IMA Journal of Numerical Analysis Page 1 of 22
doi:10.1093/imanum/drq043

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

NICHOLAS I. M. GOULD
Rutherford Appleton Laboratory, Computational Science and Engineering Department, Didcot OX11 0QX, UK
nick.gould@stfc.ac.uk

AND

DANIEL P. ROBINSON*
Mathematical Institute, University of Oxford, 24–29 St. Giles', Oxford OX1 3LB, UK
*Corresponding author: daniel.p.robinson@gmail.com

[Received on 4 December 2009; revised on 2 November 2010]

Gould and Robinson (2010, *SIAM J. Optim.*, **20**, 2023–2048; 2010, *SIAM J. Optim.*, **20**, 2049–2079) introduced a second-derivative sequential quadratic programming method (S2QP) for solving nonlinear nonconvex optimization problems. We proved that the method is globally and locally superlinearly convergent under common 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 achieving 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 it is interesting theoretically to verify that the same convergence guarantees hold when no trust-region constraint is used, our motivation is based on the practical behaviour of the algorithm. 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 that a nonmonotone strategy is incorporated.

Keywords: nonlinear programming; nonlinear inequality constraints; sequential quadratic programming; ℓ_1-penalty function; nonsmooth optimization.

1. Introduction

In Gould & Robinson (2010a,b) we presented a second-derivative sequential quadratic programming (S2QP)—a sequential inequality/equality constrained quadratic programming algorithm (an SIQP/SEQP 'hybrid') for solving the problem

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

where the constraint vector $c(x)\colon \mathbb{R}^n \to \mathbb{R}^m$ and the objective function $f(x)\colon \mathbb{R}^n \to \mathbb{R}$ are assumed to be twice continuously differentiable, σ is a positive scalar known as the penalty parameter and we have used the notation $[v]^- = \min(0, v)$ for a generic vector v (the minimum is understood to be componentwise).

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

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \quad f(x) \quad \text{subject to } c(x) \geqslant 0 \tag{NP}$$

(see Pietrzykowski (1969) and Conn & Pietrzykowski (1977) 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 defined from at most two components. The first component is the predictor step (Gould & Robinson, 2010a, Section 2.1). It is defined as the unique minimizer of a strictly *convex* quadratic approximation to ϕ subject to a trust-region constraint. To ensure global convergence the predictor step may be scaled by performing a trivial one-dimensional minimization of a second-order approximation to ϕ, resulting in a so-called Cauchy step (Gould & Robinson, 2010a, Section 2.2). Therefore the predictor step affects global efficiency since a 'better' predictor step will generally result in a better Cauchy step. In the neighbourhood of a solution, however, this distinction becomes less important provided that we compute a second (optional) component, referred to as an accelerator step, whose responsibility is to drive fast local convergence of the algorithm. If an accelerator step is computed from any of the subproblems considered in Gould & Robinson (2010a, Section 2.3), then the iterates converge superlinearly (under common assumptions; Gould & Robinson, 2010b, Theorems 3.12 and 3.14). In the special case that the accelerator step is computed from subproblem (EQP) (Gould & Robinson, 2010a, Section 2.3.2), the proof requires that the predictor step correctly identifies the set of constraints that are 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 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 Gould & Robinson (2010a,b)) 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. It is interesting theoretically to verify that the same convergence guarantees hold when no trust-region constraint is used, 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 *et al.* (2008), and a detailed comparison is given within the conclusions of Section 6.

In Section 2 we formally state and describe the modified sequential quadratic programming (SQP) method, while in Section 3 we prove that it is both globally and locally superlinearly convergent. In Section 4 we observe that, with essentially the same theory, we may prove global convergence of our method if we include a fixed predictor step trust-region constraint. In Section 5 we examine three problems from the Hock–Schittkowski test suite (Hock & Schittkowski, 1981) on which our algorithm exhibits favourable, unfavourable and typical behaviour. We conclude by giving final comments in Section 6. Before proceeding, however, we list essential notation.

1.1 *Notation*

We let $e \in \mathbb{R}^m$ denote the vector of all ones, $g(x)$ be the gradient of $f(x)$ and $\nabla_{xx} f(x)$ its (symmetric) Hessian. The matrix $\nabla_{xx} c_j(x)$ is the Hessian of $c_j(x)$, and $J(x)$ is the $m \times n$ Jacobian matrix of the constraints with ith row $\nabla c_i(x)^{\mathrm{T}}$. For convenience, we use the notation $f_k = f(x_k)$, $c_k = c(x_k)$, $g_k = g(x_k)$ and $J_k = J(x_k)$ for a given iterate x_k. The Lagrangian function associated with (NP) is $\mathcal{L}(x, y) = f(x) - y^{\mathrm{T}} c(x)$. The Hessian of the Lagrangian with respect to x is $\nabla_{xx}\mathcal{L}(x, y) = \nabla_{xx} f(x) - \sum_{j=1}^{m} y_j \nabla_{xx} c_j(x)$.

For a general vector v the notation $[v]^- = \min(0, v)$ is used, where the minimum is understood to be componentwise. 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, then we often write $V_{\mathcal{S}}(x)$ instead of $[V(x)]_{\mathcal{S}}$.

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 Gould & Robinson (2010a,b) that uses the new predictor step subproblem. The algorithm is given as Algorithm 2.1 on page 5.

We first evaluate the problem functions at the current point (x_k, y_k). Next we approximate $\nabla_{xx}\mathcal{L}(x_k, y_k)$ with a symmetric positive-definite matrix B_k (Gould & Robinson, 2010b, Section 4.1) and form the predictor step subproblem

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

By introducing elastic variables (Gill *et al.*, 2005) we may solve the equivalent strictly convex quadratic programming problem

$$\underset{s \in \mathbb{R}^n, v \in \mathbb{R}^m}{\text{minimize}} \quad f_k + g_k^{\mathrm{T}} s + \frac{1}{2} s^{\mathrm{T}} B_k s + \sigma e^{\mathrm{T}} v \quad \text{subject to } c_k + J_k s + v \geqslant 0, \quad v \geqslant 0 \tag{2.2}$$

for the predictor step s_k^{P}. We let y_k^{P} denote an optimal multiplier vector associated with the affine constraint $c_k + J_k s + v \geqslant 0$. Next we define H_k to be any symmetric approximation to $\nabla_{xx}\mathcal{L}(x_k, y_k^{\mathrm{P}})$, but for the local convergence results given in Section 3 we choose $H_k \equiv \nabla_{xx}\mathcal{L}(x_k, y_k^{\mathrm{P}})$. We note that all the results of this paper still hold if y_k^{P} is replaced by any first-order multiplier estimate y_k^{F} such that $y_k^{\mathrm{F}} - y^* = \mathrm{O}(\|x_k - x^*\|_2)$ and $[y_k^{\mathrm{F}}]_{\mathcal{I}} = 0$, where (x^*, y^*) is a local solution to problem (NP) and $\mathcal{I} \stackrel{\text{def}}{=} \{i : c_i(x^*) > 0\}$. Once H_k is defined, we define the Cauchy step $s_k^{\mathrm{CP}} = \alpha_k s_k^{\mathrm{P}}$, where α_k is the solution to

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

and

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

is the *faithful* model of ϕ. We emphasize that the predictor step computation (2.2), in contrast to Gould & Robinson (2010a,b), 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 Gould & Robinson (2010a,b), the Cauchy step $s_k^{\rm CP}$ may now have a *larger* infinity-norm than the predictor step but will always satisfy

$$\|s_k^{\rm CP}\|_\infty \leqslant \Delta_k^{\rm P}. \tag{2.5}$$

The next step is to compute the change in the faithful model at the Cauchy step, which is given by $\Delta M_k^{\rm H}(s_k^{\rm CP})$, where $\Delta M_k^{\rm H}(s) \stackrel{\rm def}{=} M_k^{\rm H}(0) - M_k^{\rm H}(s)$. We then have the option of computing an accelerator step $s_k^{\rm A}$ as the solution of any of the subproblems discussed in Gould & Robinson (2010a, Section 2.3). In particular, if we compute an accelerator step as the solution of the *equality* constrained quadratic program

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

where $\mathcal{A}(s_k^{\rm P}) = \{i\colon [c_k + J_k s_k^{\rm P}]_i \leqslant 0\}$, $\bar{f}_k^{\rm P} = f_k + g_k^{\rm T} s_k^{\rm P} + \frac{1}{2} s_k^{\rm P\,T} H_k s_k^{\rm P}$ and $\Delta_k^{\rm A} > 0$ is the accelerator trust-region radius, then we define the trial step s_k as

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

for some predefined constant $0 < \eta \leqslant 1$ (independent of k). Otherwise, if we compute $s_k^{\rm A}$ from the explicitly *inequality* constrained quadratic program

$$\begin{aligned} \underset{s\in\mathbb{R}^n}{\text{minimize}} \quad & \bar{f}_k^{\rm CP} + (g_k + H_k s_k^{\rm CP})^{\rm T} s + \frac{1}{2} s^{\rm T} H_k s + \sigma \|[c_k + J_k(s_k^{\rm CP} + s)]_{\mathcal{V}_k}^-\|_1 \\ \text{subject to} \quad & [c_k + J_k(s_k^{\rm CP} + s)]_{\mathcal{S}_k} \geqslant 0, \\ & (g_k + H_k s_k^{\rm CP} + \sigma J_k^{\rm T} z_k)^{\rm T} s \leqslant 0, \quad \|s\|_\infty \leqslant \Delta_k^{\rm A}, \end{aligned} \tag{EIQP}$$

where

$$[z_k]_i = \begin{cases} -1 & \text{if } i \in \mathcal{V}_k, \\ 0 & \text{if } i \in \mathcal{S}_k, \end{cases} \tag{2.7}$$

