and let $x^*$ be a second-order necessary KKT point for* (NP). *Then*

$$\rho > \|A(\bar{x})^\dagger Y_\sigma(\bar{x})\| \implies g(\bar{x}) = A(\bar{x})y_\sigma(\bar{x}), \quad c(\bar{x}) = 0; \tag{3.10a}$$

$$\nabla^2\phi_\sigma(x^*) \succeq 0 \iff \rho \geq \bar{\rho} := \lambda_{\max}(A(x^*)^\dagger H_L(x^*, y^*)\left(A(x^*)^\dagger\right)^T). \tag{3.10b}$$

*If $x^*$ is second-order sufficient, the inequalities in* (3.10b) *hold strictly.*

*Proof.* The proof is identical to that of Theorem 4. ∎

Using $\rho > 0$ can help cases where attaining feasibility is problematic for moderate values of $\sigma$. For simplicity we let $\rho = 0$ from now on, because it is trivial to evaluate $\phi_{\sigma,\rho}$ and its derivatives if one can compute $\phi_\sigma$.

**3.4. Scale invariance.** Note that $\phi_\sigma$ is invariant under diagonal scaling of the constraints; i.e., if $c(x)$ is replaced by $Dc(x)$ for some diagonal matrix $D$, then $\phi_\sigma$ is unchanged. It is an attractive property for $\phi_\sigma$ and $\sigma^*$ to be independent of some choices in model formulation, like the Lagrangian. However, $\phi_{\sigma,\rho}$ with $\rho > 0$ is not scale invariant, like the augmented Lagrangian, because of the quadratic term. Therefore, constraint scaling is an important consideration if $\phi_{\sigma,\rho}$ is to be used.

**4. Evaluating the penalty function.** The main challenge in evaluating $\phi_\sigma$ and its gradient is solving the shifted least-squares problem (2.2a) in order to compute $y_\sigma(x)$, and computing the Jacobian $Y_\sigma(x)$. Below we show it is possible to compute products $Y_\sigma(x)v$ and $Y_\sigma(x)^T u$ by solving structured linear systems involving the matrix used to compute $y_\sigma(x)$. If direct methods are used, a single factorization that gives the solution (2.2a) is sufficient for all products.

For this section, it is convenient to drop the arguments on the various functions and assume they are all evaluated at a point $x$ for some parameter $\sigma$. For example, $y_\sigma = y_\sigma(x),\ A = A(x),\ Y_\sigma = Y_\sigma(x),\ H_\sigma = H_\sigma(x),\ S_\sigma = S(x, g_\sigma(x))$, etc. We also express (3.5) using the shorthand notation

$$A^T A Y_\sigma^T = A^T[H_\sigma - \sigma I] + S_\sigma. \tag{4.1}$$

We first describe how to compute products $Y_\sigma u$ and $Y_\sigma^T v$ and then describe how to put those pieces together to evaluate the penalty function and its derivatives.

**4.1. Computing the product $Y_\sigma u$.** For a given $u \in \mathbb{R}^m$, we premultiply (4.1) by $u^T(A^TA)^{-1}$ to obtain

$$\begin{aligned} Y_\sigma u &= [H_\sigma - \sigma I]A(A^TA)^{-1}u + S_\sigma^T(A^TA)^{-1}u \\ &= [H_\sigma - \sigma I]v - S_\sigma^T w, \end{aligned}$$

where we define $w = -(A^TA)^{-1}u$ and $v = -Aw$. Observe that $w$ and $v$ solve the system

$$\begin{bmatrix} I & A \\ A^T & \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ u \end{bmatrix}. \tag{4.2}$$

Algorithm 1 formalizes the process.

**Algorithm 1.** Computing the matrix-vector product $Y_\sigma u$.

1: $(v, w) \leftarrow$ solution of (4.2)
2: **return** $[H_\sigma - \sigma I]v - S_\sigma^T w$

**4.2. Computing the product $Y_\sigma^T v$.** Multiplying both sides of (4.1) on the right by $v$ gives

$$A^TA(Y_\sigma^T v) = A^T([H_\sigma - \sigma I]v) + (S_\sigma v).$$

The required product $u = Y_\sigma^T v$ is in the solution of the system

$$\begin{bmatrix} I & A \\ A^T & \end{bmatrix} \begin{bmatrix} r \\ u \end{bmatrix} = \begin{bmatrix} [H_\sigma - \sigma I]v \\ -S_\sigma v \end{bmatrix}. \tag{4.3}$$

Algorithm 2 formalizes the process.

**Algorithm 2.** Computing the matrix-vector product $Y_\sigma^T v$.

1: Evaluate $[H_\sigma - \sigma I]v$ and $S_\sigma v$
2: $(r, u) \leftarrow$ solution of (4.3)
3: **return** $u$

**4.3. Computing multipliers and first derivatives.** The multiplier estimates $y_\sigma$ can be obtained from the optimality conditions for (2.2a):

$$\begin{bmatrix} I & A \\ A^T & \end{bmatrix} \begin{bmatrix} g_\sigma \\ y_\sigma \end{bmatrix} = \begin{bmatrix} g \\ \sigma c \end{bmatrix}, \tag{4.4}$$

which also gives $g_\sigma$. Algorithm 1 then gives $Y_\sigma c$ and hence $\nabla\phi_\sigma$ in (3.1a).

Observe that we can reorder operations to take advantage of specialized solvers. Consider the pair of systems

$$\begin{bmatrix} I & A \\ A^T & \end{bmatrix} \begin{bmatrix} d \\ y \end{bmatrix} = \begin{bmatrix} g \\ 0 \end{bmatrix} \qquad \text{and} \qquad \begin{bmatrix} I & A \\ A^T & \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ c \end{bmatrix}. \tag{4.5}$$

We have $g_\sigma = d + \sigma v$ and $y_\sigma = y + \sigma w$, while the computation of $Y_\sigma c$ is unchanged. The systems in (4.5) correspond to pure least-squares and least-norm problems, respectively. Specially tailored solvers may be used to improve efficiency or accuracy. This is further explored in section 4.5.

**4.4. Computing second derivatives.** We can approximate $\nabla^2\phi_\sigma$ using (3.1b) and (3.5) in two ways according to

$$\begin{aligned} \nabla^2\phi_\sigma \approx B_1 &:= H_\sigma - AY_\sigma^T - Y_\sigma A^T \\ &= H_\sigma - \widetilde{P}H_\sigma - H_\sigma\widetilde{P} + 2\sigma\widetilde{P} - A(A^TA)^{-1}S_\sigma - S_\sigma^T(A^TA)^{-1}A, \end{aligned} \tag{4.6a}$$

$$\nabla^2\phi_\sigma \approx B_2 := H_\sigma - \widetilde{P}H_\sigma - H_\sigma\widetilde{P} + 2\sigma\widetilde{P}, \tag{4.6b}$$

where $\widetilde{P} = A(A^TA)^{-1}A^T$. Note that $\widetilde{P} = P_A$ here, but this changes when regularization is used; see section 6. The first approximation ignores $\nabla[Y_\sigma(x)c]$ in (3.1b), while the second ignores $S_\sigma = S(x, g_\sigma(x))$. Because we expect $c(x) \to 0$ and $g_\sigma(x) \to 0$, $B_1$ and $B_2$ are similar to Gauss–Newton approximations to $\nabla^2\phi_\sigma(x)$, and as Fletcher (1973, Theorem 2) shows, using them in a Newton-like scheme is sufficient for quadratic convergence if (A1a) is satisfied.

Because $\widetilde{P}$ is a projection on range$(A)$, we can compute products $\widetilde{P}u$ by solving

$$\begin{bmatrix} I & A \\ A^T & \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} u \\ 0 \end{bmatrix} \tag{4.7}$$

and setting $\widetilde{P}u \leftarrow u - p$. Note that with regularization, the $(2,2)$ block of this system is modified and $\widetilde{P}$ is no longer a projection; see section 6.

The approximations (4.6a) and (4.6b) trade Hessian accuracy for computational efficiency. If the operator $S(x, v)$ is not immediately available (or not efficiently implemented), it may be avoided. Using $B_2$ requires only least-square solves, which allows us to apply specialized solvers (e.g., LSQR (Paige and Saunders, 1982)), which cannot be done when products with $Y_\sigma^T$ are required.

**4.5. Solving the augmented linear system.** We discuss some approaches to solving linear systems of the form

$$\mathcal{K}\begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} w \\ z \end{bmatrix}, \qquad \mathcal{K} := \begin{bmatrix} I & A \\ A^T & -\delta^2 I \end{bmatrix}, \tag{4.8}$$

which have repeatedly appeared in this section. Although $\delta = 0$ so far, we look ahead to regularized systems as they require only minor modification. Let $(p^*, q^*)$ solve (4.8).

Conceptually it is not important how this system is solved as long as it is solved with sufficient accuracy. However, this is the most computationally intensive part of using $\phi_\sigma$. Different solution methods have different advantages and limitations, depending on the size and sparsity of $A$, whether $A$ is available explicitly, and the prescribed solution accuracy.

One option is direct methods: factorize $\mathcal{K}$ once per iteration and use the factors to solve with each right-hand side. Several factorization-based approaches can be used with various advantages and drawbacks; see the supplementary materials for details.

In line with the goal of creating a factorization-free solver for minimizing $\phi_\sigma$, we discuss iterative methods for solving (4.8), particularly Krylov subspace solvers. This approach has two potential advantages: if a good preconditioner $\mathcal{P} \approx A^T A$ is available, then solving (4.8) could be much more efficient than with direct methods, and we can take advantage of solvers using inexact function values, gradients, or Hessian products by solving (4.8) approximately; see Heinkenschloss and Vicente (2001) and Kouri et al. (2014). For example, Pearson et al. (2012) and Simoncini (2012) describe various preconditioners for saddle-point systems arising in PDE-constrained optimization, which are closely related to the augmented systems in (4.8).

When $z = 0$, (4.8) is a (regularized) least-squares problem: $\min_q \|Aq - w\| + \delta\|q\|^2$. We use LSQR (Paige and Saunders, 1982), which ensures that the error in iterates $p_k$ and $q_k$ decreases monotonically at every iteration. (Hestenes and Stiefel (1952) show this for CG, and LSQR is equivalent to CG on the normal equations.) Furthermore, Estrin et al. (2019a) provide a way to compute an upper bound on $\|p^* - p_k\|$ and $\|q^* - q_k\|$ via LSLQ when given an underestimate of $\sigma_{\min}(A\mathcal{P}^{-1/2})$. (Note that the error norm for $q$ depends on the preconditioner.) Further discussion is in section 9.

When $w = 0$, (4.8) is a least-norm problem: $\min_{p,s} \|p\|^2 + \|s\|^2$ s.t. $A^T p + \delta s = z$. We then use CRAIG (Craig, 1955) because it minimizes the error in each Krylov subspace. Given the same underestimate of $\sigma_{\min}(A\mathcal{P}^{-1/2})$, Arioli (2013) and Estrin et al. (2019b) give a way to bound the error norms for $p$ and $q$.

