A SECOND DERIVATIVE SQP METHOD WITH IMPOSED DESCENT *

Nicholas I. M. Gould† Daniel P. Robinson‡

Oxford University Computing Laboratory
Numerical Analysis Group
Technical Report 08/09
June 2008

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 Sℓ_1QP 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

---

*Research supported by the EPSRC grants EP/E053351/1 and EP/F005369/1.

†Rutherford Appleton Laboratory, Numerical Analysis Group, Chilton, Didcot, Oxfordshire, OX11 0QX, UK (N.I.M.Gould@rl.ac.uk).

‡Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, OX1 3QD, UK (daniel.robinson@comlab.ox.ac.uk).

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 component-wise). Our motivation for solving this problem is that solutions of problem (ℓ_1-σ) correspond (under certain assumptions) to solutions of the problem

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

For more details on precisely how problems (ℓ_1-σ) and (NP) are related see [10,19].

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 QP subproblems are closely related to the optimality conditions of the underlying problem and thus give the potential for fast Newton-like 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 *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 [23], a special "elastic" mode [16], and a "feasibility restoration" phase [13], 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 updated, and a new trial step is computed. More recently, filter methods 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 $\|c(x)^-\|$. Filter 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. The flexible penalty approach may be viewed as a continuum of methods with classical 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 SEQP and SIQP methods.

Sequential equality-constrained quadratic programming (SEQP) methods solve problem (NP) by solving an equality constrained QP during each iterate. The linearized equality constraints that are included in the subproblem 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 [14], Byrd et al. [4, 5], while filter based variants have been studied by Chin and Fletcher [7].

Sequential inequality-constrained quadratic programming (SIQP) methods solve problem (NP) by solving a sequence of inequality constrained quadratic subproblems. Contrary to the strategy of SEQP methods, 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 [20]. Probably the greatest disadvantage of SIQP methods is their potential cost; to solve the inequality constrained QP subprobelm, 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 accent direction. Line-search, trust-region, and filter variants of SIQP methods have been proposed. The line-search method by Gill et al. [16] avoids unbounded and non-unique QP solutions by maintaining a 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 [13] deals with infeasibility by entering a special restoration-phase to recover from bad steps.

The algorithm we propose is an SIQP method that is most closely related to the Sℓ_1QP method proposed by Fletcher [12], 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, Fletcher's method requires the *global* minimizer of this (generally) nonconvex subproblem, which is known to be a 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: a *predictor* step s_k^p is defined as a solution to a *convex* QP subproblem; a *Cauchy* step s_k^c drives convergence of the algorithm and is computed from a special uni-variate *global* minimization problem; and an (optional) *SQP* step s_k^s is computed as a local solution of a special nonconvex QP subproblem.

The rest of the paper is organized as follows. This section proceeds to introduce requisite notation and to catalog various model functions used throughout the paper. Section 2 gives a complete description of how we generate the predictor, Cauchy and SQP steps. The algorithm for computing a first-order solution to problem (ℓ_1-σ) is given in Section 3 and the global convergence of this algorithm is considered in Section 4. Finally, Section 5 gives conclusions and future work.

1.1. Notation

Most of our notation is standard. 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 $H(x)$ its (symmetric) Hessian; the matrix $H_j(x)$ 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 component-wise. 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 $\nabla^2 \mathcal{L}(x, y) = H(x) - \sum_{j=1}^{m} y_j H_j(x)$.

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 $J_k = J(x_k)$. In addition, when given a pair of values (x_k, y_k) we define $H_k = H(x_k, y_k)$. Finally, we let B_k denote a symmetric positive semi-definite approximation to H_k.

1.2. Model functions

We define the following models of $\phi(x)$ for a given estimate x_k of a solution to problem (ℓ_1-σ).

- The *linear* model of the merit function:
  $$M_k^L(s) := M_k^L(s\,; x_k) = f_k + g_k^T s + \sigma \|(c_k + J_k s)^-\|_1.$$
- The *convex* model of the merit function:
  $$M_k^B(s) := M_k^B(s\,; x_k) = f_k + g_k^T s + \tfrac{1}{2} s^T B_k s + \sigma \|(c_k + J_k s)^-\|_1.$$
- The *faithful* model of the merit function:
  $$M_k^H(s) := M_k^H(s\,; x_k) = f_k + g_k^T s + \tfrac{1}{2} s^T H_k s + \sigma \|(c_k + J_k s)^-\|_1.$$
- The *SQP* model:
  $$M_k^S(s) := M_k^S(s\,; x_k, s_k^c) = \bar{f}_k + (g_k + H_k s_k^c)^T s + \tfrac{1}{2} s^T H_k s,$$
  where $\bar{f}_k = f_k + g_k^T s_k^c + \frac{1}{2} s_k^{cT} H_k s_k^c$ and s_k^c is the Cauchy step (see Section 2.2).
- The change in the convex model:
  $$\Delta M_k^B(s) := \Delta M_k^B(s\,; x_k) = M_k^B(0\,; x_k) - M_k^B(s\,; x_k).$$
- The change in the faithful model:
  $$\Delta M_k^H(s) := \Delta M_k^H(s\,; x_k) = M_k^H(0\,; x_k) - M_k^H(s\,; x_k).$$
- The change in the SQP model:
  $$\Delta M_k^S(s) := \Delta M_k^S(s\,; x_k, s_k^c) = M_k^S(0\,; x_k, s_k^c) - M_k^S(s\,; x_k, s_k^c).$$
- For a given trust-region radius $\Delta \geq 0$, primal variable x, and penalty parameter σ, we denote the maximum decrease in the *linear* model to be
  $$\Delta_{\max}^L(\Delta) := \Delta_{\max}^L(x, \Delta) = M_k^L(0\,; x) - \min_{\|s\|_\infty \leq \Delta} M_k^L(s\,; x). \tag{1.1}$$

Useful properties of the function Δ_{max}^L are given in the next lemma. See Borwein et al. [3] and Rockafellar [21] for more details.

Lemma 1.1. *Consider the definition of $\Delta_{\max}^L$ as given by equation (1.1). Then the following properties hold:*

(i) $\Delta^L_{\max}(x, \Delta) \geq 0$ *for all* x *and all* $\Delta \geq 0$;

(ii) $\Delta^L_{\max}(x, \cdot)$ *is a non-decreasing function;*

(iii) $\Delta^L_{\max}(x, \cdot)$ *is a concave function;*

(iv) $\Delta^L_{\max}(\cdot, \Delta)$ *is continuous;*

(v) *For any fixed* $\Delta > 0$, $\Delta^L_{\max}(x, \Delta) = 0$ *if and only if* x *is a stationary point for problem* (ℓ_1-σ).