$\mathcal{V}_k = \{i\colon [c_k + J_k s_k^{\rm CP}]_i < 0\}$, $\mathcal{S}_k = \{i\colon [c_k + J_k s_k^{\rm CP}]_i \geqslant 0\}$, $\bar{f}_k^{\rm CP} = f_k + g_k^{\rm T} s_k^{\rm CP} + \frac{1}{2} s_k^{\rm CP\,T} H_k s_k^{\rm CP}$ and $(g_k + H_k s_k^{\rm CP} + \sigma_k J_k^{\rm T} z_k)^{\rm T} s \leqslant 0$ is the so-called 'descent constraint', then we define the trial step as

$$s_k = s_k^{\rm CP} + s_k^{\rm A}. \tag{2.8}$$

Note that Gould & Robinson (2010a, Section 2.3) ensures that in both cases the resulting trial step s_k will satisfy

$$\Delta M_k^{\rm H}(s_k) \geqslant \eta \Delta M_k^{\rm H}(s_k^{\rm CP}) \geqslant 0. \tag{2.9}$$

We also note that if $|\mathcal{A}(s_k^{\rm P})| \geqslant n$, then subproblem (EQP) will generally result in the unproductive solution $s_k^{\rm A} = 0$. To prevent this unwanted outcome various alternative subproblems and strategies may be utilized. We do not consider any alternatives in this paper since subproblem (EQP) is sufficient for

proving local superlinear convergence under certain assumptions (see Theorems 3.10 and 3.11). In any case, once the full step is defined, we evaluate $\phi(x_k + s_k)$ and $\Delta M_k^{\mathrm{H}}(s_k)$.

The strategy for updating the trust-region radii and for accepting or rejecting candidate steps is similar to traditional methods (Conn *et al.*, 2000) 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 predefined constant following a successful iteration (to ensure fast asymptotic convergence) and update the accelerator trust-region radius. More precisely, if the ratio satisfies $r_k \geqslant \eta_{\mathrm{VS}}$ for some 'very successful' parameter $\eta_{\mathrm{VS}} \in (0, 1)$, then we believe that the model is a very accurate representation of the merit function within the current trust region, and therefore we increase the predictor trust-region radius by an expansion factor $\eta_{\mathrm{e}} > 1$ with the belief that the current trust-region radius may be overly restrictive. If the ratio satisfies $\eta_{\mathrm{S}} \leqslant r_k < \eta_{\mathrm{VS}}$ for a 'successful' parameter $\eta_{\mathrm{S}} \in (0, \eta_{\mathrm{VS}}]$, then we believe that the model is sufficiently accurate and keep the current predictor trust-region radius with the possibility of increasing it only to satisfy $\Delta_{k+1}^{\mathrm{p}} \geqslant \Delta_{\mathrm{RESET}}$ for some predefined 'reset' value $\Delta_{\mathrm{RESET}} > 0$. 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 that *max_fails* > 0) with the hope that the next iterate will make progress, and we say that a 'nonmonotone phase' has been entered. If we enter a nonmonotone phase, then 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, that is, 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, and this allows us to prove global convergence in Section 3. If it does not make sufficient progress, then the algorithm reverts to the best-known point, reduces the predictor trust-region radius by a contraction factor $\eta_{\mathrm{c}} \in (0, 1)$ 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. We will show that Algorithm 2.1 is globally convergent for *max_fails* $\geqslant 0$, but that *max_fails* > 0 is required to achieve fast local convergence (see Theorems 3.4, 3.10 and 3.11). In all cases we define the accelerator trust-region radius to be a constant multiple of the predictor trust-region radius, although the condition $\Delta_{k+1}^{\mathrm{A}} \leqslant \tau_f \cdot \Delta_{k+1}^{\mathrm{p}}$ for some accelerator trust-region scale factor $\tau_f \geqslant 1$ is also sufficient. For more details on nonmonotone algorithms (sometimes known as nonmonotone 'watchdog' techniques), see Conn *et al.* (2000, Chapters 10.1 and 11.3).

ALGORITHM 2.1. (Nonmonotone algorithm).
Input: (x_0, y_0)
Set parameters $0 < \eta_{\mathrm{S}} \leqslant \eta_{\mathrm{VS}} < 1$, $0 < \Delta_{\mathrm{RESET}} \leqslant \Delta_u$, $0 < \eta < 1$, $\tau_f \geqslant 1$ and $0 \leqslant$ *max_fails* $\in \mathbb{N}$.
Set expansion and contraction factors $0 < \eta_{\mathrm{c}} < 1 < \eta_{\mathrm{e}}$, fail counter *fails* $\leftarrow 0$ and counter $k \leftarrow 0$.
do
Evaluate f_k, g_k, c_k and J_k, and then compute ϕ_k.
Define B_k to be a symmetric positive-definite approximation to $\nabla_{xx}\mathcal{L}(x_k, y_k)$.
Solve problem (2.2) for the predictor step and the multipliers $(s_k^{\mathrm{p}}, y_k^{\mathrm{p}})$.
Define H_k to be a symmetric approximation to $\nabla_{xx}\mathcal{L}(x_k, y_k^{\mathrm{p}})$.
Solve problem (2.3) for s_k^{CP} and compute $\Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}})$.
Optionally, compute an accelerator step and the multipliers $(s_k^{\mathrm{A}}, y_k^{\mathrm{A}})$.
Define a full step s_k that satisfies (2.9) and then evaluate $\phi(x_k + s_k)$ and $\Delta M_k^{\mathrm{H}}(s_k)$.

if $fails = 0$ **then**
  $r_k \leftarrow (\phi(x_k) - \phi(x_k + s_k))/\Delta M_k^{\rm H}(s_k)$ [standard definition]
  If $s_k = s_k^{\rm CP}$, then set *cauchy_tried* $\leftarrow$ **true**; otherwise set *cauchy_tried* $\leftarrow$ **false**.
else
  $r_k \leftarrow (\phi_{\rm R} - \phi(x_k + s_k))/\Delta_{\rm R}^{\rm H}$ [change in ϕ based on point $x_{\rm R}$]
end if
if $r_k \geqslant \eta_{\rm VS}$ **then** [successful]
  $x_{k+1} \leftarrow x_k + s_k,\ y_{k+1} \leftarrow y_k^{\rm A}$ ($y_{k+1} \leftarrow y_k^{\rm P}$ if accelerator step not computed)
  $\Delta_{k+1}^{\rm P} \leftarrow \min(\max(\eta_{\rm e} \cdot \Delta_k^{\rm P}, \Delta_{\rm RESET}), \Delta_u)$
  $fails \leftarrow 0$
else if $r_k \geqslant \eta_{\rm S}$ **then** [successful]
  $x_{k+1} \leftarrow x_k + s_k,\ y_{k+1} \leftarrow y_k^{\rm A}$ ($y_{k+1} \leftarrow y_k^{\rm P}$ if accelerator step not computed)
  $\Delta_{k+1}^{\rm P} \leftarrow \max(\Delta_k^{\rm P}, \Delta_{\rm RESET})$
  $fails \leftarrow 0$
else
  $fails \leftarrow fails + 1$
  if $fails = 1$ **then** [save current point]
    $x_{\rm R} \leftarrow x_k,\ y_{\rm R} \leftarrow y_k,\ \phi_{\rm R} \leftarrow \phi_k,\ s_{\rm R}^{\rm CP} \leftarrow s_k^{\rm CP},\ y_{\rm R}^{\rm P} \leftarrow y_k^{\rm P}$
    $\Delta_{\rm R}^{\rm H} \leftarrow \Delta M_k^{\rm H}(s_k),\ \Delta_{\rm R}^{\rm HCP} \leftarrow \Delta M_k^{\rm H}(s_k^{\rm CP}),\ \Delta_{\rm R}^{\rm P} \leftarrow \Delta_k^{\rm P}$
  end if
  if $\leqslant fails \leqslant max\_fails$ **then** [unsuccessful]
    $x_{k+1} \leftarrow x_k + s_k,\ \Delta_{k+1}^{\rm P} \leftarrow \Delta_k^{\rm P}$
  else
    $fails \leftarrow 0$
    if *cauchy_tried* **then** [revert to saved point]
      $x_{k+1} \leftarrow x_{\rm R},\ y_{k+1} \leftarrow y_{\rm R},\ \Delta_{k+1}^{\rm P} \leftarrow \eta_{\rm c}\Delta_{\rm R}^{\rm P}$
    else
      Evaluate $\phi(x_{\rm R} + s_{\rm R}^{\rm CP})$
      if $(\phi_{\rm R} - \phi(x_{\rm R} + s_{\rm R}^{\rm CP}))/\Delta_{\rm R}^{\rm HCP} \geqslant \eta_{\rm S}$ **then** [successful Cauchy]
        $x_{k+1} \leftarrow x_{\rm R} + s_{\rm R}^{\rm CP},\ y_{k+1} \leftarrow y_{\rm R}^{\rm P},\ \Delta_{k+1}^{\rm P} \leftarrow \max(\Delta_k^{\rm P}, \Delta_{\rm RESET})$
      else [revert to saved point]
        $x_{k+1} \leftarrow x_{\rm R},\ y_{k+1} \leftarrow y_{\rm R},\ \Delta_{k+1}^{\rm P} \leftarrow \eta_{\rm c}\Delta_{\rm R}^{\rm P}$
      end if
    else if
  end if
end if
$\Delta_{k+1}^{\rm A} \leftarrow \tau_f \cdot \Delta_{k+1}^{\rm P}$
$k \leftarrow k + 1$
end do

Now that our entire algorithm has been stated, we proceed to the analysis.

