Les Cahiers du GERAD ISSN: 0711–2440

Implementing a smooth exact penalty function for constrained nonlinear optimization

R. Estrin, M. P. Friedlander,
D. Orban, M. A. Saunders

G–2019–27

April 2019

La publication de ces rapports de recherche est rendue possible grâce au soutien de HEC Montréal, Polytechnique Montréal, Université McGill, Université du Québec à Montréal, ainsi que du Fonds de recherche du Québec – Nature et technologies.

Dépôt légal – Bibliothèque et Archives nationales du Québec, 2019
– Bibliothèque et Archives Canada, 2019

The publication of these research reports is made possible thanks to the support of HEC Montréal, Polytechnique Montréal, McGill University, Université du Québec à Montréal, as well as the Fonds de recherche du Québec – Nature et technologies.

Legal deposit – Bibliothèque et Archives nationales du Québec, 2019
– Library and Archives Canada, 2019

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

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

Implementing a smooth exact penalty function for constrained nonlinear optimization

Dedicated to Roger Fletcher

Ron Estrin [a]
Michael P. Friedlander [b]
Dominique Orban [c]
Michael A. Saunders [d]

[a] *Institute for Computational and Mathematical Engineering, Stanford University, CA 94305–4042*

[b] *Department of Computer Science, University of British Columbia, Vancouver V6T 1Z4, BC, Canada*

[c] *GERAD & Department of Mathematics and Industrial Engineering, Polytechnique Montréal (Québec) Canada, H3C 3A7*

[d] *Department of Management Science and Engineering, Stanford University, CA 94305–4121*

restrin@stanford.edu
mpf@cs.ubc.ca
dominique.orban@gerad.ca
saunders@stanford.edu

April 2019
Les Cahiers du GERAD
G–2019–27

Abstract: We build upon Estrin et al. (2019) to develop a general constrained nonlinear optimization algorithm based on a smooth penalty function proposed by Fletcher (1970, 1973b). Although Fletcher's approach has historically been considered impractical, we show that the computational kernels required are no more expensive than those in other widely accepted methods for nonlinear optimization. The main kernel for evaluating the penalty function and its derivatives solves structured linear systems. When the matrices are available explicitly, we store a single factorization each iteration. Otherwise, we obtain a factorization-free optimization algorithm by solving each linear system iteratively. The penalty function shows promise in cases where the linear systems can be solved efficiently, e.g., PDE-constrained optimization problems when efficient preconditioners exist. We demonstrate the merits of the approach, and give numerical results on several PDE-constrained and standard test problems.

Résumé : Suite aux travaux de Estrin et al. (2019), nous développons une méthode pour l'optimisation avec contraintes générales basée sur une fonction de pénalité différentiable proposée par Fletcher (1970,1973b). Bien que la méthode de Fletcher fut considérée trop coûteuse par le passé, nous montrons que les noyaux de calcul ne sont pas plus coûteux que dans d'autres méthodes établies pour l'optimisation non linéaire. Le noyau principal pour évaluer la fonction de pénalité et ses dérivées consiste à résoudre des systèmes linéaires structurés. Quand les matrices sont disponibles explicitement, une seule factorisation par itération est suffisante. Dans le cas contraire, nous donnons une implémentation sans factorisation dans laquelle les systèmes sont résolus itérativement. La fonction de pénalité est prometteuse dans les cas où les systèmes linéaires se prêtent à une résolution efficace, comme c'est le cas en optimisation sous contraintes différentielles quand de bons préconditionneurs existent. Nous montrons les mérites de cette approche et donnons des résultats numériques sur des problèmes standard et plusieurs problèmes avec contraintes différentielles.

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. The work of M. P. Friedlander was supported by NSERC Discovery Grant 312104. The work of D. Orban was supported by NSERC Discovery Grant 299010-04. The research of M. A. Saunders was partially supported by the National Institute of General Medical Sciences of the National Institutes of Health [award U01GM102098].

1 Introduction

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

$$\begin{array}{lll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & f(x) & \\ \text{subject to} & c(x) = 0 & : \; y \\ & \ell \leq x \leq u & : \; z, \end{array} \tag{NP}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^m$ are smooth functions ($m \leq n$), the n-vectors ℓ and u provide (possibly infinite) bounds on x, and $y \in \mathbb{R}^m$, $z \in \mathbb{R}^n$ are Lagrange multipliers associated with the equality constraints and bounds respectively. Estrin et al. (2019) describe factorization-based and factorization-free implementations of a smooth exact penalty method proposed by Fletcher (1970) to treat equality constraints. Here, we generalize our implementation to problems with both equality and bound constraints, and hence to problems with general inequality constraints.

Fletcher's penalty function for equality constraints is the Lagrangian

$$L(x, y) = f(x) - y^T c(x), \tag{1}$$

in which the vector $y = y_\sigma(x)$ is treated as a function of x dependent on a parameter $\sigma > 0$. Fletcher (1973b) proposes an extension to inequality constraints that exhibits nonsmoothness when constraint activities change. The penalty function (1) was long considered too costly for practical use (Bertsekas, 1975; Conn et al., 2000; Nocedal and Wright, 2006), and the nonsmooth extension to inequality constraints further impacted its practicality.

We demonstrate that a certain smooth extension of Fletcher's penalty function yields a practical implementation for inequality-constrained optimization, by showing that the computational kernels are no more expensive than those in other widely accepted methods for nonlinear optimization, such as sequential quadratic programming.

The extended penalty function is *exact* because KKT points of (NP) are KKT points of the penalty problem for all values of σ larger than a finite threshold σ^*. The main computational kernel for evaluating the penalty function and its derivatives is the solution of certain structured linear systems. We show how to solve the systems efficiently by factorizing a single matrix each iteration (if the matrix is available explicitly) and reusing the factors to evaluate the penalty function and its derivatives. We also provide a *factorization-free* implementation in which linear systems are solved iteratively. This makes the penalty function particularly applicable to certain problem classes such as PDE-constrained problems, where excellent preconditioners exist; see Section 8.

The advantage of smooth exact penalty functions is that they lead to conceptually simpler algorithms compared to traditional methods for constrained problems. The original problem is replaced by a single smooth bound-constrained problem with a sufficiently large penalty parameter. This avoids complicated heuristics to trade-off primal and dual feasibility, and can avoid the need for primal feasibility restoration stages or composite-step methods. Further, because our penalty is smooth and we can compute a sufficiently accurate Hessian approximation, second-order methods with fast local convergence may be used.

Paper outline We follow the structure of Estrin et al. (2019). We introduce the penalty function in Section 2, and discuss its relationship with existing approaches in Section 3. We give the penalty function's properties and derive an explicit threshold for the penalty parameter in Section 4. In Section 5 we show how to evaluate the penalty function and its derivatives efficiently. We discuss an extension to maintain linear constraints in Section 6. Practical considerations pertaining to the penalty function appear in Section 7. We apply the penalty approach to standard and PDE-constrained problems in Section 8, and discuss future research directions in Section 9.

2 The proposed penalty function

For (NP), we propose the penalty function

$$\phi_\sigma(x) := f(x) - c(x)^T y_\sigma(x) = L(x, y_\sigma(x)), \tag{2}$$

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

$$y_\sigma(x) := \arg\min_y \tfrac{1}{2}\|A(x)y - g(x)\|^2_{Q(x)} + \sigma c(x)^T y, \qquad g(x) := \nabla f(x), \tag{3}$$

$$A(x) := \nabla c(x) = \begin{bmatrix} g_1(x) & \cdots & g_m(x) \end{bmatrix}, \qquad g_i(x) := \nabla c_i(x), \tag{4}$$

$$Y_\sigma(x) := \nabla y_\sigma(x). \tag{5}$$

Note that A and Y_σ are n-by-m matrices. We define an n-by-n diagonal matrix $Q(x) = \operatorname{diag}(q_i(x_i))$ with $\omega \in \mathbb{R}^n_+$, $\omega < u - \ell$, and

$$q_i(x_i) := \begin{cases} 1 & \text{if } \ell_i = -\infty \text{ and } u_i = \infty, \\ x_i - \ell_i - \frac{1}{2\omega_i}\left(2x_i + u_i - \ell_i - \frac{1}{2}\omega_i\right)^2 & \text{if } |u_i + \ell_i - 2x_i| \le \frac{1}{2}\omega_i, \\ \min\{x_i - \ell_i, u_i - x_i\} & \text{otherwise.} \end{cases} \tag{6}$$

The diagonal of $Q(x)$ is a smooth approximation of $\min\{x - \ell, u - x\}$, and ω controls the smoothness. We use $\omega_i = \min\{1, \frac{1}{2}(u_i - \ell_i)\}$. Note that $Q(x)$ is nonnegative on $[\ell, u]$. We describe this function in more detail below.

We assume that (NP) satisfies the following conditions:

(A1) f and c are $\mathcal{C}_3$.

(A2) The linear independence constraint qualification (LICQ) is satisfied for stationary points and at all x satisfying $\ell < x < u$. LICQ is satisfied at x if

$$\left\{\nabla c_i(x), e_j \mid x_j \in \{\ell_j, u_j\},\ i \in [m],\ j \in [n]\right\}$$

is linearly independent, where e_j is the jth column of the identity matrix, and $[n] := \{1, 2, \ldots, n\}$.

(A3) Stationary points satisfy strict complementarity. If (x^*, y^*, z^*) is a stationary point, exactly one of z^*_j and $\min\{x^*_j - \ell_j, u_j - x^*_j\}$ is zero for all $j \in [n]$.

(A4) The problem is feasible. That is, there exists x such that $\ell \le x \le u$ and $c(x) = 0$, with $\ell_j < u_j$ for all $j \in [n]$. We assume fixed variables have been eliminated from the problem.

Assumption (A1) ensures that ϕ_σ has two continuous derivatives and is typical for smooth exact penalty functions (Bertsekas, 1982, Proposition 4.16). However, at most two derivatives of f and c are required to implement this penalty function in practice (see section 5.5). Assumption (A2) guarantees that $Y_\sigma(x)$ and $y_\sigma(x)$ are uniquely defined; (A3) provides additional regularity to ensure that the threshold penalty parameter σ^* is well defined.