Properties (ii) and (iii) 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 1.1. *Let* x *be fixed. Then for all* $\Delta \geq 0$

$$\Delta^L_{\max}(\Delta) \geq \min(\Delta, 1)\Delta^L_{\max}(1). \tag{1.2}$$

Proof. First, if $\Delta \geq 1$ then part (ii) of Lemma 1.1 implies that

$$\Delta^L_{\max}(\Delta) \geq \Delta^L_{\max}(1). \tag{1.3}$$

Second, if $0 \leq \Delta < 1$ then part (iii) of Lemma 1.1 implies

$$\Delta^L_{\max}\big((1-\alpha)x + \alpha y\big) \geq (1-\alpha)\Delta^L_{\max}(x) + \alpha\Delta^L_{\max}(y)$$

for all $0 \leq \alpha \leq 1$. Choosing $x = 0$, $y = 1$, $\alpha = \Delta$, and using the fact that $\Delta^L_{\max}(0) = 0$ yields

$$\Delta^L_{\max}(\Delta) \geq \Delta \cdot \Delta^L_{\max}(1). \tag{1.4}$$

Equations (1.3) and (1.4) give the required result. ∎

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^p_k, a Cauchy step s^c_k, and an SQP step s^s_k. The predictor step is defined as the solution of a *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 the faithful model M^H_k 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 SQP step. The SQP step is computed using the faithful model and is, generally speaking, intended to increase the efficiency of the method. We begin by discussing the predictor step.

2.1. The predictor step s_k^p

The predictor step s_k^p plays a role in our method analogous to the role played by the direction of steepest descent in unconstrained trust-region methods. During each iterate of a classical 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 et al. [9]). A vector that is directly analogous 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 a solution to

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

where B_k is any symmetric positive semi-definite approximation to the Hessian, and $\Delta_k^p > 0$ is the predictor trust-region radius. If B_k is positive definite then problem (2.1) is *strictly* convex and the minimizer is unique. However, if B_k is only positive semi-definite, then the problem is convex and therefore has a unique *minimum*, but there may be more than one *minimizer*. We note that

$$\varDelta M_k^B(s_k^p) \ge 0, \tag{2.2}$$

since $M_k^B(s_k^p) \le M_k^B(0)$ and that problem (2.1) is a non-differentiable minimization problem. In fact, it is not differentiable at any point for which the constraint linearization is zero. In practice, we solve the equivalent smooth "elastic" problem defined as

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

where e is a vector of ones of length m.

Problem (2.3) is a smooth linearly-constrained convex quadratic program that may be solved using a number of software packages such as `LOQO` [22] and `QPOPT` [15], as well as the QP solvers `QPA`, `QPB`, and `QPC` that are part of the GALAHAD [17] library. In addition, if B_k is chosen to be diagonal, then the GALAHAD package `LSQP` may be used since problem (2.1) is then a separable convex quadratic program. Note that this includes the simplest choice of $B_k \equiv 0$.

The following estimate is Lemma 2.2 by Yuan [24] transcribed into our notation.

Lemma 2.1. *For a given x_k and σ_k the following inequality holds:*

$$\varDelta M_k^B(s_k^p) \ge \tfrac{1}{2} \varDelta_{\max}^L(\Delta_k^p) \min\left(1, \frac{\varDelta_{\max}^L(\Delta_k^p)}{\|B_k\|_2 {\Delta_k^p}^2}\right). \tag{2.4}$$

We note that the proof by Yuan requires the *global* minimum of the predictor subproblem. For a general symmetric matrix B_k 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 methods utilizing both *exact* second derivatives and the ℓ_1-penalty function. In our situation, however, the matrix B_k is positive semi-definite by construction and therefore the global minimum can be found efficiently.

We may further bound $\Delta M_k^B(s_k^p)$ by applying Corollary 1.1.

Corollary 2.1.

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

Proof. Follows directly from Corollary 1.1 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. Since we wish to drive convergence using the faithful model, we must derive a useful bound on the change in the *faithful* model. This essential bound is derived from the Cauchy point and is the topic of the next section.

2.2. The Cauchy step s_k^c

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. At 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). 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 $M_k^H(s)$ instead of the *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 $M_k^H(s_k^p) < 0$, which implies 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. This is done by finding the *global* minimizer of $M_k^H(\alpha s_k^p)$ for $0 \leq \alpha \leq 1$. We will see that the Cauchy step allows us to prove convergence by using the quantity $\Delta M_k^H(s_k^c)$ as a prediction of the decrease in the merit function.

To be more precise, the Cauchy step is defined as $s_k^c = \alpha_k s_k^p$ where α_k is the solution to

$$\underset{0\leq\alpha\leq 1}{\text{minimize}} \;\; M_k^H(\alpha s_k^p). \tag{2.6}$$

The function $M_k^H(\alpha s_k^p)$ 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 2.2. *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 $0 \le \alpha \le 1$:*

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

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

$$\|(c + \alpha J s)^-\|_1 = \|\big(\alpha(c + Js) + (1-\alpha)c\big)^-\|_1 \le \alpha\|(c + Js)^-\|_1 + (1-\alpha)\|c^-\|_1.$$

∎

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

Lemma 2.3. *Let s_k^p and s_k^c be defined as previously. Then*

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

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

$$\begin{aligned}
\Delta M_k^H(s_k^c) &\ge \Delta M_k^H(\alpha s_k^p) && (2.9)\\
&= \sigma\big(\|c_k^-\|_1 - \|(c_k + \alpha J_k s_k^p)^-\|_1\big) - \alpha g_k^T s_k^p - \frac{\alpha^2}{2}{s_k^p}^T H_k s_k^p && (2.10)\\
&= \sigma\big(\|c_k^-\|_1 - \|(c_k + \alpha J_k s_k^p)^-\|_1\big)\\
&\quad - \alpha g_k^T s_k^p - \frac{\alpha^2}{2}{s_k^p}^T B_k s_k^p + \frac{\alpha^2}{2}{s_k^p}^T (B_k - H_k) s_k^p. && (2.11)
\end{aligned}$$

Equation (2.9) follows since s_k^c minimizes $M_k^H(\alpha s_k^p)$ 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

$$\begin{aligned}
\Delta M_k^H(s_k^c) &\ge \sigma\big(\|c_k^-\|_1 - \alpha\|(c_k + J_k s_k^p)^-\|_1 - (1-\alpha)\|c_k^-\|_1\big)\\
&\quad - \alpha g_k^T s_k^p - \frac{\alpha}{2}{s_k^p}^T B_k s_k^p + \frac{\alpha^2}{2}{s_k^p}^T (B_k - H_k) s_k^p && (2.12)\\
&= \alpha\sigma\big(\|c_k^-\|_1 - \|(c_k + J_k s_k^p)^-\|_1\big)\\
&\quad - \alpha g_k^T s_k^p - \frac{\alpha}{2}{s_k^p}^T B_k s_k^p + \frac{\alpha^2}{2}{s_k^p}^T (B_k - H_k) s_k^p && (2.13)\\
&= \alpha \Delta M_k^B(s_k^p) + \frac{\alpha^2}{2}{s_k^p}^T (B_k - H_k) s_k^p. && (2.14)
\end{aligned}$$

Equation (2.12) follows from equation (2.11), Lemma 2.2 and the inequality $\alpha^2 \le \alpha$, which holds since $0 \le \alpha \le 1$. Finally, equations (2.13) and (2.14) follow from simplification of equation (2.12) and from the definition of $\Delta M_k^B(s_k^p)$.

