Science & Technology Facilities Council

Technical Report

A second-derivative trust-region SQP method with a “trust region free” predictor step

N I M Gould D P Robinson

November 2, 2009

RAL-TR-2009-024

Enquires about copyright, reproduction and requests for additional copies of this report should be addressed to:

Library and Information Services
SFTC Rutherford Appleton Laboratory
Harwell Science and Innovation Campus
Didcot
OX11 0QX
UK
Tel: +44 (0)1235 445384
Fax: +44(0)1235 446403
Email: library@rl.ac.uk

The STFC ePublication archive (epubs), recording the scientific output of the Chilbolton, Daresbury, and Rutherford Appleton Laboratories is available online at: http://epubs.cclrc.ac.uk/

ISSN 1358-6254

RAL-TR-2009-024

A second-derivative trust-region SQP method with a "trust-region-free" predictor step[1]

Nicholas I. M. Gould[2,3] and Daniel P. Robinson[3]

ABSTRACT

In (NAR 08/18 and 08/21, Oxford University Computing Laboratory, 2008) we introduced a second-derivative SQP method (`S2QP`) for solving nonlinear nonconvex optimization problems. We proved that the method is globally convergent and locally superlinearly convergent under standard assumptions. A critical component of the algorithm is the so-called predictor step, which is computed from a strictly convex quadratic program with a trust-region constraint. This step is essential for proving global convergence, but its propensity to identify the optimal active set is paramount for recovering fast local convergence. Thus the global *and* local efficiency of the method is intimately coupled with the quality of the predictor step.

In this paper we study the effects of removing the trust-region constraint from the computation of the predictor step; this is reasonable since the resulting problem is still strictly convex and thus well-defined. Although this is an interesting theoretical question, our motivation is based on practicality. Our preliminary numerical experience with `S2QP` indicates that the trust-region constraint occasionally degrades the quality of the predictor step and diminishes its ability to correctly identify the optimal active set. Moreover, removal of the trust-region constraint allows for re-use of the predictor step over a sequence of failed iterations thus reducing computation. We show that the modified algorithm remains globally convergent and preserves local superlinear convergence provided a nonmonotone strategy is incorporated.

[1] This work was supported by the EPSRC grants EP/E053351/1 and EP/F005369/1.

[2] Computational Science and Engineering Department, Rutherford Appleton Laboratory,
Chilton, Oxfordshire, OX11 0QX, England, EU. Email: nick.gould@stfc.ac.uk .
Current reports available from "http://www.numerical.rl.ac.uk/reports/reports.shtml".

[3] Mathematical Institute 24–29 St Giles', Oxford, OX1 3LB, England, EU.
Email: robinson@maths.ox.ac.uk

Computational Science and Engineering Department
Atlas Centre
Rutherford Appleton Laboratory
Oxfordshire OX11 0QX

November 2, 2009

1 Introduction

In [8, 9], we presented S2QP—a sequential inequality/equality constrained quadratic programming algorithm (an SIQP/SEQP "hybrid") 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). The motivation for solving this problem is that solutions of problem (ℓ_1-σ) correspond (under certain assumptions) to solutions of the nonlinear programming problem

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

see [5, 14] for more details on exactly how these problems are related. The kth iteration of the method involves the computation of a trial step, which is itself computed from three separate steps. The first step—referred to as the predictor step [9, Section 2.1]—is defined as the unique minimizer of a strictly convex quadratic approximation to ϕ subject to a trust-region constraint. Global convergence of the method depends on the predictor step since the second step—the Cauchy step [9, Section 2.2]—drives convergence and is computed from the predictor step. Moreover, the predictor step affects global efficiency since a "better" predictor step will generally result in a better Cauchy step. In the neighborhood of a solution, however, this distinction becomes less important provided we compute a third (optional) accelerator step. If an accelerator step is computed from any of the subproblems considered in [9, Section 2.3], then the iterates converge superlinearly (under standard assumptions) [8, Theorem 4.7, Theorem 4.9]. In the special case that the accelerator step is computed from a so-called SEQP subproblem [9, Section 2.3.2], the proof requires that the predictor step correctly identifies the set of constraints active at the local solution. Thus the predictor step also plays a role in guaranteeing fast local convergence. It is also clear that the quality of the predictor step is important when the (optional) accelerator step is not computed, since then the efficiency of the method globally *and* locally is entirely controlled by the predictor step. To summarize, the quality of the predictor step is extremely important *both* globally and locally.

The justification provided by the previous paragraph combined with our preliminary numerical experience with S2QP (an implementation of the algorithm outlined in [8, 9]), suggests that improvements in how we define the predictor step will lead to an improved algorithm; this is the primary purpose of this paper. To be precise, we study the effect of removing the trust-region constraint from the computation of the predictor step. This is reasonable since the problem is strictly convex and, therefore, well defined. This is an interesting theoretical question, but equally it is important from a practical point-of-view since the trust-region constraint may degrade the step quality and/or interfere with optimal active set identification. Moreover, removal of the trust-region constraint allows for re-use of the predictor step over a sequence of failed iterations thus reducing computation. Although this may be considered a "minor" change, new proofs of global convergence are needed. We must also mention that our algorithm has commonalities with the work by Morales, Nocedal, and Wu [12]; a detailed comparison is given at the end of Section 2.

In Section 2 we formally state and describe the modified SQP method, while in Section 3 prove that it is both globally and locally superlinearly convergent. We conclude by giving final comments in Section 4. Before proceeding, however, we list essential notation.

1.1 Notation

We let $g(x)$ be 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 $\nabla c_i(x)^T$. 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 $H(x,y) \overset{\text{def}}{=} \nabla^2_{xx}\mathcal{L}(x,y) = H(x) - \sum_{j=1}^m y_j H_j(x)$.

For a general vector v, the notation $[v]^- = \min(0,v)$ is used, where the minimum is understood to be component-wise; given two general vectors v and w, the notation $v \cdot w$ represents the vector whose ith component is $v_i w_i$. Given a general indexing set $\mathcal{S}$, a vector v, and a matrix V, we let $v_{\mathcal{S}}$ and $V_{\mathcal{S}}$ denote the rows of v and V that correspond to the indexing set $\mathcal{S}$. If V happens to be a function of x, we often write $V_{\mathcal{S}}(x)$ instead of $[V(x)]_{\mathcal{S}}$.

Finally, we often consider problem functions evaluated at a specific point x_k. To simplify notation we define $f_k = f(x_k)$, $c_k = c(x_k)$, $g_k = g(x_k)$ and $J_k = J(x_k)$.

2 Algorithm

In this section we state and describe our nonmonotone algorithm for minimizing problem (ℓ_1-σ). This method is a modification of that proposed in [8, 9] that uses the new predictor step subproblem; the algorithm is given as Algorithm 2.1.

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 Δ_u, predictor trust-region radius "reset" value Δ_{RESET}, expansion and contraction factors $0 < \eta_c < 1 < \eta_e$, sufficient model decrease and approximate Cauchy point tolerances $0 < \eta \le \eta_{\text{ACP}} < 1$, accelerator trust-region radius factor τ_f, and the maximum number of nonmonotone steps allowed *max_fails* are defined. With parameters set, the main iteration loop begins. First, the problem functions are evaluated at the current point (x_k, y_k). Next, we approximate $H(x_k, y_k)$ with a symmetric positive-definite matrix B_k [8, Section 2] and form the predictor step subproblem

$$\underset{s\in\mathbb{R}^n}{\text{minimize}} \;\; f_k + g_k^T s + \tfrac{1}{2} s^T B_k s + \sigma \| [c_k + J_k s]^- \|_1 \overset{\text{def}}{=} M_k^{\text{B}}(s). \tag{2.1}$$