The basis of our approach is to solve

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad \phi_\sigma(x) \quad \text{subject to} \quad \ell \le x \le u : \ z \tag{PP}$$

instead of (NP). We purposely set z to be the Lagrange multiplier for the bound constraints of both (NP) and (PP) because, as we show, they are equal at a solution.

2.1 The scaling matrix

The diagonal entries of the scaling matrix $Q(x)$ are smooth approximations of the complementarity function $\min\{x - \ell, u - x\}$ (Chen, 2000). Figure 1 plots $q(x)$ with finite ℓ and u.

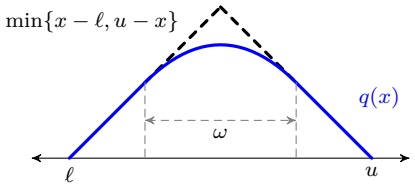

Figure 1: Plot of $q(x)$, a smooth approximation of $\min\{x-\ell, u-x\}$.

The definition of $y_\sigma(x)$ (3) can be interpreted as a smooth approximation of the complementarity conditions in the first-order KKT conditions (13d)–(13f) below. The role of $Q(x)$ is therefore to ensure that the partial derivatives of the Lagrangian corresponding to indices of inactive bounds are zero. Similar smoothing strategies can be found in the complementarity constraint literature (Anitescu, 2000; Leyffer, 2006).

For $x \in \mathbb{R}$, the derivative of $q(x)$ is

$$q'(x) = \begin{cases} 0 & \text{if } \ell = -\infty \text{ and } u = \infty, \\ 1 - \frac{2}{\omega}\left(2x + u - \ell - \frac{1}{2}\omega\right) & \text{if } |u + \ell - 2x| \leq \frac{1}{2}\omega, \\ 1 & \text{if } x - \ell < u - x, \\ -1 & \text{if } x - \ell > u - x. \end{cases} \tag{7}$$

Note that the cases in (7) are not mutually exclusive, and should checked top to bottom until a case is satisfied. The choice of $q(x)$ is not unique because any smooth concave function that is zero at $x_j \in \{\ell_j, u_j\}$ works in our framework. For instance, if $u_j - \ell_j$ is large, we could use a smooth approximation of $\min\{x_j - \ell_j, u_j - x_j, 1\}$ to avoid numerical issues that can arise if x is far from its bounds.

2.2 Notation

Denote x^* as a local stationary point of (NP), with corresponding dual solutions y^* and z^*. At x^*, define the set of active bounds as

$$\mathcal{A}(x^*) := \{j \mid x_j \in \{\ell_j, u_j\}\}, \tag{8}$$

and define the critical cones $C_\phi(x^*, z^*)$ and $C(x^*, z^*)$ as

$$\mathcal{C}_\phi(x^*, z^*) := \left\{ p \;\middle|\; \begin{array}{ll} p_j = 0 & \text{if } z_j^* \neq 0 \\ p_j \geq 0 & \text{if } x_j^* = \ell_j \\ p_j \leq 0 & \text{if } x_j^* = u_j \end{array} \right\}, \tag{9a}$$

$$\mathcal{C}(x^*, z^*) := \left\{ p \in \mathcal{C}_\phi(x^*, z^*) \,\middle|\, A(x^*)^T p = 0 \right\}. \tag{9b}$$

Observe that by (A3), $\mathcal{C}_\phi(x^*, z^*) = \{p \mid p_j = 0 \text{ if } z_j^* \neq 0\}$, so $p \in \mathcal{C}_\phi(x^*, z^*)$ if and only if $p = Q(x^*)^{1/2}\bar{p}$ for some $\bar{p} \in \mathbb{R}^n$.

Let $g(x) = \nabla f(x)$, $H(x) = \nabla^2 f(x)$, $g_i(x) = \nabla c_i(x)$, $H_i(x) = \nabla^2 c_i(x)$, and define

$$\begin{aligned} g_{\scriptscriptstyle L}(x, y) &:= g(x) - A(x)y, & g_\sigma(x) &:= g_{\scriptscriptstyle L}(x, y_\sigma(x)), \\ H_{\scriptscriptstyle L}(x, y) &:= H(x) - \sum_{i=1}^m y_i H_i(x), & H_\sigma(x) &:= H_{\scriptscriptstyle L}(x, y_\sigma(x)) \end{aligned} \tag{10}$$

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

$$R(x, v) := \nabla_x[Q(x)v] = \nabla_x \begin{bmatrix} q_1(x_1)v_1 \\ \vdots \\ q_n(x_n)v_n \end{bmatrix} = \operatorname{diag}\left(\begin{bmatrix} q_1'(x_1)v_1 \\ \vdots \\ q_n'(x_n)v_n \end{bmatrix} \right),$$

$$S(x,v) := \nabla_x[A(x)^T v] = \nabla_x \begin{bmatrix} g_1(x)^T v \\ \vdots \\ g_m(x)^T v \end{bmatrix} = \begin{bmatrix} v^T H_1(x) \\ \vdots \\ v^T H_m(x) \end{bmatrix},$$
$$T(x,w) := \nabla_x[A(x)w] = \nabla_x \left[\sum_{i=1}^m w_i g_i(x)\right] = \sum_{i=1}^m w_i H_i(x),$$

where $v \in \mathbb{R}^n$, $w \in \mathbb{R}^m$, and T 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 $A_Q(x) = Q(x)^{1/2}A(x)$ has full rank m, the operators

$$P(x) := A_Q(x)\big(A_Q(x)^T A_Q(x)\big)^{-1} A_Q(x)^T \quad \text{and} \quad \bar{P}(x) := I - P(x) \tag{11}$$

define orthogonal projectors onto range$(A_Q(x))$ and its complement respectively. More generally, for a matrix M, we define P_M and $\bar{P}_M$ as the orthogonal projectors onto range(M) and null(M) respectively.

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 of size dictated by the context.

3 Related work on penalty functions for inequality constraints

Penalty functions have long been used to solve constrained problems by replacing constraints with functions that penalize infeasibility. Estrin et al. (2019, §1.1) give an overview of other smooth exact penalty methods for equality constrained optimization and their relation to (PP). A more detailed overview is given by Di Pillo and Grippo (1984), Conn et al. (2000), and Nocedal and Wright (2006).

When $\ell = 0$ and $u = \infty$, Fletcher (1973b) proposes the penalty function

$$\psi_\sigma(x) := f(x) - c(x)^T y_\sigma(x) - z_\sigma(x)^T x,$$
$$y_\sigma(x), z_\sigma(x) := \operatorname*{arg\,min}_{y\in\mathbb{R}^m,\, z\geq 0} \tfrac{1}{2}\|A(x)y + z - g(x)\|_2^2 + \sigma c(x)^T y$$

and minimizes ψ_σ unconstrained. Although ψ_σ is exact and continuous, it is nonsmooth because of the bound constraints on z: active-set changes on those bounds correspond to non-differentiable points for ψ_σ. Solving the penalty problem requires a method for nonsmooth problems, and Maratos (1978) observes that nonsmooth merit functions may result in slow convergence.

Since Fletcher (1973a), there has been significant work on smooth exact penalty methods that handle inequality constraints (Di Pillo and Grippo, 1984, 1985; Boggs et al., 1992; Zavala and Anitescu, 2014). Many approaches replace the inequality constraints with equalities using squared slacks (Bertsekas, 1982), at which point the equality constrained problem is solved via a smooth exact penalty approach. (This is one approach for deriving ϕ_σ and (3); however, it is also possible to derive it directly from the first-order KKT conditions.) The penalty function in these cases is the augmented Lagrangian, which either keeps the dual variables explicit and penalizes the gradient of the Lagrangian (Zavala and Anitescu, 2014), or expresses the dual variables as a function of x (Di Pillo and Grippo, 1984). Our penalty function (2) takes the latter approach but defines this parametrization differently from previous approaches; rather than introducing additional dual variables for the bounds in (3), we change the norm of the least-squares problem according to the distance from the bounds, to approximate the complementarity conditions of first-order KKT points.

4 Properties of the penalty function

In this section, we show how $\phi_\sigma(x)$ naturally expresses the optimality conditions of (NP). We also give explicit expressions for the threshold value of the penalty parameter σ.

As in (Estrin et al., 2019), the gradient and Hessian of ϕ_σ may be written as

$$\nabla\phi_\sigma(x) = g_\sigma(x) - Y_\sigma(x)c(x), \tag{12a}$$
$$\nabla^2\phi_\sigma(x) = H_\sigma(x) - A(x)Y_\sigma(x)^T - Y_\sigma(x)A(x)^T - \nabla_x\left[Y_\sigma(x)c\right], \tag{12b}$$

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.

The penalty function ϕ_σ is closely related to the (partial) Lagrangian (1). To make this connection clear, we define the Karush-Kuhn-Tucker (KKT) optimality conditions for (NP) in terms of those of (PP). From the definition of ϕ_σ and y_σ and (12), we have the following definition.

Definition 1 (First-order KKT points of (NP)) *The point (x^*, z^*) is a first-order KKT point of (NP) if for any $\sigma \geq 0$ the following hold:*

$$\ell \leq x^* \leq u, \tag{13a}$$
$$c(x^*) = 0, \tag{13b}$$
$$\nabla\phi_\sigma(x^*) = z^*, \tag{13c}$$
$$z_j^* = 0, \qquad \text{if } j \notin \mathcal{A}(x^*), \tag{13d}$$
$$z_j^* \geq 0, \qquad \text{if } x_j^* = \ell_j, \tag{13e}$$
$$z_j^* \leq 0, \qquad \text{if } x_j^* = u_j. \tag{13f}$$

Then $y^* := y_\sigma(x^*)$ is the Lagrange multiplier of (NP) associated with x^*. Note that by (A3), inequalities (13e) and (13f) are strict.

Remark 1 *If (13) holds for some $\sigma \geq 0$, it necessarily holds for all $\sigma \geq 0$ because $c(x^*) = 0$. Also, the point (x^*, z^*) is a first-order KKT point of (PP) if for any $\sigma \geq 0$, (13a) and (13c)–(13f) hold.*

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

$$p^T\nabla^2\phi_\sigma(x^*)p \geq 0 \quad \text{for all } p \in \mathcal{C}(x^*, z^*). \tag{14}$$

Condition (14) is sufficient if the inequality is strict.