3. Convergence properties

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

LEMMA 3.1 The quantity

$$\chi(x, B) \stackrel{\text{def}}{=} f(x) + \sigma \|[c(x)]^-\|_1 - \min_{s \in \mathbb{R}^n} \left[f(x) + g(x)^{\mathrm{T}} s + \frac{1}{2} s^{\mathrm{T}} B s + \sigma \|[c(x) + J(x)s]^-\|_1 \right] \quad (3.1)$$

is a criticality measure in the following sense:

(i) for a given x and positive-definite matrix B it follows that $0 \leqslant \chi(x, B) < \infty$;

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

(iii) if $\{x_k\} \to x_*$, $\{B_k\}$ is a sequence of matrices such that, for some positive scalars $\lambda^{\mathrm{B}}_{\min}$ and $\lambda^{\mathrm{B}}_{\max}$, we have

$$0 < \lambda^{\mathrm{B}}_{\min} \leqslant \frac{v^{\mathrm{T}} B_k v}{v^{\mathrm{T}} v} \leqslant \lambda^{\mathrm{B}}_{\max} \qquad \text{for all } v \neq 0, \quad (3.2)$$

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 (3.1) is strictly convex and has the value $f(x) + \sigma \|[c(x)]^-\|_1$ at $s = 0$.

We now prove part (ii). The equation $\chi(x, B) = 0$ holds if and only if $s = 0$ is the unique minimizer of the strictly convex minimization problem used in equation (3.1). Optimality of that problem at the point $s = 0$ implies that

$$\text{there exists } w \in \partial \|[c(x)]^-\|_1 \text{ such that } g(x) + \sigma J(x)^{\mathrm{T}} w = 0, \quad (3.3)$$

where $\partial \|[c(x)]^-\|_1$ is the sub-differential of $\partial \|[\cdot]^-\|_1$ at the point $c(x)$ (see Fletcher, 1981, Section 14.2 for more details). This proves part (ii) since condition (3.3) 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 the entries of the lower triangular part of B (in a specified order) as $b(B) \in \mathbb{R}^{n_{\mathrm{B}}}$, where $n_{\mathrm{B}} \stackrel{\text{def}}{=} n(n+1)/2$. 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

$$\lim_{k \in K} x_k = x_* \quad \text{and} \quad \lim_{k \in K} B_k = B_* \quad \text{with } B_* \text{ positive definite}, \quad (3.4)$$

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

$$F(s, x, b(B)) = f(x) + g(x)^{\mathrm{T}} s + \frac{1}{2} s^{\mathrm{T}} B s + \sigma \|[c(x) + J(x)s]^-\|_1, \quad (3.5)$$

so that $F(s, x, b)$ is defined and continuous on $\mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{n_{\mathrm{B}}}$ and convex for each fixed (x, b). It follows from Conn *et al.* (2000, Theorem 3.2.8 using the continuous *point-to-set* map $\mathcal{C}(x, b) = \mathbb{R}^n$) that

$$F_*(x, b) \stackrel{\text{def}}{=} \min_{s \in \mathbb{R}^n} F(s, x, b) \quad (3.6)$$

is continuous, so that we may deduce from (3.1) that $\chi(x, B)$ is also continuous as a function of (x, B). Thus we have

$$\chi(x_*, B_*) = \lim_{k \in K} \chi(x_k, B_k) = 0, \tag{3.7}$$

where the first equality follows by continuity and the second by assumption. Part (ii) and (3.4) 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

$$\Delta M_k^{\mathrm{B}}(s) \stackrel{\mathrm{def}}{=} M_k^{\mathrm{B}}(0) - M_k^{\mathrm{B}}(s). \tag{3.8}$$

LEMMA 3.2 The predictor and Cauchy step satisfy

$$\Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) \geqslant \frac{1}{2} \Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) \min\left(1, \frac{\Delta_k^{\mathrm{P}}}{\|s_k^{\mathrm{P}}\|_\infty}, \frac{\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}})}{n\|B_k - H_k\|_2 \|s_k^{\mathrm{P}}\|_\infty^2}\right). \tag{3.9}$$

Proof. We consider two cases.

Case 1: $s_k^{\mathrm{P}\mathrm{T}} H_k s_k^{\mathrm{P}} \leqslant s_k^{\mathrm{P}\mathrm{T}} B_k s_k^{\mathrm{P}}$.
Subcase 1: $\|s_k^{\mathrm{P}}\|_\infty \leqslant \Delta_k^{\mathrm{P}}$.

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

$$\Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) = M_k^{\mathrm{H}}(0) - M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) \geqslant M_k^{\mathrm{B}}(0) - M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) = \Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}). \tag{3.10}$$

Subcase 2: $\|s_k^{\mathrm{P}}\|_\infty > \Delta_k^{\mathrm{P}}$.
Note that we now have $0 \leqslant \alpha_u < 1$. For a general $0 \leqslant \alpha \leqslant 1$ we have

$$\Delta M_k^{\mathrm{B}}(\alpha s_k^{\mathrm{P}}) = \sigma(\|[c_k]^-\|_1 - \|[c_k + \alpha J_k s_k^{\mathrm{P}}]^-\|_1) - \alpha g_k^{\mathrm{T}} s_k^{\mathrm{P}} - \frac{\alpha^2}{2} s_k^{\mathrm{P}\mathrm{T}} B_k s_k^{\mathrm{P}} \tag{3.11}$$

$$\geqslant \alpha\sigma(\|[c_k]^-\|_1 - \|[c_k + J_k s_k^{\mathrm{P}}]^-\|_1) - \alpha g_k^{\mathrm{T}} s_k^{\mathrm{P}} - \frac{\alpha}{2} s_k^{\mathrm{P}\mathrm{T}} B_k s_k^{\mathrm{P}} \tag{3.12}$$

$$= \alpha \Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}), \tag{3.13}$$

so that

$$\Delta M_k^{\mathrm{B}}(\alpha_u s_k^{\mathrm{P}}) \geqslant \alpha_u \Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}). \tag{3.14}$$

Equation (3.11) follows from the definition of ΔM_k^{B}, (3.12) follows since $0 \leqslant \alpha \leqslant 1$ and from Gould & Robinson (2010a, Lemma 2.3) and (3.13) follows from the definition of ΔM_k^{B}. We may then deduce that

$$\begin{aligned}
\Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) &= M_k^{\mathrm{H}}(0) - M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) \geqslant M_k^{\mathrm{H}}(0) - M_k^{\mathrm{H}}(\alpha_u s_k^{\mathrm{P}}) \quad \text{(using definition of } \Delta M_k^{\mathrm{H}} \text{ and } s_k^{\mathrm{CP}}) \\
&\geqslant M_k^{\mathrm{B}}(0) - M_k^{\mathrm{B}}(\alpha_u s_k^{\mathrm{P}}) \quad \text{(using definitions of } M_k^{\mathrm{B}} \text{ and } M_k^{\mathrm{H}} \text{ and the fact that } s_k^{\mathrm{P}\mathrm{T}} H_k s_k^{\mathrm{P}} \leqslant s_k^{\mathrm{P}\mathrm{T}} B_k s_k^{\mathrm{P}}) \\
&= \Delta M_k^{\mathrm{B}}(\alpha_u s_k^{\mathrm{P}}) \geqslant \alpha_u \Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) \quad \text{(using definition of } \Delta M_k^{\mathrm{B}} \text{ and (3.14))} \\
&= \frac{\Delta_k^{\mathrm{P}}}{\|s_k^{\mathrm{P}}\|_\infty} \Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) \quad \text{(using definition of } \alpha_u\text{).}
\end{aligned} \tag{3.15}$$

Case 2: $s_k^{\mathrm{P}\mathrm{T}} H_k s_k^{\mathrm{P}} > s_k^{\mathrm{P}\mathrm{T}} B_k s_k^{\mathrm{P}} > 0$.
Since $\alpha_k \leqslant \Delta_k^{\mathrm{P}}/\|s_k^{\mathrm{P}}\|_\infty$ by definition and $\alpha_k < 1$ as a consequence of $s_k^{\mathrm{P}\mathrm{T}} H_k s_k^{\mathrm{P}} > s_k^{\mathrm{P}\mathrm{T}} B_k s_k^{\mathrm{P}} > 0$, we conclude that $0 \leqslant \alpha_k \leqslant \min\left(1, \frac{\Delta_k^{\mathrm{P}}}{\|s_k^{\mathrm{P}}\|_\infty}\right) = \min(1, \alpha_u)$. For ease of notation, we define $\alpha_{\min} = \min(1, \alpha_u)$. For all $0 \leqslant \alpha \leqslant \alpha_{\min}$ we must have

$$\Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) \geqslant \Delta M_k^{\mathrm{H}}(\alpha s_k^{\mathrm{P}}) \tag{3.16}$$

$$= \sigma(\|[c_k]^-\|_1 - \|[c_k + \alpha J_k s_k^{\mathrm{P}}]^-\|_1) - \alpha g_k^{\mathrm{T}} s_k^{\mathrm{P}} - \frac{\alpha^2}{2} s_k^{\mathrm{P}\mathrm{T}} H_k s_k^{\mathrm{P}} \tag{3.17}$$

$$= \sigma(\|[c_k]^-\|_1 - \|[c_k + \alpha J_k s_k^{\mathrm{P}}]^-\|_1) - \alpha g_k^{\mathrm{T}} s_k^{\mathrm{P}} - \frac{\alpha^2}{2} s_k^{\mathrm{P}\mathrm{T}} B_k s_k^{\mathrm{P}} + \frac{\alpha^2}{2} s_k^{\mathrm{P}\mathrm{T}} (B_k - H_k) s_k^{\mathrm{P}}. \tag{3.18}$$

Equation (3.16) follows since s_k^{CP} minimizes $M_k^{\mathrm{H}}(\alpha s_k^{\mathrm{P}})$ for $0 \leqslant \alpha \leqslant \alpha_{\min}$, while (3.17) and (3.18) follow from the definition of ΔM_k^{H} and simple algebra. Continuing to bound the change in the faithful model, we have

$$\Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) \geqslant \alpha\sigma(\|[c_k]^-\|_1 - \|[c_k + J_k s_k^{\mathrm{P}}]^-\|_1) - \alpha g_k^{\mathrm{T}} s_k^{\mathrm{P}} - \frac{\alpha}{2} s_k^{\mathrm{P}\mathrm{T}} B_k s_k^{\mathrm{P}} + \frac{\alpha^2}{2} s_k^{\mathrm{P}\mathrm{T}} (B_k - H_k) s_k^{\mathrm{P}} \tag{3.19}$$

$$= \alpha \Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) + \frac{\alpha^2}{2} s_k^{\mathrm{P}\mathrm{T}} (B_k - H_k) s_k^{\mathrm{P}} \tag{3.20}$$

for all $0 \leqslant \alpha \leqslant \alpha_{\min}$. Equation (3.19) follows from equation (3.18), Gould & Robinson (2010a, Lemma 2.3) and the inequality $\alpha^2 \leqslant \alpha$, which holds since $0 \leqslant \alpha \leqslant \alpha_{\min} \leqslant 1$, while equation (3.20) follows from the simplification of equation (3.19) and the definition of $\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}})$.

