A SECOND DERIVATIVE SQP METHOD: LOCAL CONVERGENCE AND PRACTICAL ISSUES*

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

Abstract. Gould and Robinson [SIAM J. Optim., 20 (2010), pp. 2023–2048] proved global convergence of a second derivative SQP method for minimizing the exact $\ell_1$-merit function for a fixed value of the penalty parameter. This result required the properties of a so-called Cauchy step, which was itself computed from a so-called predictor step. In addition, they allowed for the additional computation of a variety of (optional) accelerator steps that were intended to improve the efficiency of the algorithm. The main purpose of this paper is to prove that a nonmonotone variant of the algorithm is quadratically convergent for two specific realizations of the accelerator step; this is verified with preliminary numerical results on the Hock and Schittkowski test set. Once fast local convergence is established, we consider two specific aspects of the algorithm that are important for an efficient implementation. First, we discuss a strategy for defining the positive-definite matrix $B_k$ used in computing the predictor step that is based on a limited-memory BFGS update. Second, we provide a simple strategy for updating the penalty parameter based on approximately minimizing the $\ell_1$-penalty function over a sequence of increasing values of the penalty parameter.

Key words. nonlinear programming, nonlinear inequality constraints, sequential quadratic programming, $\ell_1$-penalty function, nonsmooth optimization

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

DOI. 10.1137/080744554

1. Introduction. In [17], we presented a sequential inequality/equality constrained quadratic programming algorithm (an SIQP/SEQP “hybrid”) for solving the problem

\[
(\ell_1, \sigma) \quad \text{minimize} \quad \phi(x) = f(x) + \sigma \|c(x)\|_1,
\]

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, $\sigma$ 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 $(\ell_1, \sigma)$ correspond, under certain assumptions, to solutions of the nonlinear programming problem

\[
(NP) \quad \text{minimize} \quad f(x) \quad \text{subject to} \quad c(x) \geq 0.
\]

Our trust-region algorithm is similar to the original work of Fletcher [12]. One novel difference, however, is that we never require the global minimizer of a general indefinite quadratic program (QP). This is accomplished by computing trial steps as the sum of two well-defined steps. The first step is computed from the predictor step, which is the unique solution to a strictly convex QP. This step drives convergence of the algorithm.

*Received by the editors December 22, 2008; accepted for publication (in revised form) December 8, 2009; published electronically March 12, 2010. Research supported by the EPSRC grants EP/E053535/1 and EP/F005369/1.

†Rutherford Appleton Laboratory, Numerical Analysis Group, Chilton, Didcot, Oxfordshire, OX11 0QX, UK (nick.gould@stfc.ac.uk).

‡University of Oxford, Mathematical Institute, 24–29 St Giles’, Oxford, OX1 3LB, UK (robinson@maths.ox.ac.uk).
and is, in fact, the only step required to establish global convergence. The second step is called the accelerator step, and its purpose is to enhance global efficiency and improve local convergence. The primary purpose of this paper is to establish fast local convergence rates for two particular instances of the accelerator step suggested in [17, section 2.3].

The first instance is an equality constrained QP that relies on an estimate of the set of constraints active at a solution to \((\ell_1-\sigma)\), i.e., an optimal active set. A local analysis for this case is required for two reasons. First, the requirements on the Hessian matrix of the QP associated with the predictor step are quite minimal. Thus its ability to accurately predict an optimal active set must be established. Second, it must be shown that this particular accelerator step is accepted by the algorithm in a neighborhood of a solution, since this may be disallowed by global convergence criterion. The second accelerator step we consider is derived from an inequality constrained QP that uses an artificial so-called descent-constraint whose purpose is to ensure that the accelerator step is not an ascent direction. To establish fast local convergence we must again show that this particular accelerator step will be allowed by the algorithm within a neighborhood of a solution. Additionally, we must prove that the artificial descent-constraint does not interfere with the anticipated fast (quadratic) convergence.

The algorithm we propose has been designed with large problems in mind. Nevertheless, we expect the use of the additional, and carefully constructed, accelerator step to offer significant improvement over traditional quasi-Newton SQP methods for problems of any size.

An outline of this paper is as follows. In section 2 we provide the details of a non-monotone variant of the algorithm proposed in [17]. Section 3 then discusses the local convergence properties of that algorithm and culminates with two rate-of-convergence results. The first applies when the accelerator step is computed from an equality constrained subproblem [17, section 2.3.2], while the second applies when the accelerator step is computed from an inequality constrained subproblem [17, section 2.3.1]. In section 4 we consider other aspects of the algorithm that relate to efficiency. In particular we address strategies for adjusting the penalty parameter and for defining the positive-definite matrix required for computing the predictor step. Finally, in section 5, we provide preliminary numerical results for the proposed algorithm.

**Notation.** We let \(e\) denote the vector of all ones whose dimension is determined by the context. The gradient of \(f(x)\) is \(g(x)\) and \(\nabla_{xx} f(x)\) its (symmetric) Hessian; the matrix \(\nabla_{xx} c_j(x)\) is the Hessian of \(c_j(x)\); \(J(x)\) is the \(m \times n\) Jacobian matrix of the constraints with \(i\)th row \(\nabla c_i(x)^T\). The Lagrangian function associated with (NP) is \(L(x,y) = f(x) - y^T c(x)\). The Hessian of the Lagrangian with respect to \(x\) is \(\nabla_{xx} L(x,y) = \nabla_{xx} f(x) - \sum_{j=1}^m y_j \nabla_{xx} c_j(x)\).

For a general vector \(v\), the notation \([v]^- = \min(0,v)\) is used, where the minimum is understood to be componentwise, and \(\text{diag}(v)\) represents a diagonal matrix whose \(i\)th diagonal entry is \(v_i\); given two general vectors \(v\) and \(w\), the notation \(v \cdot w\) represents the vector whose \(i\)th component is \(v_i w_i\); given a general symmetric matrix \(A\), the notation \(A \geq \lambda\) means that the smallest eigenvalue of \(A\) is bigger than or equal to \(\lambda\); and given a set of of matrices \(A_1, A_2, \ldots, A_p\) for some \(p \geq 1\), we define \(\text{diag}(A_1, A_2, \ldots, A_p)\) to be the block-diagonal matrix whose \(i\)th block is \(A_i\).

Given a solution \(x^*\) to problem (NP), we define the indexing sets

\[
A \equiv \{ i : c_i(x^*) = 0 \} \quad \text{and} \quad I \equiv \{ i : c_i(x^*) > 0 \},
\]

which are the set of active and inactive constraints, respectively, at \(x^*\). Given a
generic vector \( v \), a generic matrix \( V \), and a generic indexing set \( S \), the notation \( v_S \) and \( V_S \) will denote the rows of \( v \) and \( V \) that correspond to the indices in \( S \); if \( v \) and \( V \) are functions of \( x \), then we sometimes write \( v_S(x) \) and \( V_S(x) \) instead of \([v(x)]_S \) and \([V(x)]_S \).