Remark 2 *If (13b) is omitted, Definition 1 corresponds to first-order KKT points of (PP). Similarly, replacing $\mathcal{C}(x^*, z^*)$ by $\mathcal{C}_\phi(x^*, z^*)$ in Definition 2 corresponds to second-order KKT points of (PP).*

The second-order KKT condition says that at a second-order KKT point of (PP), ϕ_σ has nonnegative curvature along directions in the critical cone $\mathcal{C}_\phi(x^*, z^*)$. We now show that at x^*, increasing σ increases curvature only along the normal cone to the equality constraints. We derive a threshold value for σ beyond which that ϕ_σ has nonnegative curvature even when $A(x^*)^Tp \neq 0$, as well as a condition on σ that ensures that stationary points of (PP) are primal feasible. For a given first- or second-order KKT triple (x^*, y^*, z^*) of (NP), we define

$$\sigma^* := \tfrac{1}{2}\lambda_{\max}^+\left(P(x^*)Q(x^*)^{1/2}H_L(x^*, y^*)Q(x^*)^{1/2}P(x^*)\right), \tag{15}$$

where $\lambda_{\max}^+(\cdot)$ is the maximum of the largest eigenvalue and zero. The following lemmas are similar to those of Estrin et al. (2019). Indeed, if the bounds are absent then $Q(x) = I$ and we recover the same results as in Estrin et al. (2019).

Lemma 1 *If* $c(x) \in \text{range}\left(A(x)^T Q(x)\right)$, *then* $y_\sigma(x)$ *satisfies*

$$A(x)^T Q(x) A(x) y_\sigma(x) = A(x)^T Q(x) g(x) - \sigma c(x). \tag{16}$$

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

$$\begin{aligned} &A(x)^T Q(x) A(x) Y_\sigma(x)^T \\ &\quad = A(x)^T [Q(x) H_\sigma(x) - \sigma I + R(x, g_\sigma(x))] + S(x, Q(x) g_\sigma(x)). \end{aligned} \tag{17}$$

Proof. For any x, the necessary and sufficient optimality conditions for (3) give (16). For brevity, let everything be evaluated at the same point x and drop the argument x from all operators. By differentiating both sides of (16), we obtain

$$S(QAy_\sigma) + A^T\left[R(Ay_\sigma) + QT(y_\sigma) + QAY_\sigma^T\right] = S(Qg) + A^T[R(g) + QH - \sigma I]\,.$$

By rearranging the above and using definitions (10), we obtain (17). □

Theorem 1 (Threshold penalty value) *Suppose* $(\bar{x}, \bar{z})$ *is a first-order KKT point for* (PP) *with* $Q(\bar{x})^{1/2} A(\bar{x})$ *full-rank, and let* (x^*, y^*, z^*) *be a second-order necessary KKT point for* (NP). *Then*

$$\sigma > \|A(\bar{x})^T Q(\bar{x}) Y_\sigma(\bar{x})\| \quad \Longrightarrow \quad c(\bar{x}) = 0; \tag{18a}$$

$$p^T \nabla^2 \phi_\sigma(x^*) p \geq 0 \quad \textit{for all} \quad p \in \mathcal{C}_\phi(x^*, z^*) \quad \Longleftrightarrow \quad \sigma \geq \sigma^*, \tag{18b}$$

where σ^* *is defined in* (15). *The consequence of* (18a) *is that* $\bar{x}$ *is a first-order KKT point for* (NP). *If* x^* *is second-order sufficient, the inequalities in* (18b) *hold strictly.*

Proof. Proof of (18a): By (13c)–(13f), $Q(\bar{x})\nabla\phi_\sigma(\bar{x}) = 0$, so that

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

Substituting (16) evaluated at $\bar{x}$ into this equation yields, after simplifying,

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

Taking norms of both sides and using the triangle inequality gives the inequality $\sigma\|c(\bar{x})\| \leq \|A(\bar{x})^T Q(\bar{x}) Y_\sigma(\bar{x})\|\,\|c(\bar{x})\|$, which implies that $c(\bar{x}) = 0$.

Proof of (18b): Because x^* satisfies first-order conditions (13), we have $y^* = y_\sigma(x)$ and $Q(x^*)g_\sigma(x^*) = 0$, independently of σ. Therefore $S(x, Q(x)g_\sigma(x)) = 0$. We drop the arguments from operators that take x as input and assume that they are all evaluated at x^*. By premultiplying (17) by $(A_Q^\dagger)T = Q^{1/2}A(A^TQA)^{-1}$ and postmultiplying by $Q^{1/2}$, using $H_{_L}(x^*, y^*) = H_\sigma$, and the definition of $P := P(x^*)$, we have

$$Q^{1/2} A Y_\sigma^T Q^{1/2} = (A_Q^\dagger)^T A (Q H_{_L}(x^*, y^*) Q^{1/2} - \sigma I + R(g_\sigma)) Q^{1/2} \tag{19}$$

$$= P Q^{1/2} H_{_L}(x^*, y^*) - \sigma P + (A_Q^\dagger)^T A R(g_\sigma) Q^{1/2} \tag{20}$$

Observe that if $p \in \mathcal{C}_\phi(x^*, z^*)$, then $p = Q^{1/2}\bar{p}$ for some $\bar{p} \in \mathcal{C}_\phi(x^*, z^*)$. Because $Q^{1/2} g_\sigma = 0$, we have $R(g_\sigma)Q^{1/2} = 0$. Therefore using (12b), (20), and the relation $P + \bar{P} = I$, we have

$$\begin{aligned} p^T \nabla^2 \phi_\sigma(x^*) p \geq 0 \;&\Longleftrightarrow\; \bar{p}^T Q^{1/2} \left(H_\sigma - A Y_\sigma^T - Y_\sigma A^T\right) Q^{1/2} \bar{p} \geq 0 \\ &\Longleftrightarrow\; \bar{p}^T \left(\bar{P} Q^{1/2} H_\sigma Q^{1/2} \bar{P} - P Q^{1/2} H_\sigma Q^{1/2} P + 2\sigma P\right) \bar{p} \geq 0. \end{aligned}$$

Now, because $\bar{P}\bar{p} \in \text{null}(A^T Q^{1/2})$ implies that $Q^{1/2}\bar{P}\bar{p} \in \mathcal{C}(x^*, z^*)$, the first term above is nonnegative according to Definition 2. It follows that σ must be sufficiently large that $2\sigma P - P Q^{1/2} H_{_L}(x^*, y^*) Q^{1/2} P \succeq 0$, which is equivalent to $\sigma \geq \sigma^*$. □

As in Estrin et al. (2019, Theorem 4), (18b) shows that if x^* is a second-order KKT point of (NP), there exists a threshold value σ^* beyond which x^* is also a second-order KKT point of (PP). Note that this result does not preclude the possibility that there exist minimizers of the penalty function—for any value of σ—that are not minimizers of (NP). However, these are rarely encountered in practice. Further, we can add a quadratic penalty term that, under certain conditions, ensures that KKT points of (PP) are feasible for (NP) (Estrin et al., 2019, §3.4).

5 Evaluating the penalty function

The main challenge in evaluating ϕ_σ and its gradient is the solution of the shifted weighted-least-squares problem (3) needed to compute $y_\sigma(x)$, and computation of the gradient $Y_\sigma(x)$. We show below that it is possible to compute matrix-vector products $Y_\sigma(x)v$ and $Y_\sigma(x)^T u$ by solving structured linear systems involving the same matrix. We show that this linear system may be either symmetric or unsymmetric, and discuss the tradeoffs between both approaches. In either case, if direct methods are to be used, only a single factorization that defines the solution (3) is required for all products.

For this section, it is convenient to drop the arguments on various functions and assume they are all evaluated at a point x for some parameter σ. For example, $y_\sigma = y_\sigma(x)$, $A = A(x)$, $Y_\sigma = Y_\sigma(x)$, $H_\sigma = H_\sigma(x)$, $S_\sigma = S(x, Q(x)g_\sigma(x))$, $R_\sigma = R(x, g_\sigma(x))$, etc. We express (17) using the shorthand notation

$$A^T Q A Y_\sigma^T = A^T(QH_\sigma - \sigma I + R_\sigma) + S_\sigma. \tag{21}$$

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

Every quantity of interest can be computed by solving a symmetric or unsymmetric linear system and combining the solution with the derivatives of the problem data. Typically it is preferable to solve symmetric systems; however, we find that additional Jacobian products are then needed. The additional cost may be negligible, but this matter becomes application-dependent. We therefore present both options, beginning with the symmetric case.

There are many ways to construct the right-hand sides of the linear systems presented below. One consideration is that inversions with the diagonal matrix $Q^{1/2}$ should be avoided—even though the diagonal of Q will be assumed strictly positive because of the use of an interior method (see Section 7), numerical difficulties may arise near the boundary of the feasible set if $Q^{1/2}$ contains small entries and is inverted.

5.1 Computing $\mathbf{Y_\sigma u}$

It follows from (21) that for a given m-vector u,

$$Y_\sigma u = (H_\sigma Q - \sigma I + R_\sigma)A(A^T Q A)^{-1}u + S_\sigma^T (A^T Q A)^{-1}u.$$

Let $w = -(A^T Q A)^{-1}u$ and $v = -Q^{1/2}Aw$, so that v and w are the solution of the symmetric linear system

$$\begin{bmatrix} I & Q^{1/2}A \\ A^T Q^{1/2} & \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ u \end{bmatrix}. \tag{22}$$

Then $Y_\sigma u = H_\sigma Q^{1/2} v + (\sigma I - R_\sigma)Aw - S_\sigma^T w$. Algorithm 1 formalizes this process.

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

1: $(v, w) \leftarrow$ solution of (22)
2: **return** $H_\sigma Q^{1/2} v + (\sigma I - R_\sigma)Aw - S_\sigma^T w$

5.2 Computing $Y_\sigma^T v$

Again from (21), multiplying both sides by v gives

$$Y_\sigma^T v = (A^T Q A)^{-1} A^T (Q H_\sigma - \sigma I + R_\sigma) v + (A^T Q A)^{-1} S_\sigma v.$$

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

$$\begin{bmatrix} I & Q^{1/2} A \\ A^T Q^{1/2} & \end{bmatrix} \begin{bmatrix} r \\ u \end{bmatrix} = \begin{bmatrix} Q^{1/2} H_\sigma v \\ A^T(\sigma I - R_\sigma) v - S_\sigma v \end{bmatrix}. \tag{23}$$

Algorithm 2 formalizes the process.

