SIAM J. OPTIM.
Vol. 20, No. 4, pp. 2023–2048

A SECOND DERIVATIVE SQP METHOD: GLOBAL CONVERGENCE*

NICHOLAS I. M. GOULD† AND DANIEL P. ROBINSON‡

Abstract. Sequential quadratic programming (SQP) methods form a class of highly efficient algorithms for solving nonlinearly constrained optimization problems. Although second derivative information may often be calculated, there is little practical theory that justifies exact-Hessian SQP methods. In particular, the resulting quadratic programming (QP) subproblems are often nonconvex, and thus finding their global solutions may be computationally nonviable. This paper presents a second derivative SQP method based on quadratic subproblems that are either convex, and thus may be solved efficiently, or need not be solved globally. Additionally, an explicit descent-constraint is imposed on certain QP subproblems, which "guides" the iterates through areas in which nonconvexity is a concern. Global convergence of the resulting algorithm is established.

Key words. nonlinear programming, nonlinear inequality constraints, sequential quadratic programming, ℓ_1-penalty function, nonsmooth optimization

AMS subject classifications. 49J52, 49M37, 65F22, 65K05, 90C26, 90C30, 90C55

DOI. 10.1137/080744542

1. Introduction. In this paper we present a sequential quadratic programming (SQP) method for solving the problem

$$(\ell_1\text{-}\sigma) \qquad \underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; \phi(x) = f(x) + \sigma\|[c(x)]^-\|_1,$$

where the constraint vector $c(x) : \mathbb{R}^n \to \mathbb{R}^m$ and the objective function $f(x) : \mathbb{R}^n \to \mathbb{R}$ are assumed to be twice continuously differentiable, σ is a positive scalar known as the penalty parameter, and we have used the notation $[v]^- = \min(0, v)$ for a generic vector v (the minimum is understood to be componentwise). Our motivation for solving this problem is that solutions of problem (ℓ_1-σ) correspond (under certain assumptions) to solutions of the problem

$$(\text{NP}) \qquad \underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; f(x) \quad \text{subject to} \quad c(x) \ge 0.$$

See [10, 23] for more details on exactly how problems (ℓ_1-σ) and (NP) are related.

The precise set of properties that characterize an SQP method is often author dependent. In fact, as the immense volume of literature on SQP methods continues to increase, the properties that define these methods become increasingly blurred. One may argue, however, that the backbone of every SQP method consists of "step generation" and "step acceptance/rejection." We describe these concepts in turn.

All SQP methods generate a sequence of trial steps, which are computed as solutions of cleverly chosen quadratic or quadratic-related subproblems. Typically, the quadratic programming (QP) subproblems are closely related to the optimality conditions of the underlying problem and thus give the potential for fast Newton-like

*Received by the editors December 22, 2008; accepted for publication (in revised form) December 30, 2009; published electronically March 12, 2010. Research supported by the EPSRC grants EP/E053351/1 and EP/F005369/1.
http://www.siam.org/journals/siopt/20-4/74454.html
†Rutherford Appleton Laboratory, Numerical Analysis Group, Chilton, Didcot, Oxfordshire, OX11 0QX, UK (nick.gould@stfc.ac.uk).
‡University of Oxford, Mathematical Institute, 24–29 St Giles', Oxford, OX1 3LB, UK (robinson@maths.ox.ac.uk).

convergence. More precisely, the trial steps "approximately" minimize (locally) a quadratic approximation to a Lagrangian function subject to a linearization of all or a subset of the constraint functions. Two major concerns associated with this QP subproblem are incompatible linearized constraints and unbounded solutions. There are essentially two approaches that have been used for handling unbounded solutions. The first approach is to use a positive definite approximation to the Hessian in the quadratic subproblem. The resultant *strictly convex* QP is bounded with a unique minimizer. The second approach allows for a *nonconvex* QP by explicitly bounding the solution via a trust-region constraint. Both techniques have been effective in practice. The issue of incompatible subproblems is more delicate. We first note that the QP subproblem may be "naturally" incompatible—i.e., the set of feasible points is empty. However, even if the linearized constraints are compatible, the feasible region may still be empty if a trust-region constraint is imposed; the trust-region may "cut-off" all solutions to the linear system. Different techniques, such as constraint shifting [27], a special "elastic" mode [17], and a "feasibility restoration" phase [14], have been used to deal with incompatible subproblems.

Strategies for accepting or rejecting trial steps are sometimes referred to as "globalization techniques" since they are the instrument for guaranteeing global convergence. The earliest methods used so-called merit functions to measure the quality of a trial step. A merit function is a single function that carefully balances the (usually) conflicting aims of reducing the objective function and satisfying the constraints. The basic idea is that a step is accepted if it gives sufficient decrease in the merit function; otherwise, the step is rejected, parameters are updated, and a new trial step is computed. More recently, filter methods [14] have become an attractive alternative to a merit function. Filter methods view problem (NP) as a bi-objective optimization problem—minimizing the objective function $f(x)$ and minimizing the constraint violation $\|[c(x)]^-\|$. These methods use the idea of a "filter," which is essentially a collection of pairs $(\|[c(x)]^-\|, f(x))$ such that no pair dominates another—we say that a pair $(\|[c(x_1)]^-\|, f(x_1))$ dominates a pair $(\|[c(x_2)]^-\|, f(x_2))$ if $f(x_1) < f(x_2)$ *and* $\|[c(x_1)]^-\| < \|[c(x_2)]^-\|$. Although the use of a merit function and a filter are conceptually quite different, Curtis and Nocedal [11] have shown that a "flexible" penalty approach partially bridges this gap. Their method may be viewed as a continuum of methods with traditional merit function and filter methods as the extrema.

The previous two paragraphs described two properties of all SQP methods—step computation and step acceptance or rejection—and these properties alone may differentiate one SQP method from another. In the context of problem (NP), a further fundamental distinction between SQP methods can be found in how the inequality constraints are used in the QP subproblems. This distinction has spawned a rivalry between essentially two classes of methods, which are commonly known as sequential equality-constrained quadratic programming (SEQP) and sequential inequality-constrained quadratic programming (SIQP) methods.

SEQP methods solve problem (NP) by solving an *equality* constrained QP during each iterate. The linearized equality constraints that are included may be interpreted as an approximation to the optimal active constraint set. Determining which constraints to include in each subproblem is a delicate task. The approach used by Coleman and Conn [8] includes those constraints that are nearly active at the current point. Then they solve an equality constrained QP in which a second-order approximation to the locally differentiable part of an exact penalty function is minimized subject to keeping the "nearly" active constraints fixed. An alternative approach is to use the solution of a "simpler" auxiliary subproblem as a prediction of the optimal active constraints. Often, the simpler subproblem only uses first-order information and results in a linear program. Merit function-based variants of this type have been studied by Fletcher and Sainz de la Maza [15], and Byrd et al. [4, 5], while filter-based variants have been studied by Chin and Fletcher [7].

SIQP methods solve problem (NP) by solving a sequence of inequality constrained quadratic subproblems. Unlike the SEQP philosophy, SIQP methods utilize *every* constraint in each subproblem and, therefore, avoid the precarious task of choosing which constraints to include. These methods also have the potential for fast convergence; under standard assumptions, methods of this type correctly identify the optimal active set in a finite number of iterations, and thereafter rapid convergence is guaranteed by the famous result due to Robinson [24, Theorem 3.1]. Probably the greatest disadvantage of SIQP methods is their potential cost; to solve the inequality constrained QP subproblem, both active set and interior-point algorithms may require the solution of many equality constrained quadratic programs. However, in the case of moderate-sized problems, there is much empirical evidence that indicates that the additional cost per iteration is often off set by substantially fewer function evaluations (similar evidence has yet to surface for large-sized problems). SIQP methods that utilize exact second derivatives must also deal with nonconvexity. To our knowledge, all previous second-order SIQP methods assume that global minimizers of nonconvex subproblems are computed, which is not a realistic assumption in most cases. For these methods, the computation of a local minimizer is unsatisfactory because it may yield an ascent direction. Line-search, trust-region, and filter variants of SIQP methods have been proposed. The line-search method by Gill, Murray, and Saunders [17] avoids unbounded and nonunique QP solutions by maintaining a positive definite quasi-Newton (sometimes limited-memory quasi-Newton) approximation to the Hessian of the Lagrangian. The SIQP approaches by Boggs, Kearsley, and Tolle [1, 2] modify the exact second derivatives to ensure that the reduced Hessian is sufficiently positive definite. Finally, the filter SIQP approach by Fletcher and Leyffer [14] deals with infeasibility by entering a special restoration phase to recover from bad steps.

The algorithm we propose in this paper may be considered an SIQP/SEQP hybrid that is most similar to the Sℓ_1QP method proposed by Fletcher [13], which is a second-order method designed for finding first-order critical points of problem (ℓ_1-σ). The QP subproblem studied by Fletcher is to minimize a second-order approximation to the ℓ_1-penalty function subject to a trust-region constraint. More precisely, the QP subproblem is obtained by approximating $f(x)$ and $c(x)$ in the ℓ_1-penalty function by a second- and first-order Taylor approximation, respectively. Unfortunately, the theoretical results of Fletcher's method requires the computation of the *global* minimizer of this (generally) nonconvex subproblem, which is known to be an NP-hard problem. The method we propose is also a second derivative method that is globalized via the ℓ_1-merit function, but we do not require the global minimizer of any nonconvex QP. To achieve this goal, our procedure for computing a trial step is necessarily more complicated than that used by Fletcher. Given an estimate x_k of a solution to problem (NP), a search direction is generated from a combination of three steps, all of which are tractable for large problems: a *predictor* step s_k^{P} is defined as a solution to a *strictly convex* QP subproblem; a *Cauchy* step s_k^{CP} drives convergence of the algorithm and is computed from a special univariate *global* minimization problem; and an (optional) *accelerator* step s_k^{A} is computed from a *local* solution of a special nonconvex QP subproblem. Since there is considerable flexibility in how we compute the accelerator step, we may adaptively choose the accelerator subproblem to reflect the likely success of an SIQP or SEQP approach. This observation justifies our claim that our method is an SIQP/SEQP hybrid.

The paper is organized as follows. In section 2 we provide a complete description of how the predictor, Cauchy, and accelerator steps are computed; we then show how to combine these steps to obtain a trial step. Once the trial step is defined, we then apply traditional trust-region updating strategies. The statement and description of the resulting algorithm is given in section 3. Finally, in section 4 we prove global convergence of our algorithm and draw general conclusions in section 5.

General notation and consequences. We let e denote the vector of all ones whose dimension is determined by the context. A local solution of (ℓ_1-σ) is denoted by x^*; $g(x)$ is the gradient of $f(x)$, and $\nabla_{xx} f(x)$ is its (symmetric) Hessian; the matrix $\nabla_{xx} c_j(x)$ is the Hessian of $c_j(x)$; $J(x)$ is the $m \times n$ Jacobian matrix of the constraints with ith row $\nabla c_i(x)^T$. For a general vector v, the notation $[v]^- = \min(0, v)$ is used, where the minimum is understood to be componentwise. The Lagrangian function associated with (NP) is $\mathcal{L}(x, y) = f(x) - y^T c(x)$. The Hessian of the Lagrangian with respect to x is $\nabla_{xx}\mathcal{L}(x, y) = \nabla_{xx} f(x) - \sum_{j=1}^m y_j \nabla_{xx} c_j(x)$.

We often consider problem functions evaluated at a specific point x_k. To simplify notation we define the following: $f_k = f(x_k)$, $c_k = c(x_k)$, $g_k = g(x_k)$, and $J_k = J(x_k)$. Given a pair of values (x_k, y_k), we let H_k and B_k denote symmetric approximations to $\nabla_{xx}\mathcal{L}(x_k, y_k)$ in which B_k is required additionally to be positive definite. In practice H_k may be chosen to be $\nabla_{xx}\mathcal{L}(x_k, y_k)$, but this is not necessary in what follows.

To prove global convergence to a first-order point of the merit function ϕ, we require the local linear model

$$M_k^{\text{L}}(s) \stackrel{\text{def}}{=} M_k^{\text{L}}(s\,; x_k) = f_k + g_k^T s + \sigma \|[c_k + J_k s]^-\|_1.$$

For a given step s, we then define the *change* in the linear model to be

$$\Delta M_k^{\text{L}}(s) \stackrel{\text{def}}{=} \Delta M_k^{\text{L}}(s\,; x_k) = M_k^{\text{L}}(0\,; x_k) - M_k^{\text{L}}(s\,; x_k).$$

We may now define a criticality measure for minimizing ϕ as

$$\Delta_{\max}^{\text{L}}(x, \Delta) = M_k^{\text{L}}(0\,; x) - \min_{\|s\|_\infty \le \Delta} M_k^{\text{L}}(s\,; x), \tag{1.1}$$

which is the maximum possible decrease of the linear model for a given trust-region radius $\Delta \ge 0$, primal variable x, and penalty parameter σ. Useful properties of $\Delta_{\max}^{\text{L}}$, including the fact that it is a criticality measure, are given in the next lemma. See Borwein and Lewis [3], Rockafellar [25], and Yuan [28] for more details.