Recall that $\phi_\sigma$ and $\nabla\phi_\sigma$ can be computed by solving only least-squares and least-norm problems (only one of $w$ and $z$ is nonzero at a time). Furthermore, if (4.6b) is used, the remaining solves with $\mathcal{K}$ are least-squares solves. If both $w$ and $z$ are nonzero (for products with $Y_\sigma^T$), we can shift the right-hand side of (4.8) and solve the system

$$\mathcal{K}\begin{bmatrix} \bar{p} \\ q \end{bmatrix} = \begin{bmatrix} 0 \\ z - A^T w \end{bmatrix}, \qquad p = \bar{p} + w.$$

Thus, (4.8) can be solved by CRAIG or LNLQ (Arioli, 2013; Estrin et al., 2019b) or other least-norm solvers.

Although $\mathcal{K}$ is symmetric indefinite, we do not recommend methods such as MINRES or SYMMLQ (Paige and Saunders, 1975). Orban and Arioli (2017) show that if full-space methods are applied directly to $\mathcal{K}$, then every other iteration of the solver makes little progress. However, if solves with $\mathcal{P}$ can only be performed ap-

proximately, it may be necessary to apply flexible variants of nonsymmetric full-space methods to $\mathcal{K}$, such as flexible GMRES (Saad, 1993).

**5. Maintaining explicit constraints.** We consider a variation of (NP) where some of the constraints $c(x)$ are easy to maintain explicitly; for example, some are linear. We can then maintain feasibility for a subset of the constraints, the contours of the $\phi_\sigma$ are simplified, and as we show soon, the threshold penalty parameter $\sigma^*$ is decreased. We discuss the case where some of the constraints are linear, but it is possible to extend the theory to any type of constraint.

Consider the problem

$$\text{(NP-EXP)} \qquad \underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; f(x) \;\; \text{subject to} \;\; c(x) = 0, \;\; B^T x = d,$$

where we have nonlinear constraints $c(x) = 0 \in \mathbb{R}^{m_1}$ and linear constraints $B^T x = d$ with $B \in \mathbb{R}^{n\times m_2}$, so that $m_1 + m_2 = m$. We assume that (NP-EXP) at least satisfies (A2a), so that $B$ has full column rank. We define the penalty function problem to be

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad \phi_\sigma(x) := f(x) - c(x)^T y_\sigma(x) \quad \text{subject to} \quad B^T x = d,$$
$$\begin{bmatrix} y_\sigma(x) \\ w_\sigma(x) \end{bmatrix} := \underset{y,w}{\arg\min}\, \tfrac{1}{2}\|A(x)y + Bw - g(x)\|^2 + \sigma \begin{bmatrix} c(x) \\ B^T x - d \end{bmatrix}^T \begin{bmatrix} y \\ w \end{bmatrix},$$

which is similar to (2.1) except the linear constraints are not penalized in $\phi_\sigma(x)$, and the penalty function is minimized subject to the linear constraints. A possibility is to also penalize the linear constraints while keeping them explicit; however, penalizing the linear constraints in $\phi_\sigma(x)$ introduces additional nonlinearity, and if all constraints are linear, it makes sense that the penalty function reduces to (NP-EXP).

For a given first- or second-order KKT solution $(x^*, y^*)$, the threshold penalty parameter becomes

$$\sigma^* := \tfrac{1}{2}\lambda^+_{\max}\left(\bar{P}_B P_C H_L(x^*, y^*) P_C \bar{P}_B\right) \le \tfrac{1}{2}\lambda^+_{\max}\left(P_C H_L(x^*, y^*) P_C\right), \tag{5.1}$$

where $C(x) = \begin{bmatrix} A(x) & B \end{bmatrix}$ is the Jacobian for all constraints. Inequality (5.1) holds because $\bar{P}_B$ is an orthogonal projector. If the linear constraints were not explicit, the threshold value would be the rightmost term in (5.1). Intuitively, the threshold penalty value decreases by the amount of the top eigenspace of the Lagrangian Hessian that lies in the range of $B^T$, because positive semidefiniteness of $\nabla^2\phi_\sigma(x^*)$ along that space is guaranteed by the underlying solver.

It is straightforward to adapt Theorem 4 to obtain an analogous exact penalization results for the case with explicit constraints.

**6. Regularization.** Even if $A(x^*)$ has full column rank, $A(x)$ might have low column rank or small singular values away from the solution. If $A(x)$ is rank-deficient and $c(x)$ is not in the range of $A(x)^T$, then $y_\sigma(x)$ and $\phi_\sigma(x)$ are undefined. Even if $A(x)$ has full column rank but is close to rank-deficiency, the linear systems (4.2)–(4.4) and (4.7) are ill-conditioned, threatening inaccurate solutions and impeded convergence.

We modify $\phi_\sigma$ by changing the definition of the multiplier estimates in (2.2a) to solve a regularized shifted least-squares problem with regularization parameter $\delta > 0$:

$$\phi_\sigma(x;\delta) := f(x) - c(x)^T y_\sigma(x;\delta), \tag{6.1a}$$
$$y_\sigma(x;\delta) := \underset{y}{\arg\min}\;\; \tfrac{1}{2}\|A(x)y - g(x)\|_2^2 + \sigma c(x)^T y + \tfrac{1}{2}\delta^2\|y\|_2^2. \tag{6.1b}$$

This modification is similar to the exact penalty function of di Pillo and Grippo (1986). The regularization term $\frac{1}{2}\delta^2\|y\|_2^2$ ensures that the multiplier estimate $y_\sigma(x;\delta)$ is always defined even when $A(x)$ is rank-deficient. The only computational change is that the $(2,2)$ block of the matrices in (4.2)–(4.4) and (4.7) is now $-\delta^2 I$.

Besides improving $\text{cond}(\mathcal{K})$, $\delta > 0$ has the advantage of making $\mathcal{K}$ symmetric quasi-definite. Vanderbei (1995) shows that any symmetric permutation of such a matrix possesses an $LDL^T$ factorization with $L$ unit lower triangular and $D$ diagonal indefinite. Result 2 of Gill et al. (1996) implies that the factorization is stable as long as $\delta$ is sufficiently far from zero. Various authors propose regularized matrices of this type to stabilize optimization methods in the presence of degeneracy. In particular, Wright (1998) accompanies his discussion with an update scheme for $\delta$ that guarantees fast asymptotic convergence.

We continue to assume that (NP) satisfies (A1b), but we now replace (A2b) by (A2a). For a given isolated local minimum $x^*$ of (NP), $\sigma$ sufficiently large, define

$$x(\delta) \in \arg\min_x \|x - x^*\| \text{ such that } x \text{ is a local-min of } \phi_\sigma(x;\delta)$$

for use as an analytical tool in the upcoming discussion. Although the above argmin may be set-valued, Theorem 7 shows that for sufficiently small $\delta$, $x(\delta)$ is unique.

Note that for $\delta > 0$, we would not expect that $x(\delta) = x^*$, but we want to ensure that $x(\delta) \to x^*$ as $\delta \to 0$. Note that for $x$ such that $y_\sigma(x)$ is defined,

$$\begin{aligned} y_\sigma(x;\delta) &= (A(x)^T A(x) + \delta^2 I)^{-1} A(x)^T A(x) y_\sigma(x) \\ &= y_\sigma(x) - \delta^2 (A(x)^T A(x) + \delta^2 I)^{-1} y_\sigma(x). \end{aligned}$$

Therefore, for $x$ such that $\phi_\sigma(x)$ is defined, we can write the regularized penalty function as a perturbation of the unregularized one:

$$\begin{aligned} \phi_\sigma(x;\delta) &= f(x) - c(x)^T y_\sigma(x;\delta) \\ &= f(x) - c(x)^T y_\sigma(x) + \delta^2 c(x)^T (A(x)^T A(x) + \delta^2 I)^{-1} y_\sigma(x) \\ &= \phi_\sigma(x) + \delta^2 P_\delta(x), \end{aligned} \tag{6.2}$$

where $P_\delta(x) := c(x)^T (A(x)^T A(x) + \delta^2 I)^{-1} y_\sigma(x)$. By (A1b), $P_\delta$ is bounded and has at least two continuous derivatives in a neighborhood of $x^*$.

THEOREM 7. *Suppose* (A1b) *and* (A2a) *are satisfied, $x^*$ is a second-order KKT point for* (NP)*, and $\nabla^2 \phi_\sigma(x^*) \succ 0$. Then there exists $\bar{\delta} > 0$ such that $x(\delta)$ is a $\mathcal{C}_1$ function for $0 \le \delta < \bar{\delta}$. In particular, $\|x(\delta) - x^*\| = O(\delta)$.*

*Proof.* The theorem follows from the Implicit Function Theorem (Ortega and Rheinboldt, 2000, Theorem 5.2.4) applied to $\nabla \phi_\sigma(x;\delta) = 0$. ∎

An option to recover $x^*$ using $\phi_\sigma(x;\delta)$ is to minimize a sequence of problems defined by $x_{k+1} = \arg\min_x \phi_\sigma(x;\delta_k)$ with $\delta_k \to 0$, using $x_k$ to warm-start the next subproblem. However, we show that it is possible to solve a single subproblem by decreasing $\delta$ during the subproblem iterations, while retaining fast local convergence.

To keep results independent of the minimization algorithm being used, for a family of functions $\mathcal{F}$ we define $G : \mathcal{F} \times \mathbb{R}^n \to \mathbb{R}^n$ such that for $\varphi \in \mathcal{F}$ and an iterate $x$, $G(\varphi, x)$ computes an update direction. For example, if $\mathcal{F} = \mathcal{C}_2$, we can represent Newton's method with $G(\varphi, x) = -(\nabla^2 \varphi(x))^{-1} \nabla \varphi(x)$. Define $\nu(\delta)$ as a function such that for repeated applications, $\nu^k(\delta) \to 0$ as $k \to \infty$ at a chosen rate; for example, for a quadratic rate, we let $\nu(\delta) = \delta^2$.

**Algorithm 3.** Minimization of the regularized penalty function $\phi_\sigma(x,\delta)$ with $\delta \to 0$.

1: Choose $x_1$, $\delta_0 < 1$
2: **for** $k = 1, 2, \ldots$ **do**
3: Set

$$\delta_k \leftarrow \max\left\{\min\left\{\|\nabla\phi_\sigma(x_k;\delta_{k-1})\|, \delta_{k-1}\right\}, \nu(\delta_{k-1})\right\} \tag{6.3}$$

4: $p_k \leftarrow G\left(\phi_\sigma(\cdot,\delta_k), x_k\right)$
5: $x_{k+1} \leftarrow x_k + p_k$
6: **end for**

Algorithm 3 describes how to adaptively update $\delta$ each iteration.

In order to analyze Algorithm 3, we formalize the notions of rates of convergence using definitions equivalent to those of Ortega and Rheinboldt (2000, section 9).

DEFINITION 8. *We say that $x_k \to x^*$ with order at least $\tau > 1$ if there exists $M > 0$ such that, for all sufficiently large $k$, $\|x_{k+1} - x^*\| \le M\|x_k - x^*\|^\tau$. We say that $x_k \to x^*$ with R-order at least $\tau > 1$ if there exists a sequence $\alpha_k$ such that, for all sufficiently large $k$,*

$$\|x_k - x^*\| \le \alpha_k, \qquad \alpha_k \to 0 \text{ with order at least } \tau.$$

We first show that any minimization algorithm achieving a certain local rate of convergence can be regarded as *inexact Newton* (Dembo et al., 1982).