2. The algorithm. We now describe the details of Algorithm 2.1, which is a nonmonotone variant of [17, Algorithm 3.1]. First, the user supplies an initial guess \((x_0, y_0)\) of a solution to problem \((\ell_1-\sigma)\). Next, “success” parameters \(0 < \nu_s \leq \nu_{v_s} < 1\), a reset value \( \Delta_u \) and maximum allowed value \( \Delta_v \) for the predictor trust-region radius satisfying \(0 < \Delta_u \leq \Delta_v \), sufficient model decrease tolerance and approximate Cauchy point tolerance \(0 < \eta < \eta_{\text{ACP}} < 1\), accelerator trust-region radius factor \( \tau_f \), maximum number of consecutive failures \( \max \text{ fails} \), and expansion and contraction factors \(0 < \eta_r < 1 < \eta_c \) 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 define the Cauchy step as

\[
\minimize_{s \in \mathbb{R}^n} f_k + g_k^T s + \frac{1}{2} s^T B_k s + \sigma \|c_k + J_k s\|_1, \quad \text{subject to } \|s\|_{\infty} \leq \Delta^p_k,
\]

where \( \Delta^p_k > 0 \) is the predictor trust-region radius and we have used the notation \( f_k = f(x_k), g_k = g(x_k), c_k = c(x_k) \), and \( J_k = J(x_k) \). By introducing elastic variables \([13]\), this problem is equivalent to

\[
\minimize_{s \in \mathbb{R}^n} f_k + g_k^T s + \frac{1}{2} s^T B_k s + \sigma e^T s \quad \text{subject to } c_k + J_k s + v \geq 0, \quad v \geq 0, \quad \|s\|_{\infty} \leq \Delta^p_k.
\]

Strategies for defining the positive-definite matrix \( B_k \) are discussed in section 4.1. Next, we define \( y^p_k \) to be any multiplier estimate that ultimately satisfies the conditions \( y^p_k - y^* = O(\|x_k - x^*\|_2) \) and \( [y^p_k]_x = 0 \); Lemma 3.13 shows that we may choose \( y^p_k = y^e_k \), where \( y^e_k \) is the predictor multiplier vector from (2.2), but this is not required. We prefer using this more flexible requirement on \( y^e_k \), since it allows for alternatives, such as least-squares multiplier estimates, to be tested in the future. Next, we define \( H_k \) to be any symmetric approximation to \( \nabla_{xx} \mathcal{L}(x_k, y^p_k) \), but for the local convergence results given by Theorems 3.12 and 3.14 we choose \( H_k \equiv \nabla_{xx} \mathcal{L}(x_k, y^p_k) \). We may then define the Cauchy step as \( s^c_k = \alpha_k s^p_k \), where \( \alpha_k \) is the solution to

\[
\minimize_{0 \leq \alpha \leq 1} M^0_k(\alpha s^p_k)
\]

for the so-called faithful model

\[
M^0_k(s) \overset{\text{def}}{=} M^0_k(s; x_k) = f_k + g_k^T s + \frac{1}{2} s^T H_k s + \sigma \|c_k + J_k s\|_1.
\]

The word “faithful” is used, since we are allowed to choose \( H_k \) to be the exact Hessian of the Lagrangian \( \nabla_{xx} \mathcal{L}(x_k, y^p_k) \) (note that this is generally not allowed for the predictor subproblem (2.1), since \( \nabla_{xx} \mathcal{L}(x_k, y^p_k) \) is unlikely to be positive definite). Once the Cauchy step has been computed, we compute

\[
\Delta M^0_k(s) \overset{\text{def}}{=} \Delta M^0_k(s; x_k) = M^0_k(0; x_k) - M^0_k(s; x_k),
\]

which is the change in the faithful model of \( \phi \). We mention that this notation does not allude to the dependence on the penalty parameter \( \sigma \). Next we have the option of computing an accelerator step \( s^a_k \) as the solution of any of the subproblems discussed
in [17, section 2.3]. However, in this paper we will restrict our attention to two particular instances. The first accelerator subproblem [17, section 2.3.2] is given by

\[
\text{(EQP)} \quad \begin{align*}
\text{minimize} \quad & \tilde{f}_k + (g_k + H_k s_k^p)^T s + \frac{1}{2} s^T H_k s \\
\text{subject to} \quad & [J_k s] A(s_k^p) = 0, \quad \|s\|_2 \leq \Delta_k^0,
\end{align*}
\]

where \( A(s_k^p) = \{ i : [c_k + J_k s_k^p]_i \leq 0 \}, \) \( \tilde{f}_k = f_k + g_k^T s_k^p + \frac{1}{2} s_k^p^T H_k s_k^p, \) and \( \Delta_k^0 > 0 \) is the accelerator radius. Since this subproblem defines only multipliers for the constraints whose indices are in the set \( A(s_k^p) \), we form accelerator multipliers \( y_k^A \) by “scattering” the multipliers from subproblem (EQP) into the appropriate locations of a zero-vector of length \( m \). The second accelerator subproblem [17, section 2.3.1] that we will consider is given by

\[
\text{(EIQP)} \quad \begin{align*}
\text{minimize} \quad & \tilde{f}_k + (g_k + H_k s_k^{CP})^T s + \frac{1}{2} s^{CP} H_k s + \sigma_k \| [c_k + J_k (s_k^{CP} + s)]_{V_k} \|_1 \\
\text{subject to} \quad & [c_k + J_k (s_k^{CP} + s)]_{S_k} \geq 0, \\
& (g_k + H_k s_k^{CP} + \sigma_k J_k^T z_k)^T s \leq 0, \quad \|s\|_{\infty} \leq \Delta_k^g,
\end{align*}
\]

where \( \tilde{f}_k = f_k + g_k^T s_k^{CP} + \frac{1}{2} s_k^{CP^2} H_k s_k^{CP}, \) \( z_k \) is defined componentwise as

\[
[z_k]_i = \begin{cases} 
-1 & \text{if } i \in V_k, \\
0 & \text{if } i \in S_k,
\end{cases}
\]

where \( V_k = \{ i : [c_k + J_k s_k^{CP}]_i < 0 \} \) and \( S_k = \{ i : [c_k + J_k s_k^{CP}]_i \geq 0 \} \), and \( (g_k + H_k s_k^{CP} + \sigma_k J_k^T z_k)^T s \leq 0 \) is the so-called descent-constraint. The descent-constraint guarantees that the directional derivative of \( M_k^g(s) \) in the direction \( s_k^A \) is nonpositive. In this case the accelerator multipliers \( y_k^A \) are the multipliers for the general constraints of the smooth variant of problem (EIQP). The trial step computation is completed by defining the full step \( s_k \) as

\[
\text{(2.6)} \quad \begin{align*}
\phi_k = \begin{cases} 
\frac{s_k^p + s_k^A}{\Delta M_k^p(s_k^p + s_k^A)} & \text{if } \Delta M_k^p(s_k^p + s_k^A) \geq \eta \Delta M_k^p(s_k^{CP}), \\
\frac{s_k^{CP}}{\Delta M_k^p(s_k^{CP})} & \text{otherwise},
\end{cases}
\end{align*}
\]