LEMMA 1.1. *Consider the definition of* Δ_{max}^{L} *as given by* (1.1). *Then the following properties hold:*

(i) $\Delta_{max}^{L}(x, \Delta) \ge 0$ *for all* x *and all* $\Delta \ge 0$*;*
(ii) $\Delta_{max}^{L}(x, \cdot)$ *is a nondecreasing function;*
(iii) $\Delta_{max}^{L}(x, \cdot)$ *is a concave function;*
(iv) $\Delta_{max}^{L}(\cdot, \Delta)$ *is continuous;*
(v) *For any fixed* $\Delta > 0$, $\Delta_{max}^{L}(x, \Delta) = 0$ *if and only if* x *is a stationary point for problem* (ℓ_1-σ).

Properties (ii) and (iii) of Lemma 1.1 allow us to relate the maximum decrease in the linear model for an *arbitrary* radius to the maximum decrease in the linear model for a constant radius. For convenience, we have chosen that constant to be one. The following corollary makes this precise.

COROLLARY 1.2. *Let x be fixed. Then for all $\Delta \ge 0$*

$$\Delta^{L}_{max}(x,\Delta) \ge \min(\Delta,1)\Delta^{L}_{max}(x,1). \tag{1.2}$$

Proof. See Lemma 3.1 in [5]. □

2. Step computation. During each iterate of our proposed method, we compute a trial step s_k that is calculated from three steps: a predictor step s_k^{P}, a Cauchy step s_k^{CP}, and an accelerator step s_k^{A}. The predictor step is defined as the solution of a *strictly convex* model for which the global minimum is unique and computable in polynomial time. The Cauchy step is then computed as the *global* minimizer of a specialized one-dimensional optimization problem involving a "faithful" model of ϕ and is also computable in polynomial time. It will be shown that the Cauchy step alone is enough for proving convergence, but we allow the option for computing an additional accelerator step. The accelerator step is the solution of a special QP subproblem that utilizes a less restrictive approximation H_k to $\nabla_{xx}\mathcal{L}(x_k,y_k)$ and is intended to improve the efficiency of the method by encouraging fast convergence. Once the trial step s_k has been computed, standard trust-region strategies are used to promote convergence. We begin by discussing the predictor step.

2.1. The predictor step s_k^{P}. In our algorithm, the predictor step s_k^{P} plays a role analogous to the role played by the direction of steepest descent in unconstrained trust-region methods. During each iterate of a traditional unconstrained trust-region method, a quadratic model of the objective function is minimized in the direction of steepest descent. The resulting step, known as the Cauchy step, gives a decrease in the quadratic model that is sufficient for proving convergence (see Conn, Gould, and Toint [9]). In our setting, a vector that is directly analogous to the direction of steepest descent is the vector that minimizes the linearization of the ℓ_1-merit function within a trust-region constraint. However, since we want to incorporate second-order information, we define the predictor step to be the solution to

$$\underset{s\in\mathbb{R}^n}{\text{minimize}}\ \ M_k^{\text{B}}(s) \quad \text{subject to} \quad \|s\|_\infty \le \Delta_k^{\text{P}}, \tag{2.1}$$

where the *convex* model M_k^{B} is defined by

$$M_k^{\text{B}}(s) \overset{\text{def}}{=} M_k^{\text{B}}(s\,;x_k) = f_k + g_k^T s + \tfrac{1}{2}s^T B_k s + \sigma\|[c_k + J_k s]^-\|_1$$

for any symmetric positive definite matrix approximation B_k to the Hessian matrix $\nabla_{xx}\mathcal{L}(x_k,y_k)$ and $\Delta_k^{\text{P}} > 0$ is the predictor trust-region radius; since B_k is positive definite, problem (2.1) is *strictly* convex, and the minimizer is unique. Given a step s, we define the *change* in the convex model to be

$$\Delta M_k^{\text{B}}(s) \overset{\text{def}}{=} \Delta M_k^{\text{B}}(s\,;x_k) = M_k^{\text{B}}(0\,;x_k) - M_k^{\text{B}}(s\,;x_k).$$

Note that

$$\Delta M_k^{\text{B}}(s_k^{\text{P}}) \ge 0, \tag{2.2}$$

since $M_k^{\text{B}}(s_k^{\text{P}}) \le M_k^{\text{B}}(0)$ and that problem (2.1) is a nonsmooth minimization problem. In fact, it is not differentiable at any point for which the constraint linearization is zero. In practice, we solve the equivalent "elastic" problem [17] defined as

$$\begin{aligned} &\underset{s\in\mathbb{R}^n, v\in\mathbb{R}^m}{\text{minimize}} && f_k + g_k^T s + \tfrac{1}{2}s^T B_k s + \sigma_k e^T v \\ &\text{subject to} && c_k + J_k s + v \ge 0, \ \ v \ge 0, \ \ \|s\|_\infty \le \Delta_k^{\text{P}}. \end{aligned} \tag{2.3}$$

Problem (2.3) is a *smooth* linearly constrained convex quadratic program that may be solved using a number of software packages such as LOQO [26] and QPOPT [16], as well as the QP solvers QPA, QPB, and QPC that are part of the GALAHAD [19] library. In addition, if B_k is chosen to be diagonal, then the GALAHAD package LSQP may be used, since problem (2.3) is then a separable convex quadratic program.

The following bound is [28, Lemma 2.2] transcribed into our notation.

LEMMA 2.1. *For a given x_k and σ the following inequality holds:*

$$\Delta M_k^B(s_k^P) \geq \tfrac{1}{2}\Delta^L_{max}(x_k, \Delta_k^P)\min\left(1, \frac{\Delta^L_{max}(x_k, \Delta_k^P)}{\|B_k\|_2(\Delta_k^P)^2}\right). \tag{2.4}$$

We note that the proof by Yuan requires the *global* minimum of the associated QP. For a general QP this requirement is not practical, since finding the global minimum of a nonconvex QP is NP-hard. This is likely the greatest drawback of any previous method that utilized both *exact* second derivatives and the ℓ_1-penalty function. In our situation, however, the matrix B_k is positive definite by construction, and the global minimum may be found efficiently.

We may further bound $\Delta M_k^B(s_k^P)$ by applying Corollary 1.2.

COROLLARY 2.2.

$$\Delta M_k^B(s_k^P) \geq \tfrac{1}{2}\Delta^L_{max}(x_k, 1)\min\left(1, \Delta_k^P, \frac{\Delta^L_{max}(x_k, 1)}{\|B_k\|_2}, \frac{\Delta^L_{max}(x_k, 1)}{\|B_k\|_2(\Delta_k^P)^2}\right). \tag{2.5}$$

Proof. The proof follows directly from Corollary 1.2 and Lemma 2.1. ∎

The previous corollary bounds the change in the convex model at the predictor step in terms of the maximum change in the linear model within a unit trust-region. In the next section we show how to compute a step for which the change in a *faithful* model (see subsection 2.2) of ϕ is bounded below in terms of the maximum change in the linear model within a unit trust-region; this computation is based on the predictor step.

2.2. The Cauchy step s_k^{CP}. In the beginning of section 2 we stated that the Cauchy step induces global convergence of our proposed method. However, it is also true that the predictor step may be used to drive convergence for a slightly different method; this modified algorithm may crudely be described as follows. During the computation of each iterate, the ratio of actual versus predicted decrease in the merit function is computed, where the predicted decrease is given by the change in the convex model $M_k^B(s)$ at s_k^P. Based on this ratio, the trust-region radius and iterate x_k may be updated using standard trust-region techniques. Using this idea and assuming standard conditions on the iterates generated by this procedure, one may prove convergence to a first-order solution of problem (ℓ_1-σ). This may be derived from Fletcher's work [13] by allowing a positive definite approximation to $\nabla_{xx}\mathcal{L}(x_k, y_k)$. However, our intention is to stay as *faithful* to the problem functions as possible. Therefore, in computing the ratio of actual versus predicted decrease in the merit function, we use the decrease in the *faithful* model

$$M_k^H(s) \stackrel{\text{def}}{=} M_k^H(s\,; x_k) = f_k + g_k^T s + \tfrac{1}{2}s^T H_k s + \sigma\|[c_k + J_k s]^-\|_1,$$

instead of the *strictly convex* model $M_k^B(s)$. Unfortunately, since the predictor step is computed using the approximate Hessian B_k, the point s_k^P is not directly appropriate as a means for ensuring convergence. In fact, it is possible that $M_k^H(s_k^P) > M_k^H(0)$ so that the predictor step gives an increase in the faithful model. However, a reasonable point is close-at-hand and is what we call the Cauchy step. The basic idea behind the Cauchy step is to make improvement in the *faithful* model in the direction s_k^{P} by finding the *global* minimizer of $M_k^{\mathrm{H}}(\alpha s_k^{\mathrm{P}})$ for $0 \le \alpha \le 1$. We will see that the Cauchy step allows us to prove convergence by using the *change* in the faithful model, defined to be

$$\Delta M_k^{\mathrm{H}}(s) \stackrel{\text{def}}{=} \Delta M_k^{\mathrm{H}}(s\,; x_k) = M_k^{\mathrm{H}}(0\,; x_k) - M_k^{\mathrm{H}}(s\,; x_k),$$

as a prediction of the decrease in the merit function.

To be more precise, the Cauchy step is defined as $s_k^{\mathrm{CP}} = \alpha_k s_k^{\mathrm{P}}$, where α_k is the solution to

$$\underset{0\le\alpha\le1}{\text{minimize}} \;\; M_k^{\mathrm{H}}(\alpha s_k^{\mathrm{P}}). \tag{2.6}$$

The function $M_k^{\mathrm{H}}(\alpha s_k^{\mathrm{P}})$ is a piecewise-continuous quadratic function of α for which the exact *global* minimizer may be found efficiently. Before discussing the properties of the Cauchy step, we give the following simple lemma.

LEMMA 2.3. *Let $c \in \mathbb{R}^m$, $J \in \mathbb{R}^{m\times n}$, and $s \in \mathbb{R}^n$. Then the following inequality holds for all $0 \le \alpha \le 1$:*

$$\|[c+\alpha Js]^-\|_1 \le \alpha\|[c+Js]^-\|_1 + (1-\alpha)\|[c]^-\|_1. \tag{2.7}$$

Proof. From the convexity of $\|[\cdot]^-\|_1$ it follows that

$$\|[c+\alpha Js]^-\|_1 = \|[\alpha(c+Js)+(1-\alpha)c]^-\|_1 \le \alpha\|[c+Js]^-\|_1 + (1-\alpha)\|[c]^-\|_1,$$

which is (2.7). ∎

We now give a precise lower bound for the change in the faithful model obtained from the Cauchy step.

LEMMA 2.4. *Let s_k^{P} and s_k^{CP} be defined as previously. Then*

$$\Delta M_k^{H}(s_k^{CP}) \ge \tfrac{1}{2}\Delta M_k^{B}(s_k^{P}) \min\left(1, \frac{\Delta M_k^{B}(s_k^{P})}{n\|B_k - H_k\|_2(\Delta_k^{P})^2}\right). \tag{2.8}$$

Proof. For all $0 \le \alpha \le 1$, we have

$$\Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) \ge \Delta M_k^{\mathrm{H}}(\alpha s_k^{\mathrm{P}}) \tag{2.9}$$

$$= \sigma\big(\|[c_k]^-\|_1 - \|[c_k+\alpha J_k s_k^{\mathrm{P}}]^-\|_1\big) - \alpha g_k^T s_k^{\mathrm{P}} - \frac{\alpha^2}{2} s_k^{\mathrm{P}\,T} H_k s_k^{\mathrm{P}} \tag{2.10}$$

$$\begin{aligned} &= \sigma\big(\|[c_k]^-\|_1 - \|[c_k+\alpha J_k s_k^{\mathrm{P}}]^-\|_1\big) \\ &\quad - \alpha g_k^T s_k^{\mathrm{P}} - \frac{\alpha^2}{2} s_k^{\mathrm{P}\,T} B_k s_k^{\mathrm{P}} + \frac{\alpha^2}{2} s_k^{\mathrm{P}\,T}(B_k - H_k) s_k^{\mathrm{P}}. \end{aligned} \tag{2.11}$$

Equation (2.9) follows, since s_k^{CP} minimizes $M_k^{\mathrm{H}}(\alpha s_k^{\mathrm{P}})$ for $0 \le \alpha \le 1$. Equations (2.10) and (2.11) follow from the definition of M_k^{H} and from simple algebra. Continuing to bound the change in the faithful model, we have

$$\begin{aligned}
\Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) &\geq \sigma\big(\|[c_k]^-\|_1 - \alpha\|[c_k + J_k s_k^{\mathrm{P}}]^-\|_1 - (1-\alpha)\|[c_k]^-\|_1\big) \\
&\qquad - \alpha g_k^T s_k^{\mathrm{P}} - \frac{\alpha}{2} s_k^{\mathrm{P}T} B_k s_k^{\mathrm{P}} + \frac{\alpha^2}{2} s_k^{\mathrm{P}T}(B_k - H_k)s_k^{\mathrm{P}} \qquad (2.12) \\
&= \alpha\sigma\big(\|[c_k]^-\|_1 - \|[c_k + J_k s_k^{\mathrm{P}}]^-\|_1\big) \\
&\qquad - \alpha g_k^T s_k^{\mathrm{P}} - \frac{\alpha}{2} s_k^{\mathrm{P}T} B_k s_k^{\mathrm{P}} + \frac{\alpha^2}{2} s_k^{\mathrm{P}T}(B_k - H_k)s_k^{\mathrm{P}} \qquad (2.13) \\
&= \alpha \Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) + \frac{\alpha^2}{2} s_k^{\mathrm{P}T}(B_k - H_k)s_k^{\mathrm{P}}. \qquad (2.14)
\end{aligned}$$

Equation (2.12) follows from (2.11), Lemma 2.3, and the inequality $\alpha^2 \leq \alpha$, which holds since $0 \leq \alpha \leq 1$. Finally, (2.13) and (2.14) follow from simplification of (2.12) and from the definition of $\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}})$.