LEMMA 9. *Let $\varphi(x)$ be a $\mathcal{C}_2$ function with local minimum $x^*$ and $\nabla^2\varphi(x^*) \succ 0$. Suppose we minimize $\varphi$ according to*

$$x_{k+1} = x_k + p_k, \qquad p_k = G(\varphi, x_k), \tag{6.4}$$

*such that $x_k \to x^*$ with order at least $\tau \in (1,2]$. Then in some neighborhood of $x^*$, the update procedure $G(\varphi,x)$ is equivalent to the inexact Newton iteration*

$$x_{k+1} \leftarrow x_k + p_k, \qquad \nabla^2\varphi(x_k)p_k = -\nabla\varphi(x_k) + r_k, \qquad \|r_k\| = O(\|\nabla\varphi(x_k)\|^\tau). \tag{6.5}$$

*Proof.* There exists a neighborhood $N_N(x^*)$ such that for $x_0^N \in N_N(x^*)$, the Newton update $x_{k+1}^N = x_k^N + p_k^N$ with $\nabla^2\varphi(x_k^N)p_k^N = -\nabla\varphi(x_k^N)$ converges quadratically:

$$\|x^* - x_{k+1}^N\| \le M_1\|x^* - x_k^N\|^2, \qquad x_k \in N_N(x^*).$$

Let $N_G(x^*)$ be the neighborhood where order $\tau$ convergence is obtained for (6.4) with constant $M_2$. Let $B_\epsilon(x^*) = \{x \mid \|x^* - x\| \le \epsilon\}$. Set $\epsilon < \min\{M_2^{-1/(\tau-1)}, 1\}$ such that $B_\epsilon(x^*) \subseteq N_N \cap N_G$, and observe that if $x_0 \in B_\epsilon(x^*)$, then $x_k \in B_\epsilon(x^*)$ for all $k$ because $\|x^* - x_k\|$ is monotonically decreasing (since $M_2\|x^* - x_0\|^{\tau-1} < 1$), and so by induction

$$\|x^* - x_k\| \le M_2\|x^* - x_{k-1}\|^\tau = M_2\|x^* - x_{k-1}\|^{\tau-1}\|x^* - x_{k-1}\| < \|x^* - x_{k-1}\|.$$

By continuity of $H(x)$, there exists $M_3 > 0$ such that $\|H(x)\| \le M_3$ for all $B_\epsilon(x^*)$. Then for $x_k \in B_\epsilon(x^*)$,

$$\begin{aligned}
\|r_k\| = \|\nabla^2\varphi(x_k)p_k + \nabla\varphi(x_k)\| &= \|\nabla^2\varphi(x_k)\left(x_{k+1} - x_k - p_k^N\right)\| \\
&\le \|\nabla^2\varphi(x_k)\|\|x_{k+1} - x^* + x^* - x_{k+1}^N\| \\
&\le M_3(\|x_{k+1} - x^*\| + \|x_{k+1}^N - x^*\|) \\
&\le M_3(M_1 + M_2)\|x_k - x^*\|^\tau.
\end{aligned}$$

Now, because $\varphi \in \mathcal{C}_2$ and $\nabla^2\varphi(x^*) \succ 0$, there exists a constant $M_4$ such that $\|x_k - x^*\| \leq M_4\|\nabla\varphi(x_k)\|$ for $x_k \in N_G(x^*) \cap N_N(x^*)$. Therefore $\|r_k\| \leq M_4M_3(M_1 + M_2)\|\nabla\varphi(x_k)\|^\tau$, which is the inexact Newton method, convergent with order $\tau$. ∎

Note that Lemma 9 can be modified to accommodate any form of superlinear convergence, as long as $\|r_k\|$ converges at the same rate as $x_k \to x^*$.

THEOREM 10. *Suppose that* (A1b) *and* (A2a) *are satisfied, $x^*$ is a second-order KKT point for* (NP), *$\nabla^2\phi_\sigma(x^*) \succ 0$, and there exist $\bar{\delta}$ and an open set $B(x^*)$ containing $x^*$ such that for $\widetilde{x}_0 \in B(x^*)$ and $0 < \delta \leq \bar{\delta}$, the sequence defined by $\widetilde{x}_{k+1} = \widetilde{x}_k + G(\phi_\sigma(\cdot;\delta), \widetilde{x}_k)$ converges quadratically to $x(\delta)$:*

$$\|x(\delta) - \widetilde{x}_{k+1}\| \leq M_\delta\|x(\delta) - \widetilde{x}_k\|^2.$$

*Further suppose that for $\delta \leq \bar{\delta}$, $M_\delta \leq M$ is uniformly bounded. Then there exist an open set $B'(x^*)$ that contains $x^*$ and $0 < \delta' < 1$ such that if $x \in B'(x^*)$, $\delta \leq \delta'$, and $x_k \to x^*$ for $x_k$ defined by Algorithm 3 (with $\nu(\delta) = \delta^2$), then $x_k \to x^*$ R-quadratically.*

The proof is in Appendix A.2. Although there are many technical assumptions, the takeaway message is that we need only minimize $\phi_\sigma(\cdot;\delta_k)$ until $\|\nabla\phi_\sigma\| = O(\delta_k)$, because under typical smoothness assumptions we have that $\|x(\delta) - x^*\| = O(\delta)$ for $\delta$ sufficiently small. Decreasing $\delta$ at the same rate as the local convergence rate of the method on a fixed problem should not perturb $\phi_\sigma(x;\delta)$ too much, therefore allowing for significant progress on the perturbed problem in few steps. The assumption that $M_\delta \leq M$ uniformly also appears strong, but we believe it is unavoidable—the number of iterations between updates to $\delta$ must be bounded above by a constant for overall convergence to be unimpeded. Within the basin of convergence and for a fixed $\delta > 0$, an optimization method would achieve the same local convergence rate that it would have with $\delta = 0$ fixed.

Theorem 10 can be generalized to superlinear rates of convergence using a similar proof. As long as $\nu(\cdot)$ drives $\delta \to 0$ as fast as the underlying algorithm would locally converge for fixed $\delta$, local convergence of the entire regularized algorithm is unchanged.

**7. Inexact evaluation of the penalty function.** We discuss the effects of solving (4.8) approximately, and thus evaluating $\phi_\sigma$ and its derivatives inexactly. Various optimization solvers can utilize inexact function values and derivatives while ensuring global convergence and certain local convergence rates, provided the user can compute relevant quantities to a prescribed accuracy. For example, Conn et al. (2000, sections 8–9) describe conditions on the inexactness of model and gradient evaluations to ensure convergence; Heinkenschloss and Vicente (2001) describe a trust-region SQP solver using inexact gradients; Kouri et al. (2014) describe an inexact trust-region solver using inexact function values and gradients for unconstrained problems. We focus on inexactness within trust-region methods for optimizing $\phi_\sigma$.

The accuracy and computational cost in the evaluation of $\phi_\sigma$ and its derivatives depends on the accuracy of the solves of (4.8). If the cost to solve (4.8) depends on solution accuracy (e.g., with iterative linear solvers), it is advantageous to consider optimization solvers that use inexact computations, especially for large-scale problems.

Let $\mathcal{S} \subseteq \mathbb{R}^n$ be a compact set. In this section, let $\widetilde{\phi}_\sigma(x)$, $\nabla\widetilde{\phi}_\sigma(x)$, etc. distinguish the inexact quantities from their exact counterparts. We also drop the arguments from

operators as in section 4. We consider three quantities that are computed inexactly: $g_\sigma$, $\phi_\sigma$, and $\nabla\phi_\sigma$. For given positive error tolerances $\eta_i$ (which may be relative to their corresponding quantities), we are interested in exploring termination criteria for solving (4.8) to ensure that the following conditions hold for all $x \in \mathcal{S}$:

$$\left|\phi_\sigma - \widetilde{\phi}_\sigma\right| \le M\eta_1, \tag{7.1a}$$

$$\left\|\nabla\phi_\sigma - \nabla\widetilde{\phi}_\sigma\right\| \le M\eta_2, \tag{7.1b}$$

$$\|g_\sigma - \widetilde{g}_\sigma\| \le M\eta_3, \tag{7.1c}$$

where $M > 0$ is some fixed constant (which may or may not be known). Kouri et al. (2014) give a trust-region method using inexact objective value and gradient information that guarantees global convergence provided (7.1a)–(7.1b) hold without requiring that $M$ be known a priori. We may compare this to the conditions of Conn et al. (2000, sections 8.4 and 10.6), which require more stringent conditions on (7.1a)–(7.1b). They require that $\eta_2 = \|\nabla\widetilde{\phi}_\sigma\|$ and that $M$ be known and fixed according to parameters in the trust-region method.

This leads us to the following proposition, which allows us to bound the residuals of (4.2) and (4.4) to ensure (7.1).

PROPOSITION 11. *Let $\mathcal{S}$ be a compact set, and suppose that $\sigma_{\min}(A(x)) \ge \lambda > 0$ for all $x \in \mathcal{S}$. Then for $x \in \mathcal{S}$, if*

$$\|r_1\| = \left\|\mathcal{K}\begin{bmatrix}\widetilde{g}_\sigma\\ \widetilde{y}_\sigma\end{bmatrix} - \begin{bmatrix}g\\ \sigma c\end{bmatrix}\right\| \le \min\{1, \|c\|^{-1}\}\cdot\min\{\eta_1, \eta_3\}, \tag{7.2}$$

*then* (7.1a) *and* (7.1c) *hold for some constant $M$. Also, if*

$$\|r_1\| \le \eta_2 \quad \textit{and} \quad \|r_2\| = \left\|\mathcal{K}\begin{bmatrix}\widetilde{v}\\ \widetilde{w}\end{bmatrix} - \begin{bmatrix}0\\ c\end{bmatrix}\right\| \le \min\{1, \eta_2\}, \tag{7.3}$$