The previous string of inequalities holds for *all* $0 \le \alpha \le 1$, 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

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

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

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

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

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

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

$$q(-b/2a) = \frac{\Delta M_k^B(s_k^p)^2}{2|s_k^{pT}(B_k - H_k)s_k^p|} \geq \frac{\Delta M_k^B(s_k^p)^2}{2\|B_k - H_k\|_2\|s_k^p\|_2^2} \geq \frac{\Delta M_k^B(s_k^p)^2}{2n\|B_k - H_k\|_2\|s_k^p\|_\infty^2}.$$

Finally, since $\|s_k^p\|_\infty \leq \Delta_k^p$, we have

$$q(-b/2a) \geq \frac{\Delta M_k^B(s_k^p)^2}{2n\|B_k - H_k\|_2 {\Delta_k^p}^2}.$$

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

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

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

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

Returning to equation (2.14), we have

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

which completes the proof. ∎

We note that in the special case $B_k = H_k$, the term $\Delta M_k^B(s_k^p)/\big(n\|B_k - H_k\|_2 {\Delta_k^p}^2\big)$ should be interpreted as infinity, and then Lemma 2.3 reduces to

$$\Delta M_k^H(s_k^c) \geq \tfrac{1}{2}\Delta M_k^B(s_k^p), \tag{2.16}$$

which trivially holds since $B_k = H_k$ and $s_k^c = s_k^p$.

We may further bound the change in the faithful model obtained from the Cauchy step by employing Corollary 2.1

Corollary 2.2. *Let s_k^p and s_k^c be defined as previously. Then*

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

where

$$\mathcal{S} = \Big\{\, 1,\ \Delta_k^p,\ \frac{\Delta_{\max}^L(1)}{\|B_k\|_2},\ \frac{\Delta_{\max}^L(1)}{\|B_k\|_2 \Delta_k^{p\,2}},\ \frac{\Delta_{\max}^L(1)}{2n\|B_k - H_k\|_2},\ \frac{\Delta_{\max}^L(1)}{2n\|B_k - H_k\|_2 \Delta_k^{p\,2}},$$
$$\frac{\Delta_{\max}^L(1)^3}{2n\|B_k - H_k\|_2 \|B_k\|_2^2 \Delta_k^{p\,2}},\ \frac{\Delta_{\max}^L(1)^3}{2n\|B_k - H_k\|_2 \|B_k\|_2^2 \Delta_k^{p\,6}},\ \Big\}.$$

Proof. The bound follows from Corollary 2.1 and Lemma 2.3. ∎

Corollary 2.2 provides the necessary bound for proving convergence of our proposed algorithm. However, the derivation of this bound relied on minimizing the faithful model along a single direction, namely the predictor step 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 SQP step serves this purpose.

2.3. The SQP step s_k^s

We begin by discussing three primary motivations for an SQP step s_k^s; we use the word "an" instead of "the" since we propose many reasonable alternatives. The first motivation of the SQP step is to improve the rate-of-convergence. The predictor step s_k^p uses a positive semi-definite approximation B_k to the true Hessian H_k. Since the Cauchy step s_k^c is computed as a minimization problem in the direction s_k^p, the ultimate quality of the Cauchy step is constrained by how well B_k approximates H_k (when restricted to the null-space of the Jacobian). The simplest and cheapest choice is $B_k = 0$, but this would result in at best first-order convergence. In general, if B_k is chosen to more closely approximate H_k then the predictor step s_k^p becomes more costly to compute, but would likely lead to better convergence. Of course as B_k is required to be positive semi-definite and since H_k is usually indefinite, this is typically not even possible. However, if a quasi-Newton approach was used to update B_k at each iterate using, for example, a quasi-Newton BFGS update, then one might expect to establish super-linear convergence.

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 $B_k = H_k$ and pick σ_k sufficiently large, then provided the trust-region radius Δ_k^p is inactive, the resulting Cauchy step $s_k^c (= s_k^p)$ is the classical SQP step for problem (NP). This means that the Cauchy step may be the "ideal" step. As previously stated, the choice $B_k = H_k$ will generally not be permissible. However, if a quasi-Newton or limited-memory quasi-Newton approach is used that maintains positive definite approximations B_k, then good convergence properties may be expected. We summarize by saying that the quality of the Cauchy step is strongly dependent on how well B_k approximates H_k (possibly when restricted to the null-space of the Jacobian matrix).

Unfortunately, even if the Cauchy step is an "excellent" direction, it may still suffer from the Maratos effect [9, 18]. The Maratos effect occurs when the linear approximation to the constraints in problem (2.1) 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 SQP step.

The third motivation for the SQP step is to improve the general performance of our method. Since the quadratic model used in computing the SQP step is allowed to use the exact Hessian H_k, it is generally a more faithful model of the merit function.

2.3.1. Explicitly-constrained SQP steps

This section discusses a class of SQP steps computed from *explicitly*-constrained subproblems. We use the terminology "explicitly-constrained" to emphasize that we include a "constraint-like" restriction explicitly in the subproblem. Useful estimates may be shown for rather general explicit constraints, but in terms of efficiency there are three natural choices that may be used. We define an *explicitly*-constrained SQP step as a solution to

$$
\text{(SQP-E)} \qquad \begin{array}{ll} \underset{s\in\mathbb{R}^n}{\text{minimize}} & \bar{f}_k + (g_k + H_k s_k^c)^T s + \frac{1}{2} s^T H_k s = M_k^S(s) \\ \text{subject to} & \chi(s) \geq 0 \\ & (g_k + H_k s_k^c)^T s \leq 0 \\ & \|s\|_\infty \leq \Delta_k^s, \end{array}
$$

where $\chi(s)$ is *any* concave vector-valued function defined for all $\|s\|_\infty \leq \Delta_k^s$, and $\bar{f}_k = f_k + g_k^T s_k^c + \frac{1}{2} s_k^{cT} H_k s_k^c$. The artificial constraint $(g_k + H_k s_k^c)^T s \leq 0$ is imposed to ensure that all local solutions are non-accent directions for the SQP model; it is clear that a local minimizer of a nonconvex QP may be an ascent direction. The following lemma gives a bound on the change in the SQP model $M_k^S(s)$ at a local solution of problem (SQP-E).