The previous string of inequalities holds for *all* $0 \leq \alpha \leq 1$, so it must hold for the value of α that maximizes the right-hand side. As a function of α, the right-hand side may be written as $q(\alpha) = a\alpha^2 + b\alpha$, where

$$a = \tfrac{1}{2} s_k^{\mathrm{P}T}(B_k - H_k)s_k^{\mathrm{P}} \quad \text{and} \quad b = \Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) \geq 0.$$

There are three cases to consider.

Case 1. $a \geq 0$. In this case the quadratic function $q(\alpha)$ is convex, and the maximizer on the interval $[0,1]$ must occur at $x = 1$. Thus, the maximum of q on the interval $[0,1]$ is $q(1)$ and may be bounded by

$$q(1) = a + b \geq b \geq \tfrac{1}{2} b = \tfrac{1}{2} \Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}),$$

since $b \geq 0$ and $a \geq 0$.

Case 2. $a < 0$ and $-b/2a \leq 1$. In this case the maximizer on the interval $[0,1]$ must occur at $\alpha = -b/2a$. Therefore, the maximum of q on the interval $[0,1]$ is given by

$$q\left(-\frac{b}{2a}\right) = a\frac{b^2}{4a^2} + b\frac{-b}{2a} = -\frac{b^2}{4a}.$$

Substituting for a and b, using the Cauchy–Schwarz inequality, and applying norm inequalities shows

$$q\left(-\frac{b}{2a}\right) = \frac{\big(\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}})\big)^2}{2|s_k^{\mathrm{P}T}(B_k - H_k)s_k^{\mathrm{P}}|} \geq \frac{\big(\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}})\big)^2}{2\|B_k - H_k\|_2 \|s_k^{\mathrm{P}}\|_2^2} \geq \frac{\big(\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}})\big)^2}{2n\|B_k - H_k\|_2 \|s_k^{\mathrm{P}}\|_\infty^2}.$$

Finally, since $\|s_k^{\mathrm{P}}\|_\infty \leq \Delta_k^{\mathrm{P}}$, we have

$$q\left(-\frac{b}{2a}\right) \geq \frac{\big(\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}})\big)^2}{2n\|B_k - H_k\|_2 (\Delta_k^{\mathrm{P}})^2}.$$

Case 3. $a < 0$ and $-b/2a > 1$. In this case the maximizer of q on the interval $[0,1]$ is given by $\alpha = 1$. Therefore, the maximum of q on the interval $[0,1]$ is given by $q(1)$ and is bounded by

$$q(1) = a + b > -\tfrac{1}{2} b + b = \tfrac{1}{2} b = \tfrac{1}{2} \Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}),$$

since the inequality $-b/2a > 1$ implies $a > -b/2$.

If we denote the maximizer of $q(\alpha)$ on the interval $[0,1]$ by α^*, then consideration of all three cases shows that

$$q(\alpha^*) \geq \frac{1}{2}\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) \min\left(1, \frac{\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}})}{n\|B_k - H_k\|_2(\Delta_k^{\mathrm{P}})^2}\right). \tag{2.15}$$

Returning to (2.14), we have

$$\Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) \geq q(\alpha^*) \geq \tfrac{1}{2}\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) \min\left(1, \frac{\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}})}{n\|B_k - H_k\|_2(\Delta_k^{\mathrm{P}})^2}\right),$$

which completes the proof. ☐

Note that in the special case $B_k = H_k$, the term $\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}})/\big(n\|B_k - H_k\|_2(\Delta_k^{\mathrm{P}})^2\big)$ should be interpreted as infinity, and then Lemma 2.4 reduces to

$$\Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) \geq \tfrac{1}{2}\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}), \tag{2.16}$$

which trivially holds, since $B_k = H_k$ and $s_k^{\mathrm{CP}} = s_k^{\mathrm{P}}$.

We now arrive at the desired result for the Cauchy step; the change in the faithful model obtained by the Cauchy step is bounded below in terms of the criticality measure $\Delta_{\max}^{\mathrm{L}}$.

COROLLARY 2.5. *Let s_k^P and s_k^{CP} be defined as previously. Then*

$$\Delta M_k^{H}(s_k^{CP}) \geq \tfrac{1}{4}\Delta_{max}^{L}(x_k, 1)\min(\mathcal{S}_k),$$

where

$$\mathcal{S}_k = \left\{ 1,\ \Delta_k^{P},\ \frac{\Delta_{max}^{L}(x_k,1)}{\|B_k\|_2},\ \frac{\Delta_{max}^{L}(x_k,1)}{\|B_k\|_2(\Delta_k^{P})^2},\ \frac{\Delta_{max}^{L}(x_k,1)}{2n\|B_k - H_k\|_2},\ \frac{\Delta_{max}^{L}(x_k,1)}{2n\|B_k - H_k\|_2(\Delta_k^{P})^2},\right.$$
$$\left.\frac{\big(\Delta_{max}^{L}(x_k,1)\big)^3}{2n\|B_k - H_k\|_2\|B_k\|_2^2(\Delta_k^{P})^2},\ \frac{\big(\Delta_{max}^{L}(x_k,1)\big)^3}{2n\|B_k - H_k\|_2\|B_k\|_2^2(\Delta_k^{P})^6} \right\}.$$

Proof. The bound follows from Corollary 2.2 and Lemma 2.4. ☐

Corollary 2.5 provides a bound that is sufficient for proving convergence of our proposed algorithm, but we note that an *approximate* Cauchy point s_k^{ACP} satisfying

$$\Delta M_k^{\mathrm{H}}(s_k^{\mathrm{ACP}}) \geq \eta_{\mathrm{ACP}}\Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) \quad \text{and} \quad \|s_k^{\mathrm{ACP}}\|_\infty \leq \|s_k^{\mathrm{P}}\|_\infty \quad \text{for some } 0 < \eta_{\mathrm{ACP}} < 1 \tag{2.17}$$

is also sufficient. An approximate Cauchy point may be obtained, for example, by backtracking from the predictor step s_k^{P} to the Cauchy point s_k^{CP} until (2.17) is satisfied. Since the theory for an *approximate* Cauchy point is identical to the theory for the Cauchy point (modulo a constant factor in the appropriate estimates), we focus primarily on the latter.

The derivation of the bound supplied by Corollary 2.5 relied on minimizing the faithful model along the single direction s_k^{P}. If the predictor step is a bad search direction for the faithful model (most likely because B_k is, in some sense, a poor approximate to H_k), then convergence is likely to be slow. In order to improve efficiency we may need to make "better" use of the faithful model; the accelerator step serves this purpose and more.

2.3. The accelerator step $s_k^{\rm A}$ and the full step s_k. We begin by discussing three primary motivations for an accelerator step $s_k^{\rm A}$; we use the word "an" instead of the word "the," since we propose several reasonable alternatives. The first motivation of the accelerator step is to improve the rate-of-convergence. The predictor step $s_k^{\rm P}$ uses a positive definite approximation B_k to the true Hessian $\nabla_{xx}\mathcal{L}(x_k, y_k)$, while the Cauchy step $s_k^{\rm CP}$ is computed as a minimization problem in the direction $s_k^{\rm P}$. Therefore, the quality of both the predictor step and the Cauchy step is constrained by how well B_k approximates $\nabla_{xx}\mathcal{L}(x_k, y_k)$ (possibly when restricted to the null space of the Jacobian of the active constraints). The simplest and cheapest choice is $B_k = I$, but this would likely result in a linear convergence rate. In general, if B_k is chosen to more closely approximate $\nabla_{xx}\mathcal{L}(x_k, y_k)$, then the predictor step $s_k^{\rm P}$ becomes more costly to compute but would likely lead to faster convergence. Of course, as B_k is required to be positive definite and since $\nabla_{xx}\mathcal{L}(x_k, y_k)$ is usually indefinite, this is typically not even possible. To promote efficiency, therefore, we may compute an accelerator step $s_k^{\rm A}$ from various accelerator subproblems that are formed from either the predictor step or the Cauchy step. Once the "quality" of the step $s_k^{\rm A}$ has been determined, we define the full step accordingly; sections 2.3.1, 2.3.2, and 2.3.3 make this statement precise.

The previous paragraph may do the Cauchy step injustice; not only does the Cauchy step guarantee convergence of the algorithm, but it may happen that the Cauchy step is an excellent direction. In fact, if we are allowed the choice $B_k = \nabla_{xx}\mathcal{L}(x_k, y_k)$ and choose σ sufficiently large, then provided the trust-region radius $\Delta_k^{\rm P}$ is inactive, the resulting Cauchy step $s_k^{\rm CP}(= s_k^{\rm P})$ is the classical SQP step for problem (NP). This means that the Cauchy step may be the "ideal" step. As previously stated, however, the choice $B_k = \nabla_{xx}\mathcal{L}(x_k, y_k)$ will generally not be permissible. We summarize by saying that the quality of the Cauchy step is strongly dependent on how well B_k "approximates" $\nabla_{xx}\mathcal{L}(x_k, y_k)$.

Unfortunately, even if the Cauchy step is an "excellent" direction, it may still suffer from the Maratos effect [9, 22]. The Maratos effect occurs when the linear approximation to the constraint function does not adequately capture the nonlinear behavior of the constraints. As a result, although the unit step may make excellent progress towards finding a solution of problem (NP), it is in fact rejected by the merit function, and subsequently the trust-region radius is reduced; this inhibits the natural convergence of Newton's Method. Avoiding the Maratos effect is the second motivation for the accelerator step.

The third motivation for the accelerator step is to improve the performance of our method; the quadratic model used in computing the accelerator step may use an *indefinite* approximation H_k to the Hessian of the Lagrangian and is, therefore, considered a more faithful model of the merit function.

Since the Cauchy step is good enough to guarantee convergence, we require that the accelerator step and the full step s_k be defined so that

$$\Delta M_k^{\rm H}(s_k) \ge \eta \Delta M_k^{\rm H}(s_k^{\rm CP}) \tag{2.18}$$

for some $0 < \eta \le \eta_{\rm ACP} < 1$. The method we choose for computing the accelerator step is irrelevant from a global convergence perspective, since we may set $s_k = s_k^{\rm CP}$ regardless of the merit of the kth accelerator step (this includes the case when an accelerator step is not computed) and thus satisfy (2.18) in a *trivial way*. However, from a numerical efficiency perspective we have two goals in mind. First, we want to calculate accelerator steps that lead to fast quadratic convergence locally. Second, we do not want to waste the computation of the accelerator step globally by having to resort to satisfying (2.18) in a trivial way; i.e., we want to satisfy (2.18) in a *non-trivial* way. How to obtain both these goals is not obvious, since if we simply replicate traditional SQP subproblems, we quickly find that nonconvexity may cause trial steps to be ascent directions for $M_k^{\mathrm{H}}(s)$ and thus wasteful. In the next three subsections we propose different subproblems for computing an accelerator step, two of which satisfy *all* of our required criteria. The fact that they lead to fast local convergence, however, is the topic of our companion paper [20].