The previous string of inequalities holds for all $0 \leqslant \alpha \leqslant \alpha_{\min}$. So it must hold for the value of α that maximizes the right-hand side of (3.20). As a function of α, the right-hand side may be written as $q(\alpha) = a\alpha^2 + b\alpha$, where

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

There are two subcases to consider.
Subcase 1: $-b/2a \leqslant \alpha_{\min}$.
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 that

$$q(-b/2a) = \frac{(\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}))^2}{2\left\|s_k^{\mathrm{P}\mathrm{T}}(B_k - H_k)s_k^{\mathrm{P}}\right\|} \geqslant \frac{(\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}))^2}{2\|B_k - H_k\|_2\|s_k^{\mathrm{P}}\|_2^2} \geqslant \frac{(\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}))^2}{2n\|B_k - H_k\|_2\|s_k^{\mathrm{P}}\|_\infty^2}. \tag{3.21}$$

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

$$q(\alpha_{\min}) = a\alpha_{\min}^2 + b\alpha_{\min} = \alpha_{\min}(a\alpha_{\min} + b) > \frac{\alpha_{\min}}{2} b = \frac{\min(1, \alpha_u)}{2} \Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) \tag{3.22}$$

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

If we denote the maximizer of $q(\alpha)$ on the interval $[0, \alpha_{\min}]$ by α^*, then equations (3.21) and (3.22) show that

$$q(\alpha^*) \geqslant \frac{1}{2} \Delta M_k^{\mathrm{B}}(s_k^{\mathrm{p}}) \min\left(1, \frac{\Delta_k^{\mathrm{p}}}{\|s_k^{\mathrm{p}}\|_\infty}, \frac{\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{p}})}{n\|B_k - H_k\|_2 \|s_k^{\mathrm{p}}\|_\infty^2}\right). \tag{3.23}$$

Returning to equation (3.20), we have

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

Combining this with equations (3.10) and (3.15) 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 $\{x_k\}_{k\geqslant 0} \subset \mathcal{B} \subset \mathbb{R}^n$ for some compact set $\mathcal{B}$ and that $\{B_k\}$ is any matrix sequence such that

$$0 < \lambda_{\min}^{\mathrm{B}} \leqslant \frac{v^{\mathrm{T}} B_k v}{v^{\mathrm{T}} v} \quad \text{for all v} \neq 0 \in \mathbb{R}^{\mathrm{n}} \tag{3.24}$$

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

Proof. The first-order optimality conditions for problem (2.2) are

$$\begin{aligned} g_k + B_k s &= J_k^{\mathrm{T}} y, \\ \sigma e &= y + z, \\ \min(c_k + J_k s, y) &= 0, \\ \min(v, z) &= 0, \end{aligned}$$

where y is a multiplier vector for the affine constraint $c_k + J_k s \geqslant 0$, z is the multiplier vector for the simple bound constraint $v \geqslant 0$ and the minimum is taken componentwise. From these conditions we deduce that

$$s_k^{\mathrm{p}} = -B_k^{-1}\left(g_k - J_k^{\mathrm{T}} y_k^{\mathrm{p}}\right) \text{ and } \|y_k^{\mathrm{p}}\|_\infty \leqslant \sigma, \tag{3.25}$$

where y_k^{p} is the vector of Lagrange multipliers. It is now easy to see that the required result holds by using (3.25), standard norm inequalities, (3.24), the fact that $\{x_k\}$ is contained in the compact set $\mathcal{B}$, the continuity of norms and the continuity of g and J. □

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

$$\begin{aligned} \mathcal{S} &= \{k \in \mathbb{N} \mid \text{iterate } k \text{ is labelled either successful or 'successful Cauchy' by Algorithm 2.1}\}; \\ \mathcal{U} &= \{k \in \mathbb{N} \mid \text{iterate } k \text{ is labelled 'unsuccessful' by Algorithm 2.1}\}; \\ \mathcal{R} &= \{k \in \mathbb{N} \mid \text{iterate } k \text{ is labelled 'revert to saved point' by Algorithm 2.1}\}. \end{aligned} \tag{3.26}$$

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. $\{x_k\}_{k\geqslant 0} \subset \mathcal{B} \subset \mathbb{R}^n$ for some compact set $\mathcal{B}$;
2. there exist positive constants $\lambda^{\mathrm{B}}_{\min}$, $\lambda^{\mathrm{B}}_{\max}$ and b_{H} such that $0 \leqslant \lambda^{\mathrm{B}}_{\min} \leqslant (v^{\mathrm{T}} B_k v)/\|v\|_2^2 \leqslant \lambda^{\mathrm{B}}_{\max}$ for all $v \neq 0 \in \mathbb{R}^n$, and $\|H_k\|_2 \leqslant b_{\mathrm{H}}$.

Then, either x_K is a first-order critical point for problem (ℓ_1-σ) for some $K \geqslant 0$ 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 $K \geqslant 0$ then the proof is complete. 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 $\{\Delta^{\mathrm{p}}_k\}$ that converges to zero.
Since Δ^{p}_k 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

$$\lim_{k\in\mathcal{K}} x_k = x_*, \tag{3.27}$$

$$\lim_{k\in\mathcal{K}} \Delta^{\mathrm{p}}_k = 0, \tag{3.28}$$

$$\lim_{k\in\mathcal{K}} \|s^{\mathrm{CP}}_k\|_\infty = 0 \tag{3.29}$$

and

$$r^{\mathrm{CP}}_k < \eta_{\mathrm{S}} \ \text{ for all } k \in \mathcal{K}, \tag{3.30}$$

where

$$r^{\mathrm{CP}}_k \overset{\mathrm{def}}{=} \frac{\phi_k - \phi(x_k + s^{\mathrm{CP}}_k)}{\Delta M^{\mathrm{H}}_k(s^{\mathrm{CP}}_k)}.$$

Subcase 1: There exists a subsequence of $\{\Delta M^{\mathrm{B}}_k(s^{\mathrm{p}}_k)\}_{k\in\mathcal{K}}$ 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 $\chi(x_k, B_k) \equiv \Delta M^{\mathrm{B}}_k(s^{\mathrm{p}}_k)$.
Subcase 2: There does not exist a subsequence of $\{\Delta M^{\mathrm{B}}_k(s^{\mathrm{p}}_k)\}_{k\in\mathcal{K}}$ that converges to zero.
This implies the existence of a positive scalar δ such that

$$\Delta M^{\mathrm{B}}_k(s^{\mathrm{p}}_k) \geqslant \delta > 0 \ \text{ for all } k \in \mathcal{K}. \tag{3.31}$$

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

$$f(x_k + \varepsilon v) = f_k + \varepsilon g_k^{\mathrm{T}} v + \mathrm{o}(\varepsilon) = f_k + \varepsilon g_k^{\mathrm{T}} v + \frac{\varepsilon^2}{2} v^{\mathrm{T}} H_k v + o(\varepsilon) \tag{3.32}$$

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

$$c(x_k + \varepsilon v) = c_k + \varepsilon J_k v + \mathrm{o}(\varepsilon). \tag{3.33}$$

Combining these two equations gives

$$\begin{aligned}
\phi(x_k + \varepsilon v) &= f_k + \varepsilon g_k^{\mathrm{T}} v + \frac{\varepsilon^2}{2} v^{\mathrm{T}} H_k v + \mathrm{o}(\varepsilon) + \sigma \|[c_k + \varepsilon J_k v + \mathrm{o}(\varepsilon)]^-\|_1 \\
&= f_k + \varepsilon g_k^{\mathrm{T}} v + \frac{\varepsilon^2}{2} v^{\mathrm{T}} H_k v + \sigma \|[c_k + \varepsilon J_k v]^-\|_1 + \mathrm{o}(\varepsilon) \\
&= M^{\mathrm{H}}_k(\varepsilon v) + \mathrm{o}(\varepsilon),
\end{aligned} \tag{3.34}$$

where the first equality follows from the definition of ϕ and the Taylor expansions, the second equality follows from the boundedness of $\partial \|[\cdot]^-\|_1$ and the last equality follows from the definition of $M_k^{\mathrm{H}}(\varepsilon v)$. Choosing $v = s_k^{\mathrm{CP}}/\|s_k^{\mathrm{CP}}\|_\infty$ and $\varepsilon = \|s_k^{\mathrm{CP}}\|_\infty$ in equation (3.34) yields

$$\phi(x_k + s_k^{\mathrm{CP}}) = M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) + \mathrm{o}(\|s_k^{\mathrm{CP}}\|_\infty). \tag{3.35}$$

Equation (3.35) then implies the equation

$$r_k^{\mathrm{CP}} = \frac{\phi_k - \phi(x_k + s_k^{\mathrm{CP}})}{\Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}})} = \frac{\Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) + \mathrm{o}(\|s_k^{\mathrm{CP}}\|_\infty)}{\Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}})} = 1 + \frac{\mathrm{o}(\|s_k^{\mathrm{CP}}\|_\infty)}{\Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}})} \tag{3.36}$$

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

$$\begin{aligned}
\Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}}) &\geqslant \frac{1}{2}\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}}) \min\left(1, \frac{\Delta_k^{\mathrm{P}}}{\|s_k^{\mathrm{P}}\|_\infty}, \frac{\Delta M_k^{\mathrm{B}}(s_k^{\mathrm{P}})}{n\|B_k - H_k\|_2 \|s_k^{\mathrm{P}}\|_\infty^2}\right) \\
&\geqslant \frac{\delta}{2} \min\left(1, \frac{\Delta_k^{\mathrm{P}}}{\kappa_{\mathrm{pred}}}, \frac{\delta}{n(\lambda_{\max}^{\mathrm{B}} + b_H)\kappa_{\mathrm{pred}}^2}\right) \\
&= \frac{\delta}{2\kappa_{\mathrm{pred}}}\Delta_k^{\mathrm{P}}
\end{aligned} \tag{3.37}$$

Where the first inequality follows from Lemma 3.2, the second inequality follows from (3.31), assumption 2 of this theorem, and Lemma 3.3, and the final equality follows from (3.28).