*then* (7.1b) *holds for some (perhaps different) constant $M$.*

*Proof.* Because $\mathcal{S}$ is compact and $\lambda > 0$, there exists $\bar{\lambda} > 0$ such that $\|\mathcal{K}\|$, $\|\mathcal{K}^{-1}\| \le \bar{\lambda}$ for all $x \in \mathcal{S}$. Thus, (7.1c) follows directly from (4.4) and (7.2) with $M = \bar{\lambda}$. Similarly,

$$\left|\phi_\sigma - \widetilde{\phi}_\sigma\right| = \left|c^T\left(y_\sigma - \widetilde{y_\sigma}\right)\right| \le \|c\|\,\|y_\sigma - \widetilde{y_\sigma}\| \le \bar{\lambda}\eta_1,$$

and (7.1a) holds with $M = \bar{\lambda}$. We apply a similar analysis to ensure that (7.1b) holds. Define the vector $h \in \mathbb{R}^m$ such that $h_i = \|H_i\|$. Define $v$, $w$ as the solutions to (7.3) for $r_2 = 0$, so that from (7.3) we have

$$\begin{aligned}
\|\nabla\phi_\sigma - \nabla\widetilde{\phi}_\sigma\| &\le \|g_\sigma - \widetilde{g}_\sigma\| + \|Y_\sigma c - \widetilde{Y}_\sigma c\|\\
&\le \bar{\lambda}\eta_2 + \|(H_\sigma - \sigma I)v - S_\sigma^T w - (\widetilde{H}_\sigma - \sigma I)\widetilde{v} + \widetilde{S}_\sigma^T\widetilde{w}\|\\
&\le \bar{\lambda}\eta_2 + \sigma\|v - \widetilde{v}\| + \|H_\sigma v - \widetilde{H}_\sigma\widetilde{v}\| + \|S_\sigma^T w - \widetilde{S}_\sigma^T\widetilde{w}\|\\
&\le \left(\bar{\lambda} + \sigma\bar{\lambda}\right)\eta_2 + \|H_\sigma(v - \widetilde{v}) + (H_\sigma - \widetilde{H}_\sigma)\widetilde{v}\|\\
&\quad + \|S_\sigma^T(w - \widetilde{w}) + (S_\sigma - \widetilde{S}_\sigma)^T\widetilde{w}\|\\
&\le \left(\bar{\lambda} + \sigma\bar{\lambda}\right)\eta_2 + \|H_\sigma\|\|v - \widetilde{v}\| + \left\|\sum_{i=1}^m \left((y_\sigma)_i - (\widetilde{y}_\sigma)_i\right)H_i\right\|\|\widetilde{v}\|
\end{aligned}$$

$$+ \|S_\sigma\|\|w - \widetilde{w}\| + \left\|\sum_{i=1}^{m} \widetilde{w}_i H_i\right\| \|g_\sigma - \widetilde{g}_\sigma\|$$
$$\leq \left(\bar{\lambda} + \sigma\bar{\lambda} + \|H_\sigma\|\bar{\lambda} + \bar{\lambda}\|\widetilde{v}\|\|h\| + \|S_\sigma\|\bar{\lambda} + \|\widetilde{w}\|\|h\|\bar{\lambda}\right)\eta_2.$$

Note that $\|H_\sigma\|$, $\|h\|$, $\|S_\sigma\|$, $\|\widetilde{w}\|$, and $\|\widetilde{v}\|$ are bounded uniformly in $\mathcal{S}$. ∎

In the absence of additional information, using (7.1) with unknown $M$ may be the only way to take advantage of inexact computations, because computing exact constants (such as the norm of $\mathcal{K}$ or the various operators above) is not practical. In some cases the bounds (7.1) are relative, e.g., $\eta_2 = \min\{\|\nabla\widetilde{\phi}_\sigma\|, \Delta\}$ for a certain $\Delta > 0$. It may then be necessary to compute $\|\nabla\widetilde{\phi}_\sigma\|$ and refine the solutions of (4.4) and (4.2) until they satisfy (7.2)–(7.3). However, given the expense of applying these operators, it may be more practical to use a nominal relative tolerance, as in the numerical experiments of section 9.

We include a (trivial) improvement to Proposition 11 that satisfies (7.1a) and (7.1c) with $M = 1$, given additional spectral information on $A$. If we solve (4.4) via

$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} \Delta g_\sigma \\ y_\sigma \end{bmatrix} = \begin{bmatrix} 0 \\ \sigma c - A^T g \end{bmatrix}, \qquad g_\sigma = g + \Delta g_\sigma, \tag{7.4}$$

we can use LNLQ (Estrin et al., 2019b), a Krylov subspace method for such systems, which ensures that $\|\Delta g_\sigma - \Delta\widetilde{g}_\sigma^{(j)}\|$ and $\|y_\sigma - \widetilde{y}_\sigma^{(j)}\|$ are monotonic, where $\Delta\widetilde{g}_\sigma^{(j)}$, $\widetilde{y}_\sigma^{(j)}$ are the $j$th LNLQ iterates. Given $\lambda > 0$ such that $\sigma_{\min}(A) \geq \lambda$, LNLQ can compute cheap upper bounds on $\|\Delta g_\sigma - \Delta\widetilde{g}_\sigma^{(j)}\|$ and $\|y_\sigma - \widetilde{y}_\sigma^{(j)}\|$, allowing us to terminate the solve when $\|\Delta g_\sigma - \widetilde{\Delta g_\sigma}\| \leq \eta_2\|\widetilde{\Delta g_\sigma} + g\| = \eta_2\|\widetilde{g}_\sigma\|$ and $\|y_\sigma - \widetilde{y}_\sigma\| \leq \min\{1, \|c\|^{-1}\}\eta_1$. Typically, termination criteria for the optimization solver will include a condition that $\|g_\sigma\| \leq \epsilon_d$ to determine approximate local minimizers to (NP). For such cases, we can instead require that $\|\widetilde{g}_\sigma\| \leq \frac{1}{1+\eta_2}\epsilon^d$, because then

$$\|g_\sigma\| \leq \|g_\sigma - \widetilde{g}_\sigma\| + \|\widetilde{g}_\sigma\| \leq (1 + \eta_2)\|\widetilde{g}_\sigma\| \leq \epsilon^d.$$

Similarly, we have

$$\left|\phi_\sigma - \widetilde{\phi}_\sigma\right| \leq \|c\|\|y_\sigma - \widetilde{y}_\sigma\| \leq \eta_1,$$

which now satisfies (7.1a) with $M = 1$.

Although finding suitable bounds $\lambda$ on the smallest singular value may be difficult in general, it is trivially available in some cases because of the way $\mathcal{K}$ is preconditioned (for an example, see section 9). However, a complication is that if LNLQ is used with a right-preconditioner $\mathcal{P} \approx A^T A$, then $\|y_\sigma - \widetilde{y}_\sigma\|_\mathcal{P}$ is monotonic and LNLQ provides bounds on the preconditioned norm instead of the Euclidean norm. If $\|\mathcal{P}^{-1}\|$ can be bounded, then the bound $\|y_\sigma - \widetilde{y}_\sigma\| \leq \|y_\sigma - \widetilde{y}_\sigma\|_\mathcal{P}\|\mathcal{P}^{-1}\|$ can be used.

**8. Practical considerations.** We discuss some matters related to the use of $\phi_\sigma$ in practice. In principle, nearly any smooth unconstrained solver can be used to find a local minimum of $\phi_\sigma$ because it has at least one continuous derivative, and a continuous Hessian approximation if (A1a) is satisfied. However, the structure of $\phi_\sigma$ lends itself more readily to certain optimization methods than to others, especially when the goal of creating a factorization-free solver is kept in mind.

Fletcher (1973) originally envisioned a Newton-type procedure

$$x_{k+1} \leftarrow x_k - \alpha_k B_i^{-1}(x_k)\nabla\phi_\sigma(x_k), \qquad i = 1 \quad \text{or} \quad 2,$$

where $B_1$, $B_2$ are the Hessian approximations from (4.6) and $\alpha_k > 0$ is a step-size. Fletcher (1973, Theorem 2) further proved that superlinear convergence is achieved, or quadratic convergence if the second derivatives of $f$ and $c$ are Lipschitz continuous. However, for large problems it is expensive to compute $B_i$ explicitly and solve the dense system $B_i s_k = -\nabla\phi_\sigma(x_k)$.

We instead propose using a Steihaug (1983) Newton-CG type trust-region solver to minimize $\phi_\sigma$. First, trust-region methods are preferable to linesearch methods (Nocedal and Wright, 2006, sections 3–4) for objectives with expensive evaluations; it is costly to evaluate $\phi_\sigma$ repeatedly to determine a step-size at every iteration as this requires solving a linear system. Further, $\nabla^2\phi_\sigma$ is often indefinite, and trust-region methods can take advantage of directions of negative curvature. Computing $B_i$ explicitly is not practical, but products are reasonable as they only require solving two additional linear systems with the same matrix, thus motivating the use of a Newton-CG type trust-region solver. In particular, solvers such as TRON (Lin and Moré, 1999b) and KNITRO (Byrd et al., 2006) are suitable for minimizing $\phi_\sigma$. KNITRO has the additional advantage of handling explicit linear constraints.

We have not yet addressed choosing the penalty parameter $\sigma$. Although we can provide an a posteriori threshold value for $\sigma^*$, it is difficult to know this threshold ahead of time. Mukai and Polak (1975) give a scheme for updating $\rho$ with $\phi_{\sigma,\rho}$ and $\sigma = 0$; however, they were using a Newton-like scheme that required a solve with $B_1(x)$. Further, $\sigma^*$ ensures only *local* convexity, and that a local minimizer of (NP) is a local minimizer of $\phi_\sigma$—but as with other penalty functions, $\phi_\sigma$ may be unbounded below in general for any $\sigma$. A heuristic that we employ is to ensure that the primal and dual feasibility, $\|c(x)\|$ and $\|g_L(x, y_\sigma(x))\|$, are within some factor of each other (e.g., 100) to encourage them to decrease at approximately the same rate. If primal feasibility decreases too quickly and small steps are taken, it is indicative of $\sigma$ being too large, and similarly if primal feasibility is too large, then $\sigma$ should be increased; this can be done with a multiplicative factor or by estimating $\frac{1}{2}\|P_A(x)H_\sigma(x)P_A(x)\|$ via the power method (because it is an upper bound on $\sigma^*$ when $x = x^*$; see (3.3)). Although this heuristic is often effective in our experience, in situations where the penalty function begins moving toward negative infinity, we require a different recovery strategy, which is the subject of future work. The works of Mukai and Polak (1975); Byrd et al. (2012) give promising directions for developing such strategies.

In practice, regularization (section 6) is used only if necessary. For well-behaved problems, using $\delta = 0$ typically requires fewer outer iterations than using $\delta > 0$. However, when convergence is slow and/or the Jacobians are ill-conditioned, initializing with $\delta > 0$ is often vital and can improve performance significantly.

**9. Numerical experiments.** We investigate the performance of Fletcher's penalty function on several PDE-constrained optimization problems and some standard test problems. For each test we use the stopping criterion

$$\begin{matrix}\|c(x_k)\| \le \epsilon_k^p \\ \|g_\sigma(x_k)\| \le \epsilon_k^d\end{matrix} \quad \text{or} \quad \|\nabla\phi_\sigma(x_k)\| \le \epsilon^d, \tag{9.1}$$

with $\epsilon_k^p := \epsilon\,(1 + \|x_k\|_\infty + \|c(x_0)\|_\infty)$ and $\epsilon_k^d := \epsilon\,(1 + \|y_k\|_\infty + \|g_\sigma(x_0)\|_\infty)$, where $\epsilon > 0$ is a tolerance, e.g., $\epsilon = 10^{-8}$. We also keep $\sigma$ fixed for each experiment.

Depending on the problem, the augmented systems (4.8) are solved by either direct or iterative methods. For direct methods, we use the corrected seminormal equations (Björck, 1996); see the supplementary materials for implementation details. For iterative solves, we use CRAIG (Arioli, 2013) with preconditioner $\mathcal{P}$ and two

possible termination criteria: for some positive parameter $\eta$,

$$\left\| \mathcal{K} \begin{bmatrix} p^{(k)} \\ q^{(k)} \end{bmatrix} - \begin{bmatrix} u \\ v \end{bmatrix} \right\|_{\bar{\mathcal{P}}^{-1}} \leq \eta \left\| \begin{bmatrix} u \\ v \end{bmatrix} \right\|_{\bar{\mathcal{P}}^{-1}}, \qquad \bar{\mathcal{P}} := \begin{bmatrix} I & \\ & \mathcal{P} \end{bmatrix}, \tag{9.2a}$$

$$\left\| \begin{bmatrix} p^* \\ q^* \end{bmatrix} - \begin{bmatrix} p^{(k)} \\ q^{(k)} \end{bmatrix} \right\|_{\bar{\mathcal{P}}} \leq \eta \left\| \begin{bmatrix} p^{(k)} \\ q^{(k)} \end{bmatrix} \right\|_{\bar{\mathcal{P}}}, \tag{9.2b}$$

which are respectively based on the relative residual and the relative error (obtained via LNLQ). We can use (9.2b) when a lower bound on $\sigma_{\min}(A\mathcal{P}^{-1/2})$ is available (e.g., for the PDE-constrained optimization problems).