2.3.1. An explicitly inequality-constrained accelerator subproblem. In this section we discuss an accelerator step that is the solution of an *explicitly inequality-constrained* QP subproblem (EIQP). The terminology "explicitly inequality-constrained" is used to emphasize that the subproblem has a subset of the linearized constraints as explicit constraints and that *all* these explicit constraints are imposed as *inequalities*. The subproblem is given by

$$\text{(EIQP)} \qquad \begin{array}{ll} \underset{s\in\mathbb{R}^n}{\text{minimize}} & M_k^{\mathrm{A}}(s) \\ \text{subject to} & [c_k + J_k(s_k^{\mathrm{CP}} + s)]_{\mathcal{S}_k} \geq 0, \\ & (g_k + H_k s_k^{\mathrm{CP}} + \sigma J_k^T w_k)^T s \leq 0, \\ & \|s\|_\infty \leq \Delta_k^{\mathrm{A}}, \end{array}$$

where the accelerator model M_k^{A} is given by

$$M_k^{\mathrm{A}}(s) \stackrel{\text{def}}{=} M_k^{\mathrm{A}}(s\,; x_k, s_k^{\mathrm{CP}}) = \bar{f}_k + (g_k + H_k s_k^{\mathrm{CP}})^T s + \tfrac{1}{2} s^T H_k s + \sigma \|[c_k + J_k(s_k^{\mathrm{CP}} + s)]_{\mathcal{V}_k}^- \|_1,$$

s_k^{CP} is the Cauchy step, $\bar{f}_k = f_k + g_k^T s_k^{\mathrm{CP}} + \frac{1}{2} {s_k^{\mathrm{CP}}}^T H_k s_k^{\mathrm{CP}}$, $w_k \in \mathbb{R}^m$ is defined componentwise as

$$[w_k]_i = \begin{cases} -1 & \text{if } i \in \mathcal{V}_k\ , \\ 0 & \text{if } i \in \mathcal{S}_k\ , \end{cases} \tag{2.19}$$

where $\mathcal{V}_k \stackrel{\text{def}}{=} \mathcal{V}(x_k\,; s_k^{\mathrm{CP}}) = \{i : [c_k + J_k s_k^{\mathrm{CP}}]_i < 0\}$ and $\mathcal{S}_k \stackrel{\text{def}}{=} \mathcal{S}(x_k\,; s_k^{\mathrm{CP}}) = \{i : [c_k + J_k s_k^{\mathrm{CP}}]_i \geq 0\}$, and $\Delta_k^{\mathrm{A}} > 0$ is the accelerator trust-region radius. The sets $\mathcal{V}_k$ and $\mathcal{S}_k$ contain the indices of the linearized constraints that are *violated* and *satisfied* at the Cauchy step, respectively; given a generic vector v or matrix V, the notations $[v]_{\mathcal{V}_k}$ and $[V]_{\mathcal{V}_k}$ will denote the rows of v and V, respectively, that correspond to the indices in $\mathcal{V}_k$. Analogous notation applies to the indexing set $\mathcal{S}_k$. The artificial constraint $(g_k + H_k s_k^{\mathrm{CP}} + \sigma J_k^T w_k)^T s \leq 0$ (henceforth referred to as the "descent-constraint") guarantees that the directional derivative of $M_k^{\mathrm{A}}(s)$ in the direction s_k^{A} is nonpositive. We will soon see that this condition ensures that a useful accelerator-Cauchy step s_k^{CA} may easily be defined from *any* local solution to problem (EIQP); without the descent-constraint, it is clear that a local minimizer may be an ascent direction for which M_k^{H} increases. If problem (EIQP) was smooth, then the descent-constraint would guarantee that any solution would not cause the model $M_k^{\mathrm{A}}(s)$ to increase. However, since (EIQP) is not smooth, it is possible that the model may still increase at a local solution even though it must initially decrease in that direction! Consider problem (EIQP) with data $\sigma = 2$, $s_k^{\mathrm{CP}} = 0$, $\Delta_k^{\mathrm{A}} = 2$,

$$c_k = \begin{pmatrix} -4 \\ -11 \\ -11 \end{pmatrix}, \ J_k = \begin{pmatrix} 16 & 0 \\ 6 & 10 \\ 6 & -10 \end{pmatrix}, \ g_k = \begin{pmatrix} 34 \\ 4 \end{pmatrix}, \quad \text{and} \ H_k = \begin{pmatrix} -13 & -24/11 \\ -24/11 & 1 \end{pmatrix}.$$

It can be verified that $s_k^{\mathrm{A}} = (11/6, 0)$ is a local solution and that $M_k^{\mathrm{A}}(s_k^{\mathrm{A}}) > M_k^{\mathrm{A}}(0)$, and thus the *change* in the accelerator model

$$\Delta M_k^{\mathrm{A}}(s) \stackrel{\mathrm{def}}{=} \Delta M_k^{\mathrm{A}}(s\,; x_k, s_k^{\mathrm{CP}}) = M_k^{\mathrm{A}}(0\,; x_k, s_k^{\mathrm{CP}}) - M_k^{\mathrm{A}}(s\,; x_k, s_k^{\mathrm{CP}})$$

is negative; the model has increased. However, if we define $\alpha_0 = \min(\alpha_{\mathrm{B}}, 1) > 0$, where α_{B} is defined as

$$\alpha_{\mathrm{B}} = \min\left\{\left\{\frac{-[c_k + J_k s_k^{\mathrm{CP}}]_i}{[J_k s_k^{\mathrm{A}}]_i} : i \in \mathcal{V}_k \text{ and } [J_k s_k^{\mathrm{A}}]_i > 0\right\} \cup \{\infty\}\right\}, \tag{2.20}$$

i.e., α_{B} is the distance in the direction s_k^{A} to the first point of nondifferentiability, then it follows from Lemma 2.6 that the accelerator model does not increase for the step $s_k^{\mathrm{CA}} = \alpha_s s_k^{\mathrm{A}}$, where α_s is the minimizer of $M_k^{\mathrm{A}}(\alpha s_k^{\mathrm{A}})$ for $0 \le \alpha \le \alpha_0$ (see Figure 1).

LEMMA 2.6. *Let s_k^{A} be any local solution for problem* (EIQP)*, and define $\alpha_0 = \min(\alpha_B, 1) > 0$, where α_B is defined by* (2.20)*. It follows that the vector $s_k^{CA} = \alpha_s s_k^{A}$, where α_s is the minimizer of $M_k^{A}(\alpha s_k^{A})$ for $0 \le \alpha \le \alpha_0$, satisfies*

$$\Delta M_k^{A}(s_k^{CA}) \ge -\tfrac{1}{2}(g_k + H_k s_k^{CP} + \sigma J_k^T w_k)^T s_k^{A} \cdot \min\left(\alpha_s, -\frac{(g_k + H_k s_k^{CP} + \sigma J_k^T w_k)^T s_k^{A}}{n\|H_k\|_2(\Delta_k^{A})^2}\right) \ge 0. \tag{2.21}$$

Moreover;

$$\text{if } {s_k^{A}}^T H_k s_k^{A} > 0, \text{ then } (g_k + H_k s_k^{CP} + \sigma J_k^T w_k)^T s_k^{A} < 0. \tag{2.22}$$

Proof. We first prove (2.22). Suppose that ${s_k^{\mathrm{A}}}^T H_k s_k^{\mathrm{A}} > 0$ and that $(g_k + H_k s_k^{\mathrm{CP}} + \sigma J_k^T w_k)^T s_k^{\mathrm{A}} \ge 0$. These conditions and the fact that s_k^{A} is a local solution to problem (EIQP) imply that $s_k^{\mathrm{A}} = 0$. This contradicts ${s_k^{\mathrm{A}}}^T H_k s_k^{\mathrm{A}} > 0$, and therefore (2.22) holds.

To establish (2.21) we consider the quadratic function $q(\alpha) \stackrel{\mathrm{def}}{=} M_k^{\mathrm{A}}(\alpha s_k^{\mathrm{A}}) = a\alpha^2 + b\alpha + c$ defined on the interval $[0, \alpha_0]$, where

$$a = \tfrac{1}{2}{s_k^{\mathrm{A}}}^T H_k s_k^{\mathrm{A}}, \quad b = (g_k + H_k s_k^{\mathrm{CP}} + \sigma J_k^T w_k)^T s_k^{\mathrm{A}} \le 0, \quad \text{and} \quad c = \bar{f} + \sigma(c_k + J_k s_k^{\mathrm{CP}})^T w_k. \tag{2.23}$$

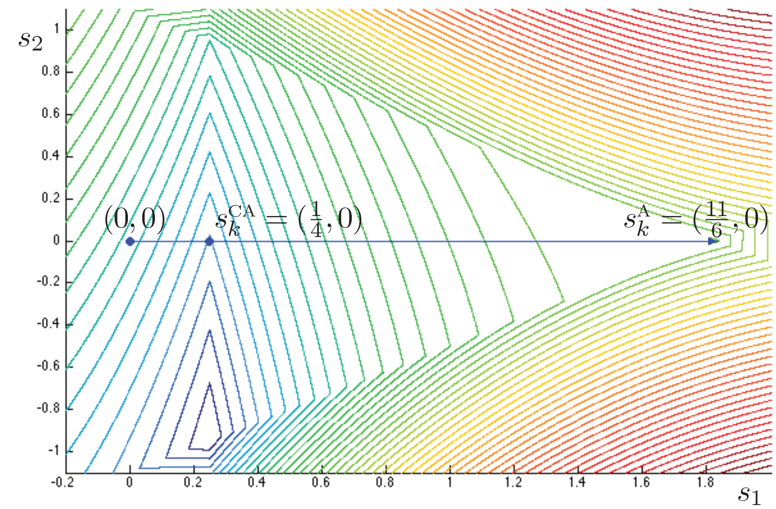
We consider three cases.

Case 1. $a > 0$ and $-b/2a > \alpha_0$. In this case we must have $\alpha_s = \alpha_0$. Using this fact, the definitions of $\Delta M_k^{\mathrm{A}}(s)$ and of $q(\alpha)$, and the inequalities $-b/2a > \alpha_0 = \alpha_s$ and $a > 0$, we have

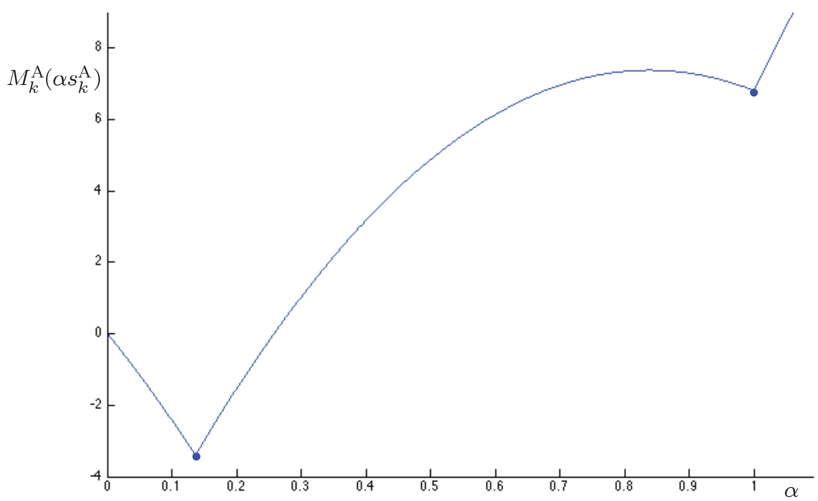
$$\Delta M_k^{\mathrm{A}}(\alpha_s s_k^{\mathrm{A}}) = q(0) - q(\alpha_s) = -a\alpha_s^2 - b\alpha_s \ge -\frac{1}{2}b\alpha_s = -\frac{\alpha_s}{2}(g_k + H_k s_k^{\mathrm{CP}} + \sigma J_k^T w_k)^T s_k^{\mathrm{A}}. \tag{2.24}$$

Case 2. $a > 0$ and $-b/2a \le \alpha_0$. In this case we must have $\alpha_s = -b/2a$. Using this fact, the definitions of $\Delta M_k^{\mathrm{A}}(s)$ and of $q(\alpha)$, norm inequalities, and the inequality $\|s_k^{\mathrm{A}}\|_\infty \le \Delta_k^{\mathrm{A}}$, we have

$$\begin{aligned}
\Delta M_k^{\mathrm{A}}(\alpha_s s_k^{\mathrm{A}}) = q(0) - q(\alpha_s) = -a\alpha_s^2 - b\alpha_s &= \frac{b^2}{4a} \\
&= \frac{\left((g_k + H_k s_k^{\mathrm{CP}} + \sigma J_k^T w_k)^T s_k^{\mathrm{A}}\right)^2}{2{s_k^{\mathrm{A}}}^T H_k s_k^{\mathrm{A}}} \\
&\ge \frac{\left((g_k + H_k s_k^{\mathrm{CP}} + \sigma J_k^T w_k)^T s_k^{\mathrm{A}}\right)^2}{2n\|H_k\|_2(\Delta_k^{\mathrm{A}})^2}.
\end{aligned} \tag{2.25}$$



(a) Contour graph of the function $M_k^{\text{A}}(s)$.



(b) Graph of $M_k^{\text{A}}(\alpha s_k^{\text{A}})$ for $0 \leq \alpha \leq 12/11$.

FIG. 1. (a) *Contour graph of* $M_k^A(s)$ *with local solution* $s_k^A = (11/6, 0)$. *The vector* $s_k^{CA} = (1/4, 0)$ *is a local minimizer when constrained to the direction* s_k^A. (b) *Graph of* $M_k^A(\alpha s_k^A)$ *for* $0 \leq \alpha \leq 12/11$. *Notice that* $\alpha_s = \alpha_B = 6/44$ *and that* $M_k^A(s_k^A) > M_k^A(0) > M_k^A(\alpha_s s_k^A) = M_k^A(s_k^{CA})$.