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

$$\begin{aligned}
\left|\frac{\mathrm{o}(\|s_k^{\mathrm{CP}}\|_\infty)}{\Delta M_k^{\mathrm{H}}(s_k^{\mathrm{CP}})}\right| &\leqslant \frac{2\kappa_{\mathrm{pred}} z_k \|s_k^{\mathrm{CP}}\|_\infty}{\delta \Delta_k^{\mathrm{P}}} \quad \text{(using (3.37) and definition of 'o')} \\
&\leqslant \frac{2\kappa_{\mathrm{pred}} z_k \Delta_k^{\mathrm{P}}}{\delta \Delta_k^{\mathrm{P}}} = \frac{2\kappa_{\mathrm{pred}}}{\delta} z_k \quad \text{(using definition of Cauchy step and simplifying)}
\end{aligned} \tag{3.38}$$

and where the subsequence $\{z_k\}_{\mathcal{K}}$ converges to zero. It then follows from (3.36) and (3.38) that

$$r_k^{\mathrm{CP}} = 1 + \mathrm{o}(1) \quad \text{for } k \in \mathcal{K}. \tag{3.39}$$

This is a contradiction since this implies that, for $k \in \mathcal{K}$ sufficiently large, the identity $r_k^{\mathrm{CP}} > \eta_{\mathrm{S}}$ holds, which violates equation (3.30). Thus Subcase 2 cannot 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 $\{\Delta_k^{\mathrm{P}}\}$ that converges to zero.

An examination of the algorithm shows that this implies the existence of a positive number δ and an infinite subsequence $\mathcal{K}_{\mathcal{S}} \subseteq \mathcal{S}$ (recall the definition of $\mathcal{S}$ given by (3.26)) such that

$$\lim_{k \in \mathcal{K}_{\mathcal{S}}} x_{k-l(k)} = x_*, \tag{3.40}$$

$$\Delta_k^{\mathrm{P}} \geqslant \delta > 0 \quad \text{for all} \quad k, \tag{3.41}$$

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

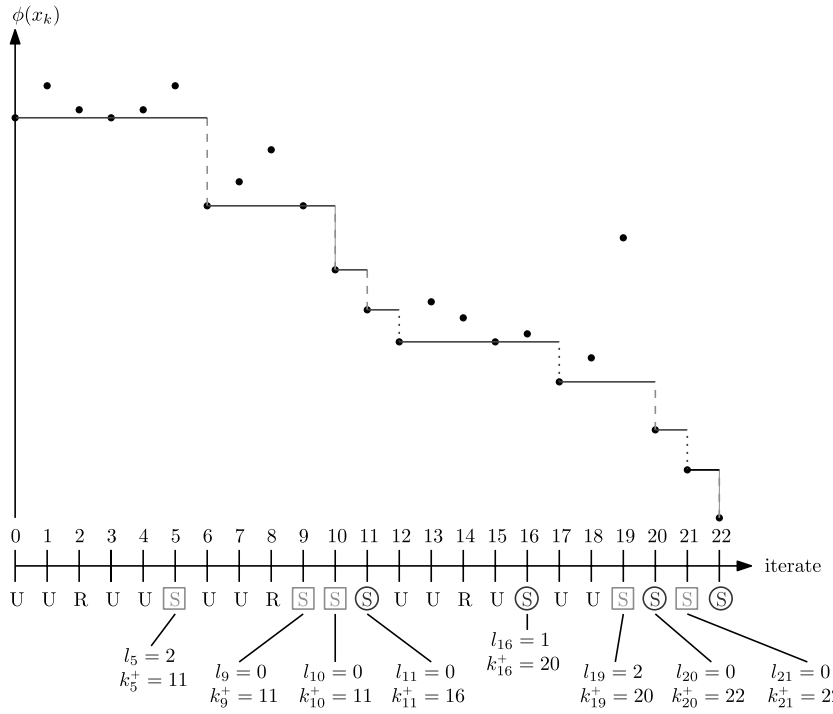

FIG. 1. Illustration of the quantities used in Case 2 of Theorem 3.4 assuming that *max_fails* $= 2$. The x-axis represents the iterate and the y-axis represents the value of the merit function ϕ for a given iterate. The labels S, U and R below the x-axis indicate whether that iterate belongs to the indexing set $\mathcal{S}$, $\mathcal{U}$ or $\mathcal{R}$ (see (3.26)), respectively. Every S is enclosed in either a circle or a 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 horizontal solid lines indicate the least value of ϕ accepted as a successful iterate up until that point. The length of a dotted vertical line or a 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 introduced the notation $l_k = l(x_k)$ and $k_l^+ = k^+(x_l)$.

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 Fig. 1). Note that this implies that

$$\phi(x_{k+1}) \geqslant \phi(x_{k^+(k)-l(k^+(k))}) \quad \text{for all } k \in \mathcal{S} \quad \text{and} \quad \lim_{k\in\mathcal{S}} k^+(k) = \infty. \tag{3.42}$$

For the remainder of this proof we write ΔM^{H} instead of ΔM_k^{H}, and ΔM^{B} instead of ΔM_k^{B}—the 'missing' argument is always assumed to be the subscript of the step, that is, $\Delta M^{\mathrm{B}}(s_j)$ means $\Delta M_j^{\mathrm{B}}(s_j)$ 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 (2.9) that

$$\phi(x_{k-l(k)}) - \phi(x_{k+1}) \geqslant \eta_{\mathrm{S}} \Delta M^{\mathrm{H}}(s_{k-l(k)}) \geqslant \eta\eta_{\mathrm{S}} \Delta M^{\mathrm{H}}(s^{\mathrm{CP}}_{k-l(k)}). \tag{3.43}$$

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

$$\phi(x_{k-l(k)}) - \phi(x_{k+1}) \geqslant \eta_{\mathrm{S}} \Delta M^{\mathrm{H}}(s^{\mathrm{CP}}_{k-l(k)}). \tag{3.44}$$

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

$$\phi(x_{k-l(k)}) - \phi(x_{k+1}) \geqslant \eta\eta_{\mathrm{S}} \Delta M^{\mathrm{H}}(s^{\mathrm{CP}}_{k-l(k)}) \quad \text{for all } k \in \mathcal{K}_{\mathcal{S}}. \tag{3.45}$$

Equation (3.45), Lemmas 3.2 and 3.3, (3.41) and assumption 2 of this theorem imply that

$$\phi(x_{k-l(k)}) - \phi(x_{k+1}) \geqslant \frac{\eta\eta_{\mathrm{S}}}{2} \Delta M^{\mathrm{B}}(s^{\mathrm{P}}_{k-l(k)}) \min\left(1, \frac{\delta}{\kappa_{\mathrm{pred}}}, \frac{\Delta M^{\mathrm{B}}(s^{\mathrm{P}}_{k-l(k)})}{(\lambda^{\mathrm{B}}_{\max} + b_{\mathrm{H}})\kappa^2_{\mathrm{pred}}}\right) \tag{3.46}$$

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

$$\sum_{k \in \mathcal{K}_{\mathcal{S}}, k \leqslant \bar{k}} \frac{\eta\eta_{\mathrm{S}}}{2} \Delta M^{\mathrm{B}}(s^{\mathrm{P}}_{k-l(k)}) \min\left(1, \frac{\delta}{\kappa_{\mathrm{pred}}}, \frac{\Delta M^{\mathrm{B}}(s^{\mathrm{P}}_{k-l(k)})}{(\lambda^{\mathrm{B}}_{\max} + b_{\mathrm{H}})\kappa^2_{\mathrm{pred}}}\right) \leqslant \sum_{k \in \mathcal{K}_{\mathcal{S}}, k \leqslant \bar{k}} \phi(x_{k-l(k)}) - \phi(x_{k+1}) \tag{3.47}$$

$$\leqslant \sum_{k \in \mathcal{S}, k \leqslant \bar{k}} \phi(x_{k-l(k)}) - \phi(x_{k+1}) \tag{3.48}$$

$$= \phi(x_0) - \phi(x_{\bar{k}+1}) \tag{3.49}$$

$$\leqslant \phi(x_0) - \phi(x_{k^+(\bar{k})-l(k^+(\bar{k}))}). \tag{3.50}$$

Equation (3.47) follows from (3.46), then (3.48) follows since we are adding more positive terms to the sum, (3.49) follows from the construction of the algorithm and the fact that $\bar{k} \in \mathcal{S}$ by assumption, and (3.50) follows from (3.42). To help the reader understand we note that, for the value $\bar{k} = 21$, the right-hand side of (3.47) is equal to the sum of the lengths of the dotted lines in Fig. 1, while the right-hand side of (3.48) is equal to the sum of the lengths of the dotted lines and the dashed lines. If we now let $\bar{k}$ converge to infinity in the previous string of inequalities and use (3.42) and (3.40), then we may conclude that