By introducing elastic variables [7], we may solve the equivalent strictly convex quadratic programming problem

$$\underset{s\in\mathbb{R}^n}{\text{minimize}}\;\; f_k + g_k^T s + \tfrac{1}{2}s^T B_k s + \sigma e^T v \;\;\text{subject to}\;\; c_k + J_k s + v \geq 0,\; v \geq 0 \tag{2.2}$$

for the predictor step s_k^{P}; we let y_k^{P} denote the multiplier vector associated with the general constraint $c_k + J_k s + v \geq 0$. Now, y_k^{F} is defined as any first-order multiplier estimate such that $y_k^{\text{F}} - y^* = O(\|x_k - x^*\|_2)$ and $[y_k^{\text{F}}]_{\mathcal{I}} = 0$, where (x^*, y^*) is a local solution to problem (NP) and $\mathcal{I} \overset{\text{def}}{=} \{i : c_i(x^*) > 0\}$; the vector y_k^{F} is only relevant when we consider local convergence, and then [8, Lemma 4.8] allows the choice $y_k^{\text{F}} \equiv y_k^{\text{P}}$. We then define H_k to be any symmetric approximation to $H(x_k, y_k^{\text{F}})$, but for the local convergence results given in Section 3 we choose $H_k \equiv H(x_k, y_k^{\text{F}})$. The Cauchy step is now defined as $s_k^{\text{CP}} = \alpha_k s_k^{\text{P}}$, where α_k is the solution to

$$\underset{0\leq\alpha\leq\alpha_u}{\text{minimize}}\;\; M_k^{\text{H}}(\alpha s_k^{\text{P}}) \quad \text{for} \quad \alpha_u \overset{\text{def}}{=} \frac{\Delta_k^{\text{P}}}{\|s_k^{\text{P}}\|_\infty}, \tag{2.3}$$

where

$$M_k^{\text{H}}(s) \overset{\text{def}}{=} f_k + g_k^T s + \tfrac{1}{2}s^T H_k s + \sigma\|[c_k + J_k s]^-\|_1 \tag{2.4}$$

is the *faithful* model of ϕ. We emphasize that the predictor step computation (2.2), in contrast to [8, 9], does *not* involve any trust-region constraint. The predictor trust-region radius Δ_k^{P} is *only* used during the Cauchy step computation (2.3) for constraining the length of the step. To further contrast [8, 9], the Cauchy step s_k^{CP} may now have a *larger* infinity-norm than the predictor step, but will always satisfy

$$\|s_k^{\text{CP}}\|_\infty \leq \Delta_k^{\text{P}}. \tag{2.5}$$

The next step is to compute the change in the faithful model at the Cauchy step, which is given by $\Delta M_k^{\text{H}}(s_k^{\text{CP}})$ where $\Delta M_k^{\text{H}}(s) \overset{\text{def}}{=} M_k^{\text{H}}(0) - M_k^{\text{H}}(s)$. We then have the option of computing an accelerator step s_k^{A} as the solution of any of the subproblems discussed in [9, Section 2.3]. If subproblem (SEQP) (see page 17) is used to compute the accelerator step, then we define the trial step as

$$s_k = \begin{cases} s_k^{\text{P}} + s_k^{\text{A}} & \text{if } \Delta M_k^{\text{H}}(s_k^{\text{P}} + s_k^{\text{A}}) \geq \eta \Delta M_k^{\text{H}}(s_k^{\text{CP}}) \\ s_k^{\text{CP}} & \text{otherwise} \end{cases} \tag{2.6}$$

for some constant $0 < \eta \leq 1$ (independent of k); otherwise, if subproblem (SIQP-E) (see page 17) is used to compute the accelerator step, we define

$$s_k = s_k^{\text{CP}} + s_k^{\text{A}}. \tag{2.7}$$

Note that this ensures in all cases that the full step s_k satisfies

$$\Delta M_k^{\text{H}}(s_k) \geq \eta \Delta M_k^{\text{H}}(s_k^{\text{CP}}) \geq 0. \tag{2.8}$$

See [9, Section 2.3] for more details. We then evaluate $\phi(x_k + s_k)$ and $\mathit{\Delta} M_k^{\mathrm{H}}(s_k)$.

The strategy for updating the trust-region radii and for accepting or rejecting candidate steps is similar to traditional methods and based on the ratio r_k of actual versus predicted decrease in ϕ. Differences are that we must account for nonmonotone steps, ensure that the predictor trust-region radius is bigger than a pre-defined constant following a successful iteration (to ensure fast asymptotic convergence), and update the accelerator trust-region radius. 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 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 keep the current predictor trust-region radius with the possibility of increasing it only to satisfy $\Delta_{k+1}^{\mathrm{P}} \geq \Delta_{\mathrm{RESET}}$. Otherwise, the ratio indicates that there is poor agreement between the model M_k^{H} and the merit function. It is precisely this case that differentiates the nonmonotone Algorithm 2.1 from its monotone variant. In fact, if every iteration is successful, then the two algorithms are identical. However, if a failure occurs then Algorithm 2.1 still accepts the step (provided $\mathit{max\_fails} > 0$) with the hope that the next iterate will make progress; we say that a "nonmonotone phase" has been entered. If we enter a nonmonotone phase, the ratio r_k of actual to predicted decrease in the merit function is computed based on the trial point $x_k + s_k$ and the *best-known* point, i.e., the solution estimate directly *before* the nonmonotone phase was entered. If the number of consecutive failures reaches the maximum number allowed (as denoted by the parameter *max_fails*), then we check whether the first Cauchy step computed during the current nonmonotone phase makes progress; this allows us to prove global convergence in Section 3. If it does not make sufficient progress, the algorithm reverts to the best-known point, reduces the predictor trust-region radius, and proceeds on. In less precise terms, the algorithm has "gone back in time" and proceeds as if we were using the monotone variant until the next failure occurs. In all cases we define the accelerator trust-region radius to be a constant multiple of the predictor trust-region radius, although the condition $\Delta_{k+1}^{\mathrm{A}} \leq \tau_f \cdot \Delta_{k+1}^{\mathrm{P}}$ for some constant τ_f is also sufficient. For more details on nonmonotone algorithms (sometimes known as nonmonotone "watchdog" techniques), see [4, Chapters 10.1 and 11.3].