\[
\text{(2.7)} \quad (\text{EQP}) \quad s_k = \begin{cases} 
\phi_k & \text{if } \Delta M_k^p(s_k^p + s_k^A) \geq \eta \Delta M_k^p(s_k^{CP}), \\
s_k^{CP} & \text{otherwise},
\end{cases}
\]

\[
\text{(2.8)} \quad (\text{EIQP}) \quad s_k = \begin{cases} 
\phi_k + s_k^A & \text{if } \Delta M_k^g(s_k^{CP} + s_k^A) \geq \eta \Delta M_k^g(s_k^{CP}), \\
s_k^{CP} & \text{otherwise},
\end{cases}
\]

for some constant \( 0 < \eta < 1 \) independent of \( k \) (see [17, section 2.3] for more details). Note that the bound

\[
\text{(2.9)} \quad \Delta M_k^p(s_k) \geq \eta \Delta M_k^p(s_k^{CP})
\]

is always satisfied and is the crucial bound used to prove global convergence in [17]. We note that although the accelerator step is optional for proving global convergence, it is generally required for establishing quadratic convergence. We then evaluate \( \phi(x_k + s_k) \) and the change in the faithful model at the full step \( \Delta M_k^p(s_k). \)

We now must decide whether to accept the trial step \( s_k \). This decision is based on the value of the ratio \( r_k \), which in turn depends on the history of “successful/unsuccesful” steps. If we assume that every iterate is successful in the sense of traditional trust-region strategies, i.e., there is “good agreement” between the actual and predicted decrease in \( M_k^u \), then our nonmonotone algorithm is identical to traditional updating strategies. However, if a failure occurs (in the traditional sense),
Algorithm 2.1 Nonmonotone Algorithm.

Input: \((x_0, y_0)\)

Set parameters \(0 < \eta_e \leq \eta_{vs} < 1\), \(0 < \Delta_R \leq \Delta_R^u \leq \Delta_U\), \(0 < \eta \leq \eta_{ACP} < 1\), and \(\tau_f \geq 1\).

Set expansion and contraction factors \(0 < \eta_c < 1 < \eta_e\) and fail counter \(\text{fails} = 0\).

Set nonmonotone parameters \(0 \leq \text{max fails} \in \mathbb{N}\) and \(\text{fails} = 0\).

\(k \leftarrow 0\)

\(\text{do}\)

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 \(\nabla_{xx} L(x_k, y_k)\).

Solve problem (2.1) for predictor step and multipliers \((s_k^p, y_k^p)\).

Define multiplier estimate \(y_k^e\) such that \(y_k^e - y^* = O(||x_k - x^*||_2)\) and \([y_k^e]_I = 0\).

Define \(H_k\) to be a symmetric approximation to \(\nabla_{xx} L(x_k, y_k^e)\).

Solve problem (2.3) for \(s_k^{cp}\) and compute \(\Delta M_k^{cp}(s_k^{cp})\).

Compute an accelerator step and multipliers \((s_k^A, y_k^A)\) (optional).

Define a full step \(s_k\) that satisfies condition (2.9).

Evaluate \(\phi(x_k + s_k)\) and \(\Delta M_k^A(s_k)\).

\(\text{if} \ \text{fails} = 0, \ \text{then}\)

\(r_k \leftarrow (\phi(x_k) - \phi(x_k + s_k))/\Delta M_k^A(s_k)\) \hspace{1cm} \text{[standard definition]}\)

\(\text{else}\)

\(r_k \leftarrow (\phi_R - \phi(x_k + s_k))/\Delta_R^u\) \hspace{1cm} \text{[change in } \phi \text{ based on point } x_k]\)

\(\text{end if}\)

\(\text{if} \ r_k \geq \eta_{vs}, \text{ then} \) \hspace{1cm} \text{[very successful]}

\(x_{k+1} \leftarrow x_k + s_k\)

\(y_{k+1} \leftarrow y_k^e\) \hspace{1cm} \text{[if accelerator step not computed]}\)

\(\Delta_{k+1}^p \leftarrow \min \left(\max(\eta_e \cdot \Delta_R^p, \Delta_R), \Delta_U\right)\) \hspace{1cm} \text{[increase } \Delta_k^p \text{ and ensure } \Delta_{k+1}^p \geq \Delta_R]\)

\(\text{fails} \leftarrow 0\)

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.
else if \( r_k \geq \eta_s \), then

[successful]
\[
x_{k+1} \leftarrow x_k + s_k \\
y_{k+1} \leftarrow y_k^A \quad (y_{k+1} \leftarrow y^p_k \text{ if accelerator step not computed}) \\
\Delta^p_{k+1} \leftarrow \max(\Delta^p_k, \Delta^v_k) \quad \text{[ensure \( \Delta^p_k \) is bigger than \( \Delta^v_k \])} \\
fails \leftarrow 0
\]

else

[if fails \( \leq \max \text{ fails} \), then]

[save current point]
\[
x_n \leftarrow x_k, \quad y_n \leftarrow y_k, \quad \phi_n \leftarrow \phi_k, \quad \Delta^u_n \leftarrow \Delta M_k(s_k) \\
\Delta^v_n \leftarrow \Delta^p_k \\
\Delta^v_{k+1} \leftarrow \eta_c \Delta^v_k \quad \text{(optional)}
\]

[revert to saved point]
\[
x_{k+1} \leftarrow x_n, \quad y_{k+1} \leftarrow y_n \\
\Delta^v_{k+1} \leftarrow \eta_c \Delta^v_n \\
fails \leftarrow 0
\]

end if

[revert to saved point]
\[
x_{k+1} \leftarrow x_n, \quad y_{k+1} \leftarrow y_n \\
\Delta^v_{k+1} \leftarrow \eta_c \Delta^v_n \\
fails \leftarrow 0
\]

end if

[decrease \( \Delta^v_k \)]

end if

\[
\Delta^v_{k+1} \leftarrow \tau_f \cdot \Delta^v_k
\]

[update accelerator radius]
\[
k \leftarrow k + 1
\]

end do

3. Local convergence. In this section we show that Algorithm 2.1 is superlinearly convergent if an accelerator step is computed from either subproblem (EQP) or (EIQP). The update to the Lagrange multiplier vector \( y_k \) is now critical, and we must consider the sequence of vector-pairs \( (x_k, y_k) \). To simplify notation, we let \( w \) denote the combined \( x \) and \( y \) vectors, i.e., \( w = (x, y) \), and we write \( w_k = (x_k, y_k) \) for the current estimate of a solution \( w^* = (x^*, y^*) \), \( w^p_k = (x^p_k, y^p_k) \) for the solution to the predictor subproblem (2.2), and \( w^A_k = (x^A_k, y^A_k) \) for the solution of the accelerator subproblem (the precise definition of \( y^A_k \) depends on which accelerator subproblem is used).