Case 3. $a \leq 0$. In this case we must have $\alpha_s = \alpha_0$. Using this fact, the definitions of $\Delta M_k^{\text{A}}(s)$ and of $q(\alpha)$, and the inequalities $a \leq 0$ and $b \leq 0$, we have

$$\begin{aligned} \Delta M_k^{\text{A}}(\alpha_s s_k^{\text{A}}) &= q(0) - q(\alpha_s) = -a\alpha_s^2 - b\alpha_s \geq \max(-a\alpha_s^2, -b\alpha_s) \\ &\geq -b\alpha_s \geq -\tfrac{1}{2}b\alpha_s = -\frac{\alpha_s}{2}(g_k + H_k s_k^{\text{CP}} + \sigma J_k^T w_k)^T s_k^{\text{A}}. \end{aligned} \tag{2.26}$$

The desired bound on $\Delta M_k^{\text{A}}(\alpha_s s_k^{\text{A}})$ follows from (2.24), (2.25), and (2.26). ∎

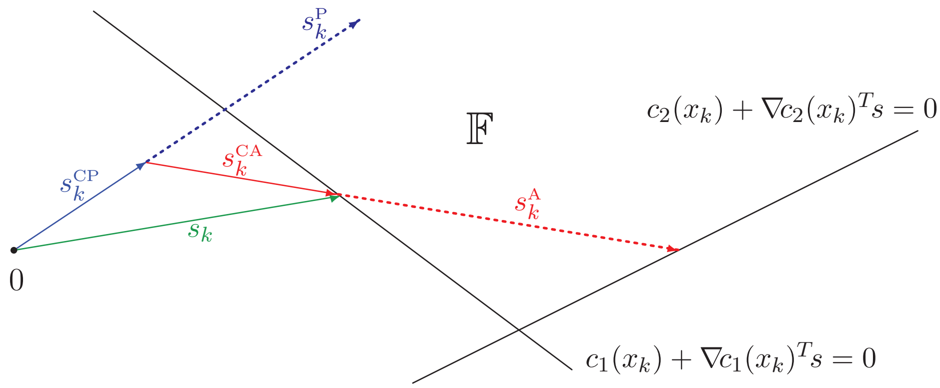


Fig. 2. *The relevant steps for subproblem* (EIQP) *with linearized constraints* $c_1(x_k) + \nabla c_1(x_k)^T s = 0$ *and* $c_2(x_k) + \nabla c_2(x_k)^T s = 0$ *and feasible region* $\mathbb{F}$. *We have illustrated the case* $\Delta M_k^A(s_k^A) < 0$ *so that the full step* s_k *is defined from the accelerator-Cauchy step as* $s_k = s_k^{CP} + s_k^{CA}$. *If the condition* $\Delta M_k^A(s_k^A) \geq 0$ *had been satisfied, then the full step would have been defined as* $s_k = s_k^{CP} + s_k^A$.

This result suggests that we define the full step as (see Figure 2)

$$s_k = \begin{cases} s_k^{\rm CP} + s_k^{\rm A} & \text{if } \Delta M_k^{\rm A}(s_k^{\rm A}) \geq 0, \\ s_k^{\rm CP} + s_k^{\rm CA} & \text{otherwise.} \end{cases} \tag{2.27}$$

The next result shows that this definition guarantees that condition (2.18) is satisfied in a nontrivial way.

Lemma 2.7. *If the* accelerator *step and full step are defined by* (2.27)*, then they satisfy*

$$\Delta M_k^H(s_k) \geq \Delta M_k^H(s_k^{CP}).$$

Proof. Using the definitions of $M_k^{\rm H}$, $\Delta M_k^{\rm H}$, $M_k^{\rm A}$, and $\Delta M_k^{\rm A}$, it follows that

$$\Delta M_k^{\rm H}(s_k) = M_k^{\rm H}(0) - M_k^{\rm H}(s_k^{\rm CP}) + M_k^{\rm H}(s_k^{\rm CP}) - M_k^{\rm H}(s_k) = \Delta M_k^{\rm H}(s_k^{\rm CP}) + \Delta M_k^{\rm A}(s_k - s_k^{\rm CP}).$$

We may then conclude that

$$\Delta M_k^{\rm H}(s_k) = \begin{cases} \Delta M_k^{\rm H}(s_k^{\rm CP}) + \Delta M_k^{\rm A}(s_k^{\rm A}) \geq \Delta M_k^{\rm H}(s_k^{\rm CP}) & \text{if } \Delta M_k^{\rm A}(s_k^{\rm A}) \geq 0, \\ \Delta M_k^{\rm H}(s_k^{\rm CP}) + \Delta M_k^{\rm A}(s_k^{\rm CA}) \geq \Delta M_k^{\rm H}(s_k^{\rm CP}) & \text{otherwise,} \end{cases}$$

by using Lemma 2.6 and (2.27). ∎

This result shows that if subproblem (EIQP) is used and the full step is defined by (2.27), then the full step is *always* at least as good as the Cauchy step, which is itself good enough to guarantee convergence. We stress that Lemma 2.7 is true as a direct result of the descent-constraint and is generally not true if the full step is defined from the traditional SQP subproblem. This property is, to our knowledge, the only result of its kind.

2.3.2. An equality-constrained accelerator subproblem. This particular accelerator subproblem includes a subset of the linearized constraints as *equality* constraints. The choice of which to include is based on a prediction of those constraints

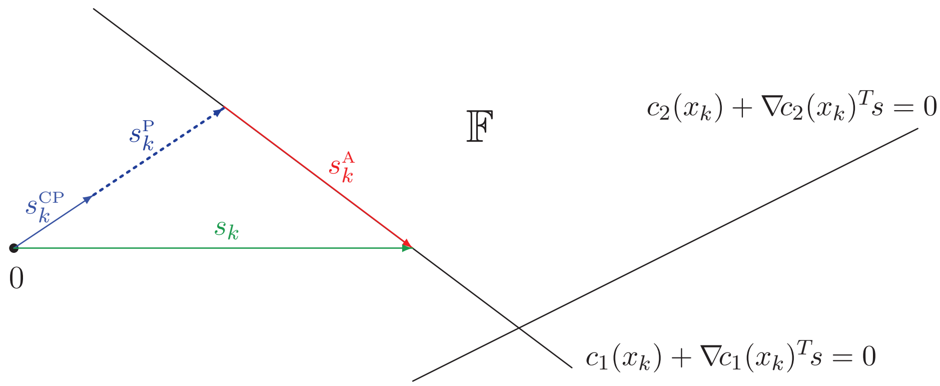


FIG. 3. *The relevant steps for subproblem* (EQP) *with linearized constraints* $c_1(x_k) + \nabla c_1(x_k)^T s = 0$ *and* $c_2(x_k) + \nabla c_2(x_k)^T s = 0$ *and feasible region* $\mathbb{F}$. *The illustration depicts the case* $\Delta M_k^H(s_k^P + s_k^A) \geq \eta \Delta M_k^H(s_k^{CP})$ *so that the full step is defined as* $s_k = s_k^P + s_k^A$. *By construction, the accelerator step* s_k^A *is contained in the null space of the constraints that are not strictly feasible at the predictor step* s_k^P, *i.e., the constraints whose indices are in the set* $\mathcal{A}(s_k^P) = \{1\}$. *Compare this with Figure* 2, *which shows that if the accelerator step* s_k^A *is computed from subproblem* (EIQP), *then it is* not *necessarily contained in the null space of the constraints that are not strictly feasible at the Cauchy step* s_k^{CP}.

that will be active at a solution to problem (NP). Our prediction is based on those constraints whose linearization is *not* strictly satisfied by the predictor step s_k^{P}. If we define the set $\mathcal{A}(s_k^{\text{P}}) = \{i : [c_k + J_k s_k^{\text{P}}]_i \leq 0\}$, then the subproblem takes the form

$$\text{(EQP)} \qquad \begin{array}{ll} \underset{s \in \mathbb{R}^n}{\text{minimize}} & \bar{f}_k + (g_k + H_k s_k^{\text{P}})^T s + \frac{1}{2} s^T H_k s \\ \text{subject to} & [J_k s]_{\mathcal{A}(s_k^{\text{P}})} = 0, \ \ \|s\|_2 \leq \Delta_k^{\text{A}}, \end{array}$$

where $\bar{f}_k = f_k + g_k^T s_k^{\text{P}} + \frac{1}{2} s_k^{\text{P}T} H_k s_k^{\text{P}}$. Note that we have used the two-norm for the trust-region constraint and that this does not change any of the theoretical results, since $\|s\|_\infty \leq \|s\|_2 \leq \Delta_k^{\text{A}}$ for all feasible vectors s. If a priori we knew the optimal active set, then we could compute a solution to problem (NP) by solving a sequence of equality constrained QP subproblems; this is equivalent to solving a sequence of subproblems of the form given by (EQP) if $\mathcal{A}(s_k^{\text{P}})$ agrees with the optimal active set. The fact that $\mathcal{A}(s_k^{\text{P}})$ does eventually agree with the optimal active set (under certain assumptions) may be deduced from Robinson [24, Theorem 3.1]. His result implies that if x^* is a solution to problem (NP) that satisfies the strong second-order sufficient conditions for optimality, then there exists a neighborhood of x^* such that if x_k is in this neighborhood, then the predictor step s_k^{P} will correctly identify the optimal active set, provided the trust-region constraint is inactive.

Let s_k^{A} denote the solution to subproblem (EQP). In an attempt to satisfy condition (2.18) in a nontrivial way, we define the full step s_k as (see Figure 3)

$$\text{(2.28)} \qquad s_k = \begin{cases} s_k^{\text{P}} + s_k^{\text{A}} & \text{if } \Delta M_k^{\text{H}}(s_k^{\text{P}} + s_k^{\text{A}}) \geq \eta \Delta M_k^{\text{H}}(s_k^{\text{CP}}), \\ s_k^{\text{CP}} & \text{otherwise.} \end{cases}$$

We note that this update strategy guarantees that the full step s_k satisfies condition (2.18), but perhaps in a trivial way.

Generally speaking, computing a solution to subproblem (EQP) is less expensive than computing a solution to subproblem (EIQP); this is certainly an advantage.

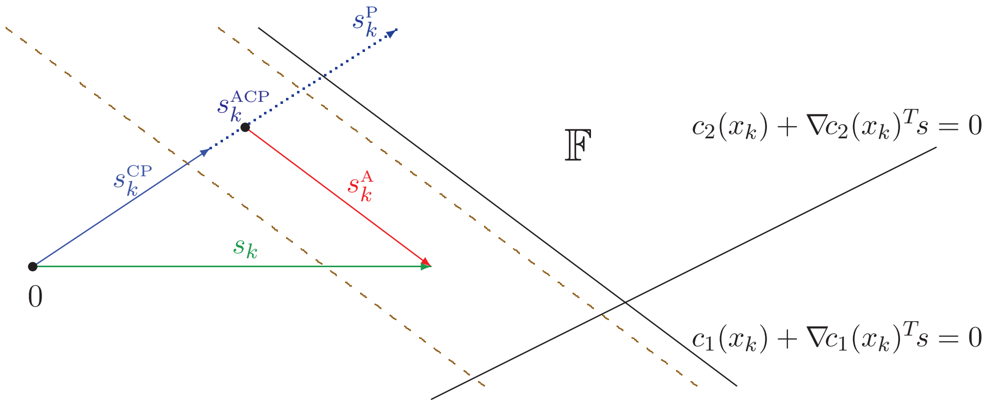


FIG. 4. *The relevant steps for subproblem* (EQP-ACP) *with linearized constraints* $c_1(x_k) + \nabla c_1(x_k)^T s = 0$ *and* $c_2(x_k) + \nabla c_2(x_k)^T s = 0$ *and feasible region* $\mathbb{F}$*. All points that lie along the direction* s_k^P *and between the two dashed lines are acceptable approximate Cauchy points as defined by* (2.17)*. By construction, the accelerator step* s_k^A *is contained in the null space of the constraints that are not strictly feasible at* s_k^{ACP}*, i.e., the constraints whose indices are in the set* $\mathcal{A}(s_k^{ACP}) = \{1\}$*. Compare this with Figure* 3.