Lemma 2.4. *Assume that $\chi(0) \geq 0$. Then if s_k^s is a local solution for problem* (SQP-E)*, the following bound on the change in the quadratic model holds at s_k^s:*

$$
\Delta M_k^S(s_k^s) = M_k^S(0) - M_k^S(s_k^s) \geq \tfrac{1}{2} \max\left(-(g_k + H_k s_k^c)^T s_k^s,\ |s_k^{sT} H_k s_k^s| \right).
$$

Moreover;

$$
\text{if } \ s_k^{sT} H_k s_k^s > 0 \ \text{ then } \ (g_k + H_k s_k^c)^T s_k^s < 0.
$$

Proof. We consider two cases.
Case 1. : ${s_k^s}^T H_k s_k^s \le 0$
In this case we have

$$\Delta M_k^S(s_k^s) = -(g_k + H_k s_k^c)^T s_k^s - \tfrac{1}{2}{s_k^s}^T H_k s_k^s.$$

Since ${s_k^s}^T H_k s_k^s \le 0$ by assumption and the inequality $(g_k + H_k s_k^c)^T s_k^s \le 0$ is enforced as an explicit constraint in problem (SQP-E), it follows that

$$\begin{aligned}\Delta M_k^S(s_k^s) &\ge \max\Big(-(g_k + H_k s_k^c)^T s_k^s,\ \tfrac{1}{2}|{s_k^s}^T H_k s_k^s|\Big)\\ &\ge \tfrac{1}{2}\max\Big(-(g_k + H_k s_k^c)^T s_k^s,\ |{s_k^s}^T H_k s_k^s|\Big),\end{aligned}$$

and case 1 is complete.
Case 2. : ${s_k^s}^T H_k s_k^s > 0$
Recall that the descent constraint ensures that the inequality $(g_k + H_k s_k^c)^T s_k^s \le 0$ holds. We first show that $(g_k + H_k s_k^c)^T s_k^s < 0$. For proof by contradiction, assume that $(g_k + H_k s_k^c)^T s_k^s = 0$. Since 0 and s_k^s are both feasible for problem (SQP-E), and since $\chi(s)$ is a concave function by assumption, it is clear that αs_k^s is feasible for problem (SQP-E) for $0 \le \alpha \le 1$. Furthermore, the directional derivative of M_k^S at s_k^s in the direction $-s_k^s$ exists and is given by

$$\nabla M_k^S(s_k^s)^T(-s_k^s) = -\big(g_k + H_k(s_k^c + s_k^s)\big)^T s_k^s = -{s_k^s}^T H_k s_k^s < 0,$$

where we have used the fact that $(g_k + H_k s_k^c)^T s_k^s = 0$ and that ${s_k^s}^T H_k s_k^s > 0$. This contradicts that s_k^s is a local solution to problem (SQP-E) since $-s_k^s$ is a feasible descent direction. Therefore, $(g_k + H_k s_k^c)^T s_k^s < 0$ must be true.

Now we show the bound on $\Delta M_k^S(s_k^s)$. We define the quadratic function

$$q(\alpha) = a\alpha^2 + b\alpha + e,$$

where

$$a = \tfrac{1}{2}{s_k^s}^T H_k s_k^s > 0, \qquad b = (g_k + H_k s_k^c)^T s_k^s < 0, \qquad \text{and} \qquad e = \bar{f}_k.$$

With this definition, it follows that $q(\alpha) = M_k^S(\alpha s_k^s)$ and that $\Delta M_k^S(s_k^s) = q(0) - q(1)$. Since $q(\alpha)$ is a strictly convex quadratic function and $q(1)$ is a minimizer for q on the interval $[0, 1]$, it follows that $-b/2a \ge 1$. Using this inequality we have

$$\Delta M_k^S(s_k^s) = -a - b \ge \max(-\tfrac{1}{2}b, a) = \tfrac{1}{2}\max\Big(-(g_k + H_k s_k^c)^T s_k^s,\ |{s_k^s}^T H_k s_k^s|\Big),$$

which completes case 2. ∎

This result shows that an (SQP-E) step will never cause the SQP model to increase and, in general, it will decrease. The only situation in which the SQP model does not decrease is when the step s_k^s is a direction of zero curvature for H_k *and* the explicit descent constraint is active. It is of interest to consider a sequence of iterates $\{x_k\}$ converging to a solution of problem (NP) for which the second-order sufficient conditions are satisfied. In this case, we expect that for k sufficiently large the condition ${s_k^s}^T H_k s_k^s > 0$ would be satisfied. Then, Lemma 2.4 implies that the artificial constraint $(g_k + H_k s_k^c)^T s \leq 0$ will be inactive. This property is essential if we expect to recover fast convergence since the artificial constraint may impede the natural convergence of Newton's Method. However, when far from a solution, the artificial constraint stabilizes the method by "guiding" the iterates through areas of indefiniteness by ensuring that the SQP step does not increase the model M_k^s.

We now provide three specific concave functions $\chi(s)$ and the resultant explicitly-constrained SQP subproblem; these choices have been made with our primary goals in mind. We use the notation $c_k^c = c(x_k + s_k^c)$ and $J_k^c = J(x_k + s_k^c)$.

- The choice
$$\chi(s) = c_k + J_k s - \min(c_k, -J_k s_k^c)$$
leads to the following explicitly-constrained SQP subproblem:
$$\text{(SQP-E}_1\text{)} \qquad \begin{array}{ll} \underset{s \in \mathbb{R}^n}{\text{minimize}} & \bar{f}_k + (g_k + H_k s_k^c)^T s + \frac{1}{2} s^T H_k s \\ \text{subject to} & c_k + J_k s \geq \min\left(c_k, -J_k s_k^c\right), \\ & (g_k + H_k s_k^c)^T s \leq 0, \\ & \|s\|_\infty \leq \Delta_k^s. \end{array}$$

- The choice
$$\chi(s) = c_k^c + J_k s - \min\big(c_k^c, 0\big)$$
leads to the following explicitly-constrained SQP subproblem:
$$\text{(SQP-E}_2\text{)} \qquad \begin{array}{ll} \underset{s \in \mathbb{R}^n}{\text{minimize}} & \bar{f}_k + (g_k + H_k s_k^c)^T s + \frac{1}{2} s^T H_k s \\ \text{subject to} & c_k^c + J_k s \geq \min\big(c_k^c, 0\big), \\ & (g_k + H_k s_k^c)^T s \leq 0, \\ & \|s\|_\infty \leq \Delta_k^s. \end{array}$$

- The choice
$$\chi(s) = c_k^c + J_k^c s - \min\big(c_k^c, 0\big)$$
leads to the following explicitly-constrained SQP subproblem:
$$\text{(SQP-E}_3\text{)} \qquad \begin{array}{ll} \underset{s \in \mathbb{R}^n}{\text{minimize}} & \bar{f}_k + (g_k + H_k s_k^c)^T s + \frac{1}{2} s^T H_k s \\ \text{subject to} & c_k^c + J_k^c s \geq \min(c_k^c, 0), \\ & (g_k + H_k s_k^c)^T s \leq 0, \\ & \|s\|_\infty \leq \Delta_k^s. \end{array}$$

First, we note that the value $s = 0$ is feasible for all three subproblems. Second, we note that subproblems (SQP-E$_2$) and (SQP-E$_3$) are closely related to subproblems typically used to avoid the Maratos effect in SQP methods for *equality* constraints (see [9], for example). However, we emphasize that we are not claiming that these subproblems avoid the Maratos effect.