The main result of this section is that if a successful iterate of Algorithm 2.1 gets close enough to a local minimizer \( w^* \) of problem (NP) that satisfies the strong second-order sufficient conditions, and if we compute an accelerator step from either subproblem (EQP) or (EIQP), then the sequence of iterates converges to \( w^* \) with the same convergence properties as Newton’s method for zero-finding applied to the function

\[
F_N(x, y_A) = \left( g(x) - J_A(x)^T y_A \right)_{c_A(x)},
\]

where the indexing set \( A \) is defined in (1.1). We accomplish this by first showing that if \( w_k \) is close enough to \( w^* \), then the predictor step accurately predicts the optimal active set and that the trust-region constraint is inactive. We then show that specific accelerator steps also identify the optimal active set and that their associated trust-region constraints are inactive. Since these steps are then equivalent to one step of
Newton’s method for zero-finding applied to \( F_N \), we deduce that \( w_{k+1} \) is closer to \( w^* \) than was \( w_k \). This process is then repeated and results in the value \( w_{k+2} \). Since Algorithm 2.1 is a nonmonotone approach, the analysis given by Conn, Gould, and Toint [10, section 15.3.2] shows that the \( \ell_1 \)-merit function will accept the value \( x_{k+2} \), and it follows that convergence may be described using classical results for Newton’s method applied to the function \( F_N \).

In the following definitions related to a solution of problem (NP), we use the notation \( c^* = c(x^*) \), \( g^* = g(x^*) \), and \( J^* = J(x^*) \).

**Definition 3.1** (first-order KKT point). *We say that the point \((x^*, y^*)\) is a first-order KKT point for problem (NP) if*

\[
g^* - J^*^T y^* = 0, \quad c^* \geq 0, \quad y^* \geq 0, \quad \text{and} \quad c^* \cdot y^* = 0. \tag{3.2}
\]

**Definition 3.2** (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 \( \lambda_{\min}^H > 0 \) such that \( s^T \nabla_{xx} L(x^*, y^*) s \geq \lambda_{\min}^H s^T s \) for all \( s \) satisfying \( J_A^* s = 0 \).

**Definition 3.3** (strict complementarity). *We say that strict complementarity holds at a KKT point \((x^*, y^*)\) for problem (NP) if \( y_A^* > 0 \).

**Definition 3.4** (linear independent constraint qualification). *We say that the linear independent constraint qualification (LICQ) holds at a KKT point \((x^*, y^*)\) for problem (NP) if \( J_A^* \) has full row rank.*

**Definition 3.5**. *We say that the strong second-order sufficient conditions hold at a point \((x^*, y^*)\) if it satisfies Definitions 3.1–3.4.*

In the ensuing analysis we prove results that hold in a neighborhood of a solution to problem (NP). It is thus convenient for us to define \( B_\epsilon(v) = \{x \in \mathbb{R}^n : \|x - v\|_2 < \epsilon \} \) and \( B_\epsilon^*(v) = \{x \in \mathbb{R}^n : \|x - v\|_2 \leq \epsilon \} \) to be the open and the closed ball centered at \( v \) of radius \( \epsilon \), respectively, for a given vector \( v \in \mathbb{R}^n \) and scalar \( \epsilon \).

### 3.1. Optimal active set identification.

The analysis that ensues requires a notion of “uniformity” for the underlying KKT systems within a neighborhood of a solution \( w^* \). This is generally not an issue for systems involving \( H_k \), since it is reasonable to expect that if \( w_k \) converges to \( w^* \), then \( H_k \) will converge to \( \nabla_{xx} L(x^*, y^*) \); this certainly occurs if \( H_k \equiv \nabla_{xx} L(x_k, y_k) \) or \( H_k \equiv \nabla_{xx} L(x_k, y_k^d) \). A similar statement does not hold for systems involving \( B_k \), since \( B_k \) is generally not a continuous function of \( w \). Moreover, we certainly cannot expect the positive-definite matrix \( B_k \) to converge to \( \nabla_{xx} L(x^*, y^*) \), since \( \nabla_{xx} L(x^*, y^*) \) is normally indefinite. The optimality conditions for problem (NP) suggest that we need the matrices \( H_k \) and \( B_k \) to be positive definite when restricted to the null space of the active constraints (note that \( B_k \) is positive definite by construction); this is essentially the uniformity that we need. To develop a general framework, we define the following sets that depend on the minimizer \( x^* \) through the indexing set \( A \) defined by (1.1):

\[
S(x; x^*) = \left\{ M = MT \in \mathbb{R}^{n \times n} : \beta_{\max} \geq \frac{s^T M s}{s^T s} \geq \lambda_{\min} \quad \text{for all} \quad s \quad \text{satisfying} \quad J_A(x)s = 0 \right\}
\]

and

\[
S_{\epsilon} = \bigcup_{w \in B_\epsilon(w^*)} S(x; x^*)
\]

for given real numbers \( \beta_{\max} \) and \( \lambda_{\min} > 0 \). Using this definition, we now state a result that supplies the required uniformity.
LEMMA 3.6. If \( w^* \) is a KKT point for problem (NP) that satisfies the LICQ, then

(i) for any \( 0 < \varepsilon < \infty \) the set

\[
S_\varepsilon = \bigcup_{w \in \mathcal{B}_\varepsilon(w^*)} S(x; x^*)
\]

is compact;

(ii) if \( \nu_1 \leq \nu_2 \), then \( S_{\nu_1} \subseteq S_{\nu_2} \);

(iii) there exists a positive number \( \varepsilon_1 \) such that if \( w \in \mathcal{B}_{\varepsilon_1}(w^*) \) and \( M \in S_{\varepsilon_1} \), then \( s^TMs \geq (\lambda_{\text{min}}/2)s^T s \) for all \( s \) satisfying \( J_A(x)s = 0 \).

If in addition, strict complementarity holds at \( w^* \), then

(iv) there exists a positive number \( \varepsilon_2 \) such that \( \varepsilon_2 \leq \varepsilon_1 \) and numbers \( \beta_0 > 0 \) and \( \beta > 0 \) such that if \( w \in \mathcal{B}_{\varepsilon_2}(w^*) \) and \( M \in S_{\varepsilon_2} \), then \( J_A(x) \) has full row rank, \( c_I(x) > 0 \), \( y_A > 0 \), and the matrices

\[
\bar{K}_M(x) = \begin{pmatrix} M & J_A(x)^T \\ J_A(x) & 0 \end{pmatrix} \quad \text{and} \quad K_M(w) = \begin{pmatrix} M & -J_A(x)^T \\ \text{diag}(y_A)J_A(x) & 0 \\ 0 & \text{diag}(c_I) \end{pmatrix}
\]

are nonsingular and satisfy

\[
\|\bar{K}_M(x)^{-1}\|_2 \leq \beta_0 \quad \text{and} \quad \|K_M(w)^{-1}\|_2 \leq \beta,
\]

where the indexing set \( \mathcal{I} \) is defined in (1.1);

(v) if \( w \in \mathcal{B}_{\varepsilon_2}(w^*) \) and \( M \in S_{\varepsilon_2} \), then it follows that

\[
s = O(||x - x^*||_2), \quad \pi_A - y_A^* = O(||x - x^*||_2), \quad \text{and} \quad \pi - y^* = O(||x - x^*||_2),
\]

where \( s \) and \( \pi_A \) satisfy

\[
\bar{K}_M(x) \begin{pmatrix} s \\ -\pi_A \end{pmatrix} = \begin{pmatrix} M & J_A(x)^T \\ J_A(x) & 0 \end{pmatrix} \begin{pmatrix} s \\ -\pi_A \end{pmatrix} = -\begin{pmatrix} g(x) \\ c_A(x) \end{pmatrix}
\]

and \( \pi \) is obtained from \( \pi_A \) by “scattering” the components of \( \pi_A \) into a zero-vector of length \( m \) as indicated by \( A \).