Algorithm 2.1. Nonmonotone algorithm.

```
Input: $(x_0, y_0)$
Set parameters $0 < \eta_S \leq \eta_{VS} < 1$, $0 < \Delta_{\mathrm{RESET}} \leq \Delta_u$, $0 < \eta \leq \eta_{\mathrm{ACP}} < 1$, $\tau_f \geq 1$, and
    $0 \leq \mathit{max\_fails} \in \mathbb{N}$.
Set expansion and contraction factors $0 < \eta_c < 1 < \eta_e$, fail counter $\mathit{fails} \leftarrow 0$, and counter
    $k \leftarrow 0$.
do
    Evaluate $f_k$, $g_k$, $c_k$, $J_k$ and then compute $\phi_k$.
    Define $B_k$ to be a symmetric positive-definite approximation to $H(x_k, y_k)$.
```

```
Solve problem (2.2) for predictor step and multipliers $(s_k^{\mathrm{P}}, y_k^{\mathrm{P}})$.
Define $y_k^{\mathrm{F}}$ to be any multiplier estimate for which $y_k^{\mathrm{F}} - y^* = O(\|x_k - x^*\|_2)$ and $[y_k^{\mathrm{F}}]_{\mathcal{I}} = 0$.
Define $H_k$ to be a symmetric approximation to $H(x_k, y_k^{\mathrm{F}})$.
Solve problem (2.3) for $s_k^{\mathrm{CP}}$ and compute $\varDelta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}})$.
Optionally, compute an accelerator step and multipliers $(s_k^{\mathrm{A}}, y_k^{\mathrm{A}})$.
Define the full step $s_k$ from (2.6)/(2.7) that satisfies (2.8) and then evaluate $\phi(x_k + s_k)$ and
    $\varDelta M_k^{\mathrm{H}}(s_k)$.
if $\mathit{fails} = 0$ then
    $r_k \leftarrow \big(\phi(x_k) - \phi(x_k + s_k)\big)/\varDelta M_k^{\mathrm{H}}(s_k)$            [standard definition]
    If $s_k = s_k^{\mathrm{CP}}$, set $\mathit{Cauchy\_tried} = \mathbf{true}$; otherwise set $\mathit{Cauchy\_tried} = \mathbf{false}$.
else
    $r_k \leftarrow \big(\phi_{\mathrm{R}} - \phi(x_k + s_k)\big)/\varDelta_{\mathrm{R}}^{\mathrm{H}}$            [change in $\phi$ based on point $x_{\mathrm{R}}$]
end if
if $r_k \ge \eta_{VS}$ then                                                          [successful]
    $x_{k+1} \leftarrow x_k + s_k,\ y_{k+1} \leftarrow y_k^{\mathrm{A}}$  ($y_{k+1} \leftarrow y_k^{\mathrm{F}}$ if accelerator step not computed)
    $\Delta_{k+1}^{\mathrm{P}} \leftarrow \min\big(\max(\eta_e \cdot \Delta_k^{\mathrm{P}}, \Delta_{\mathrm{RESET}}), \Delta_u\big)$
    $\mathit{fails} \leftarrow 0$
else if $r_k \ge \eta_S$ then                                                        [successful]
    $x_{k+1} \leftarrow x_k + s_k,\ y_{k+1} \leftarrow y_k^{\mathrm{A}}$  ($y_{k+1} \leftarrow y_k^{\mathrm{F}}$ if accelerator step not computed)
    $\Delta_{k+1}^{\mathrm{P}} \leftarrow \max(\Delta_k^{\mathrm{P}}, \Delta_{\mathrm{RESET}})$
    $\mathit{fails} \leftarrow 0$
else
    $\mathit{fails} \leftarrow \mathit{fails} + 1$
    if $\mathit{fails} = 1$ then                                                     [save current point]
        $x_{\mathrm{R}} = x_k,\ y_{\mathrm{R}} = y_k,\ \phi_{\mathrm{R}} = \phi_k,\ s_{\mathrm{R}}^{\mathrm{CP}} = s_k^{\mathrm{CP}},\ y_{\mathrm{R}}^{\mathrm{F}} = y_k^{\mathrm{F}}$
        $\varDelta_{\mathrm{R}}^{\mathrm{H}} = \varDelta M_k^{\mathrm{H}}(s_k),\ \varDelta_{\mathrm{R}}^{\mathrm{HCP}} = \varDelta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}),\ \Delta_{\mathrm{R}}^{\mathrm{P}} = \Delta_k^{\mathrm{P}}$
    end if
    if $\mathit{fails} \le \mathit{max\_fails}$ then                                 [unsuccessful]
        $x_{k+1} \leftarrow x_k + s_k,\ \Delta_{k+1}^{\mathrm{P}} \leftarrow \Delta_k^{\mathrm{P}}$
    else
        $\mathit{fails} \leftarrow 0$
        if $\mathit{Cauchy\_tried}$                                                  [revert to saved point]
            $x_{k+1} \leftarrow x_{\mathrm{R}},\ y_{k+1} \leftarrow y_{\mathrm{R}},\ \Delta_{k+1}^{\mathrm{P}} \leftarrow \eta_c \Delta_{\mathrm{R}}^{\mathrm{P}}$
        else
            Evaluate $\phi(x_{\mathrm{R}} + s_{\mathrm{R}}^{\mathrm{CP}})$
            if $\big(\phi_{\mathrm{R}} - \phi(x_{\mathrm{R}} + s_{\mathrm{R}}^{\mathrm{CP}})\big)/\varDelta_{\mathrm{R}}^{\mathrm{HCP}} \ge \eta_S$ then     [successful Cauchy]
                $x_{k+1} \leftarrow x_{\mathrm{R}} + s_{\mathrm{R}}^{\mathrm{CP}},\ y_{k+1} \leftarrow y_{\mathrm{R}}^{\mathrm{F}},\ \Delta_{k+1}^{\mathrm{P}} \leftarrow \max(\Delta_k^{\mathrm{P}}, \Delta_{\mathrm{RESET}})$
            else                                                                     [revert to saved point]
                $x_{k+1} \leftarrow x_{\mathrm{R}},\ y_{k+1} \leftarrow y_{\mathrm{R}},\ \Delta_{k+1}^{\mathrm{P}} \leftarrow \eta_c \Delta_{\mathrm{R}}^{\mathrm{P}}$
            end if
        else if
        end if
    end if
    $\Delta^{\text{A}}_{k+1} \leftarrow \tau_f \cdot \Delta^{\text{P}}_{k+1}$
    $k \leftarrow k+1$
end do
```

Before giving convergence results for our method, we compare Algorithm 2.1 with the work by Morales et al. [12]. Roughly, they compute a predictor step (without a trust-region constraint) followed by an accelerator step defined as the solution to the equality constrained subproblem we considered in [9, Section 2.3.2]. They then reduce the ℓ_1-merit function by performing a line search along the "bent" path defined by the steps s^{P}_k and s^{A}_k. Our methods differ in the following ways. Firstly, Algorithm 2.1 is based on trust-region methodology even though the predictor step is computed *without* a trust region radius; their algorithm is based on line-search philosophy. Secondly, global convergence of our algorithm is guaranteed by the Cauchy step, while convergence of their algorithm is ensured by the predictor step with a suitable line search. Thirdly, we allow and have analyzed an accelerator step computed as the minimizer of an inequality constrained subproblem, which allows for active set refinement; they have not considered such a subproblem, although one could imagine that such an analysis is possible. Finally, our algorithms differ even when using an equality constrained subproblem [9, Section 2.3.2] to compute an accelerator step. Following the rejection of a trial step, Morales et al. perform a line search in the direction of the predictor step. We, on the other hand, perform the equivalent of a backtracking line search with each trial point enhanced by a *new* accelerator direction. Since convergence of our method relies on the Cauchy point, we could easily use more sophisticated line search techniques without sacrificing convergence.

3 Convergence properties

We begin by defining a criticality measure based on the predictor step subproblem.

Lemma 3.1 *The quantity*

$$\chi(x,B) \overset{\text{def}}{=} f(x)+\sigma\|[c(x)]^-\|_1 - \min_{s\in\mathbb{R}^n} \left[f(x)+g(x)^T s + \tfrac{1}{2}s^T B s + \sigma\|[c(x)+J(x)s]^-\|_1\right] \quad (3.9)$$

is a criticality measure in the sense that

(i) $0 \le \chi(x,B) < \infty$ *for all* x *and every positive-definite matrix* B*;*

(ii) *for any positive-definite matrix* B*, it holds that* $\chi(x,B)=0$ *if and only if* x *is a first-order critical point for problem* (ℓ_1-σ)*; and*

(iii) *if* $\{x_k\}\to x_*$*,* $\{B_k\}$ *is a sequence of matrices such that for some positive scalars* κ_{min} *and* κ_{max}

$$0 < \lambda^B_{\min} \le \frac{v^T B_k v}{v^T v} \le \lambda^B_{\max} \quad \textit{for all } v \ne 0, \quad (3.10)$$

and $\chi(x_k, B_k) \to 0$, then x_* is a first-order critical point for problem (ℓ_1-σ).

Proof. Part (i) follows immediately from the fact that the minimization problem in (3.9) is strictly convex and has value $f(x) + \sigma\|[c(x)]^-\|_1$ at $s = 0$.

We now prove part (ii). The equation $\chi(x, B) = 0$ is true if and only if $s = 0$ is the unique minimizer to the strictly convex minimization problem used in equation (3.9). First order optimality implies that

$$\text{there exists } w \in \partial\,\|[c(x)]^-\|_1 \text{ such that } \big(g(x) + \sigma J(x)^T w\big)^T v \geq 0 \text{ for all } v \in \mathbb{R}^n, \tag{3.11}$$

where $\partial\,\|[c(x)]^-\|_1$ is the sub-differential [6, Section 14.3] of $\partial\,\|[\cdot]^-\|_1$ at the point $c(x)$. This proves part (ii) since condition (3.11) is precisely the first-order conditions for x to be a first-order critical point for problem (ℓ_1-σ).

Given a symmetric matrix B we define the vector obtained by stacking all of the entries of the lower triangular part of B (in a specified order) as $b(B) \in \mathbb{R}^{n_{\text{B}}}$, where $n_{\text{B}} \overset{\text{def}}{=} n(n+1)/2$. The assumption on the matrix sequence $\{B_k\}$ in part (iii) guarantees that the vector sequence $\{b(B_k)\}$ is bounded so that there exists a subsequence K such that $\lim_{k\in K} b(B_k) = b_*$. This implies that

$$\lim_{k\in K} x_k = x_* \text{ and } \lim_{k\in K} B_k = B_* \text{ with } B_* \text{ positive definite,} \tag{3.12}$$

where $B_* \overset{\text{def}}{=} b(b_*)$. Now define the function

$$F\big(s, x, b(B)\big) = f(x) + g(x)^T s + \tfrac{1}{2} s^T B s + \sigma\|[c(x) + J(x)s]^-\|_1 \tag{3.13}$$

so that $F(s, x, b)$ is defined and continuous on $\mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{n_{\text{B}}}$ and convex for each fixed (x, b). It follows from [4, Theorem 3.2.8 using the continuous *point-to-set* map $\mathcal{C}(x, b) = \mathbb{R}^n$] that

$$F_*(x, b) \overset{\text{def}}{=} \min_{s\in\mathbb{R}^n} F\big(s, x, b\big) \tag{3.14}$$

is continuous so that we may deduce from (3.9) that $\chi(x, B)$ is also continuous as a function of (x, B). Thus we have

$$\chi(x_*, B_*) = \lim_{k\in K} \chi(x_k, B_k) = 0, \tag{3.15}$$

where the first equality follows by continuity and the second by assumption. Part (ii) and (3.12) then imply that x_* is a first-order critical point for problem (ℓ_1-σ). □

We now give a lower bound for the change in the faithful model obtained from the Cauchy step; this is the essential estimate for proving global convergence of Algorithm 2.1. The result uses the change in the convex model M_k^{B}, which we define as

$$\varDelta M_k^{\text{B}}(s) \overset{\text{def}}{=} M_k^{\text{B}}(0) - M_k^{\text{B}}(s). \tag{3.16}$$

Lemma 3.2 *Let s_k^{P} and s_k^{CP} be defined as previously. Then*

$$\varDelta M_k^{H}(s_k^{CP}) \geq \tfrac{1}{2}\varDelta M_k^{B}(s_k^{P}) \min\left(1, \frac{\Delta_k^{P}}{\|s_k^{P}\|_\infty}, \frac{\varDelta M_k^{B}(s_k^{P})}{n\|B_k - H_k\|_2 \|s_k^{P}\|_\infty^2}\right). \tag{3.17}$$

Proof. We consider two cases.