Because we are using trust-region methods to minimize $\phi_\sigma$, the objective and gradient of $\phi_\sigma$ (and therefore of $f$) are evaluated once per iteration. We use KNITRO (Byrd et al., 2006) and a MATLAB implementation of TRON (Lin and Moré, 1999b). Our implementation of TRON does not require explicit Hessians (only Hessian-vector products) and is unpreconditioned. We use $B_1(x)$ (4.6a) when efficient products with $S(u, x)$ are available; otherwise we use $B_2(x)$ (4.6b). When $\phi_\sigma$ is evaluated approximately (for large $\eta$), we use the solvers without modification, thus pretending that the function and gradient are evaluated exactly. Note that when the evaluations are noisy, the solvers are no longer guaranteed to converge; the development of suitable solvers is left to future work and discussed in section 10.

**9.1. 1D Burgers' problem.** Let $\Omega = (0, 1)$ denote the physical domain and $H^1(\Omega)$ denote the Sobolev space of functions in $L^2(\Omega)$, whose weak derivatives are also in $L^2(\Omega)$. We solve the following one-dimensional ODE-constrained control problem:

$$\begin{aligned} \underset{u \in H^1(\Omega), z \in L^2(\Omega)}{\text{minimize}} \quad & \tfrac{1}{2} \int_\Omega \left(u(x) - u_d(x)\right)^2 dx + \tfrac{1}{2}\alpha \int_\Omega z(x)^2 dx \\ \text{subject to} \quad & -\nu u_{xx} + u u_x = z + h \quad \text{in } \Omega, \\ & u(0) = 0, \;\; u(1) = -1, \end{aligned} \tag{9.3}$$

where the constraint is a 1D Burgers' equation over $\Omega = (0, 1)$, with $h(x) = 2\left(\nu + x^3\right)$ and $\nu = 0.08$. The first objective term measures deviation from the data $u_d(x) = -x^2$, while the second term regularizes the control with $\alpha = 10^{-2}$. We discretize (9.3) by segmenting $\Omega$ into $n_c = 512$ equal-sized cells, and we approximate $u$ and $z$ with piecewise linear elements. This results in a nonlinearly constrained optimization problem with $n = 2n_c = 1024$ variables and $m = n_c - 1$ constraints.

We optimize $u, z$ by minimizing $\phi_\sigma$ with $\sigma = 10^3$, using $B_1(x)$ (4.6a) as Hessian approximation and $u_0 = 0$, $z_0 = 0$ as the initial point. We use TRON to optimize $\phi_\sigma$ and LNLQ to (approximately) solve (4.8). We partition the Jacobian of the discretized constraints into $A(x)^T = \begin{bmatrix} A_u(x)^T & A_z(x)^T \end{bmatrix}$, where $A_u(x) \in \mathbb{R}^{n \times n}$ and $A_z(x) \in \mathbb{R}^{m \times m}$ are the Jacobians for $u$ and $z$. We use the preconditioner $\mathcal{P}(x) = A_u(x)^T A_u(x)$, which amounts to performing two linearized Burgers' solves with a given source. For this preconditioner, $\sigma_{\min}(A\mathcal{P}^{-1/2}) \geq 1$, allowing us to bound the error via LNLQ and to use both (9.2b) and (9.2a) to terminate LNLQ. The maximum number of inner-CG iterations (for solving the trust-region subproblem) is $n$.

We choose $\epsilon = 10^{-8}$ in the stopping conditions (9.1). Table 1 records the number of Hessian- and Jacobian-vector products as we vary the accuracy of the linear system solves via $\eta$ in (9.2).

TRON required a moderate number of trust-region iterations. However, evaluating $\phi_\sigma$ and its derivatives can require many Jacobian and Hessian products, because

TABLE 1

*Results of solving* (9.3) *using TRON to minimize* $\phi_\sigma$*, with various* $\eta$ *in* (9.2b) *(left) and* (9.2a) *(right) to terminate the linear system solves. We record the number of Lagrangian Hessian (*$\#Hv$*), Jacobian (*$\#Av$*), and adjoint Jacobian (*$\#A^Tv$*) products.*

| $\eta$ | Iter. | $\#Hv$ | $\#Av$ | $\#A^Tv$ | Iter. | $\#Hv$ | $\#Av$ | $\#A^Tv$ |
|---|---|---|---|---|---|---|---|---|
| $10^{-2}$ | 37 | 19112 | 56797 | 50553 | 35 | 7275 | 29453 | 27148 |
| $10^{-4}$ | 34 | 6758 | 35559 | 33423 | 35 | 7185 | 36757 | 34482 |
| $10^{-6}$ | 35 | 7182 | 45893 | 43619 | 35 | 7194 | 47999 | 45721 |
| $10^{-8}$ | 35 | 7176 | 53296 | 51204 | 35 | 7176 | 54025 | 51753 |
| $10^{-10}$ | 35 | 7176 | 59802 | 57530 | 35 | 7176 | 59310 | 57038 |
| | Error-based termination | | | | Residual-based termination | | | |

for every product with the approximate Hessian we need to solve an augmented linear system. (Note that there are more products with $A^T$ than $A$ because we shift the right-hand side as in (7.4) prior to each system solve.) On the other hand, the linear systems did not have to be solved to full precision. As $\eta$ increased from $10^{-10}$ to $10^{-2}$, the number of Hessian-vector products stayed relatively constant, but the number of Jacobian-vector products dropped substantially, and the average number of LNLQ iterations required per solve dropped from about 9 to 5, except that when $\eta = 10^{-2}$ in (9.2b), the linear solves were too inaccurate and the number of CG iterations per trust-region subproblem increased dramatically near the solution (requiring more linear solves). Using (9.2a) usually led to more products with the Lagrangian Hessian and Jacobian, except when the linear solves were nearly exact, or extremely inexact.

**9.2. 2D inverse Poisson problem.** Let $\Omega = (-1, 1)^2$ denote the physical domain and $H_0^1(\Omega) \subset H^1(\Omega)$ be the Hilbert space of $H^1(\Omega)$ functions whose value on the boundary $\partial\Omega$ is zero. We solve the following 2D PDE-constrained control problem:

$$\begin{aligned} \underset{u \in H_0^1(\Omega),\, z \in L^\infty(\Omega)}{\text{minimize}} \quad & \tfrac{1}{2} \int_\Omega (u - u_d)^2 \, dx + \tfrac{1}{2}\alpha \int_\Omega z^2 dx \\ \text{subject to} \quad & -\nabla \cdot (z \nabla u) = h \quad \text{in } \Omega, \\ & u = 0 \quad \text{in } \partial\Omega. \end{aligned} \tag{9.4}$$

Let $c = (0.2, 0.2)$ and define $S_1 = \{x \mid \|x - c\|_2 \le 0.3\}$ and $S_2 = \{x \mid \|x - c\|_1 \le 0.6\}$. The target state $u_d$ is generated as the solution of the PDE with $z_*(x) = 1 + 0.5 \cdot I_{S_1}(x) + 0.5 \cdot I_{S_2}(x)$, where for any set $C$, $I_C(x) = 1$ if $x \in C$ and 0 otherwise.

The force term here is $h(x_1, x_2) = -\sin(\omega x_1)\sin(\omega x_2)$, with $\omega = \pi - \frac{1}{8}$. The control variable $z$ represents the diffusion coefficients for the Poisson problem that we are trying to recover based on the observed state $u_d$. We set $\alpha = 10^{-4}$ as regularization parameter. We discretize (9.4) using $P_1$ finite elements on a uniform mesh of 1089 triangles and employ an identical discretization for the optimization variables $z \in L^\infty(\Omega)$, obtaining a problem with $n_z = 1089$ controls and $n_u = 961$ states, so that $n = n_u + n_z$. There are $m = n_u$ constraints, as we must solve the PDE on every interior grid point. The initial point is $u_0 = \mathbb{1}$, $z_0 = \mathbb{1}$. Typically $z \ge 0$ is explicitly imposed, but we only consider equality constraints in the present paper (inequality constraints are treated in (Estrin et al., 2020)). For this discretization, the iterates $z_k$ remained positive throughout the minimization.

We use $\sigma = 10^{-2}$ as penalty parameter and $B_2(x)$ as Hessian approximation. We again use LNLQ for the linear solves, with the same preconditioning strategy as

TABLE 2

*Results of solving* (9.4) *using TRON to minimize* $\phi_\sigma$ *with various* $\eta$ *in* (9.2b) *(left) and* (9.2a) *(right) to terminate the linear system solves. We record the number of Lagrangian Hessian (*$\#Hv$*), Jacobian (*$\#Av$*), and adjoint Jacobian (*$\#A^Tv$*) products.*

| $\eta$ | Iter. | $\#Hv$ | $\#Av$ | $\#A^Tv$ | Iter. | $\#Hv$ | $\#Av$ | $\#A^Tv$ |
|---|---|---|---|---|---|---|---|---|
| $10^{-2}$ | 29 | 874 | 1794 | 2608 | 27 | 850 | 1772 | 2562 |
| $10^{-4}$ | 27 | 830 | 1950 | 2728 | 25 | 668 | 1649 | 2265 |
| $10^{-6}$ | 27 | 866 | 2317 | 3129 | 27 | 868 | 2356 | 3168 |
| $10^{-8}$ | 27 | 866 | 2673 | 3485 | 27 | 866 | 2784 | 3596 |
| $10^{-10}$ | 27 | 866 | 3145 | 3957 | 27 | 866 | 3251 | 4063 |
| | Error-based termination | | | | Residual-based termination | | | |

in section 9.1. The results are given in Table 2. We see a similar trend to that of Table 1, as larger $\eta$ allows TRON to converge within nearly the same number of outer iterations and Lagrangian Hessian-vector products (even when $\eta = 10^{-2}$), while significantly decreasing the number of Jacobian-vector products. We see again that using (9.2b) to terminate LNLQ tends to need less work than with (9.2a). The exception is using (9.2a) with $\eta = 10^{-4}$. The solver terminates two iterations sooner, resulting in a sharp drop in Jacobian-vector products but little change in solution quality. Note that if $\epsilon = 10^{-9}$ were used for the experiment, the runs would appear more similar to one another.