We now give a brief interpretation of $\chi(s)$ for each subproblem. For subproblem (SQP-E$_1$), the constraint $\chi(s)$ ensures that the linearized constraint violation at the step $s_k^c + s_k^s$ is no larger than the linearized constraint violation at s_k^c. We will soon see that this property results in a useful bound on $\Delta M_k^H(s_k^c + s_k^s)$. For subproblem (SQP-E$_2$), the constraint $\chi(s)$ may allow for further minimization of the model function M_k^S for all constraints i that "bend backwards", i.e. constraints i for which $c_i(x_k + s_k^c)$ is *feasible*. Finally, the constraint $\chi(s)$ for subproblem (SQP-E$_3$) is a "tilted" version of (SQP-E$_2$).

2.3.2. Implicitly-constrained SQP steps

This section discusses several choices for computing an SQP step from *implicitly*-constrained SQP subproblems. We use the terminology implicitly-constrained because we are attempting to satisfy a "constraint-like" function *implicitly* by penalizing the violation of that constraint. The primary advantage of these subproblems over *explicitly*-constrained SQP subproblems is their direct connection to standard techniques for avoiding the Maratos effect. Their main disadvantage is that we are no longer guaranteed that the *sum* of the Cauchy step and the SQP step will give us sufficient decrease in the faithful model M_k^H. However, since these steps are intended to avoid the Maratos effect, they would mostly be used asymptotically and this is precisely the situation in which we expect the implicitly-constrained SQP steps to give sufficient decrease in the faithful model.

We define an *implicitly*-constrained SQP step as a solution to

$$\text{(SQP-I)} \qquad \begin{array}{ll} \underset{s \in \mathbb{R}^n}{\text{minimize}} & \bar{f}_k + (g_k + H_k s_k^c)^T s + \frac{1}{2} s^T H_k s + \bar{\sigma} \|\chi(s)^-\|_1 \\ \text{subject to} & \|s\|_\infty \le \Delta_k^s, \end{array}$$

where $\chi(s)$ is *any* vector-valued function defined for all $\|s\|_\infty \le \Delta_k^s$, $\bar{f}_k = f_k + g_k^T s_k^c + \frac{1}{2} s_k^{cT} H_k s_k^c$, and $\bar{\sigma} > 0$ is a positive penalty parameter that may or may not be equal to σ.

We now provide two specific vector-valued functions $\chi(s)$ and the resultant implicitly-constrained SQP subproblem; these choices have been made with the Maratos effect in mind. Again, we use the notation $c_k^c = c(x_k + s_k^c)$ and $J_k^c = J(x_k + s_k^c)$.

- The choice
$$\chi(s) = c_k^c + J_k s$$
leads to the following implicitly-constrained SQP subproblem:
$$\text{(SQP-I}_1\text{)} \quad \begin{array}{ll} \underset{s \in \mathbb{R}^n}{\text{minimize}} & f_k + (g_k + H_k s_k^c)^T s + \frac{1}{2} s^T H_k s + \bar{\sigma} \|(c_k^c + J_k s)^-\|_1 \\ \text{subject to} & \|s\|_\infty \le \Delta_k^s. \end{array}$$

- The choice
$$\chi(s) = c_k^c + J_k^c s$$

leads to the following implicitly-constrained SQP subproblem:

$$\text{(SQP-I}_2\text{)} \quad \begin{array}{ll} \underset{s\in\mathbb{R}^n}{\text{minimize}} & f_k + (g_k + H_k s_k^c)^T s + \frac{1}{2} s^T H_k s + \bar{\sigma}\|(c_k^c + J_k^c s)^-\|_1 \\ \text{subject to} & \|s\|_\infty \le \Delta_k^s. \end{array}$$

2.4. The full step s_k

In Sections 2.1 and 2.2 we discussed how to compute the predictor step and the Cauchy step. The Cauchy step s_k^c was carefully constructed from the predictor step to ensure that it gave decrease in the faithful model M_k^H. Next, Section 2.3 discussed many options for computing an SQP step s_k^s; they were categorized as either explicitly- or implicitly-constrained SQP steps. This section analyzes the full step $s_k = s_k^c + s_k^s$.

We first examine the full step s_k when the SQP step is computed from any of the explicitly-constrained SQP subproblems. These subproblems were carefully constructed to ensure that any local minimizer results in a decrease in the model M_k^S. We now must investigate the decrease in the *faithful* model obtained from the *full* step. The next lemma gives a condition that guarantees that the decrease in the faithful model obtained from the full step is at least as great as the decrease obtained from the Cauchy point.

Lemma 2.5. *If s_k^s is computed from an* explicitly-*constrained SQP subproblem and if the following inequality holds*

$$\|(c_k + J_k s_k)^-\|_1 \equiv \big\|\big(c_k + J_k(s_k^c + s_k^s)\big)^-\big\|_1 \le \|(c_k + J_k s_k^c)^-\|_1 \tag{2.17}$$

then the following three estimates hold

$$\Delta M_k^H(s_k) \ge \Delta M_k^S(s_k^s) + \Delta M_k^H(s_k^c), \tag{2.18}$$
$$\Delta M_k^H(s_k) \ge \Delta M_k^S(s_k^s), \tag{2.19}$$
$$\Delta M_k^H(s_k) \ge \Delta M_k^H(s_k^c). \tag{2.20}$$

Proof. We begin by noticing that equations (2.19) and (2.20) follow immediately from equation (2.18) since $\Delta M_k^H(s_k^c) \ge 0$ and $\Delta M_k^S(s_k^s) \ge 0$ by Lemma 2.3 and Lemma 2.4. It remains to show (2.18).

Using the definition of ΔM_k^H and simplifying, we have

$$\begin{aligned}
\Delta M_k^H(s_k) &= M_k^H(0) - M_k^H(s_k) && (2.21)\\
&= \sigma\left(\|c_k^-\|_1 - \|(c_k + J_k s_k)^-\|_1\right) - g_k^T s_k - \tfrac{1}{2} s_k^T H_k s_k && (2.22)\\
&= \sigma\left(\|c_k^-\|_1 - \|(c_k + J_k s_k)^-\|_1\right) - s_k^{sT}(g_k + H_k s_k^c) \\
&\quad - \tfrac{1}{2} s_k^{sT} H_k s_k^s - g_k^T s_k^c - \tfrac{1}{2} s_k^{cT} H_k s_k^c && (2.23)\\
&= \Delta M_k^S(s_k^s) + \sigma\left(\|c_k^-\|_1 - \|(c_k + J_k s_k)^-\|_1\right) - g_k^T s_k^c - \tfrac{1}{2} s_k^{cT} H_k s_k^c && (2.24)\\
&\ge \Delta M_k^S(s_k^s) + \sigma\left(\|c_k^-\|_1 - \|(c_k + J_k s_k^c)^-\|_1\right) - g_k^T s_k^c - \tfrac{1}{2} s_k^{cT} H_k s_k^c && (2.25)\\
&= \Delta M_k^S(s_k^s) + \Delta M_k^H(s_k^c). && (2.26)
\end{aligned}$$

Equations (2.21) and (2.22) follow from the definitions of ΔM_k^H and M_k^H. Equation (2.23) follows from the definition of s_k and from gathering like terms, while equation (2.24) follows from the definition of ΔM_k^S. Finally, equations (2.25) and (2.26) follow from the assumption in this lemma and the definition of ΔM_k^H. ∎

The previous lemma has the following interpretation: if the linearized constraint violation at the full step is no greater than the linearized constraint violation at the Cauchy step, then the decrease in the faithful model at the full step is no less than the decrease in the faithful model obtained from the Cauchy step. The next lemma gives a condition that guarantees that inequality (2.17) is satisfied.