Case 1: $s_k^{\mathrm{P}T} H_k s_k^{\mathrm{P}} \le s_k^{\mathrm{P}} B_k s_k^{\mathrm{P}}$

Subcase 1: $\|s_k^{\mathrm{P}}\|_\infty \le \Delta_k^{\mathrm{P}}$

This subcase implies $\alpha_u \ge 1$ so that $\alpha = 1$ is in the domain of the Cauchy step computation (2.3). This implies $M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) \le M_k^{\mathrm{H}}(s_k^{\mathrm{P}}) \le M_k^{\mathrm{B}}(s_k^{\mathrm{P}})$, where the second inequality follows since $s_k^{\mathrm{P}T} H_k s_k^{\mathrm{P}} \le s_k^{\mathrm{P}} B_k s_k^{\mathrm{P}}$ by assumption. Since $M_k^{\mathrm{H}}(0) = M_k^{\mathrm{B}}(0)$, we conclude that

$$\mathit{\Delta} M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) = M_k^{\mathrm{H}}(0) - M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) \ge M_k^{\mathrm{B}}(0) - M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) = \mathit{\Delta} M_k^{\mathrm{B}}(s_k^{\mathrm{P}}). \tag{3.18}$$

Subcase 2: $\|s_k^{\mathrm{P}}\|_\infty > \Delta_k^{\mathrm{P}}$

Note that we now have $0 \le \alpha_u < 1$. For a general $0 \le \alpha \le 1$, we have

$$\begin{aligned}
\mathit{\Delta} M_k^{\mathrm{B}}(\alpha s_k^{\mathrm{P}}) &= \sigma\big(\|[c_k]^-\|_1 - \|[c_k + \alpha J_k s_k^{\mathrm{P}}]^-\|_1\big) - \alpha g_k^T s_k^{\mathrm{P}} - \frac{\alpha^2}{2} s_k^{\mathrm{P}T} B_k s_k^{\mathrm{P}} && (3.19)\\
&\ge \alpha\sigma\big(\|[c_k]^-\|_1 - \|[c_k + J_k s_k^{\mathrm{P}}]^-\|_1\big) - \alpha g_k^T s_k^{\mathrm{P}} - \frac{\alpha}{2} s_k^{\mathrm{P}T} B_k s_k^{\mathrm{P}}. && (3.20)\\
&= \alpha \mathit{\Delta} M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) && (3.21)
\end{aligned}$$

so that

$$\mathit{\Delta} M_k^{\mathrm{B}}(\alpha_u s_k^{\mathrm{P}}) \ge \alpha_u \mathit{\Delta} M_k^{\mathrm{B}}(s_k^{\mathrm{P}}). \tag{3.22}$$

Equation (3.19) follows from the definition of $\mathit{\Delta} M_k^{\mathrm{B}}$, (3.20) follows since $0 \le \alpha \le 1$ and from [9, Lemma 2.2], and (3.21) follows from the definition of $\mathit{\Delta} M_k^{\mathrm{B}}$. We may then deduce

$$\begin{aligned}
\mathit{\Delta} M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) &= M_k^{\mathrm{H}}(0) - M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) \ge M_k^{\mathrm{H}}(0) - M_k^{\mathrm{H}}(\alpha_u s_k^{\mathrm{P}}) \qquad \big(\text{using def. of } \mathit{\Delta} M_k^{\mathrm{H}} \text{ and } s_k^{\mathrm{CP}}\big)\\
&\ge M_k^{\mathrm{B}}(0) - M_k^{\mathrm{B}}(\alpha_u s_k^{\mathrm{P}}) \qquad \big(\text{using defs. of } M_k^{\mathrm{B}} \text{ and } M_k^{\mathrm{H}} \text{ and the fact } s_k^{\mathrm{P}T} H_k s_k^{\mathrm{P}} \le s_k^{\mathrm{P}T} B_k s_k^{\mathrm{P}}\big)\\
&= \mathit{\Delta} M_k^{\mathrm{B}}(\alpha_u s_k^{\mathrm{P}}) \ge \alpha_u \mathit{\Delta} M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) \qquad \big(\text{using def. of } \mathit{\Delta} M_k^{\mathrm{B}} \text{ and (3.22)}\big)\\
&= \frac{\Delta_k^{\mathrm{P}}}{\|s_k^{\mathrm{P}}\|_\infty} \mathit{\Delta} M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) \qquad \big(\text{using def. of } \alpha_u\big).
\end{aligned} \tag{3.23}$$

Case 2: $s_k^{\mathrm{P}T} H_k s_k^{\mathrm{P}} > s_k^{\mathrm{P}} B_k s_k^{\mathrm{P}} > 0$

Since $\alpha_k \le \Delta_k^{\mathrm{P}} / \|s_k^{\mathrm{P}}\|_\infty$ by definition and $\alpha_k < 1$ as a consequence of $s_k^{\mathrm{P}T} H_k s_k^{\mathrm{P}} > s_k^{\mathrm{P}} B_k s_k^{\mathrm{P}} > 0$, we conclude that $0 \le \alpha_k \le \min\left(1, \frac{\Delta_k^{\mathrm{P}}}{\|s_k^{\mathrm{P}}\|_\infty}\right) = \min(1, \alpha_u)$; for ease of notation, we define $\alpha_{\min} = \min(1, \alpha_u)$. For all $0 \le \alpha \le \alpha_{\min}$, we must have

$$\begin{aligned}
\mathit{\Delta} M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) &\ge \mathit{\Delta} M_k^{\mathrm{H}}(\alpha s_k^{\mathrm{P}}) && (3.24)\\
&= \sigma\big(\|[c_k]^-\|_1 - \|[c_k + \alpha J_k s_k^{\mathrm{P}}]^-\|_1\big) - \alpha g_k^T s_k^{\mathrm{P}} - \frac{\alpha^2}{2} s_k^{\mathrm{P}T} H_k s_k^{\mathrm{P}} && (3.25)\\
&= \sigma\big(\|[c_k]^-\|_1 - \|[c_k + \alpha J_k s_k^{\mathrm{P}}]^-\|_1\big) - \alpha g_k^T s_k^{\mathrm{P}} - \frac{\alpha^2}{2} s_k^{\mathrm{P}T} B_k s_k^{\mathrm{P}} + \frac{\alpha^2}{2} s_k^{\mathrm{P}T} (B_k - H_k) s_k^{\mathrm{P}}. && (3.26)
\end{aligned}$$

Equation (3.24) follows since s_k^{CP} minimizes $M_k^{\mathrm{H}}(\alpha s_k^{\mathrm{P}})$ for $0 \le \alpha \le \alpha_{\min}$, while (3.25) and (3.26) follow from the definition of $\mathit{\Delta} M_k^{\mathrm{H}}$ and simple algebra. Continuing to bound the change in the faithful model, we have

$$\Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) \geq \alpha\sigma\big(\|[c_k]^-\|_1 - \|[c_k + J_k s_k^{\mathrm{P}}]^-\|_1\big) - \alpha g_k^T s_k^{\mathrm{P}} - \frac{\alpha}{2} s_k^{\mathrm{P}T} B_k s_k^{\mathrm{P}} + \frac{\alpha^2}{2} s_k^{\mathrm{P}T}(B_k - H_k)s_k^{\mathrm{P}} \tag{3.27}$$

$$= \alpha \Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) + \frac{\alpha^2}{2} s_k^{\mathrm{P}T}(B_k - H_k)s_k^{\mathrm{P}} \tag{3.28}$$

for all $0 \leq \alpha \leq \alpha_{\min}$. Equation (3.27) follows from equation (3.26), [9, Lemma 2.2], and the inequality $\alpha^2 \leq \alpha$, which holds since $0 \leq \alpha \leq \alpha_{\min} \leq 1$, while equation (3.28) follows from the simplification of equation (3.27) and the definition of $\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}})$.