**9.3. 2D Poisson–Boltzmann problem.** We now solve a control problem where the constraint is a 2D Poisson–Boltzmann equation:

$$\begin{aligned} \underset{u\in H_0^1(\Omega), z\in L^2(\Omega)}{\text{minimize}} \quad & \tfrac{1}{2}\int_\Omega (u-u_d)^2\,dx + \tfrac{1}{2}\alpha\int_\Omega z^2 dx \\ \text{subject to} \quad & -\Delta u + \sinh(u) = h + z \quad \text{in } \Omega, \\ & u = 0 \quad \text{in } \partial\Omega. \end{aligned} \tag{9.5}$$

We use the same notation and $\Omega$ as in section 9.2, with forcing term $h(x_1, x_2) = -\sin(\omega x_1)\sin(\omega x_2)$, $\omega = \pi - \frac{1}{8}$, and target state

$$u_d(x) = \begin{cases} 10 & \text{if } x \in [0.25, 0.75]^2, \\ 5 & \text{otherwise.} \end{cases}$$

We discretize (9.5) using $P_1$ finite elements on a uniform mesh with 10201 triangles, resulting in a problem with $n = 20002$ variables and $m = 9801$ constraints. We use $u_0 = \mathbb{1}$, $z_0 = \mathbb{1}$ as the initial point.

We perform the experiment described in section 9.2 using $\sigma = 10^{-1}$ and record the results in Table 3. The results are similar to Table 2, where the number of Jacobian products decreases with $\eta$, while the number of outer iterations and Lagrangian-Hessian products stays fairly constant. We see that with stopping criterion (9.2a), the LNLQ iterations increase somewhat compared to (9.2b), as it is a tighter criterion.

**9.4. Regularization.** We next solve problems where $A(x)$ is rank-deficient for some iterates, requiring that $\phi_\sigma$ be regularized (section 6). We use the corrected seminormal equations to solve the linear systems (to machine precision), with $B_2(x)$ as the Hessian approximation.

For problem `hs061` ($n = 3$ variables, $m = 3$ constraints) from the CUTEst test set (Gould et al., 2015) we use $x_0 = 0$, $\sigma = 10^2$, $\delta_0 = 10^{-1}$. For problem `mss1` ($n = 90$,

TABLE 3

*Results of solving* (9.5) *using TRON to minimize* $\phi_\sigma$ *with various* $\eta$ *in* (9.2b) *(left) and* (9.2a) *(right) to terminate the linear system solves. We record the number of Lagrangian Hessian (*$\#Hv$*), Jacobian (*$\#Av$*), and adjoint Jacobian (*$\#A^Tv$*) products.*

| $\eta$ | Iter. | $\#Hv$ | $\#Av$ | $\#A^Tv$ | Iter. | $\#Hv$ | $\#Av$ | $\#A^Tv$ |
|---|---|---|---|---|---|---|---|---|
| $10^{-2}$ | 57 | 1150 | 2300 | 3392 | 58 | 1166 | 2427 | 3534 |
| $10^{-4}$ | 58 | 1166 | 2479 | 3586 | 57 | 1150 | 2728 | 3820 |
| $10^{-6}$ | 57 | 1150 | 2812 | 3904 | 57 | 1150 | 3210 | 4302 |
| $10^{-8}$ | 57 | 1150 | 3357 | 4449 | 57 | 1150 | 3721 | 4813 |
| $10^{-10}$ | 57 | 1150 | 3811 | 4903 | 57 | 1150 | 4265 | 5357 |
| | Error-based termination | | | | Residual-based termination | | | |

TABLE 4

*Convergence results for* `hs061` *(left) and* `mss1` *(right) when TRON and KNITRO minimize* $\phi_\sigma(\cdot;\delta)$*. The first rows show the iteration at which* $\delta$ *is updated, and the last two rows record the final primal and dual infeasibilities.*

| $\delta$ | TRON | KNITRO |
|---|---|---|
| $10^{-1}$ | 22 | 12 |
| $10^{-2}$ | 23 | 13 |
| $10^{-4}$ | 24 | 14 |
| $\|c(\bar{x})\|$ | $10^{-10}$ | $10^{-9}$ |
| $\|g_\sigma(\bar{x})\|$ | $10^{-7}$ | $10^{-8}$ |

| $\delta$ | TRON | KNITRO |
|---|---|---|
| $10^{-2}$ | 46 | 33 |
| $10^{-4}$ | 52 | 36 |
| $10^{-7}$ | 53 | 37 |
| $\|c(\bar{x})\|$ | $10^{-12}$ | $10^{-14}$ |
| $\|g_\sigma(\bar{x})\|$ | $10^{-8}$ | $10^{-9}$ |

$m = 73$) we use $x_0 = 0$, $\sigma = 10^3$, $\delta_0 = 10^{-2}$. In both cases we decrease $\delta$ according to $\nu(\delta) = \delta^2$ to retain local quadratic convergence. For both problems, $A(x_0)$ is rank-deficient and $\phi_\sigma$ is undefined, so the trust-region solvers terminate immediately. We therefore regularize $\phi_\sigma$ and record the iteration at which $\delta$ changed. For `mss1`, we set $\delta_{\min} = 10^{-7}$ to avoid ill-conditioning. The results are in Table 4.

The regularized problems converge with few iterations between $\delta$ updates, showing evidence of quadratic convergence. Note that a large $\delta$ can perturb $\phi_\sigma(\cdot;\delta)$ substantially, so that $\delta_0$ may need to be chosen judiciously. We use $\delta_0 = 10^{-2}$ because neither TRON nor KNITRO would converge for `mss1` when $\delta_0 = 10^{-1}$.

**10. Discussion and concluding remarks.** The smooth exact penalty approach is promising for nonlinearly constrained optimization particularly when the augmented linear systems (4.8) can be solved efficiently. However, several potential drawbacks remain as avenues for future work.

One property of $\phi_\sigma$ observed from our experiments is that it is highly nonlinear and nonconvex. Even though superlinear or quadratic local convergence can be achieved, the high nonlinearity potentially results in many globalization iterations and small step-sizes. Further, $\phi_\sigma$ is usually nonconvex, even if (NP) is convex.

Another aspect not yet discussed is preconditioning the trust-region subproblem. This is particularly nontrivial because the (approximate) Hessian is available only as an operator. Traditional approaches based on incomplete factorizations (Lin and Moré, 1999a) are not applicable. One possibility is to use a preconditioner for the Lagrangian Hessian $H_\sigma$ as a preconditioner for the approximate penalty Hessian $B_i$ (4.6a)–(4.6b). This may be effective when $m \ll n$ because $H_\sigma$ and $B_i$ would differ only by a low-rank update; otherwise $H_\sigma$ can be a poor approximation to $B_i$. Preconditioning is vital for trust-region solvers and is a direction for future work.

Products with $B_i$ (4.6a)–(4.6b) are generally more expensive than typical Hessian-

vector products, as they require solving a linear system. Products with a quasi-Newton approximation would be significantly faster. Also, exact curvature away from the solution may be less important than near the solution for choosing good directions; therefore a hybrid scheme that begins with quasi-Newton and switches to $B_i$ may be effective. Another strategy, similar to Morales and Nocedal (2000), is to use quasi-Newton approximations to precondition the trust-region subproblems involving $B_i$: the approximation for iteration $k$ can be updated with every $B_i(x_{k-1})$ product, or with every update step $x_k - x_{k-1}$.

Further improvements that would make our approach applicable to a wider range of problems include developing suitable solvers based on the work of Heinkenschloss and Vicente (2001) and Kouri et al. (2014) as these properly handle noisy function and gradient evaluations; a robust update for the penalty parameter $\sigma$; termination criteria on the augmented systems in order to integrate $\phi_\sigma$ fully into a solver in the style of Kouri et al. (2014); and relaxing (A2a) to weaker constraint qualifications. Even if $\sigma \geq \sigma^*$, it is possible for $\phi_\sigma$ to be unbounded, because $\sigma$ only guarantees local convexity. Diverging iterates must be detected sufficiently early because by the time unboundedness is detected, it may be too late to update $\sigma$ and we would need to backtrack several iterates. To relax (A2a), it may be possible to combine our regularization (6.1a) with a dual proximal-point approach.

The next obvious extension is to handle inequality constraints—the subject of Estrin et al. (2020). Fletcher (1973) proposed a nonsmooth extension to $\phi_\sigma$ for this purpose, but it is less readily applicable to most solvers.

Our MATLAB implementation can be found at https://github.com/optimizers/FletcherPenalty. To highlight the flexibility of Fletcher's approach, we implemented several options for applying various solvers to the penalty function and for solving the augmented systems, and other options discussed along the way.

**Appendix A. Technical details.** We provide proofs and technical details that were omitted earlier.

**A.1. Example problem with a spurious local minimum.** Consider the feasibility problem (NP) with $f(x) = 0$ and $c(x) = x^3 + x - 2$. The only minimizer is $x^* = 1$. The penalty function

$$\phi_\sigma(x) = \sigma \frac{(x^3 + x - 2)^2}{(3x^2 + 1)^2}$$

is defined everywhere and has local minimizers at $x_1 = 1$ (the solution) and $x_2 \approx -1.56$. Because the stationary points are independent of $\sigma$ in this case, $\phi_\sigma$ always has the spurious local minimizer $x_2$.

**A.2. Proof of Theorem 10.** We repeat the assumptions of Theorem 10:

(B1) (NP) satisfies (A1b) and (A2a).
(B2) $x^*$ is a second-order KKT point for (NP) satisfying $\nabla^2 \phi_\sigma(x^*) \succ 0$.
(B3) There exist $\bar{\delta} > 0$ and an open set $B(x^*)$ containing $x^*$ such that if $\widetilde{x}_0 \in B(x^*)$ and $\delta \leq \bar{\delta}$, the sequence $\widetilde{x}_{k+1} = \widetilde{x}_k + G(\phi_\sigma(\cdot; \delta), \widetilde{x}_k)$ converges quadratically to $x(\delta)$ with constant $M$ independent of $\delta$.
(B4) Assume $\delta, \delta_k \geq 0$ throughout so that we can avoid indicating this everywhere.

LEMMA 12. *Under the assumptions of Theorem* 10*:*

1. *$\phi_\sigma(\cdot; \delta)$ has two continuous derivatives for $\delta > 0$ and $x \in \mathbb{R}^n$ by* (B1)*.*
2. *There exists an open set $B_1(x^*)$ containing $x^*$ such that $\phi_\sigma(x)$ is well defined and has two continuous derivatives for all $x \in B_1(x^*)$ by* (B1)*.*

3. $\nabla^2\phi_\sigma(x^*) = \nabla^2\phi_\sigma(x^*;0) \succ 0$ *and* $\phi_\sigma(x;\delta)$ *is* $\mathcal{C}_2$ *in both* $x$ *and* $\delta$, *so by assumption* (B2) *there exists an open set* $B_2(x^*)$ *containing* $x^*$ *and* $\widetilde{\delta} > 0$ *such that* $\nabla^2\phi_\sigma(x;\delta) \succ 0$ *for* $(x,\delta) \in B_2(x^*) \times [0,\widetilde{\delta}]$.
4. *By Theorem* 7, *there exists* $\hat{\delta}$ *such that for* $\delta \le \hat{\delta}$, $x(\delta)$ *is continuous in* $\delta$. *Therefore there exists an open set* $B_3(x^*)$ *such that* $x(\delta) \in B_3(x^*)$ *for* $\delta \le \hat{\delta}$.
5. *There exists a neighborhood* $B_4(x^*)$ *where Newton's method is quadratically convergent (with factor* $N$*) on* $\phi_\sigma(x)$ *by* (B2).
6. *Given* $\delta_0 \le \min\{\bar{\delta}, \hat{\delta}\}$, *where* $\bar{\delta}$ *is defined in* (B3), *there exists a neighborhood* $B_5(x^*)$ *such that* $\|\nabla\phi_\sigma(x;\delta_0)\| \le \delta_0$ *for all* $x \in B_5(x^*)$.