Proof. We first prove part (i) of Lemma 3.6. Since it is clear that \( S_\varepsilon \) is bounded, we show only that \( S_\varepsilon \) is closed. Let \( \{M_k\} \) be a sequence in \( S_\varepsilon \) such that \( \lim_{k \to \infty} M_k = M \). This implies the existence of a sequence \( \{w_k\} \in \mathcal{B}_\varepsilon(w^*) \) such that \( M_k \in S(x_k; x^*) \) and \( v^T M_k v \geq \lambda_{\text{min}} v^T v \) for all \( v \) such that \( J_A(x_k)v = 0 \). The set \( \mathcal{B}_\varepsilon(w^*) \) is compact, and, therefore, we can pass to a subsequence \( K \) such that \( \lim_{k \in K} w_k = \bar{w} \in \mathcal{B}_\varepsilon(w^*) \). Since \( J \) is continuous and \( J_A(x^*) \) has full row rank, [7, Theorem 2.3] implies the existence of a locally continuous null space basis function \( Z(\cdot) \) such that \( J_A(x_k)Z(x_k) = 0 \), \( \lim_{k \in K} Z(x_k) = Z \), and \( J_A(x_k)Z = 0 \). This implies that \( Z(x_k)^T M_k Z(x_k) \geq \lambda_{\text{min}} \) and upon taking limits that \( Z^T M Z \geq \lambda_{\text{min}} \). Since it is clear that \( M \) is symmetric and satisfies \( \|M\|_2 \leq \beta_{\text{max}} \), we have \( \bar{M} \in S(x; x^*) \subseteq S_\varepsilon \). Thus, \( S_\varepsilon \) is closed.

Part (ii) of Lemma 3.6 follows immediately from the definitions of \( S_{\nu_1} \) and \( S_{\nu_2} \).

We now prove part (iii) of Lemma 3.6. If part (iii) of Lemma 3.6 was not true, then there would exist a monotonically decreasing and strictly positive sequence \( \{\delta_k\} \to 0 \) and associated sequences \( \{w_k\}, \{s_k\}, \text{ and } \{M_k\} \) such that \( w_k \in \mathcal{B}_{\delta_k}(w^*) \), \( M_k \in S_{\delta_k} \subseteq S_{\delta_1}, \ J_A(x_k)s_k = 0, \ ||s_k||_2 = 1, \) and \( s_k^T M_k s_k < \lambda_{\text{min}}/2 \). It follows from these properties, part (i) of Lemma 3.6, and the fact that the sequence \( \{s_k\} \) belongs to a compact set,
that there exists a subsequence \( K_2 \), a matrix \( M^* \in S_{\delta_1} \), and a unit vector \( s^* \) such that
\[
\lim_{k \in K_2} w_k = w^*, \quad \lim_{k \in K_2} M_k = M^*, \quad \lim_{k \in K_2} s_k = s^*, \quad J_A^* s^* = 0, \quad \text{and} \quad s^T M^* s^* \leq \lambda_{\min}/2.
\]

Since \( M_k \in S_{\delta_1} \) and \( \{\delta_k\} \to 0 \), there also exists a sequence \( \{\tilde{x}_k\} \to x^* \) such that \( s^T M_k s \geq \lambda_{\min} s^2s \) for all \( s \) satisfying \( J_A(\tilde{x}_k)s = 0 \). Using the same argument as in the first paragraph of this proof, we find that \( Z^* M^* Z^* \succeq \lambda_{\min} \), where the columns of \( Z^* \) form a basis for the null space of \( J_A^* \). This contradicts (3.8), and thus part (iii) of Lemma 3.6 must be true.

To show that part (iv) of Lemma 3.6 holds, we first note that strict complementarity and the LICQ imply that there exists a number \( \varepsilon_s \) such that \( 0 < \varepsilon_s \leq \varepsilon_1 \) and
\[
c_i(x) \geq \frac{1}{2} c_i^* > 0 \quad \text{for} \quad i \in I, \quad y_i \geq \frac{1}{2} y^*_i > 0 \quad \text{for} \quad i \in A, \quad \text{and} \quad J_A(x) \text{ has full row rank}
\]
for all \( w \in B_{\varepsilon_s}(w^*) \). Under the current assumptions, it follows from parts (ii) and (iii) of Lemma 3.6, and [1, Lemma 1.27] that
\[
\|\tilde{K}_M(x_k)^{-1}\|_2 \geq k \quad \text{for all} \quad k \geq 0.
\]

Assume that (3.5a) does not hold for any \( \varepsilon_2 \leq \varepsilon_s \) so that there exists a monotonically decreasing sequence \( \{\delta_k\} \to 0 \) such that \( 0 < \delta_k \leq \varepsilon_s \) and associated sequences \( \{w_k\} \in B_{\delta_k}(w^*) \) and \( \{M_k\} \in S_{\delta_k} \subseteq S_{\varepsilon_s} \) such that
\[
\|\tilde{K}_M(x_k)^{-1}\|_2 \geq k \quad \text{for all} \quad k \geq 0.
\]

Since \( \{\delta_k\} \to 0 \) and \( S_{\varepsilon_s} \) is compact as a result of part (i) of Lemma 3.6, there exists a subsequence \( K_3 \) such that \( \lim_{k \in K_3} w_k = w^* \) and \( \lim_{k \in K_3} M_k = M^* \in S_{\varepsilon_s} \). It then follows from (3.10) that \( \tilde{K}_M(x^*) \) is nonsingular. Since \( \lim_{k \in K_3} \tilde{K}_M(x_k) = \tilde{K}_M(x^*) \), [14, Theorem 8.64] implies that the singular values of \( \tilde{K}_M(x_k) \) are uniformly bounded away from zero for \( k \in K_3 \) sufficiently large. Therefore, \( \|\tilde{K}_M(x_k)^{-1}\|_2 \) must be bounded above for all \( k \in K_3 \), which contradicts (3.11). Thus, (3.5a) holds for some \( \varepsilon_1 \geq \varepsilon_s \geq \varepsilon_2 > 0 \) and \( \beta_0 > 0 \). It also follows from (3.9) that \( J_A(x) \) has full row rank, \( c_I(x) > 0 \), and \( y_A > 0 \) for all \( w \in B_{\varepsilon_2}(w^*) \).