The previous string of inequalities holds for *all* $0 \leq \alpha \leq \alpha_{\min}$, so it must hold for the value of α that maximizes the right-hand-side of (3.28). As a function of α, the right-hand-side may be written as $q(\alpha) = a\alpha^2 + b\alpha$ where

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

There are two sub-cases to consider.

Subcase 1 : $-b/2a \leq \alpha_{\min}$

In this case the maximizer on the interval $[0, \alpha_{\min}]$ must occur at $\alpha = -b/2a$ so that the maximum is

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

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

$$q(-b/2a) = \frac{\big(\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}})\big)^2}{2|s_k^{\mathrm{P}T}(B_k - H_k)s_k^{\mathrm{P}}|} \geq \frac{\big(\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}})\big)^2}{2\|B_k - H_k\|_2\|s_k^{\mathrm{P}}\|_2^2} \geq \frac{\big(\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}})\big)^2}{2n\|B_k - H_k\|_2\|s_k^{\mathrm{P}}\|_\infty^2}. \tag{3.29}$$

Subcase 2 : $-b/2a > \alpha_{\min}$

In this case the maximizer of q on the interval $[0, \alpha_{\min}]$ is $\alpha = \alpha_{\min}$ and the maximum is bounded by

$$q(\alpha_{\min}) = a\alpha_{\min}^2 + b\alpha_{\min} = \alpha_{\min}(a\alpha_{\min} + b) > \frac{\alpha_{\min}}{2} b = \frac{\min(1, \alpha_u)}{2} \Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) \tag{3.30}$$

since the inequality $-b/2a > \alpha_{\min}$ implies $a\alpha_{\min} > -b/2$ because $a < 0$.

If we denote the maximizer of $q(\alpha)$ on the interval $[0, \alpha_{\min}]$ by α^*, then equations (3.29) and (3.30) show that

$$q(\alpha^*) \geq \tfrac{1}{2} \Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) \min\left(1, \frac{\Delta_k^{\mathrm{P}}}{\|s_k^{\mathrm{P}}\|_\infty}, \frac{\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}})}{n\|B_k - H_k\|_2\|\Delta_k^{\mathrm{P}}\|_\infty^2}\right). \tag{3.31}$$

Returning to equation (3.28), we have

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

Combining this with equations (3.18) and (3.23) gives the required result. □

We now show that under reasonable assumptions on the convex models M_k^{B}, the predictor steps will be uniformly bounded.

Lemma 3.3 *Let f and c be continuously differentiable functions. Assume that $\{x_k\}$ is* any *vector sequence such that $\{x_k\}_{k\geq 0} \subset \mathcal{B} \subset \mathbb{R}^n$ for some compact set $\mathcal{B}$, and that $\{B_k\}$ is* any *matrix sequence such that*

$$0 < \lambda_{\min}^{B} \leq \frac{v^T B_k v}{v^T v} \quad \textit{for all } v \neq 0 \in \mathbb{R}^n \tag{3.32}$$

for some positive constant $\lambda_{\min}^B$ independent of k. Then there exists a positive constant κ_{pred} such that $\|s_k^{\mathrm{P}}\|_\infty \leq \kappa_{pred}$ where s_k^{P} is the predictor step, i.e., the unique minimizer of problem (2.2)*.*

Proof. Writing down the optimality conditions for problem (2.2) shows that

$$s_k^{\mathrm{P}} = -B_k^{-1}\left(g_k - J_k^T y_k^{\mathrm{P}}\right) \quad \text{and} \quad \|s_k^{\mathrm{P}}\|_\infty \leq \sigma, \tag{3.33}$$

where y_k^{P} is the vector of Lagrange multipliers. It is now easy to see that the required result holds by using (3.33), standard norm inequalities, (3.32), the fact that $\{x_k\}$ is contained in the compact set $\mathcal{B}$, continuity of norms, and continuity of g and J. □

The following global convergence proof for Algorithm 2.1 requires the following sets:

$$\begin{aligned}
\mathcal{S} &= \{k \in \mathbb{N} \mid \text{ iterate } k \text{ is labeled either "successful" or "successful Cauchy" by Alg. 2.1}\};\\
\mathcal{U} &= \{k \in \mathbb{N} \mid \text{ iterate } k \text{ is labeled "unsuccessful" by Algorithm 2.1}\}; \text{ and}\\
\mathcal{R} &= \{k \in \mathbb{N} \mid \text{ iterate } k \text{ is labeled "revert to saved point" by Algorithm 2.1}\}.
\end{aligned} \tag{3.34}$$

Theorem 3.4 *Let f and c be twice continuously differentiable functions, and let $\{x_k\}$, $\{H_k\}$, $\{B_k\}$, $\{\Delta_k^{\mathrm{P}}\}$, and $\{\Delta_k^{\mathrm{A}}\}$, be sequences generated by* Algorithm 2.1*. Assume that the following conditions hold:*

1. *$\{x_k\}_{k\geq 0} \subset \mathcal{B} \subset \mathbb{R}^n$ for some compact set $\mathcal{B}$; and*
2. *there exist positive constants $\lambda_{\min}^B$, $\lambda_{\max}^B$ and b_H such that $0 \leq \lambda_{\min}^B \leq (v^T B_k v)/\|v\|_2^2 \leq \lambda_{\max}^B$ for all $v \neq 0 \in \mathbb{R}^n$, and $\|H_k\|_2 \leq b_H$.*

Then, either x_K is a first-order critical 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 $\{\Delta_k^{\text{P}}\}$ that converges to zero.
Since Δ_k^{P} is only decreased following an unsuccessful nonmonotone phase and since the first Cauchy step of each nonmonotone phase is always checked for sufficient progress, we may conclude that there exists a subsequence $\mathcal{K} \subseteq \mathbb{N}$ such that

$$\lim_{k\in\mathcal{K}} x_k = x_*, \tag{3.35}$$

$$\lim_{k\in\mathcal{K}} \Delta_k^{\text{P}} = 0, \tag{3.36}$$

$$\lim_{k\in\mathcal{K}} \|s_k^{\text{CP}}\|_\infty = 0, \text{ and} \tag{3.37}$$

$$r_k^{\text{CP}} < \eta_S \text{ for all } k \in \mathcal{K}, \tag{3.38}$$

where

$$r_k^{\text{CP}} \stackrel{\text{def}}{=} \frac{\phi_k - \phi(x_k + s_k^{\text{CP}})}{\mathit{\Delta} M_k^{\text{H}}(s_k^{\text{CP}})}.$$

Subcase 1: There exists a subsequence of $\{\mathit{\Delta} M_k^{\text{B}}(s_k^{\text{P}})\}_{k\in\mathcal{K}}$ that converges to zero.
It follows immediately from Lemma 3.1 that x_* is a first-order critical point for problem (ℓ_1-σ) since $\chi(x_k, B_k) \equiv \mathit{\Delta} M_k^{\text{B}}(s_k^{\text{P}})$.
Subcase 2: There does not exist a subsequence of $\{\mathit{\Delta} M_k^{\text{B}}(s_k^{\text{P}})\}_{k\in\mathcal{K}}$ that converges to zero.
This implies the existence of a positive scalar δ such that

$$\mathit{\Delta} M_k^{\text{B}}(s_k^{\text{P}}) \geq \delta > 0 \text{ for all } k \in \mathcal{K}. \tag{3.39}$$

A Taylor expansion of f at x_k in a general direction v gives

$$f(x_k + \varepsilon v) = f_k + \varepsilon g_k^T v + o(\varepsilon) = f_k + \varepsilon g_k^T v + \frac{\varepsilon^2}{2} v^T H_k v + o(\varepsilon) \tag{3.40}$$

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

$$c(x_k + \varepsilon v) = c_k + \varepsilon J_k v + o(\varepsilon). \tag{3.41}$$

Combining these two equations gives

$$\begin{aligned} \phi(x_k + \varepsilon v) &= f_k + \varepsilon g_k^T v + \frac{\varepsilon^2}{2} v^T H_k v + o(\varepsilon) + \sigma \|[c_k + \varepsilon J_k v + o(\varepsilon)]^-\|_1 \\ &= f_k + \varepsilon g_k^T v + \frac{\varepsilon^2}{2} v^T H_k v + \sigma \|[c_k + \varepsilon J_k v]^-\|_1 + o(\varepsilon) \\ &= M_k^{\text{H}}(\varepsilon v) + o(\varepsilon), \end{aligned} \tag{3.42}$$