*We define*

$$B'(x^*) := B(x^*) \cap \left(\bigcap_{i=1}^{5} B_i(x^*)\right) \qquad \textit{and} \qquad \delta' < \min\{\bar{\delta}, \widetilde{\delta}, \hat{\delta}, 1\}$$

*and note that* $x_k$ *defined by Algorithm* 3 *satisfies* $x_k \in B'(x^*)$ *for all* $k$ *by* (B3). *Because* $\phi_\sigma(x;\delta)$ *is* $\mathcal{C}_2$ *in* $B'(x^*) \times [0,\delta']$, *there exist positive constants* $K_1, \dots, K_5$ *such that*

7. $\|\nabla\phi_\sigma(x;\delta)\| \le K_1 < 1$, $\|\nabla^2\phi_\sigma(x;\delta)\|, \|\nabla^2\phi_\sigma(x;\delta)^{-1}\| \le K_2$ *for* $x \in B'(x^*)$ *and* $\delta \le \delta'$;
8. $\|\nabla P_\delta(x)\| \le K_3$, $\|\nabla^2 P_\delta(x)\| \le K_4$ *for* $x \in B'(x^*)$ *and* $\delta \le \delta'$;
9. $\|x_k - x^*\| \le K_5\|\nabla\phi_\sigma(x_k)\|$.

*Proof.* Statements 1–2 follow from (B1). Statements 3 and 5 follow from (B2). Statement 4 follows from Theorem 7.

Now consider statement 6. For a given $\delta_0$, we have $\nabla\phi_\sigma(x(\delta_0);\delta_0) = 0$ and so there exists a neighborhood $\widetilde{B}$ around $x(\delta_0)$ such that $\|\nabla\phi_\sigma(x;\delta_0)\| \le \delta_0$ for all $x \in \widetilde{B}$. Further, $x(\delta_0) \in B_3(x^*)$, so let $B_5(x^*) = \widetilde{B} \cap B_3(x^*)$. ∎

We first give some technical results. All assume that $x_k \in B'(x^*)$ and $\delta_k \le \delta'$.

LEMMA 13. *Assume* $\delta_{k-1} \le \delta_0 \le \delta'$. *For* $\delta_k$ *defined according to* (6.3),

$$\|\nabla\phi_\sigma(x_k;\delta_k)\| = O(\delta_k).$$

*Proof.* The result holds for $k = 0$ in view of observation 6 of Lemma 12.

Because $x_k \in B'(x^*)$ and $\delta_k$, $\delta_{k-1} \le \delta'$, observation 8 of Lemma 12 gives $\|\nabla P_{\delta_{k-1}}(x_k)\| \le K_3$ and $\|\nabla P_{\delta_k}(x_k)\| \le K_3$. Using (6.2), we have

$$\begin{aligned}
\|\nabla\phi_\sigma(x_k;\delta_k)\| &= \|\nabla\phi_\sigma(x_k;\delta_{k-1}) - \delta_{k-1}^2\nabla P_{\delta_{k-1}}(x_k) + \delta_k^2\nabla P_{\delta_k}(x_k)\| \\
&\le \|\nabla\phi_\sigma(x_k;\delta_{k-1})\| + \delta_{k-1}^2\|\nabla P_{\delta_{k-1}}(x_k)\| + \delta_k^2\|\nabla P_{\delta_k}(x_k)\| \\
&\le \|\nabla\phi_\sigma(x_k;\delta_{k-1})\| + (\delta_{k-1}^2 + \delta_k^2)K_3 \\
&= \|\nabla\phi_\sigma(x_k;\delta_{k-1})\| + O(\delta_k), \qquad \text{(A.1)}
\end{aligned}$$

where the last inequality follows from $\delta_k^2 \le \delta_k$ and (6.3), which implies that $\delta_k \ge \nu(\delta_{k-1}) = \delta_{k-1}^2$.

We consider two cases. If $\|\nabla\phi_\sigma(x_k;\delta_{k-1})\| \le \delta_{k-1}$, (6.3) implies that

$$\delta_k = \max(\|\nabla\phi_\sigma(x_k;\delta_{k-1})\|, \delta_{k-1}^2) \ge \|\nabla\phi_\sigma(x_k;\delta_{k-1})\|,$$

and therefore (A.1) gives $\|\nabla\phi_\sigma(x_k;\delta_k)\| = O(\delta_k)$.

Otherwise, (6.3) yields $\delta_k = \max(\delta_{k-1}, \delta_{k-1}^2) = \delta_{k-1}$, and there exists $\ell \le k-1$ such that $\delta_k = \delta_{k-1} = \cdots = \delta_\ell < \delta_{\ell-1}$, or $\ell = 1$. If $\delta_\ell < \delta_{\ell-1}$, step 3 of Algorithm 3

implies that $\|\nabla\phi_\sigma(x_\ell;\delta_{\ell-1})\| < \delta_{\ell-1}$, which by the above sequence of inequalities implies that $\|\nabla\phi_\sigma(x_\ell;\delta_\ell)\| = O(\delta_\ell)$. Then, because $\delta_k = \delta_\ell$,

$$\|\nabla\phi_\sigma(x_\ell,\delta_k)\| = \|\nabla\phi_\sigma(x_\ell,\delta_\ell)\| = O(\delta_\ell) = O(\delta_k).$$

Define the sequence $\{\widetilde{x}_j\}$ with $\widetilde{x}_0 = x_\ell$ and $\widetilde{x}_{j+1} = \widetilde{x}_j + G(\phi_\sigma(\cdot;\delta_\ell),\widetilde{x}_j)$. By (B3), $\widetilde{x}_j \to x(\delta_\ell)$ quadratically, so after $j$ iterations of this procedure, by Taylor's theorem, for some $\widetilde{M}$, we have

$$\|\nabla\phi_\sigma(\widetilde{x}_j;\delta_\ell)\| \le \widetilde{M}^j\|\nabla\phi_\sigma(\widetilde{x}_0;\delta_\ell)\|^{2^j} \le \widetilde{M}^j K_1^{2^j-1}\|\nabla\phi_\sigma(\widetilde{x}_0;\delta_\ell)\|^2.$$

Then after $j = O(1)$ iterations of this procedure (depending only on $\widetilde{M}$ and $K_1$), we have $\widetilde{M}^j K_1^{2^j-1} \le 1$ (because $K_1 < 1$), so that

$$\|\nabla\phi_\sigma(\widetilde{x}_j;\delta_k)\| \le \|\nabla\phi_\sigma(\widetilde{x}_0;\delta_k)\|^2 = \|\nabla\phi_\sigma(x_\ell;\delta_k)\|^2 = O(\delta_k^2) < O(\delta_k).$$

Therefore, $k - \ell \le j = O(1)$, and by (B3) and the fact that $\delta_{k-1} = \delta_k$,

$$\|\nabla\phi_\sigma(x_k;\delta_{k-1})\| = O\left(M^{k-\ell}\|\nabla\phi_\sigma(x_\ell;\delta_{k-1})\|^{2^{k-\ell}}\right) = O\left(M^{k-\ell}\delta_k^{2^{k-\ell}}\right) = O(\delta_k).$$

The second equality follows from the fact that $\delta_{k-1} = \delta_k$, and so $\|\nabla\phi_\sigma(x_\ell;\delta_{k-1})\| = O(\delta_k)$ by the induction assumption. ∎

LEMMA 14. *For $p_k$ defined by step 4 of Algorithm 3, $\|p_k\| = O(\delta_k)$.*

*Proof.* According to (B3), we may apply Lemma 9 with $\tau = 2$ and view step 4 of Algorithm 3 as an inexact Newton step, i.e., there exists a constant $N_2 > 0$ such that

$$\nabla^2\phi_\sigma(x_k;\delta_k)p_k = -\nabla\phi_\sigma(x_k;\delta_k) + r_k, \qquad \|r_k\| \le N_2\|\nabla\phi_\sigma(x_k;\delta_k)\|^2.$$

Therefore by Lemma 13,

$$\begin{aligned}\|p_k\| &= \|\nabla^2\phi_\sigma(x_k;\delta_k)^{-1}(-\nabla\phi_\sigma(x_k;\delta_k) + r_k)\| \\ &\le \|\nabla^2\phi_\sigma(x_k;\delta_k)^{-1}\|\,(\|\nabla\phi_\sigma(x_k;\delta_k)\| + \|r_k\|) \\ &\le K_2\left(\|\nabla\phi_\sigma(x_k;\delta_k)\| + N_2\|\nabla\phi_\sigma(x_k;\delta_k)\|^2\right) \\ &\le K_2\left(O(\delta_k) + O(\delta_k^2)\right) = O(\delta_k).\end{aligned}$$ ∎

LEMMA 15. *Let $p_k$ be defined by step 4 of Algorithm 3 and $q_k$ be the Newton direction for the unregularized penalty function defined by $\nabla^2\phi_\sigma(x_k)q_k = -\nabla\phi_\sigma(x_k)$. Then $\|p_k - q_k\| \in O(\delta_k^2)$.*

*Proof.* According to (B3), we may apply Lemma 9 with $\tau = 2$ and view step 4 of Algorithm 3 as an inexact Newton step, i.e.,

$$\nabla^2\phi_\sigma(x_k;\delta_k)p_k = -\nabla\phi_\sigma(x_k;\delta_k) + r_k, \tag{A.2a}$$

$$\|r_k\| = O(\|\nabla\phi_\sigma(x_k;\delta_k)\|^2). \tag{A.2b}$$

We premultiply (A.2a) by $\nabla^2\phi_\sigma(x_k)^{-1}$ and use (6.2) to obtain

$$p_k + \delta_k^2\nabla^2\phi_\sigma(x_k)^{-1}\nabla^2 P_{\delta_k}(x_k)p_k = q_k + \delta_k^2\nabla^2\phi_\sigma(x_k)^{-1}\nabla P_{\delta_k}(x_k) + \nabla^2\phi_\sigma(x_k)^{-1}r_k.$$

Lemma 13, Lemma 14, and the triangle inequality then yield

$$\begin{aligned}\|p_k - q_k\| &= \left\|\delta_k^2\nabla^2\phi_\sigma(x_k)^{-1}\left(\nabla P_{\delta_k}(x_k) - \nabla^2 P_{\delta_k}(x_k)p_k\right) + \nabla^2\phi_\sigma(x_k)^{-1}r_k\right\| \\ &\le \delta_k^2\|\nabla^2\phi_\sigma(x_k)^{-1}\|\left(\|\nabla P_{\delta_k}(x_k)\| + \|\nabla^2 P_{\delta_k}(x_k)p_k\|\right) + \|\nabla^2\phi_\sigma(x_k)^{-1}r_k\| \\ &\le \delta_k^2 K_2\left(K_3 + O(\delta_k)\right) + O(\delta_k^2) = O(\delta_k^2).\end{aligned}$$ ∎

Using the previous technical results, we are in position to establish our main result.