We now show that (3.5b) holds for \( \varepsilon_2 \). Let \( w \in B_{\varepsilon_2}(w^*) \) and \( M \in S_{\varepsilon_2} \). Equation (3.9) implies that the matrices
\[
N_F = \begin{pmatrix} I & 0 & J_I(x^T) \text{diag}(c_I(x^*))^{-1} \\ 0 & \text{diag}(y_A)^{-1} & 0 \\ 0 & 0 & I \end{pmatrix}, \quad N_M = \begin{pmatrix} M & J_A(x)^T & 0 \\ J_A(x) & 0 & 0 \\ 0 & 0 & \text{diag}(c_I(x^*)) \end{pmatrix},
\]
and \( N_S = \text{diag}(I, -I, I) \) are nonsingular: they satisfy \( N_F K_M(w) N_S = N_M \) so that \( \|K_M(w)^{-1}\|_2 \leq \|N_M^{-1}\|_2 \|N_F\|_2 \). It is also clear from (3.9) that the quantity \( \|N_F\|_2 \) is bounded for all \( w \in B_{\varepsilon_2}(w^*) \), so to bound \( \|K_M(w)^{-1}\|_2 \) we must bound \( \|N_M^{-1}\|_2 \), but it is sufficient to bound \( \|\tilde{K}_M(x)^{-1}\|_2 \), since (3.9) guarantees that \( c_I(x) \geq \frac{1}{2} c^*_I > 0 \) (componentwise). The result follows from (3.5a), and, therefore, there exists a number \( \beta > 0 \) such that \( \|K_M(w)^{-1}\|_2 \leq \beta \) for all \( w \in B_{\varepsilon_2}(w^*) \) and \( M \in S_{\varepsilon_2} \).

Finally, we prove part (v) of Lemma 3.6. Let \( w_k \in B_{\varepsilon_2}(w^*) \) and \( M \in S_{\varepsilon_2} \). Since \( c_A^* = 0 \), it follows that system (3.7) is equivalent to
\[
(3.12) \quad \begin{pmatrix} M & J_A(x)^T \\ J_A(x) & 0 \end{pmatrix} \begin{pmatrix} s \\ y_A - \pi_A \end{pmatrix} = - \begin{pmatrix} g(x) - J_A(x)^T y_A^* \\ c_A(x) - c_A^* \end{pmatrix}.
\]
Inequality (3.5a), norm inequalities, and Taylor expansions for \(g(x), c_A(x),\) and \(J_A(x)\) at the point \(x^*\) yield \(s = O(\|x - x^*\|_2)\) and \(\pi_A - y_A^* = O(\|x - x^*\|_2)\). The fact that \(\pi - y^* = O(\|x - x^*\|_2)\) follows, since \(\pi_x = 0\) by construction and \(y^*_x = 0\) from the optimality conditions for problem (NP).

Our next aim is to prove a result concerning active set identification. Given a vector \(w\), we define the function

\[
F_{\text{KKT}}(w) = \left( \begin{array}{c} g(x) - J(x)^T y \\ c(x) \cdot y \end{array} \right).
\]

**Lemma 3.7.** Let \(w^*\) be a solution to problem (NP) that satisfies strict complementarity and the LICQ. Then there exist numbers \(\mu > 0\) and \(\beta > 0\) such that if \(w_k \in B_{\mu/2}(w^*), M \in S_{\mu/2}, \) and \(4\beta \|F_{\text{KKT}}(w_k)\|_2 \leq \mu,\) then there exists a unique closest minimizer \((x_k(M), y_k(M)) = w_k(M)\) to the point \(w_k\) for the problem

\[
\begin{array}{l}
\text{minimize} \quad \frac{1}{2}(x - x_k)^T M(x - x_k) + g_k^T(x - x_k) \\
\text{subject to} \quad c_k + J_k(x - x_k) \geq 0
\end{array}
\]

with the following properties:

(i) \(\|x_k(M) - x_k\|_\infty \leq \|w_k(M) - w_k\|_2 \leq 2\beta \|F_{\text{KKT}}(w_k)\|_2;\)
(ii) the set of constraints active at \(x_k(M)\) for problem (3.14) are the same as the indices in \(A;\)
(iii) the solution \(w_k(M)\) satisfies strict complementarity; and
(iv) \(J_A(x_k)\) has full row rank.

**Proof.** We begin by letting \(\varepsilon_1 \geq \varepsilon_2 > 0\) and \(\beta > 0\) be the constants guaranteed by Lemma 3.6. Given any vector-pair \((w, \bar{w})\) and symmetric matrix \(M,\) we define

\[
F_M(w; \bar{w}) = \left( \begin{array}{c} M(x - \bar{x}) + g(\bar{x}) - J(\bar{x})^T y \\ (c(\bar{x}) + J(\bar{x})(x - \bar{x})) \cdot y \end{array} \right).
\]

Differentiating (3.15) we have

\[
F'_M(w; \bar{w}) = \begin{pmatrix} M & -J(\bar{x})^T \\ \text{diag}(y)J(\bar{x}) & \text{diag}(c(\bar{x}) + J(\bar{x})(x - \bar{x})) \end{pmatrix}.
\]

Choosing \((w, \bar{w}) = (w^*, \bar{w}^*)\) we have

\[
F'_M(w^*; \bar{w}^*) = \begin{pmatrix} M & -J_A^T & -J_x^T \\ \text{diag}(y_A^*)J_A^* & 0 & 0 \\ 0 & 0 & \text{diag}(c_x^*) \end{pmatrix},
\]

since optimality conditions at \(w^*\) imply \(c_A^* = 0\) and \(y_x^* = 0.\) It follows from (3.5a) with the choice \(w = w^*\) that the matrix \(F'_M(w^*; w^*)\) is nonsingular and satisfies

\[
\|F'_M(w^*; w^*)^{-1}\| \leq \beta \quad \text{for all } M \in S_{\varepsilon_2}.
\]

Next, choose a number \(\mu\) such that \(0 < \mu \leq \varepsilon_2,\) and if \(w\) and \(\bar{w}\) are contained in \(B_\mu(w^*),\) then the following conditions are satisfied:

C1. if \(c^*_i > 0,\) then \(|c(\bar{x}) + J_k(x - \bar{x})|_i > 0;\)
C2. if \(y^*_i > 0,\) then \(y_i > 0;\)
C3. \(||F'_M(w; \bar{w}) - F'_M(w^*; w^*)||_2 \leq 1/(2\beta)\) (this estimate holds for all \(M).\)
Let $w_k \in \mathcal{B}_{\mu/2}(w^*)$ and $M \in S_{\mu/2}$. Since $\mu < \mu/2 \leq \varepsilon_2 \leq \varepsilon_1$, it follows from parts (ii) and (iii) of Lemma 3.6 that $J_A(x_k)$ has full row rank and that estimate (3.16) holds for $M$; thus part (iv) of Lemma 3.7 is true. Using the argument by Robinson [29, Lemma 1], we now show that $F_M(w; w_k)$ has a unique zero in $\mathcal{B}_{\mu/2}(w_k)$. Note that

(3.17) \[ \mathcal{B}_{\mu/2}(w_k) \subset \mathcal{B}_{\mu}(w^*), \]

since if $w \in \mathcal{B}_{\mu/2}(w_k)$, then

\[ \|w - w^*\|_2 \leq \|w - w_k\|_2 + \|w_k - w^*\|_2 < \mu/2 + \mu/2 \leq \mu. \]