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 $M_k^{\text{H}}(\varepsilon v)$. Choosing $v = s_k^{\text{CP}}/\|s_k^{\text{CP}}\|_\infty$ and $\varepsilon = \|s_k^{\text{CP}}\|_\infty$ in equation (3.42) yields

$$\phi(x_k + s_k^{\text{CP}}) = M_k^{\text{H}}(s_k^{\text{CP}}) + o(\|s_k^{\text{CP}}\|_\infty). \tag{3.43}$$

Equation (3.43) then implies the equation

$$r_k^{\text{CP}} = \frac{\phi_k - \phi(x_k + s_k^{\text{CP}})}{\Delta M_k^{\text{H}}(s_k^{\text{CP}})} = \frac{\Delta M_k^{\text{H}}(s_k^{\text{CP}}) + o(\|s_k^{\text{CP}}\|_\infty)}{\Delta M_k^{\text{H}}(s_k^{\text{CP}})} = 1 + \frac{o(\|s_k^{\text{CP}}\|_\infty)}{\Delta M_k^{\text{H}}(s_k^{\text{CP}})} \tag{3.44}$$

since $\phi_k = M_k^{\text{H}}(0)$. We now proceed to bound $\Delta M_k^{\text{H}}(s_k^{\text{CP}})$. For all $k \in \mathcal{K}$ sufficiently large, we have for some constant $\kappa_{\text{pred}} > 0$ that

$$\begin{aligned}\Delta M_k^{\text{H}}(s_k^{\text{CP}}) &\geq \frac{1}{2}\Delta M_k^{\text{B}}(s_k^{\text{P}}) \min\left(1, \frac{\Delta_k^{\text{P}}}{\|s_k^{\text{P}}\|_\infty}, \frac{\Delta M_k^{\text{B}}(s_k^{\text{P}})}{n\|B_k - H_k\|_2\|s_k^{\text{P}}\|_\infty^2}\right) \quad \big(\text{use Lemma 3.2}\big)\\ &\geq \frac{\delta}{2}\min\left(1, \frac{\Delta_k^{\text{P}}}{\kappa_{\text{pred}}}, \frac{\delta}{n(\lambda_{\max}^B + b_H)\kappa_{\text{pred}}^2}\right) \quad \big(\text{use (3.39), assumption 2, and Lemma 3.3}\big)\\ &= \frac{\delta}{2\kappa_{\text{pred}}}\Delta_k^{\text{P}} \quad \big(\text{use (3.36)}\big).\end{aligned} \tag{3.45}$$

It now follows that there exists a positive sequence $\{z_k\}$ such that for $k \in \mathcal{K}$ sufficiently large

$$\begin{aligned}\left|\frac{o(\|s_k^{\text{CP}}\|_\infty)}{\Delta M_k^{\text{H}}(s_k^{\text{CP}})}\right| &\leq \frac{2\kappa_{\text{pred}} z_k \|s_k^{\text{CP}}\|_\infty}{\delta\Delta_k^{\text{P}}} \quad \big(\text{use (3.45) and definition of "little-oh"}\big)\\ &\leq \frac{2\kappa_{\text{pred}} z_k \Delta_k^{\text{P}}}{\delta\Delta_k^{\text{P}}} = \frac{2\kappa_{\text{pred}}}{\delta} z_k \quad \big(\text{use definition of Cauchy step and simplify}\big)\end{aligned} \tag{3.46}$$

and where the subsequence $\{z_k\}_\mathcal{K}$ converges to zero. It then follows from (3.44) and (3.46) that

$$r_k^{\text{CP}} = 1 + o(1) \text{ for } k \in \mathcal{K}. \tag{3.47}$$

This is a contradiction since this implies that for $k \in \mathcal{K}$ sufficiently large the identity $r_k^{\text{CP}} > \eta_S$ holds, which violates equation (3.38); thus subcase 2 can not occur. Therefore, if Case 1 occurs, then x^* is a first-order critical point as shown in subcase 1.

Case 2 : there does not exist 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 $\mathcal{K}_\mathcal{S} \subseteq \mathcal{S}$ (recall the definition of $\mathcal{S}$ given by (3.34)) such that

$$\lim_{k \in \mathcal{K}_\mathcal{S}} x_{k-l(k)} = x_*, \tag{3.48}$$

$$\Delta_k^{\text{P}} \geq \delta > 0 \text{ for all } k, \tag{3.49}$$

where for each $k \in \mathcal{S}$ we define $l(k)$ to be the number of fails that occurred in that nonmonotone phase before that successful iteration was computed. For consistency, if iterate k was successful but was not part of a nonmonotone phase, then we define $l(k) = 0$. Thus every successful iterate is part of a nonmonotone sequence, but it may have length zero. Also, for each $k \in \mathcal{S}$ we define $k^+(k) \in \mathcal{K}_\mathcal{S}$ to be the smallest number in $\mathcal{K}_\mathcal{S}$ that is *strictly* greater than k (see Figure 3.1). Note that this implies

$$\phi(x_{k+1}) \geq \phi\big(x_{k^+(k)-l(k^+(k))}\big) \text{ for all } k \in \mathcal{S} \quad \text{and} \quad \lim_{k \in \mathcal{S}} k^+(k) = \infty. \tag{3.50}$$

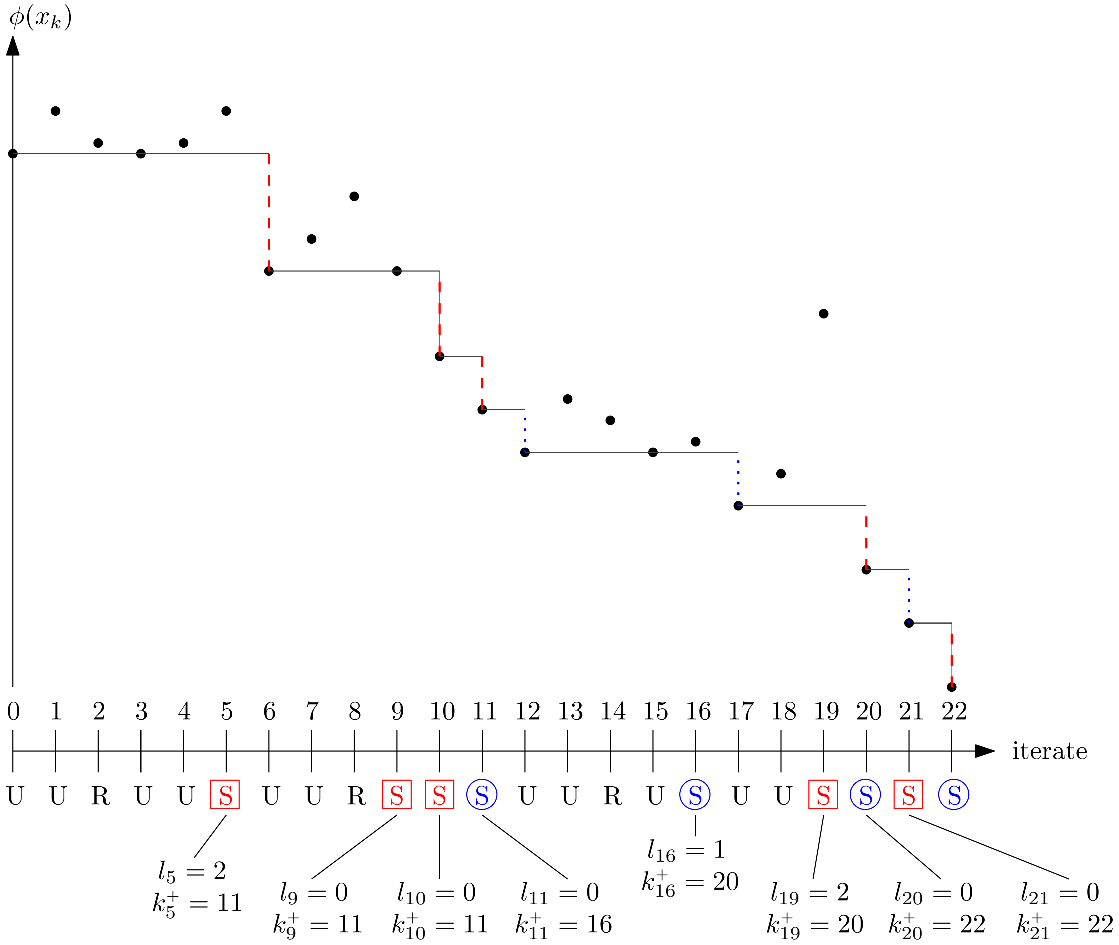

Figure 3.1: Illustration of the quantities used in Case 2 of Theorem 3.4 assuming that *max_fails* = 2. The x-axis represents the iterate and the y-axis the value of the merit function ϕ for a given iterate. The values S, U, and R below the x-axis denote whether that iterate belongs to the indexing set $\mathcal{S}$, $\mathcal{U}$, or $\mathcal{R}$ (see (3.34)), respectively. Every S is enclosed in either a (blue) circle or a (red) square—the circle indicates that the corresponding iterate is in the subsequence $\mathcal{K}_{\mathcal{S}}$, while the square indicates that the iterate was *not* in $\mathcal{K}_{\mathcal{S}}$. The (dark-grey) horizontal solid lines indicate the least value of ϕ accepted as a successful iterate up until that point. The length of a (blue) dotted vertical line or a (red) dashed vertical line located above an iterate k indicates the improvement in the merit function obtained from the successful step s_{k-1} as compared to the previous best successful value. We have used the short-hand notation $l_k = l(x_k)$ and $k_l^+ = k^+(x_l)$.