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

1: Evaluate $Q^{1/2} H_\sigma v$ and $A^T(\sigma I + R_\sigma)v - S_\sigma v$
2: $(r, u) \leftarrow$ solution of (23)
3: **return** u

5.3 Unsymmetric linear system

We briefly comment on how to use unsymmetric systems in place of (22) and (23). We can compute products of the form $Y_\sigma u = (H_\sigma - \sigma I + R_\sigma)\bar{v} - S_\sigma^T w$ (where $w = -(A^T Q A)^{-1} u$ and $\bar{v} = -Aw$), and products $u = Y_\sigma^T v$ by solving the respective linear systems:

$$\begin{bmatrix} I & A \\ A^T Q & \end{bmatrix} \begin{bmatrix} \bar{v} \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ u \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} I & QA \\ A^T & \end{bmatrix} \begin{bmatrix} \bar{r} \\ u \end{bmatrix} = \begin{bmatrix} (QH_\sigma - \sigma I - R_\sigma)v \\ -S_\sigma v \end{bmatrix}. \tag{24}$$

Algorithms 1 and 2 can then be appropriately modified to use the above linear systems.

5.4 Computing multipliers and first derivatives

The multiplier estimates y_σ and Lagrangian gradient can be obtained from one of the following linear systems:

$$\begin{bmatrix} I & Q^{1/2} A \\ A^T Q^{1/2} & \end{bmatrix} \begin{bmatrix} d \\ y_\sigma \end{bmatrix} = \begin{bmatrix} Q^{1/2} g \\ \sigma c \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} I & A \\ A^T Q & \end{bmatrix} \begin{bmatrix} g_\sigma \\ y_\sigma \end{bmatrix} = \begin{bmatrix} g \\ \sigma c \end{bmatrix}. \tag{25}$$

Observe that in the unsymmetric case we obtain g_σ immediately. The symmetric system yields $d = Q^{1/2} g_\sigma$. As noted earlier, computing $g_\sigma \leftarrow Q^{-1/2} d$ may be inaccurate. An alternative would be to compute $g_\sigma \leftarrow g - A y_\sigma$, which costs an extra Jacobian product.

The penalty gradient $\nabla \phi_\sigma = g_\sigma - Y_\sigma c$ can then be computed using g_σ and computing $Y_\sigma c$ via Algorithm 1 or its unsymmetric variant.

5.5 Computing second derivatives

We approximate $\nabla^2 \phi_\sigma$ from (12b) using the same approaches as Estrin et al. (2019):

$$\begin{aligned} \nabla^2 \phi_\sigma \approx B_1 &:= H_\sigma - A Y_\sigma^T - Y_\sigma A^T && \text{(26a)} \\ &= H_\sigma - \widetilde{P}(QH_\sigma + R_\sigma - \sigma I) - (H_\sigma Q + R_\sigma - \sigma I)\widetilde{P} \\ &\quad - A(A^T Q A)^{-1} S_\sigma - S_\sigma^T (A^T Q A)^{-1} A \\ &\approx B_2 := H_\sigma - \widetilde{P}(QH_\sigma + R_\sigma - \sigma I) - (H_\sigma Q + R_\sigma - \sigma I)\widetilde{P}, && \text{(26b)} \end{aligned}$$

where $\widetilde{P} = A(A^T Q A)^{-1} A$. The first approximation drops the third derivative term $\nabla[Y_\sigma c]$ in (12b), while the second approximation drops the term $S_\sigma(x, Q g_\sigma)$, because those terms are zero at a solution. Thus, B_1 and B_2 can be interpreted as Gauss-Newton approximations of $\nabla^2 \phi_\sigma$. Using similar arguments

to those made by Fletcher (1973a, Theorem 2), we expect those approximations to result in quadratic convergence when $f, c \in \mathcal{C}_3$, and superlinear convergence when $f, c \in \mathcal{C}_2$.

Computing products with B_1 only requires products with Y_σ and Y_σ^T, which can be handled by Algorithms 1 and 2. To compute a product $\widetilde{P}u$, we can solve

$$\begin{bmatrix} I & Q^{1/2}A \\ A^TQ^{1/2} & \end{bmatrix} \begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} 0 \\ A^Tu \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} I & A \\ A^TQ & \end{bmatrix} \begin{bmatrix} \bar{p} \\ q \end{bmatrix} = \begin{bmatrix} u \\ 0 \end{bmatrix}, \qquad \widetilde{P}u = -Aq. \tag{27}$$

As before, using the unsymmetric system avoids an additional Jacobian product, which may be negligible compared to solving an unsymmetric system.

5.6 Solving the augmented linear system

We comment on various approaches for solving the necessary linear systems

$$\mathcal{K} \begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} w \\ z \end{bmatrix}, \quad \text{where} \quad \mathcal{K} = \begin{bmatrix} I & Q^{1/2}A \\ A^TQ^{1/2} & \end{bmatrix} \text{ or } \begin{bmatrix} I & A \\ A^TQ & \end{bmatrix}. \tag{28}$$

This is the most computationally intensive step in the penalty approach. Note that when direct methods are used, a single factorization is needed to evaluate ϕ_σ and its (approximate) derivatives.

Estrin et al. (2019, §4.5) describe several approaches for solving the symmetric system (using both direct and iterative methods), so we do not repeat this discussion here. For unsymmetric systems, any sparse factorization of $\mathcal{K}$ may be used; also, we could factorize $Q^{1/2}A$ with a Q-less QR factorization and use the (refined) semi-normal equations (Björck and Paige, 1994) as in the symmetric case (as long as multiplications with $Q^{-1/2}$ are avoided).

If iterative methods are used, the unsymmetric system requires unsymmetric iterative methods such as GMRES (Saad and Schultz, 1986), SPMR (Estrin and Greif, 2018), or QMR (Freund and Nachtigal, 1991), where the choice of method depends on considerations such as short- vs. long-recurrence, available preconditioners, or robustness. Note that preconditioners approximating $\mathcal{P} \approx A^TQA$ apply to both the symmetric and unsymmetric systems; however, unsymmetric solvers may allow inexact preconditioner solves, while short-recurrence symmetric solvers may not.

If optimization solvers that accept inexact function and derivative evaluations are used (e.g., Conn et al. (2000, §8–9) or Heinkenschloss and Ridzal (2014)), the results of Estrin et al. (2019, §7) apply here as well; that is, bounding the residual norm of the linear systems is sufficient to bound the function and derivative evaluation error up to a constant (under mild assumptions). This is useful in cases where solving the linear system exactly every iteration is prohibitively expensive. Further, when the symmetric system is used, it is possible to use methods that upper bound the solution error. For example, Arioli (2013) develop error bounds for CRAIG (Craig, 1955), and Estrin et al. (2018) develop error bounds for LNLQ when an underestimate of the smallest singular value of the preconditioned Jacobian is available.

6 Maintaining explicit constraints

We consider a variation of (NP) where some of the constraints $c(x)$ are easy to maintain explicitly; for example, linear equality constraints. We show below that maintaining subsets of constraints explicitly decreases the threshold penalty parameter σ^* in Theorem 1. Instead of (NP), consider the problem with explicit linear equality constraints

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) \quad \text{subject to} \quad c(x) = 0, \quad B^Tx = d, \quad \ell \leq x \leq u, \tag{NP-EXP}$$

where $c(x) \in \mathbb{R}^{m_1}$ and $B^T x = d$ with $B \in \mathbb{R}^{n \times m_2}$, so that $m_1 + m_2 = m$. We assume that (NP-EXP) at least satisfies (A2), so that B has full column rank. We define the penalty problem as

$$\begin{aligned} &\underset{x \in \mathbb{R}^n}{\text{minimize}} \;\; \phi_\sigma(x) := f(x) - c(x)^T y_\sigma(x) \;\; \text{subject to} \;\; B^T x = d, \;\; \ell \le x \le u, \\ &\begin{bmatrix} y_\sigma(x) \\ w_\sigma(x) \end{bmatrix} := \underset{y,w}{\arg\min} \, \tfrac{1}{2} \|A(x)y + Bw - g(x)\|^2_{Q(x)} + \sigma \begin{bmatrix} c(x) \\ B^T x - d \end{bmatrix}^T \begin{bmatrix} y \\ w \end{bmatrix}, \end{aligned} \tag{29}$$

which is similar to (PP) except that the linear constraints are not penalized in $\phi_\sigma(x)$, and the linear constraints are explicitly present. Another possibility is to penalize the linear constraints as well, while keeping them explicit; however, this introduces additional nonlinearity in ϕ_σ. Further, if all constraints are linear, it is desirable for the penalty function to reduce to (NP-EXP).

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

$$\sigma^* := \tfrac{1}{2} \lambda^+_{\max} \left(\bar{P}_{Q^{1/2}B} P_{Q^{1/2}C} Q^{1/2} H_L(x^*, y^*) Q^{1/2} P_{Q^{1/2}C} \bar{P}_{Q^{1/2}B} \right) \tag{30}$$

$$\le \tfrac{1}{2} \lambda^+_{\max} \left(P_{Q^{1/2}C} Q^{1/2} H_L(x^*, y^*) Q^{1/2} P_{Q^{1/2}C} \right), \tag{31}$$

where $Q := Q(x^*)$, $C := \begin{bmatrix} A(x^*) & B \end{bmatrix}$ is the Jacobian for all constraints. Inequality (31) holds because $\bar{P}_{Q^{1/2}B}$ is an orthogonal projector. If the linear constraints were not explicit, the threshold value would be (31). Intuitively, the threshold penalty value decreases because positive semidefiniteness of $\nabla^2 \phi_\sigma(x^*)$ is only required on a lower-dimensional subspace.

The following result is analogous to Theorem 1 with the smaller threshold value.