Define the function

(3.18) \[ T_M(w) = w - F'_M(w^*; w^*)^{-1}F_M(w; w_k) \]

so that

\[ T'_M(w) = I - F'_M(w^*; w^*)^{-1}F'_M(w; w_k) = F'_M(w^*; w^*)^{-1}(F'_M(w^*; w^*) - F'_M(w; w_k)). \]

It follows that

\[ \|T'_M(w)\|_2 \leq \beta\|F'_M(w^*; w^*) - F'_M(w; w_k)\|_2 \leq \frac{1}{4} \quad \text{(use (3.16), (3.17), and C3)} \]

for all $w \in \mathcal{B}_{\mu/2}(w_k)$, which implies that $T_M$ is a contraction. It also follows that

(3.19) \[ \|T_M(w_k) - w_k\|_2 \leq \beta\|F_M(w_k; w_k)\|_2 \quad \text{(use (3.18) and (3.16)).} \]

Using the triangle inequality, the fact that $T_M(w)$ is a contraction with contraction factor $1/2$, (3.19), and the assumption that $4\beta\|F_M(w_k; w_k)\|_2 \leq \mu$, we have that for all $w \in \mathcal{B}_{\mu/2}(w_k)$ the estimate

\[ \|T_M(w) - w_k\|_2 \leq \|T_M(w) - T_M(w_k)\|_2 + \|T_M(w_k) - w_k\|_2 \]

\[ \leq \frac{1}{2}\|w - w_k\|_2 + \beta\|F_M(w_k; w_k)\|_2 \leq \frac{\mu}{2}, \]

which implies $T_M : \mathcal{B}_{\mu/2}(w_k) \rightarrow \mathcal{B}_{\mu/2}(w_k)$. We may now apply the well-known fixed point result [31, Theorem 9.23] which states that $T_M$ has a unique fixed point $w_k(M)$ in $\mathcal{B}_{\mu/2}(w_k)$ and that

\[ \|x_k(M) - x_k\|_\infty \leq \|x_k(M) - x_k\|_2 \leq \|w_k(M) - w_k\|_2 \quad \text{(use norm inequalities)} \]

\[ \leq 2\|T_M(w_k) - w_k\|_2 \quad \text{(estimate from fixed-point theorem)} \]

\[ \leq 2\beta\|F_M(w_k; w_k)\|_2 \quad \text{(use (3.19)),} \]

which proves part (i) of Lemma 3.7. Since $w_k(M)$ is a fixed point for $T_M(w)$, (3.18) implies that

(3.20) \[ F_M(w_k(M); w_k) = 0. \]

Thus $w_k(M)$ satisfies the equality conditions for being a first-order KKT point for problem (3.14). We now show that the point $w_k(M)$ is actually a first-order KKT point for problem (3.14). Since $w_k(M) \in \mathcal{B}_{\mu/2}(w_k) \subset \mathcal{B}_{\mu}(w^*)$, we may deduce the following: if $y^*_i > 0$, then C2 implies $[y_k(M)]_i > 0$ and then (3.20) implies $[c_k + J_k(x_k(M) - x_k)]_i = 0$; if $c^*_i > 0$, then C1 implies $[c_k + J_k(x_k(M) - x_k)]_i > 0$ and then (3.20) implies $[y_k(M)]_i = 0$. Since strict complementarity holds at $w^*$ by assumption,
one of these two cases must hold, and, therefore, \( w_k(M) \) is a first-order KKT point for the problem (3.14) that satisfies strict complementarity and correctly identifies the optimal active set; this establishes parts (ii) and (iii). The fact that \( x_k(M) \) is a minimizer follows from parts (ii) and (iii) of Lemma 3.6. Finally, \( w_k(M) \) is the unique closest solution, since any other solution would be a KKT point and, therefore, a zero of the function \( F_M(w; w_k) \). However, \( w_k(M) \) is the unique zero inside \( \mathcal{B}_{\mu/2}(w_k) \).  

### 3.2. Local descent properties

In this section we show that, in a neighborhood of a solution \( w^* \), directions related to the traditional SQP step are descent directions for the underlying model functions; this result is critical for proving that Algorithm 2.1 has a fast rate-of-convergence. We use the following definition.

**Definition 3.8.** Given a vector \( v \in \mathbb{R}^n \) and a subspace \( V \subseteq \mathbb{R}^n \), we define

\[
\theta(v, V) = \begin{cases} 
\tan^{-1}(\|v_n\|_2/\|v_s\|_2) & \text{if } \|v_n\|_2 \neq 0, \\
\pi/2 & \text{otherwise},
\end{cases}
\]

for the underlying model functions; this result is critical for proving that Algorithm 2.1 has a fast rate-of-convergence. We use the following definition.

\[
(3.21) \quad \theta(v, V) = \begin{cases} 
\tan^{-1}(\|v_n\|_2/\|v_s\|_2) & \text{if } \|v_n\|_2 \neq 0, \\
\pi/2 & \text{otherwise},
\end{cases}
\]

where \( v = v_n + v_s \) is the unique orthogonal decomposition of \( v \) such that \( v_n \in V \) and \( v_n \perp V \).

The next result essentially says how close a vector \( s \) must be to the null space of the active constraints to guarantee positive curvature in a neighborhood of a solution.

**Lemma 3.9.** Let \( w^* \) be a solution to problem (NP) that satisfies the LICQ. Then, there exists a number \( \varepsilon_2 > 0 \) such that if \( w, s, \) and \( M \) satisfy \( w \in \mathcal{B}_{\varepsilon_2}(w^*), M \in S_{\varepsilon_2}, \) and

\[
(3.22) \quad \theta(s, \text{null}(J_A(x))) \leq \theta \overset{\text{def}}{=} \min \left( \frac{\pi}{2}, \tan^{-1}\left( \frac{\lambda_{\min}}{24\beta_{\max}} \right) \right),
\]

then \( s^T M s \geq (\lambda_{\min}/8)s^T s \).

**Proof.** Let \( \varepsilon_2 \) be defined as in part (iv) of Lemma 3.6 so that \( J_A(x) \) has full row-rank for all \( w \in \mathcal{B}_{\varepsilon_2}(w^*) \). Suppose that \( w \in \mathcal{B}_{\varepsilon_2}(w^*), M \in S_{\varepsilon_2}, \) and \( s \) satisfy (3.22). If we write \( s = s_N + s_h \) for \( s_N \in \text{null}(J_A(x)) \) and \( s_h \in \text{Range}(J_A(x)^T) \), it follows from (3.21) and (3.22) that \( \theta = \theta(s, \text{null}(J_A(x))) \) satisfies

\[
(3.23) \quad \frac{\|s_h\|_2}{\|s_N\|_2} = \tan(\theta) \leq 1.
\]

Using the orthogonal decomposition of \( s \), parts (ii) and (iii) of Lemma 3.6, the Cauchy–Schwarz inequality, the definition of \( \beta_{\max} \), and (3.23) and (3.22), we have

\[
\frac{s^T M s}{s^T s} = \frac{s_N^T M s_N + s_h^T M s_h + 2s_N^T M s_h}{\|s_N\|^2_2 + \|s_h\|^2_2} \\
\geq \frac{(\lambda_{\min}/2)s_N^T s_N - \beta_{\max}\|s_h\|^2_2 - 2\beta_{\max}\|s_N\|_2\|s_h\|_2}{\|s_N\|^2_2 + \|s_h\|^2_2} \\
\geq \frac{\lambda_{\min}}{4} - \beta_{\max}\tan^2(\theta) - 2\beta_{\max}\tan(\theta) \\
\geq \frac{\lambda_{\min}}{4} - 3\beta_{\max}\tan(\theta) \geq \frac{\lambda_{\min}}{8},
\]

which completes the proof.