For the remainder of this proof, we write ΔM^{H} instead of ΔM^{H}_k and ΔM^{B} instead of ΔM^{B}_k—the "missing" argument is always assumed to be the subscript of the step, i.e., $\Delta M^{\text{B}}(s_j)$ means $\Delta M^{\text{B}}_j(s_j)$ for any iterate j.

If $k \in \mathcal{K}_{\mathcal{S}}$ and k is classified as a "successful" iteration by Algorithm 2.1, then it follows from (2.8) that

$$\phi(x_{k-l(k)}) - \phi(x_{k+1}) \geq \eta_S \Delta M^{\text{H}}(s_{k-l(k)}) \geq \eta\eta_S \Delta M^{\text{H}}(s^{\text{CP}}_{k-l(k)}). \tag{3.51}$$

On the other hand, if $k \in \mathcal{K}_{\mathcal{S}}$ and k is classified as a "successful Cauchy" iteration by Algorithm 2.1, then we have by construction that

$$\phi(x_{k-l(k)}) - \phi(x_{k+1}) \geq \eta_S \Delta M^{\text{H}}(s^{\text{CP}}_{k-l(k)}). \tag{3.52}$$

Since $\eta \in (0,1)$, we conclude that

$$\phi(x_{k-l(k)}) - \phi(x_{k+1}) \geq \eta\eta_S \Delta M^{\text{H}}(s^{\text{CP}}_{k-l(k)}) \text{ for all } k \in \mathcal{K}_{\mathcal{S}}. \tag{3.53}$$

Equation (3.53), Lemmas 3.2 and 3.3, (3.49), and assumption 2 of this theorem imply

$$\phi(x_{k-l(k)}) - \phi(x_{k+1}) \geq \frac{\eta\eta_S}{2} \Delta M^{\text{B}}\big(s^{\text{P}}_{k-l(k)}\big) \min\left(1, \frac{\delta}{\kappa_{\text{pred}}}, \frac{\Delta M^{\text{B}}\big(s^{\text{P}}_{k-l(k)}\big)}{(\lambda^B_{\max} + b_H)\kappa^2_{\text{pred}}}\right) \tag{3.54}$$

for some positive constant κ_{pred} independent of k. If we let $\bar{k} \in \mathcal{S}$ and sum over all $k \in \mathcal{K}_{\mathcal{S}}$ less than $\bar{k}$, we have

$$\begin{aligned}
\sum_{k\in\mathcal{K}_{\mathcal{S}}, k\leq\bar{k}} \frac{\eta\eta_S}{2} \Delta M^{\text{B}}\big(s^{\text{P}}_{k-l(k)}\big) \min\left(1, \frac{\delta}{\kappa_{\text{pred}}}, \frac{\Delta M^{\text{B}}\big(s^{\text{P}}_{k-l(k)}\big)}{(\lambda^B_{\max} + b_H)\kappa^2_{\text{pred}}}\right) &\leq \sum_{k\in\mathcal{K}_{\mathcal{S}}, k\leq\bar{k}} \phi\big(x_{k-l(k)}\big) - \phi(x_{k+1}) && (3.55)\\
&\leq \sum_{k\in\mathcal{S}, k\leq\bar{k}} \phi\big(x_{k-l(k)}\big) - \phi(x_{k+1}) && (3.56)\\
&= \phi(x_0) - \phi(x_{\bar{k}+1}) && (3.57)\\
&\leq \phi\big(x_0\big) - \phi\big(x_{k^+(\bar{k}) - l(k^+(\bar{k}))}\big). && (3.58)
\end{aligned}$$

Equation (3.55) follows from (3.54), (3.56) since we are adding more positive terms to the sum, (3.57) follows from the construction of the algorithm and the fact that $\bar{k} \in \mathcal{S}$ by assumption, and (3.58) follows from (3.50). To help the reader understand, we note that for the value $\bar{k} = 21$ the right-hand-side of (3.55) is equal to the sum of the lengths of the (blue) dotted lines in Figure 3.1, while the right-hand-side of (3.56) is equal to the sum of the lengths of the (blue) dotted lines *and* the (red) dashed lines. If we now let $\bar{k}$ converge to infinity in the previous string of inequalities and use (3.50) and (3.48), we may conclude that

$$\sum_{k\in\mathcal{K}_{\mathcal{S}}} \frac{\eta\eta_S}{2}\Delta M^{\mathrm{B}}\big(s^{\mathrm{P}}_{k-l(k)}\big)\min\left(1,\frac{\delta}{\kappa_{\mathrm{pred}}},\frac{\Delta M^{\mathrm{B}}\big(s^{\mathrm{P}}_{k-l(k)}\big)}{(\lambda^B_{\max}+b_H)\kappa^2_{\mathrm{pred}}}\right)\le \phi(x_0)-\phi(x^*), \tag{3.59}$$

which implies

$$\lim_{k\in\mathcal{K}_{\mathcal{S}}}\Delta M^{\mathrm{B}}(s^{\mathrm{P}}_{k-l(k)})=0 \tag{3.60}$$

because the series on the left-hand-side is convergent. Since (3.48) states $\lim_{k\in\mathcal{K}_{\mathcal{S}}} x_{k-l(k)} = x_*$ and it follows from (3.9) and (3.60) that

$$\lim_{k\in\mathcal{K}_{\mathcal{S}}}\chi\big(x_{k-l(k)},B_{k-l(k)}\big)=\lim_{k\in\mathcal{K}_{\mathcal{S}}}\Delta M^{\mathrm{B}}(s^{\mathrm{P}}_{k-l(k)})=0, \tag{3.61}$$

we conclude from part (iii) of Lemma 3.1 that x_* is a first-order critical point for problem (ℓ_1-σ).

In both cases we have shown that there exists a limit point x_* that is a first-order critical point for problem (ℓ_1-σ). We are done since one of these cases must occur. □

We conclude this section by giving local convergence results for problem (NP). These results assume that the penalty parameter σ is sufficiently large so that minimizers of the ℓ_1-penalty function correspond to minimizers of problem (NP) (see [5,14] for more details on exactly how these two problems are related). We note that many authors have provided frameworks for guaranteeing that this condition holds in practice [1–3,8,10,11,13,15–17]. We use the following definitions related to a solution of problem (NP).

Definition 1 (First-order KKT point) *We say that the point* (x^*,y^*) *is a first-order KKT point for problem* (NP) *if*

$$g(x^*)-J(x^*)^Ty^*=0,\quad c(x^*)\ge 0,\quad y^*\ge 0,\quad \text{and}\quad c(x^*)\cdot y^*=0. \tag{3.62}$$

Given a first-order KKT point (x^*,y^*), we let $\mathcal{A}\overset{\text{def}}{=}\{i : c_i(x^*)=0\}$ denote the index set of constraints active at x^*.

Definition 2 (Second-order sufficient conditions) *A point* (x^*,y^*) *satisfies the second-order sufficient conditions for problem* (NP) *if* (x^*,y^*) *is a first-order KKT point and if there exists* $\lambda^H_{\min}>0$ *such that* $s^TH(x^*,y^*)s\ge\lambda^H_{\min}s^Ts$ *for all* s *satisfying* $J_{\mathcal{A}}(x^*)s=0$.

Definition 3 (Strict complementarity) *We say that strict complementarity holds at a KKT point* (x^*,y^*) *for problem* (NP) *if* $y^*_{\mathcal{A}}>0$.