$$\sum_{k \in \mathcal{K}_{\mathcal{S}}} \frac{\eta\eta_{\mathrm{S}}}{2} \Delta M^{\mathrm{B}}(s^{\mathrm{P}}_{k-l(k)}) \min\left(1, \frac{\delta}{\kappa_{\mathrm{pred}}}, \frac{\Delta M^{\mathrm{B}}(s^{\mathrm{P}}_{k-l(k)})}{(\lambda^{\mathrm{B}}_{\max} + b_H)\kappa^2_{\mathrm{pred}}}\right) \leqslant \phi(x_0) - \phi(x^*), \tag{3.51}$$

which implies that

$$\lim_{k \in \mathcal{K}_{\mathcal{S}}} \Delta M^{\mathrm{B}}(s^{\mathrm{P}}_{k-l(k)}) = 0 \tag{3.52}$$

because the series on the left-hand side is convergent. Since (3.40) states that $\lim_{k \in \mathcal{K}_{\mathcal{S}}} x_{k-l(k)} = x_*$ and it follows from (3.1) and (3.52) that

$$\lim_{k \in \mathcal{K}_{\mathcal{S}}} \chi(x_{k-l(k)}, B_{k-l(k)}) = \lim_{k \in \mathcal{K}_{\mathcal{S}}} \Delta M^{\mathrm{B}}(s^{\mathrm{P}}_{k-l(k)}) = 0, \tag{3.53}$$

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). The proof is now complete 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 Pietrzykowski (1969) and Conn & Pietrzykowski (1977) 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 (Powell, 1978b; Mayne & Maratos, 1979; Sahba, 1987; Pantoja & Mayne, 1991; Burke, 1992; Mongeau & Sartenaer, 1995; Yuan, 1995; Byrd *et al.*, 2005; Byrd *et al.*, 2008; Gould & Robinson, 2010b). We use the following definitions related to a solution of problem (NP).

DEFINITION 3.5 [First-order Karush-Kuhn-Tucker (KKT) point]. We say that the point (x^*, y^*) is a first-order KKT point for problem (NP) if

$$g(x^*) - J(x^*)^{\mathrm{T}} y^* = 0, \quad c(x^*) \geqslant 0, \quad y^* \geqslant 0 \ \text{ and } \ c(x^*) \cdot y^* = 0. \tag{3.54}$$

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

DEFINITION 3.6 (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 $\lambda^{\mathrm{H}}_{\min} > 0$ such that $s^{\mathrm{T}} \nabla_{xx} \mathcal{L}(x^*, y^*) s \geqslant \lambda^{\mathrm{H}}_{\min} s^{\mathrm{T}} s$ for all s satisfying $J_{\mathcal{A}}(x^*) s = 0$.

DEFINITION 3.7 (Strict complementarity). We say that strict complementarity holds at a KKT point (x^*, y^*) for problem (NP) if $y^*_{\mathcal{A}} > 0$.

DEFINITION 3.8 (Linear independence constraint qualification). We say that the linear independence constraint qualification holds at a KKT point (x^*, y^*) for problem (NP) if the matrix $J_{\mathcal{A}}(x^*)$ has full row rank.

DEFINITION 3.9 We say that the *strong* second-order sufficient conditions hold at a point (x^*, y^*) if it satisfies Definitions 3.5–3.8.

Our local superlinear convergence results require that *max_fails* > 0, that is, that Algorithm 2.1 is truly nonmonotone. This ensures that trial steps will be temporarily accepted in the neighbourhood of a local solution even if the merit function ϕ increases. This approach guarantees that the so-called Maratos effect (Maratos, 1978), which is known to lead to slow local convergence, is avoided.

For our first result we assume that an accelerator step is computed from subproblem (EQP) as given in Section 2 and discussed in Gould & Robinson (2010a, Section 2.3.2). Since this subproblem only defines multipliers y^{A}_k for the constraints whose indices are in the set $\mathcal{A}(s^{\mathrm{p}}_k)$ as defined in Section 2, we form multipliers for problem (NP) by 'scattering' y^{A}_k into the appropriate locations of a zero vector of length m. The following theorem is the same as Theorem 3.12 in Gould & Robinson (2010b).

THEOREM 3.10 ((EQP) local convergence result). Let (x^*, y^*) be a minimizer for problem (NP) that satisfies the strong second-order sufficient conditions as given by Definition 3.9. Let the assumptions of Theorem 3.4 hold and suppose that $\sigma > \|y^*\|_\infty$, the accelerator step is computed from subproblem (EQP) with the choice $H_k \equiv \nabla_{xx} \mathcal{L}(x_k, y^{\mathrm{p}}_k)$, and *max_fails* $\geqslant 1$ in Algorithm 2.1. It follows that there exists an open neighbourhood 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 neighbourhood, 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 an R-superlinear rate, respectively. Moreover, if $\nabla_{xx} \mathcal{L}(x, y)$ is Lipschitz continuous in a neighbourhood of (x^*, y^*), then they convergence at a Q-quadratic and an R-quadratic rate, respectively.

Proof. The only reason why the proof of Theorem 3.12 in Gould & Robinson (2010b) would not apply here 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 (EIQP) as given in Section 2 and described in Gould & Robinson (2010a, Section 2.3.1).

THEOREM 3.11 ((EIQP) local convergence result). Let (x^*, y^*) be a minimizer for problem (NP) that satisfies the strong second-order sufficient conditions as given by Definition 3.9. Let the assumptions of Theorem 3.4 hold and assume that $\sigma > \|y^*\|_\infty$, the accelerator step is computed from subproblem (EIQP) with the choice $H_k \equiv \nabla_{xx}\mathcal{L}(x_k, y_k^{\mathrm{p}})$, and *max_fails* $\geqslant 1$ in Algorithm 2.1. It follows that there exists an open neighbourhood 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 neighbourhood, then the sequences of iterates $\{x_k\}$ and $\{y_k\}$ converge to x^* and y^* at a Q-superlinear and an R-superlinear rate, respectively. Moreover, if $\nabla_{xx}\mathcal{L}(x, y)$ is Lipschitz continuous in a neighbourhood of (x^*, y^*), then they converge at a Q-quadratic and an R-quadratic rate, respectively.

The proof of Theorem 3.11 follows the proof of Theorem 3.14 in Gould & Robinson (2010b) exactly.

4. Inclusion of a trust region

The 'trust-region-free' method described in Section 2 depends on the sequence $\{B_k\}$ of positive-definite matrices. If this sequence is chosen well, then the algorithm typically performs very well in practice. However, we recognize that by removing the trust-region constraint altogether, we have removed the 'safety net' enjoyed by trust-region methods. If the sequence $\{B_k\}$ is defined poorly (or degeneracy exists), then tests indicate that Algorithm 2.1 without a predictor trust-region constraint may occasionally perform poorly when compared with its counterpart (Gould & Robinson, 2010a,b) that includes one. (It also depends on precisely how one implements the nonmonotone behaviour of the algorithm.) Therefore, in this section we examine the inclusion of a trust-region constraint of the form

$$\|s\|_\infty \leqslant \max(\Delta_{\mathrm{LB}}, \Delta_k^{\mathrm{p}}) \tag{4.1}$$

in the computation of the predictor step, where $\Delta_{\mathrm{LB}} > 0$ is a lower bound on the size of the trust-region radius. Note that this is counterintuitive since trust-region methods may need to make the trust-region radius relatively small to make progress. The predictor subproblem becomes

$$\underset{s\in\mathbb{R}^n, v\in\mathbb{R}^m}{\text{minimize}} \quad f_k + g_k^{\mathrm{T}}s + \frac{1}{2}s^{\mathrm{T}}B_k s + \sigma e^{\mathrm{T}}v \quad \text{subject to} \quad c_k + J_k s + v \geqslant 0,\ v \geqslant 0,\ \|s\|_\infty \leqslant \max(\Delta_{\mathrm{LB}}, \Delta_k^{\mathrm{p}}). \tag{4.2}$$

Provided that Δ_{LB} is chosen to be reasonably large, say $\Delta_{\mathrm{LB}} \geqslant 1$, then this new subproblem shares the advantages of Algorithm 2.1 and that described in Gould & Robinson (2010a,b), that is, (i) the 'large' trust region will not interfere with optimal active set identification near a solution, (ii) predictor steps may be 'recycled' during a sequence of failed iterations, preventing excessive computation, and (iii) the trust-region constraint prevents excessively large steps that may occasionally result from poor choices of B_k, degeneracy, or otherwise.

Two observations are needed for the proofs of Theorems 3.4, 3.10 and 3.11 to hold when Algorithm 2.1 uses subproblem (4.2) in place of (2.2). First, definition (3.1) of the criticality measure $\chi(x, B)$ must be modified to include the new trust-region constraint. Second, Lemma 3.3 still holds since a predictor step computed from subproblem (4.2) cannot be larger than that computed from subproblem (2.2). The proofs of Theorems 3.4, 3.10 and 3.11 now follow exactly as in Section 3. Thus Algorithm 2.1 with

a predictor step computed from subproblem (4.2) is both globally and superlinearly convergent when an accelerator step is computed from either subproblem (EQP) or (EIQP).

5. Numerical experiments

A numerical comparison of Algorithm 2.1 with and without a predictor step trust-region constraint is difficult since computational nonlinear programming is sensitive to choices of the initial point and adjustments of control parameters. These often lead to a dramatically different sequence of iterates. Therefore we do not expect either variant of our algorithm to be uniformly better. However, this paper does provide us the freedom to remove the predictor step trust-region constraint as we see appropriate.