Theorem 2 (Threshold penalty value with explicit constraints) *Suppose $(\bar{x}, \bar{z})$ is a first-order necessary KKT point for Equation (29), and let (x^*, y^*, z^*) be a second-order necessary KKT point for (NP-EXP). Define $\mathcal{C}^*_\phi := \mathcal{C}_\phi(x^*, z^*) \cap \text{null}(B^T)$, $Q := Q(\bar{x})$, and $\bar{P}_{Q^{1/2}B} := \bar{P}_{Q^{1/2}B}(\bar{x})$. Then*

$$\sigma > \|A(\bar{x})^T Q^{1/2} \bar{P}_{Q^{1/2}B} Q^{1/2} Y_\sigma(\bar{x})\| \quad \Longrightarrow \quad c(\bar{x}) = 0; \tag{32a}$$

$$p^T \nabla^2 \phi_\sigma(x^*) p \succeq 0 \quad \textit{for all} \quad p \in \mathcal{C}^*_\phi \quad \Longleftrightarrow \quad \sigma \ge \sigma^*, \tag{32b}$$

where σ^* is defined in (30). The consequence of (32a) is that $\bar{x}$ is a KKT point for (NP). If x^* is second-order sufficient, the inequalities in (32b) hold strictly.

The proof of the theorem, and details of evaluating the penalty function with explicit constraints, are given in Appendix A. Although we only considered the linear case here, explicit nonlinear constraints can be handled with minor modifications.

7 Practical considerations

So far we have demonstrated that for sufficiently large σ, minimizers of (NP) are minimizers of (PP), and we showed how to evaluate ϕ_σ and its derivatives. By (A2) we know that ϕ_σ is defined for all $\ell < x < u$. Although it may appear that any optimization solver can be applied to minimize (PP), the structure of ϕ_σ lends itself more readily to certain types of solvers.

First, we recommend interior solvers rather than exterior or active-set methods. For $\phi_\sigma(x)$ to be defined, we require that $Q(x) \succeq 0$ (thus disqualifying exterior point methods) and that $Q(x)^{1/2} A(x)$ have full column-rank (so that at most $n - m$ components of x can be at one of their bounds). Even if (A2) is satisfied, an active-set method may choose a poor active set that causes $\phi_\sigma(x)$ to be undefined (or it may have too many active bounds). On the other hand, interior methods ensure that $Q(x) \succ 0$ and avoid this issue (at least until x converges and approaches the bounds).

As in (Estrin et al., 2019), Newton-CG type trust-region solvers (Steihaug, 1983) should be used to minimize (PP). Products with approximations of $\nabla^2 \phi_\sigma(x)$ can be efficiently computed, but computing

the Hessian itself is not practical. Also, trust-region methods are better equipped to deal with negative curvature than linesearch methods (ϕ_σ typically has an indefinite Hessian). Finally, evaluating ϕ_σ at several points (such as during a linesearch) is expensive because every evaluation requires solving a different linear system. Given these considerations, a solver like KNITRO (Byrd et al., 2006) is ideal for solving (PP).

It remains future work to determine a robust procedure for updating σ if it is too small (causing ϕ_σ to be unbounded) or too large (causing small steps to be taken). For the following experiments, we choose an initial σ specific to each problem and keep it constant. We also have the same heuristic available that is discussed by Estrin et al. (2019, §8) to update σ, which often works in practice.

8 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

$$\begin{aligned} \|c(x)\|_\infty &\le \epsilon_p \\ \|N(x) g_\sigma(x)\|_\infty &\le \epsilon_d \end{aligned} \qquad \text{or} \qquad \|N(x)\nabla\phi_\sigma(x)\|_\infty \le \epsilon_d, \tag{33}$$

with $N(x) = \operatorname{diag}(\min\{x - \ell, u - x, \mathbb{1}\})$, $\epsilon_p := \epsilon\,(1 + \|x\|_\infty + \|c(x_0)\|_\infty)$, and $\epsilon_d := \epsilon\,(1 + \|y\|_\infty + \|g_\sigma(x_0)\|_\infty)$, where x_0 is the initial point, $y_0 = y_0(x_0)$, and $\epsilon = 10^{-8}$.

For the standard test problems, we use the semi-normal equations with one step of iterative refinement (Björck and Paige, 1994). For the PDE-constrained problems, we use LNLQ with the CRAIG transfer point (Estrin et al., 2018; Craig, 1955; Arioli, 2013) to solve the symmetric augmented system (28) with preconditioner $\mathcal{P}$ and two possible termination criteria:

$$\left\| \begin{bmatrix} p^* \\ q^* \end{bmatrix} - \begin{bmatrix} p^{(k)} \\ q^{(k)} \end{bmatrix} \right\|_{\bar{\mathcal{P}}} \le \eta \left\| \begin{bmatrix} p^{(k)} \\ q^{(k)} \end{bmatrix} \right\|_{\bar{\mathcal{P}}}, \qquad \bar{\mathcal{P}} := \begin{bmatrix} I & \\ & \mathcal{P} \end{bmatrix}, \tag{34a}$$

$$\left\| \mathcal{K} \begin{bmatrix} p^{(k)} \\ q^{(k)} \end{bmatrix} - \begin{bmatrix} u \\ v \end{bmatrix} \right\|_{\bar{\mathcal{P}}^{-1}} \le \eta \left\| \begin{bmatrix} u \\ v \end{bmatrix} \right\|_{\bar{\mathcal{P}}^{-1}}, \tag{34b}$$

which are based on the relative residual and the relative error (obtained via LNLQ (Estrin et al., 2018)), respectively. We can use (34a) when a lower bound on $\sigma_{\min}(\mathcal{P}^{-1/2}A)$ is available, which is the case in the PDE-constrained optimization problems below.

We use KNITRO (Byrd et al., 2006) to solve (PP). When ϕ_σ is evaluated approximately (when η is large), we use such solvers without modification, thus pretending that the function and gradient are evaluated exactly. The use of inexact linear solves is discussed in (Estrin et al., 2019, §7); the following experiments using inexactness are similar to those in (Estrin et al., 2019, §9).

8.1 2D inverse Poisson problem

Let $\Omega = (0,1)^2$ represent the physical domain and $H^1(\Omega)$ denote the Sobolev space of functions in $L^2(\Omega)$, whose weak derivatives are also in $L^2(\Omega)$. Let $H^1_0(\Omega) \subset H^1(\Omega)$ be the Hilbert space of functions whose value on the boundary $\partial\Omega$ is zero. We solve the following 2D PDE-constrained control problem:

$$\begin{aligned} \underset{u \in H^1_0(\Omega),\, z \in L^2(\Omega)}{\text{minimize}} \quad & \tfrac{1}{2}\int_\Omega (u - u_d)^2 \,\mathrm{d}x + \tfrac{1}{2}\alpha \int_\Omega z^2 \,\mathrm{d}x \\ \text{subject to} \quad & -\nabla\cdot(z\nabla u) = h \quad \text{in } \Omega, \\ & u = 0 \quad \text{on } \partial\Omega, \\ & z \ge 0 \quad \text{in } \Omega. \end{aligned} \tag{35}$$

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

The force term 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 Poisson diffusion coefficients that we are trying to recover from the observed state u_d. We set $\alpha = 10^{-4}$ as the regularization parameter. The problem is almost identical to that of Estrin et al. (2019, §9.2) but with an additional bound constraint on the control variables (to ensure positivity of the diffusion coefficients).

We discretize (35) in two ways using P_1 finite elements on a uniform mesh of 1089 (resp. 10201) triangular elements and employ an identical discretization for the optimization variables $z \in L^2(\Omega)$, obtaining a problem with $n_z = 1089$ ($n_z = 10201$) controls and $n_u = 961$ ($n_u = 9801$) 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.

We compute $x = (u, z)$ by applying KNITRO to (PP) with $\sigma = 10^{-2}$, using $B_2(x)$ as the Hessian approximation (26b) and initial point $u_0 = \mathbb{1}$, $z_0 = \mathbb{1}$. We partition the Jacobian of the discretized constraints as $A(x)^T = \begin{bmatrix} A_u(x)^T & A_z(x)^T \end{bmatrix}$, where $A_u(x) \in \mathbb{R}^{n \times n}$, $A_z(x) \in \mathbb{R}^{m \times n}$ are the Jacobians for variables u, z respectively. We use the preconditioner $\mathcal{P}(x) = A_u(x)^T A_u(x)$, which amounts to performing two solves of a variable-coefficient Poisson equation. For this preconditioner, because the only bound constraints are $z \geq 0$, $Q(x) = \text{blkdiag}(I, Z)$ with $Z = \text{diag}(z)$, so that

$$\begin{aligned} \mathcal{P}^{-1} A(x)^T Q(x) A(x) &= \mathcal{P}^{-1}(A_u(x)^T A_u(x) + A_z(x) Z A_z(x)) \\ &= I + \mathcal{P}^{-1} A_z(x) Z A_z(x). \end{aligned}$$

Thus $\sigma_{\min}(A(x)\mathcal{P}^{-1/2}) \geq 1$, allowing us to bound the error via LNLQ and to use both (34b) and (34a) as termination criteria.

We choose $\epsilon = 10^{-8}$ in the stopping conditions (33). In Table 1 we vary η to control the accuracy of the linear system solves (34), and we record the number of Hessian- and Jacobian-vector products.

Table 1: Results from solving (35) using KNITRO to solve (PP) with various η in (34a) (left) and (34b) (right) to terminate the linear system solves. The top (resp. bottom) table records results for the smaller problem with $n = 2050$, $m = 1089$ (resp. larger problem with $n = 20002$, $m = 10201$). We record the number of function/gradient evaluations ($\#f, g$), Lagrangian Hessian ($\#Hv$), Jacobian ($\#(Av)$, and adjoint Jacobian ($\#A^Tv$) products.