*Proof of Theorem* 10. We show that for $x_0 \in B'(x^*)$ we achieve R-quadratic convergence, by showing that $\|x_k - x^*\| = O(\delta_k)$ and that $\delta_k \to 0$ quadratically. By observation 9 of Lemma 12, (6.2), the triangle inequality, Lemma 13, and observation 8 of Lemma 12, we have

$$\begin{aligned}\|x_k - x^*\| &\le K_5\|\nabla\phi_\sigma(x_k)\| \\ &= K_5\|\nabla\phi_\sigma(x_k;\delta_k) - \delta_k^2\nabla P_{\delta_k}(x_k)\| \\ &\le K_5(\|\nabla\phi_\sigma(x_k;\delta_k)\| + \delta_k^2\|\nabla P_{\delta_k}(x_k)\|) \\ &\le K_5\left(O(\delta_k) + \delta_k^2 K_3\right) = O(\delta_k).\end{aligned}$$

Let $q_k$ be the Newton direction defined in Lemma 15. There exists a constant $N > 0$ such that

$$\begin{aligned}\|x_{k+1} - x^*\| &= \|x_k + p_k - x^*\| \\ &\le \|x_k + q_k - x^*\| + \|p_k - q_k\| \\ &\le N\|x_k - x^*\|^2 + \|p_k - q_k\| = O(\delta_k^2).\end{aligned}$$

It remains to show that $\delta_k$ decreases quadratically. If $\|\nabla\phi_\sigma(x_{k+1},\delta_k)\| \le \delta_k^2$,

$$\delta_{k+1} = \max\{\min\{\|\nabla\phi_\sigma(x_{k+1},\delta_k)\|,\delta_k\},\delta_k^2\} \le \max\{\|\nabla\phi_\sigma(x_{k+1},\delta_k)\|,\delta_k^2\} = \delta_k^2.$$

Assume now that $\|\nabla\phi_\sigma(x_{k+1},\delta_k)\| > \delta_k^2$. We have from (6.2) and observations 7–8 of Lemma 12 that

$$\begin{aligned}\delta_{k+1} &= \max\{\min\{\|\nabla\phi_\sigma(x_{k+1},\delta_k)\|,\delta_k\},\delta_k^2\} \\ &\le \|\nabla\phi_\sigma(x_{k+1},\delta_k)\| \\ &\le \|\nabla\phi_\sigma(x_{k+1})\| + \delta_k^2\|\nabla P_{\delta_k}(x_{k+1})\| \\ &\le K_2^{-1}\|x_{k+1} - x^*\| + \delta_k^2 K_3 = O(\delta_k^2).\end{aligned}$$

Thus we have $\|x_k - x^*\| = O(\delta_k)$ and $\delta_{k+1} = O(\delta_k^2)$, which means that $x_k \to x^*$ R-quadratically. ∎

**Acknowledgments.** We would like to express our deep gratitude to Drew Kouri for supplying PDE-constrained optimization problems in MATLAB, for helpful discussions throughout this project, and for hosting the first author for two weeks at Sandia National Laboratories. We are also extremely grateful to the reviewers for their unusually careful reading and their many helpful questions and suggestions.

## REFERENCES

M. Arioli (2013), *Generalized Golub–Kahan bidiagonalization and stopping criteria*, SIAM J. Matrix Anal. Appl., 34, pp. 571–592, https://doi.org/10.1137/120866543.

D. P. Bertsekas (1975), *Necessary and sufficient conditions for a penalty method to be exact*, Math. Program., 9, pp. 87–99.

D. P. Bertsekas (1982), *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York.

Å. Björck (1996), *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, https://doi.org/10.1137/1.9781611971484.

R. H. Byrd, J. Nocedal, and R. A. Waltz (2006), *KNITRO: An integrated package for nonlinear optimization*, in Large-Scale Nonlinear Optimization, G. di Pillo and M. Roma, eds., Springer-Verlag, New York, pp. 35–59.

R. H. Byrd, G. Lopez-Calva, and J. Nocedal (2012), *A line search exact penalty method using steering rules*, Math. Program., 133, pp. 39–73.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint (2000), *Trust-Region Methods*, MOS-SIAM Ser. Optim. 1, SIAM, Philadelphia, https://doi.org/10.1137/1.9780898719857.

R. Courant (1943), *Variational methods for the solution of problems with equilibrium and variation*, Bull. Amer. Math. Soc., 49, pp. 1–23.

J. E. Craig (1955), *The N-step iteration procedures*, J. Math. Phys., 34, pp. 64–73.

R. S. Dembo, S. C. Eisenstat, and T. Steihaug (1982), *Inexact Newton methods*, SIAM J. Numer. Anal., 19, pp. 400–408, https://doi.org/10.1137/0719025.

G. di Pillo and L. Grippo (1984), *A class of continuously differentiable exact penalty function algorithms for nonlinear programming problems*, in System Modelling and Optimization, E. P. Toft-Christensen, ed., Springer-Verlag, Berlin, pp. 246–256.

G. di Pillo and L. Grippo (1986), *An exact penalty function method with global convergence properties for nonlinear programming problems*, Math. Program., 36, pp. 1–18.

R. Estrin, D. Orban, and M. A. Saunders (2019a), *LSLQ: An iterative method for linear least-squares with an error minimization property*, SIAM J. Matrix Anal. Appl., 40, pp. 254–275, https://doi.org/10.1137/17M1113552.

R. Estrin, D. Orban, and M. A. Saunders (2019b), *LNLQ: An iterative method for linear least-norm problems with an error minimization property*, SIAM J. Matrix Anal. Appl., 40, pp. 1102–1124, https://doi.org/10.1137/18M1194948.

R. Estrin, M. P. Friedlander, D. Orban, and M. A. Saunders (2020), *Implementing a smooth exact penalty function for inequality-constrained nonlinear optimization*, SIAM J. Sci. Comput., 42, pp. A1836–A1859, https://doi.org/10.1137/19M1255069.

R. Fletcher (1970), *A class of methods for nonlinear programming with termination and convergence properties*, in Integer and Nonlinear Programming, J. Abadie, ed., North-Holland, Amsterdam, pp. 157–175.

R. Fletcher (1973), *A class of methods for nonlinear programming:* III. *Rates of convergence*, in Numerical Methods for Nonlinear Optimization, F. A. Lootsma, ed., Academic Press, New York.

R. Fletcher (1985), *An $l_1$ penalty method for nonlinear constraints*, in Numerical Optimization, 1984 (Boulder, CO, 1984), SIAM, Philadelphia, pp. 26–40.

P. E. Gill, M. A. Saunders, and J. R. Shinnerl (1996), *On the stability of Cholesky factorization for symmetric quasidefinite systems*, SIAM J. Matrix Anal. Appl., 17, pp. 35–46, https://doi.org/10.1137/S0895479893252623.

G. H. Golub and V. Pereyra (1973), *The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate*, SIAM J. Numer. Anal., 10, pp. 413–432, collection of articles dedicated to the memory of George E. Forsythe, https://doi.org/10.1137/0710036.

N. I. M. Gould, D. Orban, and Ph. L. Toint (2015), *CUTEst: A constrained and unconstrained testing environment with safe threads for mathematical optimization*, Comput. Optim. Appl., 60, pp. 545–557, https://doi.org/10.1007/s10589-014-9687-3.

M. Heinkenschloss and L. N. Vicente (2001), *Analysis of inexact trust-region SQP algorithms*, SIAM J. Optim., 12, pp. 283–302, https://doi.org/10.1137/S1052623499361543.

M. R. Hestenes (1969), *Multiplier and gradient methods*, J. Optim. Theory Appl., 4, pp. 303–320.

M. R. Hestenes and E. Stiefel (1952), *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Standards, 49, pp. 409–436 (1953).

D. P. Kouri, M. Heinkenschloss, D. Ridzal, and B. G. van Bloemen Waanders (2014), *Inexact objective function evaluations in a trust-region algorithm for PDE-constrained optimization under uncertainty*, SIAM J. Sci. Comput., 36, pp. A3011–A3029, https://doi.org/10.1137/140955665.

C.-J. Lin and J. J. Moré (1999a), *Incomplete Cholesky factorizations with limited memory*, SIAM J. Sci. Comput., 21, pp. 24–45, https://doi.org/10.1137/S1064827597327334.

C.-J. Lin and J. J. Moré (1999b), *Newton's method for large bound-constrained optimization problems*, SIAM J. Optim., 9, pp. 1100–1127, https://doi.org/10.1137/S1052623498345075.

N. Maratos (1978), *Exact Penalty Function Algorithms for Finite Dimensional and Optimization Problems*, Ph.D. thesis, Imperial College of Science and Technology, London, UK.

J. L. Morales and J. Nocedal (2000), *Automatic preconditioning by limited memory quasi-Newton updating*, SIAM J. Optim., 10, pp. 1079–1096, https://doi.org/10.1137/S1052623497327854.

H. Mukai and E. Polak (1975), *A quadratically convergent primal-dual algorithm with global convergence properties for solving optimization problems with equality constraints*, Math. Programming, 9, pp. 336–349.

J. Nocedal and S. J. Wright (2006), *Numerical Optimization*, 2nd ed., Springer, New York.
D. Orban and M. Arioli (2017), *Iterative Solution of Symmetric Quasi-Definite Linear Systems*, SIAM Spotlights 3, SIAM, Philadelphia, https://doi.org/10.1137/1.9781611974737.
J. M. Ortega and W. C. Rheinboldt (2000), *Iterative Solution of Nonlinear Equations in Several Variables*, Classics Appl. Math. 30, SIAM, Philadelphia, https://doi.org/10.1137/1.9780898719468.
C. C. Paige and M. A. Saunders (1975), *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12, pp. 617–629, https://doi.org/10.1137/0712047.
C. C. Paige and M. A. Saunders (1982), *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Softw., 8, pp. 43–71.
J. W. Pearson, M. Stoll, and A. J. Wathen (2012), *Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems*, SIAM J. Matrix Anal. Appl., 33, pp. 1126–1152, https://doi.org/10.1137/110847949.
T. Pietrzykowski (1969), *An exact potential method for constrained maxima*, SIAM J. Numer. Anal., 6, pp. 299–304, https://doi.org/10.1137/0706028.
M. J. D. Powell (1969), *A method for nonlinear constraints in minimization problems*, in Optimization, R. Fletcher, ed., Academic Press, London, New York, Chapter 19, pp. 283–298.
Y. Saad (1993), *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14, pp. 461–469, https://doi.org/10.1137/0914028.
V. Simoncini (2012), *Reduced order solution of structured linear systems arising in certain PDE-constrained optimization problems*, Comput. Optim. Appl., 53, pp. 591–617.
T. Steihaug (1983), *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20, pp. 626–637, https://doi.org/10.1137/0720042.
Ph. L. Toint (1981), *Towards an efficient sparsity exploiting Newton method for minimization*, in Sparse Matrices and Their Uses, I. S. Duff, ed., Academic Press, London, pp. 57–88.
R. J. Vanderbei (1995), *Symmetric quasidefinite matrices*, SIAM J. Optim., 5, pp. 100–113, https://doi.org/10.1137/0805005.
S. J. Wright (1998), *Superlinear convergence of a stabilized SQP method to a degenerate solution*, Comput. Optim. Appl., 11, pp. 253–275.
V. M. Zavala and M. Anitescu (2014), *Scalable nonlinear programming via exact differentiable penalty functions and trust-region Newton methods*, SIAM J. Optim., 24, pp. 528–558, https://doi.org/10.1137/120888181.