Lemma 2.6. *Let $s_k = s_k^c + s_k^s$. Then inequality* (2.17) *holds if*

$$J_k s_k^s \geq \min\big(0, -(c_k + J_k s_k^c)\big). \tag{2.27}$$

Proof. Inequality (2.17) holds if

$$\min(0, c_k + J_k s_k^c) \leq \min\big(0, c_k + J_k(s_k^c + s_k^s)\big). \tag{2.28}$$

We consider a generic component i. If $[c_k + J_k s_k^c]_i \geq 0$, then inequality (2.28) holds if and only if $[J_k s_k^s]_i \geq -[c_k + J_k s_k^c]_i$. On the contrary, if $[c_k + J_k s_k^c]_i < 0$, then inequality (2.28) holds if and only if $[J_k s_k^s]_i \geq 0$. These conditions are precisely those given by inequality (2.27). ∎

We now give a bound on the decrease in the model M_k^H provided the explicitly-constrained SQP step is computed from problem (SQP-E$_1$).

Lemma 2.7. *Define $s_k = s_k^c + s_k^s$, where s_k^c is computed as described in* Section 2.2 *and s_k^s is* any *feasible point for problem* (SQP-E$_1$). *Then the following bounds on the decrease of $M_k^H(s_k)$ hold:*

$$\begin{aligned}
\Delta M_k^H(s_k) &\geq \Delta M_k^S(s_k^s) + \Delta M_k^H(s_k^c),\\
\Delta M_k^H(s_k) &\geq \Delta M_k^S(s_k^s),\\
\Delta M_k^H(s_k) &\geq \Delta M_k^H(s_k^c).
\end{aligned}$$

In particular, if s_k^s is a local solution to problem (SQP-E$_1$), *then the previous estimates hold.*

Proof. Subtracting the term c_k from both sides of the general constraint for problem (SQP-E$_1$) shows that any feasible point satisfies equation (2.27) and therefore inequality (2.17) holds. Lemma 2.5 then implies the result. ∎

The previous lemma shows that the full step $s_k = s_k^c + s_k^s$ is *guaranteed* to produce a good decrease in the model H_k. More specifically, the lemma shows that the decrease in the model M_k^H obtained from the full step s_k is at least as large as the decrease obtained from the Cauchy point, which in turn was carefully constructed to guarantee convergence.

We are not guaranteed that inequality (2.17) holds for the explicitly-constrained SQP subproblems (SQP-E$_2$) and (SQP-E$_3$) and therefore an estimate like that found in Lemma 2.7 is not guaranteed to be satisfied; the same situation exists for every *implicitly*-constrained SQP subproblem. Hence, when the SQP step is computed from any of these subproblems we should monitor the change in the model M_k^H to ensure that the change is "sufficient". By sufficient, we mean that the inequality

$$\Delta M_k^H(s_k) \ge \eta \Delta M_k^H(s_k^c) \tag{2.29}$$

is satisfied for some constant $0 < \eta \le 1$ independent of k. If subproblem (SQP-E$_1$) is used to compute the SQP step, then Lemma 2.7 guarantees that inequality (2.29) holds with $\eta = 1$. For any other SQP subproblem, if inequality (2.29) is satisfied then we defined $s_k = s_k^c + s_k^s$; otherwise, we set $s_k^s = 0$ so that $s_k = s_k^c$ and inequality (2.29) holds for $\eta = 1$.

3. The Algorithm

This section presents an algorithm for minimizing problem (ℓ_1-σ); the algorithm is given by Algorithm 3.1. First, the user supplies an initial guess (x_0, y_0) of a solution to problem (ℓ_1-σ). Next, "success" parameters $0 < \eta_S \le \eta_{VS} < 1$, a maximum allowed predictor trust-region radius $\bar{\Delta}$, and expansion and contraction factors $0 < \tau_c < 1 < \tau_e$ 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 semi-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 zero matrix, the identity matrix, or perhaps a scaled diagonal matrix that attempts to model the "essential properties" of the matrix H_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^c. As given, the Hessian H_k is evaluated at (x_k, y_k). However, it is also possible to compute the matrix H_k *after* the predictor step is computed using the multiplier vector from the predictor subproblem. In either case, once the Cauchy step is computed we calculate the decrease in the model M_k^H at the Cauchy step, which is given by $\Delta M_k^H(s_k^c)$. Next, we must compute an SQP step satisfying inequality (2.29). This may be done in three ways. First, the SQP subproblem may be skipped entirely so that $s_k^s = 0$ and condition (2.29) is trivially satisfied. Second, the SQP step may be defined as the solution to the SQP problem (SQP-E$_1$), since Lemma 2.7 guarantees that the full step will satisfy condition (2.29). Third, we may solve any of the other SQP subproblems discussed in Section 2.3 and check a-posteriori whether condition (2.29) is satisfied. If the condition is satisfied we accept the step; otherwise, we set $s_k^s = 0$ so that condition (2.29) is once again satisfied. Once the SQP step is computed, we set $s_k = s_k^c + s_k^s$, evaluate $\phi(x_k + s_k)$ and $\Delta M_k^H(s_k)$, and compute 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 [12] and is determined by the ratio r_k. More precisely, if the ratio r_k of actual versus predicted decrease in the ℓ_1-merit function 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. Therefore we decrease the predictor trust-region radius with the hope that the model will accurately capture the behavior of 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. We note that the precise update used for the dual variables y_{k+1} is not important for proving convergence; we do not specify any particular update in the algorithm. However, the precise update used is essential when considering performance; the multiplier vector from the SQP subproblem is the most obvious candidate. In the case that the SQP step is not computed, then the most obvious multiplier update becomes the multiplier vector from the predictor subproblem. We also note that a least-squares multiplier update is also possible, but would require solving a specialized inequality-constrained linear program.

Finally, we have the additional responsibility of updating the SQP trust-region radius. In Algorithm 3.1 we set the SQP trust-region radius to a constant multiple of the predictor trust-region radius although the condition $\Delta_{k+1}^s \leq \tau_f \cdot \Delta_{k+1}^p$ 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. The delicate situation is when the predictor trust-region radius is converging to zero on some subsequence. Since the predictor step must also be converging to zero, it seems natural to require that the full step also converges to zero. Therefore it seems intuitive to require that if $\{x_{k_j}\}_{j\geq 0}$ is any subsequence such that $\lim_{j\to\infty} \|s_{kj}^p\|_\infty = 0$, then the sequence

$$\{\Delta_{kj}^s / \|s_{kj}^p\|_\infty\}_{j\geq 0} \text{ remain bounded.} \tag{3.1}$$

A simple way to ensure this condition is by defining the SQP trust-region radius as $\Delta_{k+1}^s \leftarrow \tau_f \cdot \|s_k^p\|_\infty$, i.e. set the SQP 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 $\Delta_{k+1}^s \leftarrow \tau_f \cdot \Delta_{k+1}^p$, i.e. set the SQP trust-region radius to be a constant multiple of the size of predictor *radius*. Asymptotically they are equivalent since Corollary 4.1 shows that if we are not converging to a solution, then $\|s_k^p\|_\infty = \Delta_k^p$ for Δ_k^p sufficiently small. However, the update $\Delta_{k+1}^s \leftarrow \tau_f \cdot \Delta_{k+1}^p$ allows for larger value of Δ_k^s globally and has been observed to perform better in our initial tests.