In this section we highlight our preliminary numerical experience by considering three optimization problems from the Hock–Schittkowski test set (Hock & Schittkowski, 1981). These problems have been chosen carefully to display certain interesting behaviour. The first example shows that some problems may be solved much more efficiently by removing the trust-region constraint in the predictor step computation. By contrast, the second example shows that no trust-region constraint in the predictor step combined with a poor choice of the positive-definite matrix B_k may lead to bad performance. However, this poor performance may be rectified by using an artificial trust-region constraint simply as a safety net as described in Section 4. Finally, we give an example that exhibits, in our experience, the *typical* behaviour of Algorithm 2.1 when executed with and without a predictor step trust-region constraint.

The following parameters were used in all cases: sufficient model decrease factor $\eta = 0.5$, successful/very successful tolerances $\eta_S = 0.01$ and $\eta_{VS} = 0.7$, maximum predictor trust-region radius $\Delta_u = 1000$, trust-region reset radius $\Delta_{\text{RESET}} = 10^{-4}$, accelerator trust-region scale factor $\tau_f = 3.0$, number of nonmonotone steps allowed *max_fails* $= 1$, trust-region contraction and expansion factors $\eta_c = 0.1$ and $\eta_e = 5.0$, and an initial penalty parameter of $\sigma = 1.0$. All runs are terminated when the primal feasibility, dual feasibility and complementarity violations are all less than10^{-5}.

Since our examples are of small dimension, we chose to update the positive-definite matrix B_k in the predictor subproblem (2.2) by using the Broyden-Fletcher-Goldfarb-Shanno update (Nocedal & Wright, 1999). To perform this update we used the trial step s_k and $d_k = \nabla_x \mathcal{L}(x_k + s_k, y_{k+1}) - \nabla_x \mathcal{L}(x_k, y_{k+1})$. If these vectors did not result in a sufficiently positive-definite update, then we used the damping technique introduced by Powell (1978a). For simplicity, we chose $B_0 = I$.

We solved problem (EQP) (see Section 2) to compute an accelerator step. Using this subproblem, instead of subproblem (EIQP) (see Section 2), allows us to study how effectively the predictor step identifies the optimal active set.

Finally, we updated the penalty parameter using the concept of 'steering' as described in Byrd *et al.* (2008). Consequently, to update the penalty parameter we occasionally solved more than one predictor subproblem for a given iterate.

5.1 *Hock–Schittkowski* #85

We first ran Algorithm 2.1 with a predictor trust-region constraint as described in Gould & Robinson (2010a,b). The problem was successfully solved but required 97 function evaluations and 67 gradient evaluations. For comparison, we ran Algorithm 2.1 without the trust-region constraint. This version only needed 13 function evaluations and 12 gradient evaluations.

Careful examination of the output from Algorithm 2.1 with a predictor step trust-region constraint revealed that the trust-region first became active on iterate 8. The remaining 89 iterates alternated between successful and unsuccessful iterates and entered many nonmonotone phases. Moreover, the predictor

TABLE 1 *Output from Algorithm 2.1 (no predictor trust-region constraint) on test problem Hock-Schittkowski #85 with initial point given by the eighth iterate of the algorithm when using a trustregion constraint*

Iter	Penalty	Merit	Primal	Dual	Comp	|Y|	TRpred	#QP
0	1.00E+00	–1.87E+00	8.92E-02	2.44E-03	1.80E-04	3.92E-02	1.00E+02	–
1	1.00E+00	–1.97E+00	2.72E-06	3.15E-02	0.00E+00	0.00E+00	1.00E+02	1
2	1.00E+00	3.82E+00	5.93E+00	6.24E-03	5.34E-03	4.97E-02	1.00E+02	1
3	1.00E+00	–2.21E+00	1.63E-03	1.27E-04	9.54E-05	5.86E-02	9.68E+02	1
4	1.00E+00	–2.22E+00	3.86E-07	2.61E-06	2.32E-08	6.02E-02	9.68E+02	1

steps during successful iterates were constrained by the trust-region constraint, which appeared to impede progress. For verification, we used the eighth iterate from Algorithm 2.1 with a trust-region constraint as a starting value for the same algorithm without a trust-region constraint. The resulting output was as shown in Table 1, where column Iter represents the number of major iterations, Penalty represents the value of the penalty parameter, Merit represents the value of the ℓ_1 merit function ϕ, Primal represents the two-norm of the primal feasibility violation, Dual the two-norm of the dual feasibility violation, Comp represents the two-norm of the complementarity slackness, $|Y|$ represents the infinity-norm of the current multiplier estimate, TRpred represents the value of Δ_k^{p} and #QP represents the number of predictor step quadratic programs solved during that iteration. The output clearly shows that this problem benefits greatly from removing the trust-region constraint from the computation of the predictor step. In particular, if we remove the trust-region constraint on iteration eight, then the algorithm solves the problem with $8 + 4 = 12$ function evaluations instead of the original 97.

5.2 *Hock–Schittkowski #6*

Here, once again, we first ran Algorithm 2.1 with a predictor trust-region constraint. The problem was successfully solved and required 10 function evaluations, 6 gradient evaluations and the computation of 9 predictor steps. (Note that this problem has a single *equality* constraint and thus the two-norm of the complementarity violation is always zero by definition.) Next we removed the trust-region constraint and resolved the problem. This resulted in 120 function evaluations, 65 gradient evaluations and the computation of 65 predictor steps. Thus, for our current implementation the trust-region constraint is vital for solving this problem efficiently. Upon careful inspection of the output, we observed that our algorithm generated very poor approximations B_k that led to relatively large steps p_k. Additionally, the trial steps were poor directions of descent, the resulting Cauchy steps were very small, and progress was slow.

Inefficient descent directions resulting from poor positive-definite approximations B_k may easily be remedied. For example, we may include a stabilizing trust-region constraint such as (4.1) in the predictor step subproblem. This strategy with the choice $\Delta_{\mathrm{LB}} = 5.0$ resulted in the output shown in Table 2.
This solution required 20 function evaluations, 8 gradient evaluations and 7 predictor QP subproblems. For this particular problem we may conclude that the stabilizing trust-region constraint described in Section 4 remedies poor approximations B_k and results in superior performance.

5.3 *Hock–Schittkowski #100*

This example exhibits the ‘average’ performance of Algorithm 2.1 with and without a predictor step trust-region constraint for our current implementation. The resulting output with a predictor trust-region

TABLE 2 *Output from Algorithm 2.1 on test problem Hock-Schittkowski #6 with a stabilized predictor trust-region constraint given by (4.1)*

Iter	Penalty	Merit	Primal	Dual	Comp	|Y|	TRpred	#QP
0	1.00E+00	9.24E+00	4.40E+00	4.40E+00	0.00E+00	0.00E+00	1.00E+02	–
1	1.00E+00	9.24E+00	4.40E+00	1.85E+00	0.00E+00	1.85E-01	7.49E-01	1
2	1.00E+00	9.24E+00	4.40E+00	1.70E+00	0.00E+00	1.70E-01	3.74E-01	–
3	1.00E+00	9.24E+00	4.40E+00	1.63E+00	0.00E+00	1.63E-01	1.87E-01	–
4	1.00E+00	8.40E+00	4.85E+00	1.60E+00	0.00E+00	1.60E-01	3.74E-01	–
5	1.00E+00	8.40E+00	4.85E+00	1.60E+00	0.00E+00	1.60E-01	3.37E-01	1
6	1.00E+00	8.40E+00	4.85E+00	1.72E+00	0.00E+00	1.72E-01	1.68E-01	–
7	1.00E+00	8.28E+00	5.94E+00	1.67E+00	0.00E+00	1.67E-01	3.37E-01	–
8	1.00E+00	8.28E+00	5.94E+00	1.67E+00	0.00E+00	1.67E-01	3.03E-01	1
9	1.00E+00	8.28E+00	5.94E+00	1.73E+00	0.00E+00	1.73E-01	1.52E-01	–
10	1.00E+00	8.28E+00	5.94E+00	1.63E+00	0.00E+00	1.63E-01	7.58E-02	–
11	1.00E+00	8.07E+00	6.39E+00	1.66E+00	0.00E+00	1.58E-01	1.52E-01	–
12	1.00E+00	8.07E+00	6.39E+00	1.79E+00	0.00E+00	1.79E-01	1.36E-01	1
13	1.00E+00	8.07E+00	6.39E+00	1.87E+00	0.00E+00	1.21E-01	6.82E-02	–
14	1.00E+00	7.75E+00	6.64E+00	1.98E+00	0.00E+00	1.18E-01	1.36E-01	–
15	1.00E+00	7.75E+00	6.64E+00	2.08E+00	0.00E+00	2.63E-02	1.23E-01	1
16	1.00E+00	7.75E+00	6.64E+00	2.08E+00	0.00E+00	2.29E-02	6.14E-02	–
17	1.00E+00	7.50E+00	6.84E+00	1.70E+00	0.00E+00	2.26E-02	1.23E-01	–
18	1.00E+00	6.55E+00	6.55E+00	6.53E-05	0.00E+00	4.33E-19	2.17E+00	1
19	1.00E+00	1.07E-08	1.07E-08	1.68E-09	0.00E+00	4.37E-23	2.17E+01	1

TABLE 3 *Output from Algorithm 2.1 on test problem Hock-Schittkowski #100 with a predictor trust-region constraint*