However, problem (EQP) has at least two disadvantages. First, its success depends on correctly identifying the optimal active set, which may not always occur. If fact, if any of the strong second-order sufficient conditions do not hold at a minimizer, then the predictor step is not guaranteed to correctly identify the optimal active set. Second, the strategy for defining s_k may resort to satisfying condition (2.18) in a trivial way. When this occurs, the cost in computing s_k^{A} has essentially been wasted. To avoid this, we consider the alternative subproblem

$$\text{(EQP-ACP)} \qquad \begin{array}{ll} \underset{s\in\mathbb{R}^n}{\text{minimize}} & \bar{f} + (g_k + H_k s_k^{\mathrm{ACP}})^T s + \tfrac{1}{2} s^T H_k s \\ \text{subject to} & [J_k s]_{\mathcal{A}(s_k^{\mathrm{ACP}})} = 0, \ \ \|s\|_2 \le \Delta_k^{\mathrm{A}}, \end{array}$$

where s_k^{ACP} is an approximate Cauchy point as given by (2.17), $1 > \eta_{\mathrm{ACP}} \ge \eta > 0$, and $\mathcal{A}(s_k^{\mathrm{ACP}}) = \{i : [c_k + J_k s_k^{\mathrm{ACP}}]_i \le 0\}$ (see Figure 4). It follows that if we define

$$s_k = s_k^{\mathrm{ACP}} + \alpha s_k^{\mathrm{A}}, \tag{2.29}$$

where $0 < \alpha \le 1$ ensures that the step αs_k^{A} is feasible for linearized constraints i such that $i \notin \mathcal{A}(s_k^{\mathrm{ACP}})$, then

$$\Delta M_k^{\mathrm{H}}(s_k) \ge \Delta M_k^{\mathrm{H}}(s_k^{\mathrm{ACP}}) \ge \eta_{\mathrm{ACP}} \Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) \ge \eta \Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}). \tag{2.30}$$

The first inequality follows, since $\Delta M_k^{\mathrm{H}}(s_k) = \Delta M_k^{\mathrm{H}}(s_k^{\mathrm{ACP}}) + \Delta M_k^{\mathrm{A}}(s_k^{\mathrm{A}})$ and $\Delta M_k^{\mathrm{A}}(s_k^{\mathrm{A}}) \ge 0$; the latter is ensured, since the *global* solution s_k^{A} of problem (EQP-ACP) may be found. We note, however, that an appropriate *approximate* solution to problem (EQP) may be found by using GLTR [18] with a constraint preconditioner [9, section 7.5.4]; GLTR guarantees model decrease during every step of its iterative process. The second and third inequalities follow from (2.17) and the inequality $\eta \le \eta_{\mathrm{ACP}}$ that holds by construction. Therefore, s_k is guaranteed to satisfy condition (2.18) in a nontrivial way, and computation is never wasted. Note that $s_k^{\mathrm{ACP}} = s_k^{\mathrm{CP}}$ satisfies (2.17) trivially. However, since we show in [20] that the *predictor* step ultimately correctly identifies the set of constraints active at a minimizer of ϕ, it may be more efficient to define an approximate Cauchy point by performing a backtracking search from the predictor step; this is allowed, provided we satisfy (2.17). We conclude this section by mentioning that the GALAHAD library contains the software package EQP, which may be used to solve subproblems (EQP) and (EQP-ACP).

2.3.3. An implicitly inequality-constrained accelerator subproblem. In this section we briefly discuss a third possibility for computing an accelerator step. Motivated by traditional SQP *correction* steps [9, section 15.3.2.3] which are intended to avoid the Maratos effect [22], we would like to formulate accelerator subproblems with *shifted* variants of the linearized constraints. However, this is not completely reasonable, since the shifted constraints may lead to infeasible accelerator subproblems. An alternative is to try to satisfy the shifted linearized constraints *implicitly* by moving them into the objective function via an ℓ_1-penalty term. This assures us that the accelerator subproblem will be feasible, but the resultant accelerator step may not decrease the faithful model M_k^{H}. Therefore, in this case we define the full step as

$$s_k = \begin{cases} s_k^{\text{CP}} + s_k^{\text{A}} & \text{if } \Delta M_k^{\text{H}}(s_k^{\text{CP}} + s_k^{\text{A}}) \geq \Delta M_k^{\text{H}}(s_k^{\text{CP}}), \\ s_k^{\text{CP}} & \text{otherwise.} \end{cases} \tag{2.31}$$

Note that use of this strategy may result in some iterations satisfying (2.18) in a trivial way. However, since these subproblems would be intended to avoid the Maratos effect, they are likely to be used asymptotically, and this is precisely the situation for which we may expect them to give sufficient decrease. For further details see [21, section 2.3.3].

3. The algorithm. In this section we describe Algorithm 3.1—an SQP algorithm for computing a first-order critical point for problem (ℓ_1-σ). First, the user supplies an initial estimate (x_0, y_0) of a solution and then initial trust-region radii Δ_0^{P} and Δ_0^{A},“success” parameters $0 < \eta_S \leq \eta_{VS} < 1$, a maximum allowed predictor trust-region radius Δ_u, expansion and contraction factors $0 < \eta_c < 1 < \eta_e$, sufficient model decrease and approximate Cauchy point tolerances $0 < \eta \leq \eta_{\text{ACP}} < 1$, and accelerator trust-region radius factor τ_f are defined. With parameters set, the main “do-while” loop begins. First, the problem functions are evaluated at the current point (x_k, y_k). Next, a symmetric positive definite matrix B_k is defined, and the predictor step s_k^{P} is computed as a solution to problem (2.1). Simple choices for B_k would be the identity matrix or perhaps a scaled diagonal matrix that attempts to model the “essential properties” of the matrix $\nabla_{xx}\mathcal{L}(x_k, y_k)$. However, computing B_k via a limited-memory quasi-Newton update is an attractive option. We leave further discussion of the matrix B_k to a separate paper. Next, we solve problem (2.6) for the Cauchy step s_k^{CP}, calculate the decrease in the model M_k^{H} as given by $\Delta M_k^{\text{H}}(s_k^{\text{CP}})$, and compute an accelerator step if we believe that it will be advantageous. The step computation is completed by defining a full step s_k that satisfies condition (2.18). This may be done in three ways. First, the accelerator step may be skipped and then we simply define $s_k = s_k^{\text{CP}}$; this satisfies condition (2.18) in a trivial way. Second, we may solve either of the subproblems discussed in sections 2.3.1 or 2.3.2 and then define s_k according to (2.27) or (2.28) and (2.29), respectively. Third, we may compute an accelerator step as briefly described in section 2.3.3 and then define s_k according to (2.31). In all cases we are guaranteed that s_k satisfies condition (2.18). Next, we compute $\phi(x_k + s_k)$ and $\Delta M_k^{\text{H}}(s_k)$ and then calculate the ratio r_k of actual versus predicted decrease in the merit function.

Our strategy for updating the predictor trust-region radius and for accepting or rejecting candidate steps is identical to that used by Fletcher [13] and is determined by the ratio r_k. More precisely, if the ratio r_k is larger than η_{VS}, then we believe that the model is a very accurate representation of the true merit function within the current trust-region; therefore, we increase the predictor trust-region radius with the belief that the current trust-region radius may be overly restrictive. If the ratio is greater than η_S, then we believe the model is sufficiently accurate, and we keep the predictor trust-region radius fixed. Otherwise, the ratio indicates that there is poor agreement between the model M_k^{H} and the merit function, and, therefore, we decrease the predictor trust-region radius, with the hope that the model will accurately approximate the merit function over the smaller trust-region. As for step acceptance or rejection, we accept any iterate for which r_k is positive, since this indicates that the merit function has decreased. Next, the dual vector y_{k+1} is updated, but for proving *global convergence* the particular choice is not important. To emphasize this point, we do not specify any particular update in Algorithm 3.1. However, a reasonable strategy would be to use the multiplier vector from the solution of the smooth predictor subproblem (2.3). The specific update to y_k becomes essential when considering the rate-of-convergence and then the most obvious choice becomes the multiplier vector from whichever accelerator subproblem is used. In fact, we show in a companion paper [20] that updating y_k with the multipliers from various accelerator subproblems in section 2.3 ensures quadratic *local convergence*, under certain assumptions.

Finally, we have the additional responsibility of updating the accelerator trust-region radius. In Algorithm 3.1 we set the accelerator trust-region radius to a constant multiple of the predictor trust-region radius, although the condition $\Delta_{k+1}^{\text{A}} \le \tau_f \cdot \Delta_{k+1}^{\text{P}}$ for some constant τ_f is also sufficient. Although this update is simple and may be viewed as "obvious," we believe that it deserves extra discussion. If the predictor trust-region radius is *not* converging to zero on any subsequence, then the algorithm must be making good progress in reducing the merit function. However, a delicate situation arises when the trust-region radius does converge to zero on some subsequence. Since the predictor step must also be converging to zero, it seems natural to require that the full step s_k also converge to zero. Therefore it seems intuitive to require that if $\{x_{k_j}\}_{j\ge 0}$ is any subsequence such that $\lim_{j\to\infty} \|s_{kj}^{\text{P}}\|_\infty = 0$, then the sequence

$$\{\Delta_{kj}^{\text{A}} / \|s_{kj}^{\text{P}}\|_\infty\}_{j\ge 0} \text{ remains bounded.} \tag{3.1}$$

A simple way to ensure this condition is by defining the accelerator trust-region radius as $\Delta_{k+1}^{\text{A}} \leftarrow \tau_f \cdot \|s_k^{\text{P}}\|_\infty$; i.e., set the accelerator trust-region radius to be a constant multiple of the size of the predictor *step*. This condition is sufficient for proving convergence, but we prefer the alternate update $\Delta_{k+1}^{\text{A}} \leftarrow \tau_f \cdot \Delta_{k+1}^{\text{P}}$; i.e., set the accelerator trust-region radius to be a constant multiple of the size of predictor *radius*. Corollary 4.2 shows that, in fact, they are equivalent asymptotically, but the update $\Delta_{k+1}^{\text{A}} \leftarrow \tau_f \cdot \Delta_{k+1}^{\text{P}}$ allows for a larger value of Δ_k^{A} globally and has been observed to perform better during initial testing.

ALGORITHM 3.1 MINIMIZING THE ℓ_1-PENALTY FUNCTION.
Input: (x_0, y_0)
Set parameters $0 < \eta_S \le \eta_{VS} < 1$, $\Delta_u > 0$, $0 < \eta \le \eta_{\text{ACP}} < 1$, and $\tau_f > 0$.
Initialize predictor radius Δ_0^{P} and then set accelerator radius $\Delta_0^{\text{A}} \leftarrow \tau_f \Delta_0^{\text{P}}$.
Set expansion and contraction factors $0 < \eta_c < 1 < \eta_e$.
$k \leftarrow 0$.

do
Evaluate f_k, g_k, c_k, J_k, and then compute ϕ_k.
Define B_k to be a positive definite symmetric approximation to $\nabla_{xx}\mathcal{L}(x_k, y_k)$.
Solve problem (2.1) for s_k^{P}.
Define H_k to be a symmetric approximation to $\nabla_{xx}\mathcal{L}(x_k, y_k)$.
Solve problem (2.6) for s_k^{CP} and compute $\Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}})$.
Compute an accelerator step s_k^{A} (optional) as described in section 2.
if an accelerator step from section 2.3.1 is computed, **then**
define s_k by (2.27);
else if an accelerator step is computed from problem (EQP) of section 2.3.2, **then**
define s_k by (2.28);
else if an accelerator step is computed from problem (EQP-ACP) of section 2.3.2, **then**
define s_k by (2.29);
else if an accelerator step from section 2.3.3 is computed, **then**
define s_k by (2.31);
else
set $s_k = s_k^{\mathrm{CP}}$.
end if
Evaluate $\phi(x_k + s_k)$ and $\Delta M_k^{\mathrm{H}}(s_k)$.
Compute $r_k = \big(\phi_k - \phi(x_k + s_k)\big)/\Delta M_k^{\mathrm{H}}(s_k)$.
if $r_k \geq \eta_{VS}$, **then** [very successful]
$\Delta_{k+1}^{\mathrm{P}} \leftarrow \min(\eta_e \cdot \Delta_k^{\mathrm{P}}, \Delta_u)$ [increase predictor radius]
else if $r_k \geq \eta_S$, **then** [successful]
$\Delta_{k+1}^{\mathrm{P}} \leftarrow \Delta_k^{\mathrm{P}}$ [keep predictor radius]
else [unsuccessful]
$\Delta_{k+1}^{\mathrm{P}} \leftarrow \eta_c \cdot \Delta_k^{\mathrm{P}}$ [decrease predictor radius]
end
if $r_k > 0$, **then** [accept step]
$x_{k+1} \leftarrow x_k + s_k$
$y_{k+1} \leftarrow$ any reasonably chosen estimate
else [reject step]
$x_{k+1} \leftarrow x_k$
$y_{k+1} \leftarrow y_k$
end
$\Delta_{k+1}^{\mathrm{A}} \leftarrow \tau_f \cdot \Delta_{k+1}^{\mathrm{P}}$ [update accelerator radius]
$k \leftarrow k + 1$
end do

4. Convergence. This section shows that Algorithm 3.1 is globally convergent. Our main result is that under certain assumptions, there exists a subsequence of the iterates generated by Algorithm 3.1 that converges to a first-order solution of problem (ℓ_1-σ). The proof requires two preliminary results as well as two estimates. First, since $f(x)$ and $c(x)$ are continuously differentiable by assumption, there exists a positive constant M such that

$$\left\| \begin{pmatrix} g(x)^T \\ J(x) \end{pmatrix} \right\|_\infty \leq M \quad \text{for all } x \in \mathcal{B}, \tag{4.1}$$

where $\mathcal{B}$ is a closed, bounded, convex subset of $\mathbb{R}^n$. Second, since the function $h(f,c) = f + \sigma\|[c]^-\|_1$ is convex, there exists a positive constant L such that

$$|h(f_1,c_1) - h(f_2,c_2)| \le L \left\| \begin{pmatrix} f_1 - f_2 \\ c_1 - c_2 \end{pmatrix} \right\|_\infty \tag{4.2}$$

for all (f_1,c_1) and $(f_2,c_2) \in \big(f(\mathcal{B}), c(\mathcal{B})\big)$ [25, Theorem 10.4]. Using these bounds we may now state the following lemma, which provides a lower bound on the size of the predictor step. This is essentially [28, Lemma 3.2], except for the use of the infinity norm.