Definition 4 (Linear independent constraint qualification) *We say that the linear independent constraint qualification* (LICQ) *holds at a KKT point* (x^*,y^*) *for problem* (NP) *if the matrix* $J_{\mathcal{A}}(x^*)$ *has full row rank.*

Definition 5 *We say that the* strong *second-order sufficient conditions hold at a point* (x^*, y^*) *if it satisfies Definitions* 1 – 4.

For our first result we assume that an accelerator step is computed from subproblem (SEQP) as discussed in [9, Section 2.3.2]. We restate this subproblem for convenience:

$$\text{(SEQP)} \qquad \underset{s\in\mathbb{R}^n}{\text{minimize}} \quad \bar{f}_k + (g_k + H_k s_k^{\text{P}})^T s + \tfrac{1}{2} s^T H_k s \\ \text{subject to} \quad [J_k s]_{\mathcal{A}(s_k^{\text{P}})} = 0, \;\; \|s\|_2 \le \Delta_k^{\text{A}},$$

where $\mathcal{A}(s_k^{\text{P}}) = \{i : [c_k + J_k s_k^{\text{P}}]_i \le 0\}$ and $\bar{f}_k = f_k + g_k^T s_k^{\text{P}} + \frac{1}{2} s_k^{\text{P}T} H_k s_k^{\text{P}}$. Since this subproblem only defines multipliers for the constraints whose indices are in the set $\mathcal{A}(s_k^{\text{P}})$, we form accelerator multipliers y_k^{A} by "scattering" the multipliers from subproblem (SEQP) into the appropriate locations of a zero-vector of length m. The following theorem is the same as [8, Theorem 4.7].

Theorem 3.5 (SEQP local convergence result) *Let* (x^*, y^*) *be a minimizer for problem* (NP) *that satisfies the strong second-order sufficient conditions as given by Definition* 5. *Let the assumptions of Theorem* 3.4 *hold and suppose that* $\sigma > \|y^*\|_\infty$, *the accelerator step is computed from subproblem* (SEQP) *with the choice* $H_k \equiv H(x_k, y_k^F)$, *and max_fails* ≥ 1 *in Algorithm* 2.1. *It follows that there exists an open neighborhood of* (x^*, y^*) *such that if the accelerator step is computed for every iteration once the first successful iterate of Algorithm* 2.1 *is contained in this neighborhood then the sequences of iterates* $\{x_k\}$ *and* $\{y_k\}$ *generated by Algorithm* 2.1 *converge to* x^* *and* y^* *at a Q-superlinear and R-superlinear rate, respectively. Moreover, if* $H(x,y)$ *is Lipschitz continuous in a neighborhood of* (x^*, y^*), *then they convergence at a Q-quadratic and R-quadratic rate, respectively.*

Proof. The only reason why the proof of [8, Theorem 4.7] would not carry-over is because of the modification to how the predictor step is computed. However, the only property required is that the predictor trust-region constraint ultimately is inactive following a successful step. Since we have removed the trust-region constraint altogether, the result is immediate. □

Finally, we consider the rate of convergence of Algorithm 2.1 when the accelerator step is computed from subproblem (SIQP-E) as described in [9, Section 2.3.1]. We restate this subproblem for convenience:

$$\text{(SIQP-E)} \quad \underset{s\in\mathbb{R}^n}{\text{minimize}} \quad \bar{f}_k + (g_k + H_k s_k^{\text{CP}})^T s + \tfrac{1}{2} s^T H_k s + \sigma \|[c_k + J_k(s_k^{\text{CP}} + s)]_{\mathcal{V}_k}^-\|_1 \\ \text{subject to} \quad [c_k + J_k(s_k^{\text{CP}} + s)]_{\mathcal{S}_k} \ge 0, \\ (g_k + H_k s_k^{\text{CP}} + \sigma J_k^T z_k)^T s \le 0, \quad \|s\|_\infty \le \Delta_k^{\text{A}},$$

where

$$[z_k]_i = \begin{cases} -1 & \text{if } i \in \mathcal{V}_k\ , \\ 0 & \text{if } i \in \mathcal{S}_k\ , \end{cases} \tag{3.63}$$

$\mathcal{V}_k = \{i : [c_k + J_k s_k^{\text{CP}}]_i < 0\}$, $\mathcal{S}_k = \{i : [c_k + J_k s_k^{\text{CP}}]_i \ge 0\}$, $\bar{f}_k = f_k + g_k^T s_k^{\text{CP}} + \frac{1}{2} s_k^{\text{CP}T} H_k s_k^{\text{CP}}$, and $(g_k + H_k s_k^{\text{CP}} + \sigma_k J_k^T z_k)^T s \le 0$ is the so-called "descent-constraint".

Theorem 3.6 (SIQP-E local convergence result) *Let* (x^*, y^*) *be a minimizer for problem* (NP) *that satisfies the strong second-order sufficient conditions as given by Definition* 5. *Let the assumptions of Theorem* 3.4 *hold and assume that* $\sigma > \|y^*\|_\infty$, *the accelerator step is computed from subproblem* (SIQP-E) *with the choice* $H_k \equiv H(x_k, y_k^F)$, *and max_fails* ≥ 1 *in Algorithm* 2.1. *It follows that there exists an open neighborhood of* (x^*, y^*) *such that if the accelerator step is a solution of minimal-norm and is computed for every iteration once the first successful iterate of Algorithm* 2.1 *enters the open neighborhood, then the sequences of iterates* $\{x_k\}$ *and* $\{y_k\}$ *converge to* x^* *and* y^* *at a Q-superlinear and R-superlinear rate, respectively. Moreover, if* $H(x, y)$ *is Lipschitz continuous in a neighborhood of* (x^*, y^*), *then they converge at a Q-quadratic and R-quadratic rate, respectively.*

Proof. Follows exactly as in [8, Theorem 4.9]. □

4 Conclusions and future work

In [8, 9] we introduced S2QP—a second derivative trust-region SQP method for solving nonlinear nonconvex optimization problems. This method utilizes a so-called predictor step for proving both global and fast local convergence. Computation of this step involves solving a strictly convex quadratic program with a trust-region constraint. This is not ideal since the trust-region constraint i) may occasionally degrade the quality of the predictor step; ii) may diminish its ability to identify an optimal active set; and iii) prevents re-use of the same predictor step during a sequence of *unsuccessful* iterates. In this paper we have removed the trust-region constraint and proved that the resulting algorithm is still globally convergent, while maintaining local superlinear convergence. We consider Lemma 3.5 to be additional evidence that removing the trust-region constraint is the "right" thing to do since the result follows naturally; this is in contrast to the analogous result [8, Theorem 4.7] for which special consideration of the trust-region constraint was required.

References

[1] J. V. Burke, *A robust trust region method for constrained nonlinear programming problems*, SIAM J. Optim., 2 (1992), pp. 324–347.

[2] 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.

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

[4] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust-Region Methods*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[5] 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.

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

[7] 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.

[8] N. I. M. Gould and D. P. Robinson, *A second derivative SQP method: local convergence*, Numerical Analysis Report 08/21, Computational Laboratory, University of Oxford, Oxford, UK, 2008.

[9] ——, *A second derivative SQP method: theoretical issues*, Numerical Analysis Report 08/18, Computational Laboratory, University of Oxford, Oxford, UK, 2008.

[10] D. Q. Mayne and N. Maratos, *A first-order, exact penalty function algorithm for equality constrained optimization problems*, Math. Program., 16 (1979), pp. 303–324.

[11] M. Mongeau and A. Sartenaer, *Automatic decrease of the penalty parameter in exact penalty function methods*, European Journal of Operational Research, 83 (1995), pp. 686–699.

[12] J. L. Morales, J. Nocedal, and Y. Wu, *A sequential quadratic programming algorithm with an additional equality constrained phase*, Tech. Rep. OTC-05, Northwestern University, 2008.

[13] J. F. A. Pantoja and D. Q. Mayne, *Exact penalty functions with simple updating of the penalty parameter*, J. Optim. Theory Appl., 69 (1991), pp. 441–467.

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

[15] M. J. D. Powell, *A fast algorithm for nonlinearly constrained optimization calculations*, in Numerical Analysis, Dundee 1977, G. A. Watson, ed., no. 630 in Lecture Notes in Mathematics, Heidelberg, Berlin, New York, 1978, Springer Verlag, pp. 144–157.

[16] M. Sahba, *Globally convergent algorithm for nonlinearly constrained optimization problems*, J. Optim. Theory Appl., 52 (1987), pp. 291–309.

[17] Y. X. Yuan, *On the convergence of a new trust region algorithm*, Numer. Math., 70 (1995), pp. 515–539.