Iter	Penalty	Merit	Primal	Dual	Comp	|Y|	TRpred	#QP
0	1.00E+00	7.14E+02	0.00E+00	1.00E+02	0.00E+00	0.00E+00	1.00E+02	–
1	1.00E+01	4.13E+14	2.42E+04	1.33E+13	1.48E+01	2.61E-01	1.00E+02	2
2	1.00E+01	7.14E+02	0.00E+00	1.00E+02	0.00E+00	0.00E+00	5.77E+00	1
3	1.00E+01	1.25E+03	3.40E+01	7.56E+02	0.00E+00	0.00E+00	1.00E+01	1
4	1.00E+01	7.14E+02	0.00E+00	1.00E+02	0.00E+00	0.00E+00	9.04E-01	1
5	1.00E+01	3.46E+03	3.77E+01	5.28E+03	2.94E+01	1.05E+00	1.00E+01	1
6	1.00E+01	7.14E+02	0.00E+00	1.00E+02	0.00E+00	0.00E+00	4.52E-01	1
7	1.00E+01	8.16E+02	8.87E+00	6.60E+01	9.34E+00	1.05E+00	1.00E+01	1
8	1.00E+01	7.14E+02	0.00E+00	1.00E+02	0.00E+00	0.00E+00	2.26E-01	1
9	1.00E+01	7.03E+02	2.11E+00	8.67E+00	2.30E+00	1.09E+00	4.52E-01	1
10	1.00E+01	7.00E+02	1.12E+00	3.27E+01	5.65E-01	1.15E+00	9.04E-01	1
11	1.00E+01	6.82E+02	2.13E-02	9.44E+00	2.41E-02	1.13E+00	9.04E+00	1
12	1.00E+01	6.81E+02	7.64E-03	2.07E+00	2.82E-03	1.14E+00	9.04E+01	1
13	1.00E+01	6.81E+02	8.27E-05	2.08E-01	9.42E-05	1.14E+00	9.04E+02	1
14	1.00E+01	6.81E+02	1.49E-06	2.89E-03	1.25E-06	1.14E+00	9.04E+03	1
15	1.00E+01	6.81E+02	3.27E-10	5.82E-07	3.72E-10	1.14E+00	1.00E+04	1

constraint is as shown in Table 3, and took 16 function evaluations, 12 gradient evaluations and computed 16 predictor steps. The output without a predictor trust-region constraint is as shown in Table 4, and took 22 function evaluations, 12 gradient evaluations and computed 12 predictor steps. Therefore, by

TABLE 4 *Output from Algorithm 2.1 on test problem Hock-Schittkowski #100 without a predictor trust-region constraint*

Iter	Penalty	Merit	Primal	Dual	Comp	|Y|	TRpred	#QP
0	1.00E+00	7.14E+02	0.00E+00	1.00E+02	0.00E+00	0.00E+00	1.00E+02	–
1	1.00E+01	4.13E+14	2.42E+04	1.33E+13	1.48E+01	2.61E-01	1.00E+02	2
2	1.00E+01	7.14E+02	0.00E+00	1.00E+02	0.00E+00	0.00E+00	5.77E+00	1
3	1.00E+01	1.09E+03	3.30E+01	1.04E+02	0.00E+00	0.00E+00	1.00E+01	–
4	1.00E+01	7.14E+02	0.00E+00	1.00E+02	0.00E+00	0.00E+00	1.13E+00	1
5	1.00E+01	7.50E+02	7.07E+00	1.02E+02	0.00E+00	0.00E+00	1.00E+01	–
6	1.00E+01	7.14E+02	0.00E+00	1.00E+02	0.00E+00	0.00E+00	5.67E-01	1
7	1.00E+01	7.01E+02	1.48E+00	2.07E+01	4.80E+00	1.07E+00	1.13E+00	–
8	1.00E+01	7.01E+02	1.48E+00	7.62E+00	2.60E+00	1.02E+00	4.22E-01	1
9	1.00E+01	7.01E+02	1.48E+00	1.03E+01	1.52E+00	1.03E+00	2.11E-01	–
10	1.00E+01	7.01E+02	1.48E+00	9.88E+00	1.61E+00	1.09E+00	1.05E-01	–
11	1.00E+01	6.97E+02	1.57E+00	7.78E+00	1.68E+00	1.07E+00	2.11E-01	–
12	1.00E+01	6.97E+02	1.57E+00	6.12E+00	1.73E+00	1.09E+00	1.90E-01	1
13	1.00E+01	6.97E+02	1.57E+00	7.12E+00	1.81E+00	1.15E+00	9.49E-02	–
14	1.00E+01	6.97E+02	1.57E+00	6.97E+00	1.84E+00	1.17E+00	4.74E-02	–
15	1.00E+01	6.95E+02	1.52E+00	6.04E+00	1.73E+00	1.14E+00	2.37E-01	–
16	1.00E+01	6.95E+02	1.52E+00	4.89E+00	1.66E+00	1.09E+00	1.19E-01	1
17	1.00E+01	6.91E+02	1.04E+00	1.87E+00	1.19E+00	1.14E+00	2.37E-01	–
18	1.00E+01	6.84E+02	3.70E-01	8.18E-01	1.36E-01	1.14E+00	4.74E-01	1
19	1.00E+01	6.81E+02	3.47E-02	1.14E-01	1.28E-02	1.14E+00	4.74E+00	1
20	1.00E+01	6.81E+02	1.24E-03	3.71E-03	4.56E-04	1.14E+00	4.74E+01	1
21	1.00E+01	6.81E+02	1.50E-06	4.31E-06	5.55E-07	1.14E+00	4.74E+02	1

removing the predictor trust-region constraint, we have increased the number of function evaluations but decreased the number of predictor step quadratic programs solved. We believe this trend accurately characterizes the 'average' performance of Algorithm 2.1 with and without a predictor step trust-region constraint.

6. Conclusions and future work

In Gould & Robinson (2010a,b) 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 always 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 the 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 conclude by comparing our method with the work by Morales *et al.* (2008). Roughly, they computed a predictor step s_k^{P} (without a trust-region constraint) followed by an accelerator step s_k^{A} defined as the solution to the equality constrained subproblem (EQP) defined in Section 2 and discussed in Gould & Robinson (2010a, Section 2.3.2). They then reduced the ℓ_1-merit function by performing a line search

along the 'bent' path defined by the steps s_k^{P} and s_k^{A}. 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, while 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 analysed 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 the equality constrained subproblem (EQP) for computing an accelerator step. Following the rejection of a trial step, Morales *et al.* (2008) performed a line search in the direction of the predictor step. We, on the other hand, performed 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.

Acknowledgment

The authors thank two anonymous referees whose comments and suggestions greatly improved the presentation.

Funding

Engineering and Physical Sciences Research Council (EP/E053351/1, EP/F005369/1).

References

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

BYRD, R. H., GOULD, N. I. M., NOCEDAL, J. & WALTZ, R. A. (2005) On the convergence of successive linear–quadratic programming algorithms. *SIAM J. Optim.*, **16**, 471–489.

BYRD, R. H., NOCEDAL, J. & WALTZ, R. A. (2008) Steering exact penalty methods for nonlinear programming. *Optim. Methods Softw.*, **23**, 197–213.

CONN, A. R., GOULD, N. I. M. & TOINT, PH. L. (2000) *Trust-Region Methods*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM).

CONN, A. R. & PIETRZYKOWSKI, T. (1977) A penalty function method converging directly to a constrained optimum. *SIAM J. Numer. Anal.*, **14**, 348–375.

FLETCHER, R. (1981) *Practical Methods of Optimization, Volume 2: Constrained Optimization*. Chichester: Wiley.

GILL, P. E., MURRAY, W. & SAUNDERS, M. A. (2005) SNOPT: an SQP algorithm for large-scale constrained optimization. *SIAM Rev.*, **47**, 99–131.

GOULD, N. I. M. & ROBINSON, D. P. (2010a) A second derivative SQP method: global convergence. *SIAM J. Optim.*, **20**, 2023–2048.

GOULD, N. I. M. & ROBINSON, D. P. (2010b) A second derivative SQP method: local convergence and practical issues. *SIAM J. Optim.*, **20**, 2049–2079.

HOCK, W. & SCHITTKOWSKI, K. (1981) *Test Examples for Nonlinear Programming Codes*. Lecture Notes in Economics and Mathematical Systems, vol. 187. Berlin: Springer.

MARATOS, N. (1978) Exact penalty function algorithms for finite-dimensional and control optimization problems. *Ph.D. Thesis*, Department of Computing and Control, University of London.

MAYNE, D. Q. & MARATOS, N. (1979) A first-order, exact penalty function algorithm for equality constrained optimization problems. *Math. Program.*, **16**, 303–324.

MONGEAU, M. & SARTENAER, A. (1995) Automatic decrease of the penalty parameter in exact penalty function methods. *Eur. J. Oper. Res.*, **83**, 686–699.

MORALES, J. L., NOCEDAL, J. & WU, Y. (2008) A sequential quadratic programming algorithm with an additional equality constrained phase. *Technical Report OTC-05*. Optimization Center, McCormick School of Engineering and Applied Science, Northwestern University, Evanston, IL: (to appear in *IMA J. Numer. Anal.*).

NOCEDAL, J. & WRIGHT, S. J. (1999) *Numerical Optimization*. New York: Springer.

PANTOJA, J. F. A. & MAYNE, D. Q. (1991) Exact penalty functions with simple updating of the penalty parameter. *J. Optim. Theory Appl.*, **69**, 441–467.

PIETRZYKOWSKI, T. (1969) An exact potential method for constrained maxima. *SIAM J. Numer. Anal.*, **6**, 299–304.

POWELL M. J. D. (1978a) The convergence of variable metric methods for nonlinearly constrained optimization calculations. *Nonlinear Programming 3* (O. L. Mangasarian, R. R. Meyer & S. M. Robinson eds). New York: Academic Press, pp. 27–63.

POWELL, M. J. D. (1978b) A fast algorithm for nonlinearly constrained optimization calculations. *Numerical Analysis* (G. A. Watson ed.). Lecture Notes in Mathematics, vol. 630. Berlin: Springer, pp. 144–157.

SAHBA, M. (1987) Globally convergent algorithm for nonlinearly constrained optimization problems. *J. Optim. Theory Appl.*, **52**, 291–309.

YUAN, Y.-X. (1995) On the convergence of a new trust region algorithm. *Numer. Math.*, **70**, 515–539.