Algorithm 3.1. Minimizing the ℓ_1-penalty function
Input: (x_0, y_0)
Set parameters $0 < \eta_S \le \eta_{VS} < 1$, and $\bar{\Delta} > 0$.
Set expansion and contraction factors $0 < \tau_c < 1 < \tau_e$.
$k \leftarrow 0$
do
  Evaluate f_k, g_k, c_k, J_k, H_k, and then compute ϕ_k.
  Define B_k to be a positive semi-definite approximation to H_k.
  Solve problem (2.1) for s_k^p.
  Solve problem (2.6) for s_k^c and compute $\Delta M_k^H(s_k^c)$.
  Compute an SQP-correction step s_k^s satisfying (2.29).
  $s_k \leftarrow s_k^c + s_k^s$
  Evaluate $\phi(x_k + s_k)$ and $\Delta M_k^H(s_k)$.
  Compute $r_k = \big(\phi_k - \phi(x_k + s_k)\big)/\Delta M_k^H(s_k)$.
  if $r_k \ge \eta_{VS}$ [very successful]
    $\Delta_{k+1}^p \leftarrow \min(\tau_e \cdot \Delta_k^p, \bar{\Delta})$ [increase predictor radius]
  else if $r_k \ge \eta_S$ [successful]
    $\Delta_{k+1}^p \leftarrow \Delta_k^p$ [keep predictor radius]
  else [unsuccessful]
    $\Delta_{k+1}^p \leftarrow \tau_c \cdot \Delta_k^p$ [decrease predictor radius]
  end
  if $r_k > 0$ [accept step]
    $x_{k+1} \leftarrow x_k + s_k$
    $y_{k+1} \leftarrow$ whatever you want
  else [reject step]
    $x_{k+1} \leftarrow x_k$
    $y_{k+1} \leftarrow y_k$
  end
  $\Delta_{k+1}^s \leftarrow \tau_f \cdot \Delta_{k+1}^p$ [update SQP radius]
  $k \leftarrow k + 1$
end do

4. Convergence

This section shows that Algorithm 3.1 is globally convergent. Our man 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

$$\left\| \begin{pmatrix} g(x)^T \\ J(x) \end{pmatrix} \right\|_2 \le M \quad \text{for all } x \in \mathcal{B}, \tag{4.1}$$

where $\mathcal{B}$ is a closed and bounded 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

$$|h(f_1,c_1) - h(f_2,c_2)| \leq L \left\| \begin{pmatrix} f_1 - f_2 \\ c_1 - c_2 \end{pmatrix} \right\|_2 \tag{4.2}$$

for all (f_1, c_1) and $(f_2, c_2) \in \big(f(\mathcal{B}), c(\mathcal{B})\big)$ [21, 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 [24, Lemma 3.2] except for the use of the infinity norm.

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

$$\|s_k^p\|_\infty \geq \tfrac{1}{2}\Delta^L_{\max}(x_k, 1) \min\left(\frac{1}{LM}, \frac{1}{n(1+\bar{\Delta})\|B_k\|}\right). \tag{4.3}$$

Corollary 4.1. *Suppose that $\{x_k\}_{k\geq 0} \subset \mathcal{B}$ and that K is a subsequence of the integers such that the following hold:*

(i) *there exists a number δ such that $\Delta^L_{\max}(x_k, 1) \geq \delta > 0$ for all $k \in K$;*

(ii) *there exists a positive constant b_B such that $\|B_k\| \leq b_B$ for all $k \in K$;*

(iii) $\lim_{k\in K} \Delta_k^p = 0$.

Then

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

Proof. Equation (4.3), (i), and (ii) imply that $\|s_k^p\|_\infty$ is *strictly* bounded away from zero for all $k \in K$. However, this contradicts assumption (iii) for $k \in K$ sufficiently large since $\|s_k^p\|_\infty \leq \Delta_k^p$. Therefore, Lemma 4.1 implies that $\|s_k^p\|_\infty = \Delta_k^p$ for all $k \in K$ sufficiently large. ∎

We may now state our main result. The organization of the proof is based on Theorem 14.5.1 by Fletcher [12] and the proof of case 1 is nearly identical to that given by Fletcher.

Theorem 4.1. *Let f and c be twice continuously differentiable functions, and let $\{x_k\}$, $\{H_k\}$, $\{B_k\}$, $\{\Delta_k^p\}$, and $\{\Delta_k^s\}$, be sequences generated by* Algorithm 3.1. *Assume that the following conditions hold:*

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

Then, either x_K is a first-order point for problem (ℓ_1-σ) for some $K \geq 0$, 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 $K \geq 0$ 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 $\{\Delta_k^p\}$ that converges to zero.
Examination of the algorithm shows that this implies the existence of a subsequence S of the integers such that

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

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

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

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

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 sub-differential of $\|(\cdot)^-\|_1$ at the point c_* (see [12, 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

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

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 $M_k^H(\varepsilon v)$. The same argument using B_k in place of H_k gives the estimate

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

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

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

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

Equation (4.14) then implies the equation

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

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

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

Inequalities (4.17) and (4.18) follow from assumption (2.29) and since the Cauchy step maximizes $\Delta M_k^H(s)$ 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, equation (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 $\varepsilon_k = \|s_k^p\|_\infty$. It follows that $\|\varepsilon_k s\|_\infty = \|s_k^p\|_\infty$ and, therefore, the vector $\varepsilon_k s$ is feasible for the kth predictor subproblem. It now follows that for all $k \in S$ sufficiently large we have

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

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 equation (4.22) and since s_k^p is a global minimizer for the kth predictor subproblem. Equation (4.24) follows from equation (4.15), while inequality (4.25) follows from [12, corollary to Lemma 14.5.1]. Finally, equations (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

$$\left|\frac{o(\|s_k\|_\infty)}{\Delta M_k^H(s_k)}\right| \le \left|\frac{o(\|s_k\|_\infty)}{\eta\rho\|s_k^p\|_\infty + o(\|s_k^p\|_\infty)}\right| \tag{4.28}$$

$$\le \frac{z_k\|s_k\|_\infty}{\frac{1}{2}\eta\rho\|s_k^p\|_\infty} \tag{4.29}$$

$$\le \frac{2z_k(\|s_k^c\|_\infty + \|s_k^s\|_\infty)}{\eta\rho\|s_k^p\|_\infty} \tag{4.30}$$

$$\le \frac{2z_k(\|s_k^p\|_\infty + \|s_k^s\|_\infty)}{\eta\rho\|s_k^p\|_\infty} \tag{4.31}$$

$$= \frac{2z_k}{\eta\rho}\left(1 + \frac{\|s_k^s\|_\infty}{\|s_k^p\|_\infty}\right) \tag{4.32}$$

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 definition of "little-oh". Inequality (4.30) follows from the triangle-inequality and inequalities (4.31) and (4.32) follow from how the Cauchy point s_k^c is computed and simplification.