η	Its.	$\#f, g$	$\#Hv$	$\#Av$	$\#A^Tv$	Its.	$\#f, g$	$\#Hv$	$\#Av$	$\#A^Tv$
10^{-2}	46	64	2856	8436	8611	67	81	4374	12915	13145
10^{-4}	43	55	2168	6642	6796	36	51	1458	4642	4781
10^{-6}	35	46	2120	6876	7004	29	35	1194	4138	4238
10^{-8}	39	50	2322	7833	7973	47	71	7062	22150	22340
10^{-10}	37	47	2236	8110	8242	43	58	3170	11565	11725
10^{-2}	144	176	3662	12395	12892	100	126	3716	11702	12055
10^{-4}	131	177	4002	14470	14956	83	117	2752	9264	9582
10^{-6}	103	135	4386	15035	15409	88	132	4170	14421	14774
10^{-8}	73	103	3250	11960	12244	101	133	3726	13878	14246
10^{-10}	79	109	4088	15527	15825	104	139	5378	20291	20674
		error-based termination					residual-based termination			

We observed that for the smaller problem, KNITRO converged in a moderate number of outer iterations in all cases. When (34a) was used, we see that the number of Jacobian products tended to decrease, except when $\eta = 10^{-2}$ (for which the linear solves were too inaccurate). Using (34b) showed a less clear trend. In cases with comparable outer iteration numbers, larger η resulted in fewer Jacobian products. However, for moderate η the number of outer iterations proved to be significantly smaller, resulting in a more efficient solve than when η was too small or too large.

For the larger problem with termination condition (34a), the number of outer iterations increased with η, the number of Lagrangian Hessian products fluctuated somewhat, and Jacobian products tended to decrease. The exception was $\eta = 10^{-8}$, which hit the sweet spot of solving the linear systems sufficiently accurately to avoid many additional outer iterations, but without performing too many iterations for each linear solve. Using residual-based termination (34b) showed a less clear trend; Jacobian products roughly decreased with increasing η while the Hessian products tended to oscillate. The sweet spot was hit with $\eta = 10^{-4}$, where the fewest outer iterations and operator products were performed. For this problem, it appears that the dependence of performance on the accuracy of the linear solves as measured by the residual (34b) is much more nonlinear than when the linear solves are terminated according to the error (34a).

8.2 2D Poisson-Boltzmann problem

We now solve a control problem where the constraint is a 2D Poisson-Boltzmann equation:

$$\begin{aligned} \underset{u\in H_0^1(\Omega),\ z\in L^2(\Omega)}{\text{minimize}} \quad & \tfrac{1}{2}\int_\Omega (u-u_d)^2\,\mathrm{d}x + \tfrac{1}{2}\alpha\int_\Omega z^2\,\mathrm{d}x \\ \text{subject to} \quad & -\Delta u + \sinh(u) = h + z \quad \text{in } \Omega, \\ & u = 0 \quad \text{on } \partial\Omega, \\ & z \geq 0 \quad \text{in } \Omega. \end{aligned} \tag{36}$$

We use the same notation and Ω as in Section 8.1, 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 discretized (36) using P_1 finite elements on two uniform meshes with 1089 (resp. 10201) triangular elements, resulting in a problem with $n = 2050$ ($n = 20002$) variables and $m = 961$ ($m = 9801$) constraints. The initial point was $u_0 = \mathbb{1}$, $z_0 = \mathbb{1}$.

We performed the same experiment as in Section 8.1 using $\sigma = 10^{-1}$, and recorded the results in Table 2. We see that the results for both problems are more robust to changes in the accuracy of the linear solves. In all cases, the number of outer iterations and function/gradient evaluations were the same, and the number of Lagrangian Hessian products changed little. The number of Jacobian products steadily decreased with increasing η, with a 20–30% drop in Jacobian products from $\eta = 10^{-10}$ to $\eta = 10^{-2}$.

Table 2: Results from solving (36) using KNITRO to optimize (PP) with various η in (34a) (left) and (34b) (right) to terminate the linear system solves. The top (resp. bottom) table records results for the smaller problem with $n = 2050$, $m = 1089$ (resp. larger problem with $n = 20002$, $m = 10201$). We record the number of function/gradient evaluations ($\#f, g$), Lagrangian Hessian ($\#Hv$), Jacobian ($\#(Av)$, and adjoint Jacobian ($\#A^Tv$) products.

η	Its.	$\#f,g$	$\#Hv$	$\#Av$	$\#A^Tv$	Its.	$\#f,g$	$\#Hv$	$\#Av$	$\#A^Tv$
10^{-2}	19	20	1242	3648	3708	19	20	1242	3669	3729
10^{-4}	19	20	1252	3753	3813	19	20	1244	3762	3822
10^{-6}	19	20	1236	3868	3928	19	20	1234	3916	3976
10^{-8}	19	20	1244	4169	4229	19	20	1236	4286	4346
10^{-10}	19	20	1238	4725	4785	19	20	1250	4986	5046

10^{-2}	30	37	1524	4426	4531	30	37	1524	4468	4573
10^{-4}	30	37	1524	4574	4679	30	37	1524	4632	4737
10^{-6}	30	37	1524	4813	4918	30	37	1558	5033	5138
10^{-8}	30	37	1550	5396	5501	30	37	1550	5610	5715
10^{-10}	30	37	1550	6224	6329	30	37	1558	6582	6687

error-based termination residual-based termination

8.3 2D topology optimization

We now solve the following 2D topology optimization problem from Gersborg-Hansen et al. (2006):

$$\begin{aligned} \underset{u\in H_0^1(\Omega),\, z\in L^2(\Omega)}{\text{minimize}} \quad & \int_\Omega f u \,\mathrm{d}x \\ \text{subject to} \quad & \int_\Omega z \,\mathrm{d}x \le V \\ & -\nabla\cdot(k(z)\nabla u) = f \quad \text{in } \Omega, \\ & u = 0 \quad \text{on } \partial\Omega, \\ & 0 \le z \le 1 \quad \text{in } \Omega, \end{aligned} \tag{37}$$

where $k(z) : \Omega \to \Omega$ defined by $k(z)(x) = 10^{-3} + (1 - 10^{-3})z(x)^3$ for $x \in \Omega$. The domain is $\Omega = [0, 1]^2$, with load vector $f = 10^{-2}$, and $V = 0.4$. We discretize (37) using finite elements on a 64×64 grid as described by Gersborg-Hansen et al. (2006), resulting in a problem with 8321 variables and 4096 equality constraints. After discretization, we add a slack variable $s \ge 0$ for the first inequality constraint, so we have only equality constraints and bounds. The final problem has $n = 8322$ variables and $m = 4096$ constraints, with bounds on z and s.

We perform the same experiment as in Section 8.1 (but only on one mesh), using $\sigma = 10^{-1}$ as the penalty parameter, with initial point $u_0 = \frac{1}{2}V\mathbb{1}$, $z_0 = \frac{1}{2}V\mathbb{1}$, and $s_0 = V - \sum z_i = 0.2$. The linear constraint is kept explicit as in Section 6. The results are recorded in Table 3. With (34a), the trend is like before: as η increases the number of Jacobian products decreases (and in this case, so do the numbers of outer iterations and Lagrangian Hessian products), but this is only true until η becomes too large and the linear solves become too coarse, causing slowed convergence. When (34b) was used, we see a similar trend, except that when the linear solves are too coarse, KNITRO fails to converge.

Table 3: Results from solving (37) using KNITRO to optimize (PP) with various η in (34a) (left) and (34b) (right) to terminate the linear system solves. We record the number of function/gradient evaluations ($\#f, g$), Lagrangian Hessian ($\#Hv$), Jacobian ($\#(Av)$, and adjoint Jacobian ($\#A^Tv$) products. The symbol "*" indicates that the problem failed to converge to a feasible point after 500 iterations.

	error-based termination					residual-based termination				
η	Its.	$\#f, g$	$\#Hv$	$\#Av$	$\#A^Tv$	Its.	$\#f, g$	$\#Hv$	$\#Av$	$\#A^Tv$
10^{-2}	217	340	4340	13966	15204	*	*	*	*	*
10^{-4}	226	348	4396	14068	15204	*	*	*	*	*
10^{-6}	176	272	3232	11218	12211	191	291	3508	18326	19391
10^{-8}	185	289	3356	11582	12635	196	296	3700	20888	21973
10^{-10}	204	298	4626	15412	16511	190	286	3480	23979	25028

8.4 Explicit linear constraints

We investigate the effect of maintaining the linear constraints explicitly (Section 6), using some problems from the CUTEst test set (Gould et al., 2003) that have linear constraints. We use KNITRO to minimize ϕ_σ with and without linear constraints, because it can handle them explicitly. We use the corrected semi-normal equations to perform linear solves, and Hessian approximation $B_1(x)$ (26a). The threshold penalty parameters (15) and (30) are computed from earlier optimal solutions when the linear constraints were kept implicit (σ^*_{impl}) and explicit (σ^*_{expl}) respectively. The results are recorded in Table 4.

We observe that maintaining the linear constraints explicitly decreases the penalty parameter for all problems except `Channel400` ($\sigma^* = 0$ in both cases). KNITRO fails to find an optimal solution when the linear constraints are implicit and $\sigma < \sigma^*_{\text{impl}}$. This is because in the equality-constrained case ϕ_σ is unbounded, and otherwise KNITRO stalls without converging to a feasible solution. When σ is sufficiently large, both versions converge (with and without explicit constraints); in most cases keeping

Table 4: Results for problems with linear constraints (first three rows have only equality constraints). m_{lin} and m_{nln} are the number of linear and nonlinear constraints; σ^*_{impl} and σ^*_{expl} are threshold penalty parameters when the linear constraints are handled implicitly and explicitly; σ is the penalty parameter. The last two columns give the number of iterations before convergence; the symbol "$*$" indicates that unboundedness was detected, and "-" that 100 iterations were performed without converging. The solver exits when unboundedness is detected or an iterate satisfies (33) with $\epsilon = 10^{-8}$.

Problem	n	m_{lin}	m_{nln}	σ^*_{impl}	σ^*_{expl}	σ	Impl.	Expl.
`Chain400`	802	402	1	0.0012	0	10^{-3} 0.002	$*$ 7	10 10
`Channel400`	1600	800	800	0	0	10^{-3} 1	– –	5 5
`hs113`	18	3	5	6.61	3.39	6 7	$*$ 28	42 17
`prodpl0`	69	25	4	211.9	13.7	40 300	– –	43 30
`prodpl1`	69	25	4	60.8	3.56	10 70	– 89	22 41
`synthes3`	38	23	19	6.00	0.66	2 7	– 35	12 18

the constraints requires fewer iterations, except for `Chain400`. Although positive semidefiniteness of $\nabla^2\phi_\sigma(x^*)$ is guaranteed in the relevant critical cone when $\sigma > \sigma^*$ (in either the implicit or explicit case), a larger value of σ may sometimes be required because the curvature of ϕ_σ away from the solution may be larger or ill-behaved.