LEMMA 4.1. *Let $x_k \in \mathcal{B}$ so that (4.1) and (4.2) hold. Then, if $\|s_k^{\scriptscriptstyle P}\|_\infty < \Delta_k^{\scriptscriptstyle P}$, then*

$$\|s_k^{\scriptscriptstyle P}\|_\infty \ge \frac{1}{2} \Delta_{max}^{\scriptscriptstyle L}(x_k, 1) \min\left(\frac{1}{LM}, \frac{1}{n(1+\Delta_u)\|B_k\|_2} \right). \tag{4.3}$$

COROLLARY 4.2. *Suppose that $\{x_k\}_{k\ge 0} \subset \mathcal{B}$ so that (4.1) and (4.2) hold and that K is a subsequence of the integers such that the following hold:*

(i) *There exists a number δ such that $\Delta_{max}^{\scriptscriptstyle L}(x_k, 1) \ge \delta > 0$ for all $k \in K$;*
(ii) *There exists a positive constant b_B such that $\|B_k\|_2 \le b_B$ for all $k \in K$;*
(iii) $\lim_{k\in K} \Delta_k^{\scriptscriptstyle P} = 0$.

Then

$$\|s_k^{\scriptscriptstyle P}\|_\infty = \Delta_k^{\scriptscriptstyle P} \ \textit{for all } k \in K \textit{ sufficiently large.} \tag{4.4}$$

Proof. If (4.3) holds, then (i) and (ii) of Corollary 4.2 imply that $\|s_k^{\rm P}\|_\infty$ is *strictly* bounded away from zero for all $k \in K$. However, this contradicts assumption (iii) of Corollary 4.2 for $k \in K$ sufficiently large, since $\|s_k^{\rm P}\|_\infty \le \Delta_k^{\rm P}$. Therefore, (4.3) must not be true, and Lemma 4.1 implies that $\|s_k^{\rm P}\|_\infty = \Delta_k^{\rm P}$ for all $k \in K$ sufficiently large. ∎

We may now state our main result. The organization of the proof is based on Fletcher [13, Theorem 14.5.1], and the proof of Case 1 of Theorem 4.3 is nearly identical.

THEOREM 4.3. *Let f and c be twice continuously differentiable functions, and let $\{x_k\}$, $\{H_k\}$, $\{B_k\}$, $\{\Delta_k^{\scriptscriptstyle P}\}$, and $\{\Delta_k^{\scriptscriptstyle A}\}$ be sequences generated by Algorithm 3.1. Assume that the following conditions hold:*

1. *$\{x_k\}_{k\ge 0} \subset \mathcal{B} \subset \mathbb{R}^n$, where $\mathcal{B}$ is a closed, bounded, convex set; and*
2. *There exists positive constants b_B and b_H such that $\|B_k\|_2 \le b_B$ and $\|H_k\|_2 \le b_H$ for all $k \ge 0$.*

Then, either x_K is a first-order critical point for problem (ℓ_1-σ) for some $K \ge 0$, or there exists a subsequence of $\{x_k\}$ that converges to a first-order solution of problem (ℓ_1-σ).

Proof. If x_K is a first-order point for problem (ℓ_1-σ) for some $K \ge 0$, then we are done. Therefore, we assume that x_k is not a first-order solution to problem (ℓ_1-σ) for all k. We consider two cases.

Case 1. There exists a subsequence of $\{\Delta_k^{\rm P}\}$ that converges to zero. Examination of the algorithm shows that this implies the existence of a subsequence S of the integers such that

$$\lim_{k\in S} x_k = x_*, \tag{4.5}$$

$$\lim_{k\in S} \Delta_k^{\rm P} = 0, \tag{4.6}$$

$$\lim_{k\in S} \|s_k^{\rm P}\|_\infty = 0, \ \text{and} \tag{4.7}$$

$$r_k < \eta_{\scriptscriptstyle S} \ \text{for all} \ k \in S. \tag{4.8}$$

For a proof by contradiction, we suppose that x_* is not a first-order critical point. This implies that there exists a direction s and a scalar $\rho > 0$ such that $\|s\|_\infty = 1$ and

$$\max_{y\in\partial\,\|[c_*]^-\|_1} s^T(g_* + \sigma J_*^T y) = -\rho, \tag{4.9}$$

where $\partial\,\|[c_*]^-\|_1$ is the subdifferential of $\|[\cdot]^-\|_1$ at the point c_* (see [13, section 14.3] for more details). A Taylor expansion of f at x_k in a general direction v gives

$$f(x_k + \varepsilon v) = f_k + \varepsilon g_k^T v + o(\varepsilon) = f_k + \varepsilon g_k^T v + \frac{\varepsilon^2}{2} v^T H_k v + o(\varepsilon), \tag{4.10}$$

since $\{H_k\}$ is bounded by assumption, while a Taylor expansion of c at x_k gives

$$c(x_k + \varepsilon v) = c_k + \varepsilon J_k v + o(\varepsilon). \tag{4.11}$$

Combining these two equations gives

$$\begin{aligned}\phi(x_k + \varepsilon v) &= f_k + \varepsilon g_k^T v + \frac{\varepsilon^2}{2} v^T H_k v + o(\varepsilon) + \sigma\|[c_k + \varepsilon J_k v + o(\varepsilon)]^-\|_1 \\ &= f_k + \varepsilon g_k^T v + \frac{\varepsilon^2}{2} v^T H_k v + \sigma\|[c_k + \varepsilon J_k v]^-\|_1 + o(\varepsilon) \\ &= M_k^{\mathrm{H}}(\varepsilon v) + o(\varepsilon),\end{aligned} \tag{4.12}$$

where the first equality follows from the definition of ϕ and the Taylor expansions, the second equality follows from the boundedness of $\partial\,\|[\cdot]^-\|_1$, and the last equality follows from the definition of $M_k^{\mathrm{H}}(\varepsilon v)$. The same argument using B_k in place of H_k gives the estimate

$$\phi(x_k + \varepsilon v) = M_k^{\mathrm{B}}(\varepsilon v) + o(\varepsilon). \tag{4.13}$$

Choosing $v = s_k/\|s_k\|_\infty$ and $\varepsilon = \|s_k\|_\infty$ in (4.12) and $v = s$ and $\varepsilon = \varepsilon_k$ (we have not yet defined ε_k) in (4.13) yields

$$\phi(x_k + s_k) = M_k^{\mathrm{H}}(s_k) + o(\|s_k\|_\infty) \text{ and} \tag{4.14}$$

$$\phi(x_k + \varepsilon_k s) = M_k^{\mathrm{B}}(\varepsilon_k s) + o(\varepsilon_k). \tag{4.15}$$

Equation (4.14) then implies the equation

$$r_k = \frac{\phi_k - \phi(x_k + s_k)}{\Delta M_k^{\mathrm{H}}(s_k)} = \frac{\Delta M_k^{\mathrm{H}}(s_k) + o(\|s_k\|_\infty)}{\Delta M_k^{\mathrm{H}}(s_k)} = 1 + \frac{o(\|s_k\|_\infty)}{\Delta M_k^{\mathrm{H}}(s_k)}. \tag{4.16}$$

We now proceed to bound $\Delta M_k^{\mathrm{H}}(s_k)$. For all $k \in S$ we have

$$\begin{aligned}\Delta M_k^{\mathrm{H}}(s_k) &\geq \eta \Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) && (4.17)\\ &\geq \eta \Delta M_k^{\mathrm{H}}(s_k^{\mathrm{P}}) && (4.18)\\ &= \eta\big(M_k^{\mathrm{H}}(0) - M_k^{\mathrm{H}}(s_k^{\mathrm{P}})\big) && (4.19)\\ &= \eta\big(M_k^{\mathrm{B}}(0) - M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) - \tfrac{1}{2}{s_k^{\mathrm{P}}}^T(H_k - B_k)s_k^{\mathrm{P}}\big) && (4.20)\\ &= \eta \Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) - \frac{\eta}{2}{s_k^{\mathrm{P}}}^T(H_k - B_k)s_k^{\mathrm{P}} && (4.21)\\ &= \eta \Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) + o(\|s_k^{\mathrm{P}}\|_\infty). && (4.22)\end{aligned}$$

Inequalities (4.17) and (4.18) follow from assumption (2.18) and since the Cauchy step maximizes $\Delta M_k^{\mathrm{H}}(s)$ in the direction s_k^{P}. Equations (4.19)–(4.21) follow from the definitions of ΔM_k^{H} and ΔM_k^{B} and by introducing B_k. Finally, (4.22) follows, since $\{B_k\}$ and $\{H_k\}$ are bounded by assumption.

We now define the scalar-valued sequence $\{\varepsilon_k\}_{k\geq 0}$ such that $\varepsilon_k = \|s_k^{\mathrm{P}}\|_\infty$. It follows that $\|\varepsilon_k s\|_\infty = \|s_k^{\mathrm{P}}\|_\infty$, and, therefore, the vector $\varepsilon_k s$ is feasible for the kth predictor subproblem. It follows that for all $k \in S$ sufficiently large we have

$$\begin{aligned}
&(4.23) && \Delta M_k^{\mathrm{H}}(s_k) \geq \eta \Delta M_k^{\mathrm{B}}(\varepsilon_k s) + o(\|s_k^{\mathrm{P}}\|_\infty) \\
&(4.24) && \qquad = \eta\big(\phi_k - \phi(x_k + \varepsilon_k s)\big) + o(\|s_k^{\mathrm{P}}\|_\infty) \\
&(4.25) && \qquad \geq \eta \varepsilon_k \big(\rho + o(1)\big) + o(\|s_k^{\mathrm{P}}\|_\infty) \\
&(4.26) && \qquad = \eta\rho\varepsilon_k + o(\varepsilon_k) + o(\|s_k^{\mathrm{P}}\|_\infty) \\
&(4.27) && \qquad = \eta\rho\|s_k^{\mathrm{P}}\|_\infty + o(\|s_k^{\mathrm{P}}\|_\infty),
\end{aligned}$$

where we have used the convention $\zeta(\varepsilon_k) = o(1)$ to mean that $\zeta(\varepsilon_k) \to 0$ as $\varepsilon_k \to 0$. Inequality (4.23) follows from (4.22) and since s_k^{P} is a global minimizer for the kth predictor subproblem. Equation (4.24) follows from (4.15), while inequality (4.25) follows from [13, Corollary to Lemma 14.5.1]. Finally, (4.26) and (4.27) follow from algebra and definition of ε_k.

Equation (4.27) implies the existence of a positive sequence $\{z_k\}$ such that for $k \in S$ sufficiently large

$$\begin{aligned}
&(4.28) && \left|\frac{o(\|s_k\|_\infty)}{\Delta M_k^{\mathrm{H}}(s_k)}\right| \leq \left|\frac{o(\|s_k\|_\infty)}{\eta\rho\|s_k^{\mathrm{P}}\|_\infty + o(\|s_k^{\mathrm{P}}\|_\infty)}\right| \\
&(4.29) && \qquad \leq \frac{z_k\|s_k\|_\infty}{\frac{1}{2}\eta\rho\|s_k^{\mathrm{P}}\|_\infty} \\
&(4.30) && \qquad \leq \frac{2z_k(\|s_k^{\mathrm{P}}\|_\infty + \|s_k^{\mathrm{A}}\|_\infty)}{\eta\rho\|s_k^{\mathrm{P}}\|_\infty} \\
&(4.31) && \qquad = \frac{2z_k}{\eta\rho}\left(1 + \frac{\|s_k^{\mathrm{A}}\|_\infty}{\|s_k^{\mathrm{P}}\|_\infty}\right)
\end{aligned}$$

and where $\{z_k\}_S$ is a subsequence that converges to zero as $k \to \infty$. Inequality (4.28) follows from inequality (4.27), while inequality (4.29) follows from the definition of "little-oh." Inequality (4.30) follows from the triangle-inequality, the updates (2.27), (2.28), (2.29), and (2.31), and the construction of the steps s_k^{CP}, s_k^{ACP}, and s_k^{CA}. Finally, inequality (4.31) follows from simplification.