We now show that the assumptions in Corollary 4.1 are satisfied. Since x_* is not first-order optimal by assumption, it follows that $\Delta^L_{\max}(x_*, 1) \ne 0$. By continuity it follows that $\Delta^L_{\max}(x_k, 1)$ is strictly bounded away from zero for $k \in S$ sufficiently large; this is assumption (i) of the Corollary. Assumptions (ii) and (iii) follow directly from the assumptions in this theorem and the case we are considering.

Equation (4.32), Corollary 4.1, and the SQP trust-region radius update used in Algorithm 3.1 imply

$$\left|\frac{o(\|s_k\|_\infty)}{\Delta M_k^H(s_k)}\right| \le \frac{2z_k}{\eta\rho}\left(1 + \frac{\|s_k^s\|_\infty}{\Delta_k^p}\right) \le \frac{2(1+\tau_f)z_k}{\eta\rho}. \tag{4.33}$$

Finally, inequalities (4.16) and (4.33) show that

$$r_k = 1 + o(1) \quad \text{for} \quad k \in S. \tag{4.34}$$

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

Case 2 : there does not exists a subsequence of $\{\Delta_k^p\}$ 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

$$\lim_{k \in S} x_k = x_*, \tag{4.35}$$

$$\Delta_k^p \ge \delta > 0, \quad \text{for all} \quad k \tag{4.36}$$

$$r_k \ge \eta_S \quad \text{for all} \quad k \in S. \tag{4.37}$$

Equation (2.29) and the fact that each $k \in S$ is a successful iterate imply

$$\phi_k - \phi(x_k + s_k) \geq \eta_S \Delta M_k^H(s_k) \geq \eta\eta_S \Delta M_k^H(s_k^c). \tag{4.38}$$

Corollary 2.2, equation (4.36), the bounds b_B and b_H on B_k and H_k, and the bound $\Delta_k^p \leq \bar{\Delta}$ imply

$$\phi_k - \phi(x_k + s_k) \geq \frac{\eta\eta_S}{4} \Delta_{\max}^L(x_k, 1) \min(\mathcal{S}) \tag{4.39}$$

where

$$\mathcal{S} = \Big\{ 1,\ \delta,\ \frac{\Delta_{\max}^L(x_k,1,\sigma)}{b_B}, \frac{\Delta_{\max}^L(x_k,1,\sigma)}{b_B\bar{\Delta}^2}, \frac{\Delta_{\max}^L(x_k,1,\sigma)}{2n(b_B+b_H)}, \frac{\Delta_{\max}^L(x_k,1,\sigma)}{2n(b_B+b_H)\bar{\Delta}^2}, \frac{\Delta_{\max}^L(x_k,1,\sigma)^3}{2n(b_B+b_H)b_B^2\bar{\Delta}^2}, \frac{\Delta_{\max}^L(x_k,1,\sigma)^3}{2n(b_B+b_H)b_B^2\bar{\Delta}^6}, \Big\}.$$

Summing over all $k \in S$ yields

$$\sum_{k\in S} \phi_k - \phi(x_k + s_k) \geq \sum_{k\in S} \frac{\eta\eta_S}{4} \Delta_{\max}^L(x_k, 1) \min(\mathcal{S}). \tag{4.40}$$

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

$$\sum_{k\in S} \phi_k - \phi(x_k + s_k) = \sum_{k\in S} \phi_k - \phi(x_{k+1}) \leq \phi(x_0) - \phi(x_*). \tag{4.41}$$

Combining the two previous inequalities gives

$$\phi(x_0) - \phi(x_*) \geq \sum_{k\in S} \frac{\eta\eta_S}{4} \Delta_{\max}^L(x_k, 1) \min(\mathcal{S}), \tag{4.42}$$

which implies

$$\lim_{k\in S} \Delta_{\max}^L(x_k, 1) = 0 \tag{4.43}$$

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

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

As stated previously, the proof of case 1 is nearly identical to that given by Fletcher. However, Fletcher's proof for case 2 does not carry over to our setting. Examination of his proof indicates that the break down occurs when Fletcher essentially requires the global minimizer of M_k^H over the trust-region defined by radius Δ_k^p; we only compute the global minimizer of M_k^H in the *single* direction s_k^p.

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 further understanding of these methods by showing how a relatively simple idea may be used to avoid the pitfalls typically associated with second-derivative SQP algorithms.

We presented an ℓ_1-SQP method that is based on the work by Fletcher [12]. In Section 2, we described how to compute trial steps as a combination of a Cauchy step and an SQP step. Two classes of SQP steps were considered. Section 2.3.1 discussed the class of explicitly-constrained SQP steps that were designed to enhance efficiency, while Section 2.3.2 considered the class of implicitly-constrained SQP steps that were designed to avoid the Maratos effect. We feel that our method provides a natural framework for avoiding the Maratos effect that is less ad-hoc than traditional means. 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.

Yuan [24] shows that Fletcher's method is globally convergent under weaker assumptions on the matrices H_k. Similar conclusions are true for our method and will be covered in a separate paper. In addition, we plan to discuss 1) mechanisms for updating the penalty parameter; 2) local convergence issues; and 3) strategies for defining convex approximations to the Hessian of the Lagrangian in the large-scale case. We note that Byrd, Nocedal, and Waltz [6] and Byrd et al. [4] have already published clever techniques for updating the penalty parameter, and this will likely influence our developments. Finally, we aim to give details of numerical experiments with our evolving GALAHAD package TRIMSQP.

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] ———, *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] ———, *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*, Opt. Methods & Software., 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 Ph. L. Toint, *Trust-Region Methods*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 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 Journal of Numerical Analysis Advance Access, (2008).

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

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

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

[15] 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.

[16] ——, *SNOPT: An SQP algorithm for large-scale constrained optimization*, SIAM Rev., 47 (2005), pp. 99–131.

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

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

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

[20] S. M. Robinson, *A quadratically-convergent algorithm for general nonlinear programming problems*, Math. Program., 3 (1972), pp. 145–156.

[21] R. T. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, New Jersey, 1970.

[22] R. J. Vanderbei, *LOQO: an interior-point code for quadratic programming*, Optimization Methods and Software, 11 (1999), pp. 451–484.

[23] A. Vardi, *A trust region algorithm for equality constrained minimization: convergence properties and implementation*, SIAM J. Numer. Anal., 22 (1985), pp. 575–591.

[24] Y. Yuan, *Conditions for convergence of trust region algorithms for nonsmooth optimization*, Math. Programming, 31 (1985), pp. 220–228.