For the `Channel` problems, the threshold parameter is zero in both cases. However, KNITRO converges quickly when the linear constraints are kept explicit, but otherwise fails to converge in a reasonable number of iterations. This phenomenon for the `Channel` problems appears to be independent of σ (more values were investigated than are reported here). Even if the penalty parameter does not decrease, it appears beneficial to maintain some of the constraints explicitly.

9 Discussion and concluding remarks

We derived a smooth extension of the penalty function by Fletcher (1970) as an extension to the implementation of Estrin et al. (2019) for the case of inequality constraints. Our implementation is particularly promising for problems where augmented linear systems (28) can be solved efficiently. We further demonstrated the merits of the approach on several PDE-constrained optimization problems.

Some limitations that are shared with the equality-constrained case are avenues for future work. These include dealing with the highly nonlinear nature of the penalty function, developing robust penalty parameter updates and linear solve tolerance rules (for inexact optimization solvers), preconditioning the trust-region subproblems, and using cheaper second-derivative approximations (e.g., quasi-Newton updates) in conjunction with Hessian approximations (26a)–(26b). Possible approaches for dealing with these issues are discussed by Estrin et al. (2019, §10).

Bound constraints provide additional challenges for future work on top of the equality-constrained case. For example, we would like to extend the theory to problems with weaker constraint qualifications than (A2)–(A3). A regularization approach as in (Estrin et al., 2019, §6) can be employed when bound constraints are present, but it may need to be refined to obtain similar convergence guarantees when (A2) applies only at KKT points.

Another challenge is the possible numerical instability when iterates are close to the bounds, if the quantity $A(x)^TQ(x)A(x)$ becomes ill-conditioned. It would help to develop a specialized bound-constrained interior-point Newton-CG trust-region solver for (PP) that carefully controls the distance to the bounds and attempts to minimize the number of approximate penalty Hessian products (as Hessian products are the most computationally intensive operation requiring two linear solves). We can also investigate other functions $Q(x)$ to approximate the complementarity conditions for KKT points, as different forms may have different advantages and limitations; for example, (6) may cause premature termination if x^* is far from its bounds.

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.

A Maintaining explicit constraints

We discuss technical details about the penalty function when some of the constraints are linear and maintained explicitly as in (29). We define $W_\sigma(x) = \nabla w_\sigma(x) \in \mathbb{R}^{n\times m_2}$, and $C(x) = \begin{bmatrix} A(x) & B \end{bmatrix}$ as the Jacobian of all constraints. The operators $g_\sigma(x)$, $H_\sigma(x)$, $S(x,v)$ and $T(x,w)$ are still defined over all constraints (e.g., $g_\sigma(x) := g(x) - A(x)y_\sigma(x) - Bw_\sigma(x)$), not just the nonlinear ones, and so they act on $C(x)$ and not just $A(x)$. Define

$$g^y_\sigma(x) = g(x) - A(x)y_\sigma(x) \tag{38}$$

as the gradient of the partial Lagrangian with respect to the nonlinear constraints $c(x)$ only (note that the linear constraints do not affect H_σ). The gradient and Hessian of the penalty function become

$$\nabla\phi_\sigma(x) = g^y_\sigma(x) - Y_\sigma(x)c(x), \tag{39a}$$

$$\nabla^2\phi_\sigma(x) = H_\sigma(x) - A(x)Y_\sigma(x)^T - Y_\sigma(x)A(x)^T - \nabla_x\left[Y_\sigma(x)c\right]. \tag{39b}$$

We restate the optimality conditions for (NP-EXP) in terms of the penalty function. To do so, define the critical cones for (NP-EXP) and (29), respectively, as

$$\bar{\mathcal{C}}_\phi(x^*,z^*) = \mathcal{C}_\phi(x^*,z^*) \cap \{p \mid B^Tp = 0\}, \qquad \bar{\mathcal{C}}(x^*,z^*) = \mathcal{C}(x^*,z^*) \cap \{p \mid B^Tp = 0\}.$$

Definition 3 (First-order KKT point) *A point (x^*,z^*) is a first-order KKT point of (NP-EXP) if for any $\sigma \geq 0$ the following hold:*

$$\ell \leq x^* \leq u, \tag{40a}$$

$$c(x^*) = 0, \tag{40b}$$

$$B^Tx^* = d, \tag{40c}$$

$$\nabla\phi_\sigma(x^*) = Bw^* + z^*, \tag{40d}$$

$$z^*_j = 0 \qquad \text{if } j \notin \mathcal{A}(x^*), \tag{40e}$$

$$z^*_j \geq 0 \qquad \text{if } x^*_j = \ell_j, \tag{40f}$$

$$z^*_j \leq 0 \qquad \text{if } x^*_j = u_j. \tag{40g}$$

Then $y^* := y_\sigma(x^*)$ and $w^* := w_\sigma(x^*)$ comprise the Lagrange multipliers of (NP-EXP) associated with x^*. Note that by (A3), inequalities (40f) and (40g) are strict.

Definition 4 (Second-order KKT point) *The first-order KKT point (x^*,z^*) satisfies the second-order necessary KKT condition for (NP-EXP) if for any $\sigma \geq 0$,*

$$p^T\nabla^2\phi_\sigma(x^*)p \geq 0 \quad \text{for all } p \in \bar{\mathcal{C}}(x^*,z^*). \tag{41}$$

The condition is sufficient if the inequality is strict.

Remark 3 *As before, if (40b) is omitted, Definition 3 defines first-order KKT points of (29). Similarly, replacing $\bar{\mathcal{C}}(x^*, z^*)$ by $\bar{\mathcal{C}}_\phi(x^*, z^*)$ in Definition 4 defines second-order KKT points of (29).*

A.1 Proof of Theorem 2

Observe that the multiplier estimates $y_\sigma(x)$ and $w_\sigma(x)$ satisfy

$$C(x)^T Q(x) C(x) \begin{bmatrix} y_\sigma(x) \\ w_\sigma(x) \end{bmatrix} = C(x)^T Q(x) g(x) - \sigma \begin{bmatrix} c(x) \\ B^T x - d \end{bmatrix}. \tag{42}$$

Proof of (32a): We drop the argument x from operators and assume that all are evaluated at $\bar{x}$. Because $\bar{x}$ is a first-order KKT point for (29), we need only show that $c(\bar{x}) = 0$. Further, $Q(\nabla \phi_\sigma - Bw^*) = 0$ at $\bar{x}$, or equivalently,

$$QBw^* = Q\left(g - Ay_\sigma - Y_\sigma c\right).$$

Multiplying both sides by C^T and using (42) we have

$$\begin{bmatrix} A^TQBw^* \\ B^TQBw^* \end{bmatrix} = \sigma \begin{bmatrix} c \\ 0 \end{bmatrix} + \begin{bmatrix} A^TQBw_\sigma \\ B^TQBw_\sigma \end{bmatrix} - \begin{bmatrix} A^TQY_\sigma c \\ B^TQY_\sigma c \end{bmatrix},$$

so that $w_\sigma = w^* + (B^TQB)^{-1}B^TQY_\sigma c$. Substituting $w_\sigma(\bar{x})$ into the first block of equations and rearranging gives

$$AQ^{1/2}\bar{P}_{Q^{1/2}B}Q^{1/2}Y_\sigma c = \sigma c.$$

The triangle inequality gives $\sigma\|c\| \leq \|A^TQ^{1/2}\bar{P}_{Q^{1/2}B}Q^{1/2}Y_\sigma\|\|c\|$, implying $c = 0$. Then $w_\sigma = w^*$ and $\bar{x}$ is a first-order KKT point for (NP-EXP).

Proof of (32b): As in the proof of (18b), we differentiate (42) to obtain

$$\begin{aligned} &C(x)^TQ(x)C(x)\begin{bmatrix} Y_\sigma(x)^T \\ W_\sigma(x)^T \end{bmatrix} \\ &\quad = C(x)^T[Q(x)H_\sigma(x) - \sigma I + R(x, g_\sigma(x))] + S(x, Q(x)g_\sigma(x)). \end{aligned} \tag{43}$$

For the remainder of the proof, we assume all operators are evaluated at x^*. Because x^* satisfies first-order conditions (40), $Qg_\sigma = 0$ independently of σ, so $S(Qg_\sigma) = 0$. Let $P_{Q^{1/2}C} := P_{Q^{1/2}C(x^*)}(x^*)$, so that from (43) we have

$$Q^{1/2}\left(AY_\sigma^T + BW_\sigma^T\right)Q^{1/2} = P_{Q^{1/2}C}\left[Q^{1/2}H_\sigma Q^{1/2} - \sigma I + R(g_\sigma)Q^{1/2}\right]. \tag{44}$$

Observe that if $p \in \bar{\mathcal{C}}_\phi(x^*, z^*)$, then $p = Q^{1/2}\bar{p}$ for some $\bar{p} \in \bar{\mathcal{C}}_\phi(x^*, z^*)$. Because $Q^{1/2}g_\sigma = 0$, we have $R(g_\sigma)p = 0$.