We now show that the assumptions in Corollary 4.2 are satisfied. Since x_* is not first-order optimal by assumption, it follows from part (v) of Lemma 1.1 that $\Delta_{\max}^{\mathrm{L}}(x_*, 1) \neq 0$. From part (iv) of Lemma 1.1 it follows that $\Delta_{\max}^{\mathrm{L}}(x_k, 1)$ is strictly bounded away from zero for $k \in S$ sufficiently large; this is assumption (i) of Corollary 4.2. Assumptions (ii) and (iii) of Corollary 4.2 follow directly from the assumptions in this theorem and the case we are considering.

Equation (4.31), Corollary 4.2, and the accelerator trust-region radius update used in Algorithm 3.1 imply

$$(4.32) \qquad \left|\frac{o(\|s_k\|_\infty)}{\Delta M_k^{\mathrm{H}}(s_k)}\right| \leq \frac{2z_k}{\eta\rho}\left(1 + \frac{\|s_k^{\mathrm{A}}\|_\infty}{\Delta_k^{\mathrm{P}}}\right) \leq \frac{2(1+\tau_f)z_k}{\eta\rho}.$$

Finally, inequalities (4.16) and (4.32) show that

$$r_k = 1 + o(1) \quad \text{for} \quad k \in S. \tag{4.33}$$

This is a contradiction, since this implies that for $k \in S$ sufficiently large, the identity $r_k > \eta_S$ holds, which violates (4.8). Thus, x^* is a first-order critical point if Case 1 of Theorem 4.3 occurs.

Case 2. There does not exists a subsequence of $\{\Delta_k^{\text{P}}\}$ that converges to zero. Examination of the algorithm shows that this implies the existence of a positive number δ and of an infinite subsequence S of the integers such that

$$\lim_{k \in S} x_k = x_*, \tag{4.34}$$

$$\Delta_k^{\text{P}} \geq \delta > 0 \quad \text{for all} \quad k, \tag{4.35}$$

$$r_k \geq \eta_S \quad \text{for all} \quad k \in S. \tag{4.36}$$

Equation (2.18) and the fact that each $k \in S$ is a successful iterate imply

$$\phi_k - \phi(x_k + s_k) \geq \eta_S \Delta M_k^{\text{H}}(s_k) \geq \eta\eta_S \Delta M_k^{\text{H}}(s_k^{\text{CP}}). \tag{4.37}$$

Corollary 2.5, (4.35), the bounds b_B and b_H on B_k and H_k, and the bound $\Delta_k^{\text{P}} \leq \Delta_u$ imply

$$\phi_k - \phi(x_k + s_k) \geq \frac{\eta\eta_S}{4} \varDelta_{\text{max}}^{\text{L}}(x_k, 1) \min(\mathcal{S}_k), \tag{4.38}$$

where

$$\mathcal{S}_k = \left\{ 1,\ \delta,\ \frac{\varDelta_{\text{max}}^{\text{L}}(x_k,1)}{b_B},\ \frac{\varDelta_{\text{max}}^{\text{L}}(x_k,1)}{b_B\Delta_u^2},\ \frac{\varDelta_{\text{max}}^{\text{L}}(x_k,1)}{2n(b_B+b_H)},\ \frac{\varDelta_{\text{max}}^{\text{L}}(x_k,1)}{2n(b_B+b_H)\Delta_u^2},\right.$$
$$\left. \frac{\left(\varDelta_{\text{max}}^{\text{L}}(x_k,1)\right)^3}{2n(b_B+b_H)b_B^2\Delta_u^2},\ \frac{\left(\varDelta_{\text{max}}^{\text{L}}(x_k,1)\right)^3}{2n(b_B+b_H)b_B^2\Delta_u^6} \right\}.$$

Summing over all $k \in S$ yields

$$\sum_{k \in S} \phi_k - \phi(x_k + s_k) \geq \sum_{k \in S} \frac{\eta\eta_S}{4} \varDelta_{\text{max}}^{\text{L}}(x_k, 1) \min(\mathcal{S}_k). \tag{4.39}$$

Next, using the monotonicity of $\{\phi(x_k)\}_{k \geq 0}$ it follows that

$$\sum_{k \in S} \phi_k - \phi(x_k + s_k) = \sum_{k \in S} \phi_k - \phi(x_{k+1}) \leq \phi(x_0) - \phi(x_*). \tag{4.40}$$

Combining the two previous inequalities gives

$$\phi(x_0) - \phi(x_*) \geq \sum_{k \in S} \frac{\eta\eta_S}{4} \varDelta_{\text{max}}^{\text{L}}(x_k, 1) \min(\mathcal{S}_k), \tag{4.41}$$

which implies

$$\lim_{k \in S} \varDelta_{\text{max}}^{\text{L}}(x_k, 1) = 0, \tag{4.42}$$

since the series on the right-hand side is convergent. Parts (iv) and (v) of Lemma 1.1 then imply that $\varDelta_{\text{max}}^{\text{L}}(x_*, 1) = 0$ and that x_* is a first-order critical point.

In both cases we have shown that there exists a limit point x_* that is a first-order critical point. We are done, since one of these cases must occur. ∎

As stated previously, the proof of Case 1 of Theorem 4.3 is nearly identical to that given by Fletcher. However, his proof for Case 2 of Theorem 4.3 does not carry over to our setting because he essentially requires the global minimizer of M_k^{H} over the trust-region defined by radius Δ_k^{P}, while we compute only the global minimizer of M_k^{H} in the *single* direction s_k^{P}.

It is possible to weaken the assumption on the boundedness of $\|H_k\|_2$ in Theorem 4.3. In fact, Algorithm 3.1 is still globally convergent if there exists positive constants c_1 and c_2 such that either

$$\|H_k\|_2 \leq c_1 + c_2 \sum_{i=1}^{k} \|s_k\|_\infty \quad \text{or} \quad \|H_k\|_2 \leq c_1 + c_2 k \quad \text{for all } k \geq 0,$$

provided we use the modified update

$$\Delta_k^{\text{A}} \leftarrow \tau_f \|s_k^{\text{P}}\|_\infty$$

for the accelerator trust-region radius. It can be shown [12] that these conditions hold for many quasi-Newton updating formula under reasonable assumptions. We give no further details here but rather point the reader to [21].

5. Conclusions and future work. Research on second derivative SQP methods is very active. The optimization community continues to tangle with the difficulties associated with nonconvex subproblems in an attempt to further our understanding of these methods. This paper has provided additional understanding by showing that the relatively simple idea of imposing descent (via an artificial constraint) guarantees that certain pitfalls typically associated with second derivative SQP algorithms may be avoided.

We presented an SQP method that is based on the work by Fletcher [13]. In section 2, we described how to compute trial steps using a predictor step, a Cauchy step, and an (optional) accelerator step. Since there is considerable flexibility in defining the accelerator step from various subproblems, we explored three options: section 2.3.1 discussed an explicitly inequality-constrained quadratic programming (EIQP) subproblem that was enhanced by an artificial descent-constraint; section 2.3.2 considered an equality-constrained quadratic programming (EQP) subproblem; and section 2.3.3 briefly described a class of implicitly inequality-constrained quadratic programming subproblems that were motivated by traditional strategies for avoiding the Maratos effect. We feel that the flexibility in our algorithm provides a natural framework for avoiding the Maratos effect that is less ad hoc than traditional means. The key to an effective and efficient implementation of our method is the careful utilization of the advantages that each subproblem enjoys.

In section 4, we proved that our method is globally convergent without having to compute the global minimizer of a nonconvex quadratic program; this is arguably the greatest contribution of this paper. Moreover, we provided two reasonable strategies for computing accelerator steps that are *guaranteed* to *not* be an ascent direction for the ℓ_1-merit function. This result does not hold for traditional second derivative SQP methods.

In a companion paper we plan to (1) discuss strategies for updating the penalty parameter; (2) investigate local convergence issues; (3) discuss mechanisms for defining convex approximations to the Hessian of the Lagrangian in the large-scale case;

and (4) provide numerical experiments with our evolving GALAHAD package S2QP. We note that Byrd, Nocedal, and Waltz [6] and Byrd et al. [4] have published clever techniques for updating the penalty parameter, and this will influence our developments.

Acknowledgments. The authors are grateful to two anonymous referees for their helpful comments on this paper.

REFERENCES

[1] P. T. Boggs, A. J. Kearsley, and J. W. Tolle, *A global convergence analysis of an algorithm for large-scale nonlinear optimization problems*, SIAM J. Optim., 9 (1999), pp. 833–862.

[2] P. T. Boggs, A. J. Kearsley, and J. W. Tolle, *A practical algorithm for general large scale nonlinear optimization problems*, SIAM J. Optim., 9 (1999), pp. 755–778.

[3] J. M. Borwein and A. S. Lewis, *Convex Analysis and Nonlinear Optimization: Theory and Examples*, Springer-Verlag, New York, 2000.

[4] R. H. Byrd, N. I. M. Gould, J. Nocedal, and R. A. Waltz, *An algorithm for nonlinear optimization using linear programming and equality constrained subproblems*, Math. Program., 100 (2004), pp. 27–48.

[5] R. H. Byrd, N. I. M. Gould, J. Nocedal, and R. A. Waltz, *On the convergence of successive linear-quadratic programming algorithms*, SIAM J. Optim., 16 (2005), pp. 471–489.

[6] R. H. Byrd, J. Nocedal, and R. A. Waltz, *Steering exact penalty methods for nonlinear programming*, Optim. Methods Softw., 23 (2008), pp. 197–213.

[7] C. M. Chin and R. Fletcher, *On the global convergence of an SLP-filter algorithm that takes EQP steps*, Math. Program., 96 (2003), pp. 161–177.

[8] T. F. Coleman and A. R. Conn, *Nonlinear programming via an exact penalty function: Asymptotic analysis*, Math. Program., 24 (1982), pp. 123–136.

[9] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*, SIAM, Philadelphia, 2000.

[10] A. R. Conn and T. Pietrzykowski, *A penalty function method converging directly to a constrained optimum*, SIAM J. Numer. Anal., 14 (1977), pp. 348–375.

[11] F. E. Curtis and J. Nocedal, *Flexible penalty functions for nonlinear constrained optimization*, IMA J. Numer. Anal., 28 (2008), pp. 749–769.

[12] R. Fletcher, *An algorithm for solving linearly constrained optimization problems*, Math. Program., 2 (1972), pp. 133–165.

[13] R. Fletcher, *Practical Methods of Optimization: Vol.* 2, Constrained Optimization, John Wiley and Sons, New York, 1981.

[14] R. Fletcher and S. Leyffer, *Nonlinear programming without a penalty function*, Math. Program., 91 (2002), pp. 239–269.

[15] R. Fletcher and E. Sainz de la Maza, *Nonlinear programming and nonsmooth optimization by successive linear programming*, Math. Program., 43 (1989), pp. 235–256.

[16] P. E. Gill, W. Murray, and M. A. Saunders, *User's Guide for QPOPT* 1.0*: A Fortran Package for Quadratic Programming*, Report SOL 95-4, Department of Operations Research, Stanford University, Stanford, CA, 1995.

[17] P. E. Gill, W. Murray, and M. A. Saunders, *SNOPT: An SQP algorithm for large-scale constrained optimization*, SIAM Rev., 47 (2005), pp. 99–131.

[18] N. I. M. Gould, S. Lucidi, M. Roma, and P. L. Toint, *Solving the trust-region subproblem using the Lanczos method*, SIAM J. Optim., 9 (1999), pp. 504–525.

[19] N. I. M. Gould, D. Orban, and P. L. Toint, *GALAHAD, a library of thread-safe Fortran* 90 *packages for large-scale nonlinear optimization*, ACM Trans. Math. Software, 29 (2003), pp. 353–372.

[20] N. I. M. Gould and D. P. Robinson, *A second derivative SQP method: Local convergence and practical issues*, SIAM J. Optim., 20 (2010), pp. 2049–2079.

[21] N. I. M. Gould and D. P. Robinson, *A Second Derivative SQP Method: Theoretical Issues*, Numerical analysis report 08/18, Computational Laboratory, University of Oxford, Oxford, UK, 2008.

[22] N. Maratos, *Exact Penalty Function Algorithms for Finite-Dimensional and Control Optimization Problems*, Ph.D. thesis, Department of Computing and Control, University of London, London, 1978.

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

[24] S. M. Robinson, *Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear programming algorithms*, Math. Program., 7 (1974), pp. 1–16.
[25] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, 1970.
[26] R. J. Vanderbei, *LOQO: An interior-point code for quadratic programming*, Optim. Methods Softw., 11 (1999), pp. 451–484.
[27] A. Vardi, *A trust region algorithm for equality constrained minimization: Convergence properties and implementation*, SIAM J. Numer. Anal., 22 (1985), pp. 575–591.
[28] Y.-X. Yuan, *Conditions for convergence of trust region algorithms for nonsmooth optimization*, Math. Program., 31 (1985), pp. 220–228.