Substituting (44) into (39b), and $P_{Q^{1/2}C} + \bar{P}_{Q^{1/2}C} = I$ gives

$$\begin{aligned} &p^T\nabla^2\phi_\sigma(x^*)p \geq 0 \\ \iff &\bar{p}^TQ^{1/2}\left(H_\sigma - AY_\sigma^T - Y_\sigma A^T\right)Q^{1/2}\bar{p} \geq 0 \\ \iff &\bar{p}^T\Big(\bar{P}_{Q^{1/2}C}Q^{1/2}H_\sigma Q^{1/2}\bar{P}_{Q^{1/2}C} - P_{Q^{1/2}C}Q^{1/2}H_\sigma Q^{1/2}P_{Q^{1/2}C} + 2\sigma P_{Q^{1/2}C}\Big)\bar{p} \\ &\qquad - p^T\left(BW_\sigma^T + W_\sigma B^T\right)p \geq 0. \end{aligned}$$

Because $H_\sigma(x^*) = H_{\scriptscriptstyle L}(x^*, y^*)$, $0 = B^Tp = B^TQ^{1/2}\bar{p}$, we can write $\bar{p} = \bar{P}_{\bar{B}}q$ with $\bar{B} = Q^{1/2}B$ and hence

$$\begin{aligned} 0 \leq p^T\nabla^2\phi_\sigma(x_2^*)p \iff 0 \preceq &\bar{P}_{\bar{B}}\bar{P}_{Q^{1/2}C}H_{\scriptscriptstyle L}(x^*, y^*)\bar{P}_{Q^{1/2}C}\bar{P}_{\bar{B}} \\ &- \bar{P}_{\bar{B}}P_{Q^{1/2}C}H_{\scriptscriptstyle L}(x^*, y^*)P_{Q^{1/2}C}\bar{P}_{\bar{B}} + 2\sigma\bar{P}_{\bar{B}}P_{Q^{1/2}C}\bar{P}_{\bar{B}}. \end{aligned}$$

As before, the first term is positive semi-definite, so we only need that

$$-\bar{P}_{\bar{B}}P_{Q^{1/2}C}H_{\scriptscriptstyle L}(x^*, y^*)P_{Q^{1/2}C}\bar{P}_{\bar{B}} + 2\sigma\bar{P}_{\bar{B}}P_{Q^{1/2}C}\bar{P}_{\bar{B}} \succeq 0,$$

which is equivalent to $\sigma \geq \sigma^*$. □

A.1.1 Evaluating the penalty function and derivatives

We again drop the arguments on functions and assume they are evaluated at a point x for some σ:

$$y = y_\sigma(x), \quad A = A(x), \quad Y_\sigma = Y_\sigma(x), \quad H_\sigma = H_\sigma(x), \quad S_\sigma = S_\sigma(x, g_\sigma(x)), \quad \text{etc.}$$

We focus on the nonsymmetric linear systems; the corresponding symmetric linear systems can be derived similarly to Section 5.

The multipliers for evaluating the penalty function are obtained by solving

$$\begin{bmatrix} I & A & B \\ A^TQ & & \\ B^TQ & & \end{bmatrix} \begin{bmatrix} g_\sigma \\ y_\sigma \\ w_\sigma \end{bmatrix} = \begin{bmatrix} g \\ \sigma c \\ \sigma(Bx - d) \end{bmatrix}. \tag{45}$$

To compute the gradient and Hessian products, we use the identity

$$C^TQC \begin{bmatrix} Y_\sigma^T \\ W_\sigma^T \end{bmatrix} = C^T[QH_\sigma - \sigma I + R_\sigma] + S_\sigma \tag{46}$$

to obtain the necessary products with Y_σ and Y_σ^T. Observe that

$$Y_\sigma u = \begin{bmatrix} Y_\sigma & W_\sigma \end{bmatrix} \begin{bmatrix} u \\ 0 \end{bmatrix}, \qquad Y_\sigma^T v = \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} Y_\sigma^T \\ W_\sigma^T \end{bmatrix} v,$$

so that Algorithm 1 and Algorithm 2 can be applied.

Note that to compute the gradient in (39a), g_σ^y is not available directly from the solution to (45) and must be computed explicitly using (38).

Approximate products with $\nabla^2\phi_\sigma$ can be computed via

$$\begin{aligned} \nabla^2\phi_\sigma \approx B_1 &:= H_\sigma - AY_\sigma^T - Y_\sigma A^T \\ &= H_\sigma - \begin{bmatrix} A & 0 \end{bmatrix} (C^TQC)^{-1} C^T(QH_\sigma - \sigma I + R_\sigma) - \begin{bmatrix} A & 0 \end{bmatrix} (C^TQC)^{-1} S_\sigma \\ &\quad - (H_\sigma Q - \sigma I + R_\sigma) C (C^TQC)^{-1} \begin{bmatrix} A^T \\ 0 \end{bmatrix} - S_\sigma (C^TQC)^{-1} \begin{bmatrix} A^T \\ 0 \end{bmatrix} \\ \approx B_2 &:= H_\sigma - \begin{bmatrix} A & 0 \end{bmatrix} (C^TQC)^{-1} C^T(QH_\sigma - \sigma I + R_\sigma) \\ &\quad - (H_\sigma Q - \sigma I + R_\sigma) C (C^TQC)^{-1} \begin{bmatrix} A^T \\ 0 \end{bmatrix}. \end{aligned}$$

For products with the weighted-pseudoinverse and its transpose, we can compute

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = (C^TQC)^{-1} C^T v, \qquad v = C(C^TQC)^{-1} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

by solving the respective block systems

$$\begin{bmatrix} I & QA & QB \\ A^T & & \\ B^T & & \end{bmatrix} \begin{bmatrix} t \\ u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} v \\ 0 \\ 0 \end{bmatrix}, \qquad \begin{bmatrix} I & A & B \\ A^TQ & & \\ B^TQ & & \end{bmatrix} \begin{bmatrix} v \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} 0 \\ -u_1 \\ -u_2 \end{bmatrix}. \tag{47}$$

Thus we can obtain the same types of Hessian approximations as (26), again with two augmented system solves per product.

References

M. Anitescu. On solving mathematical programs with complementarity constraints as nonlinear programs. Preprint ANL/MCS–P864–1200, Argonne National Laboratory, 2000.

M. Arioli. Generalized Golub-Kahan bidiagonalization and stopping criteria. SIAM J. Matrix Anal. Appl., 34 (2):571–592, 2013. DOI: 10.1137/120866543.

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

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

A. Björck and C. C. Paige. Solution of augmented linear systems using orthogonal factorizations. BIT, 34(1): 1–24, 1994. DOI: 10.1007/BF01935013.

P. T. Boggs, J. W. Tolle, and A. J. Kearsley. A merit function for inequality constrained nonlinear programming problems. Internal Report NISTIR 4702, Applied and Computational Mathematics Division, National Institute of Standards and Technology, Gaithersburg, MD, USA, 1992.

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

X. Chen. Smoothing methods for complementarity problems and their applications: a survey. J. Oper. Res. Soc. Japan, 43(1):32–47, 2000. ISSN 0453-4514. DOI: 10.1016/S0453-4514(00)88750-5. New trends in mathematical programming (Kyoto, 1998).

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Trust-Region Methods. MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2000.

J. E. Craig. The N-step iteration procedures. Journal of Mathematics and Physics, 34(1):64–73, 1955.

G. Di Pillo and L. Grippo. A class of continuously differentiable exact penalty function algorithms for nonlinear programming problems. In E. P. Toft-Christensen, editor, System Modelling and Optimization, page 246–256. Springer-Verlag, Berlin, 1984.

G. Di Pillo and L. Grippo. A continuously differentiable exact penalty function for nonlinear programming problems with inequality constraints. SIAM J. Control Optim., 23(1):72–84, 1985. ISSN 0363-0129. DOI: 10.1137/0323007.

R. Estrin and C. Greif. SPMR: a family of saddle-point minimum residual solvers. SIAM J. Sci. Comput., 40(3):A1884–A1914, 2018. ISSN 1064–8275.

R. Estrin, D. Orban, and M. A. Saunders. LNLQ: An iterative method for least-norm problems with an error minimization property. Cahier du GERAD G–2018–40, GERAD, Montréal (Québec), Canada, 2018.

R. Estrin, M. P. Friedlander, D. Orban, and M. A. Saunders. Implementing a smooth exact penalty function for equality-constrained nonlinear optimization. Cahier du GERAD G–2019–04, GERAD, Montréal (Québec), Canada, 2019.

R. Fletcher. A class of methods for nonlinear programming with termination and convergence properties. In J. Abadie, editor, Integer and Nonlinear Programming, pages 157–175. North-Holland, Amsterdam, 1970.

R. Fletcher. A class of methods for nonlinear programming: III. Rates of convergence. In F. A. Lootsma, editor, Numerical Methods for Nonlinear Optimization. Academic Press, New York, 1973a.

R. Fletcher. An exact penalty function for nonlinear programming with inequalities. Math. Programming, 5: 129–150, 1973b. ISSN 0025–5610. DOI: 10.1007/BF01580117.

R. W. Freund and N. M. Nachtigal. QMR: a quasi-minimal residual method for non-Hermitian linear systems. Numer. Math., 60(3):315–339, 1991. ISSN 0029-599X. DOI: 10.1007/BF01385726. URL https://doi.org/10.1007/BF01385726.

A. Gersborg-Hansen, M. P. Bendsøe, and O. Sigmund. Topology optimization of heat conduction problems using the finite volume method. Struct. Multidiscip. Optim., 31(4):251–259, 2006. ISSN 1615-147X. DOI: 10.1007/s00158-005-0584-3.

N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEr and SifDec: A constrained and unconstrained testing environment, revisited. ACM Trans. Math. Softw., 29(4):373–394, Dec. 2003.

M. Heinkenschloss and D. Ridzal. A matrix-free trust-region SQP method for equality constrained optimization. SIAM J. Optim., 24(3):1507–1541, 2014. DOI: 10.1137/130921738.

S. Leyffer. Complementarity constraints as nonlinear equations: theory and numerical experience. In Optimization with Multivalued Mappings, volume 2 of Springer Optim. Appl., pages 169–208. Springer, New York, 2006. DOI: 10.1007/0-387-34221-4˙9.

N. Maratos. Exact Penalty Function Algorithms for Finite Dimensional and Optimization Problems. PhD thesis, Imperial College of Science and Technology, London, UK, 1978.

J. Nocedal and S. J. Wright. Numerical Optimization. Springer, New York, second edition, 2006.

Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Statist. Comput., 7(3):856–869, 1986. DOI: 10.1137/0907058.

T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. SIAM J. Numer. Anal., 20(3):626–637, 1983. DOI: 10.1137/0720042.

V. M. Zavala and M. Anitescu. Scalable nonlinear programming via exact differentiable penalty functions and trust-region Newton methods. SIAM J. Optim., 24(1):528–558, 2014.