We now show that in the neighborhood of a solution \( w^* \), the (unique) solution to

\[
(3.24) \quad \text{minimize } g(x)^T s + \frac{1}{2}s^T M s \text{ subject to } c_A(x) + J_A(x)s = 0
\]
satisfies a certain “descent” property for the underlying models (under certain assumptions).

**Lemma 3.10.** Let \( w^* \) be a minimizer for problem (NP) that satisfies the LICQ and strict complementarity, and suppose that \( \sigma > \| y^* \|_{\infty} \). It follows that there exist positive numbers \( c_2 \) and \( \varepsilon_3 \) such that if \( w \in B_{\varepsilon_3}(w^*) \) and \( M \in S_{\varepsilon_3} \), then problem (3.24) is well defined and the solution \( s_T \) satisfies

\[
(g(x) + \sigma J(x)^T z)^T s_T < -c_2 \| s_T \|_2^2 \quad \text{for} \quad z = \begin{cases} 0 & \text{if } c(x) \geq 0, \\ -1 & \text{otherwise.} \end{cases}
\]

**Proof.** Strict complementarity implies the existence of a scalar \( \kappa_8 > 0 \) such that

\[
y^*_A \geq \kappa_8 e > 0.
\]

We define \( \bar{\theta} \) as in Lemma 3.9 and choose positive scalars \( \kappa_1 \) and \( \varepsilon_3 \) so that the following hold for all \( w \in B_{\varepsilon_3}(w^*) \) and \( M \in S_{\varepsilon_3} \):
1. \( \varepsilon_3 \leq \varepsilon_2 \), where \( \varepsilon_2 \) is defined in Lemma 3.9;
2. \( \| J(x)^T \|_2 (\| J(x)J(x)^T \|_2)^{-1} \leq \kappa_3 \);  
3. the system

\[
(3.27) \quad \left( \begin{array}{cc} M & J_A(x)^T \\ J_A(x) & 0 \end{array} \right) \left( \begin{array}{c} s \\ -q \end{array} \right) = - \left( \begin{array}{c} g(x) - J_A(x)^T y_A \\ c_A(x) \end{array} \right)
\]

has a unique solution \((s, q)\) that satisfies

a. \( (\kappa_8/2)e \leq y_A + q \leq \sigma(1 - \kappa_8)e \) for some \( \kappa_\sigma > 0 \);

b. \( \| s \|_2 \leq \min(1, c_1) \), where

\[
(3.28) \quad c_1 = \frac{\kappa \sin(\bar{\theta})}{2\kappa_1 \beta_{\text{max}}} > 0 \quad \text{and} \quad \kappa = \min\left(\frac{\kappa_8}{2}, \sigma \kappa_\sigma\right) > 0; \quad \text{and}
\]

c. if \( c_1 > 0 \), then \( c_1(x) + \nabla c_i(x)^T s > 0 \).

Condition 2 can be satisfied, since \( J_A^T \) has full row rank. Condition 1 is well defined, since the assumptions of this theorem imply that the assumptions of Lemma 3.9 hold.

Since \( \varepsilon_3 \leq \varepsilon_2 \), parts (ii), (iii), and (iv) of Lemma 3.6 combined with [1, Lemma 1.27] guarantee that problem (3.24) has a unique solution, say \( s_T \), and the optimality conditions show that \((s_T, q_T)\) satisfies system (3.27), where \( q_T \) is the step from \( y \) to the Lagrange multiplier vector for problem (3.24). Note that we can make the solution \((s_T, q_T)\) arbitrarily small in norm, since the target vector in system (3.27) converges to zero as \( w \) converges to \( w^* \). This observation, (3.26), and the assumption \( \sigma > \| y^* \|_{\infty} \) guarantee that we can satisfy conditions 3a and 3b for some \( \kappa_\sigma > 0 \).

Now let \( w \in B_{\varepsilon_3}(w^*) \), \( M \in S_{\varepsilon_3} \), and \((s_T, q_T)\) denote the solution to problem (3.24) so that it satisfies system (3.27). For convenience we “scatter” the vector \( q_T \), which has length equal to the size of the indexing set \( A \), into a vector \( q_{i_T} \in \mathbb{R}^m \) so that \( [q_{i_T}]_i = 0 \) if \( i \notin A \). We also partition the constraints up into four types: I, II, III, and IV (see Figure 1); condition 3c and the properties of \( s_T \) guarantee that these are the only possibilities. Note that \( \nabla c_i(x)^T s_T < 0 \) for \( i \in I \), \( \nabla c_i(x)^T s_T = 0 \) for \( i \in II \), and \( \nabla c_i(x)^T s_T > 0 \) for \( i \in III \). It then follows from system (3.27), the definitions of \( q_{i_T} \)
and $z$, condition 3a, and the definition of $\kappa$ that

\[
(g(x) + \sigma J(x)^T z) s_T
= -s_T^T M s_T + (J_A(x)s_T)^T[y + q_T]_A + \sigma s_T^T (J(x)s_T)
= -s_T^T M s_T + \sum_{i \in I}(\nabla c_i(x)^T s_T)[y + q_T]_i + \sum_{i \in III}(\nabla c_i(x)^T s_T)[y + q_T - \sigma c]_i
\leq -s_T^T M s_T + \frac{\kappa}{2} \sum_{i \in I}(\nabla c_i(x)^T s_T) - \kappa \sigma \sum_{i \in III}(\nabla c_i(x)^T s_T)
\leq -s_T^T M s_T - \kappa \|J_A(x)s_T\|_1.
\]

(3.29)

We now develop a lower bound on $\|J_A(x)s_T\|_1$. If we let $s_T = s_T^N + s_T^F$ be the orthogonal decomposition of $s_T$ such that $s_T^N \in \text{Range}(J_A(x)^T)$ and $s_T^F \in \text{null}(J_A(x))$, then it follows that there exists a vector $r$ such that $J_A(x)^T r = s_T^N$, and, therefore,

(3.30) \quad $\|s_T^N\|_2 \leq \|J_A(x)^T\|_2 \|r\|_2$ and $J_A(x)s_T = J_A(x)s_T^N = J_A(x)J_A(x)^T r$.

Using the nonsingularity of $J_A(x)J_A(x)^T$ and norm inequalities, we have

(3.31) \quad $\|r\|_2 \leq \|\(J_A(x)J_A(x)^T\)^{-1}\|_2 \|J_A(x)s_T\|_2$.

This inequality, (3.30), and condition 2 imply

(3.32) \quad $\|J_A(x)s_T\|_2 \geq \frac{\|r\|_2}{\|\(J_A(x)J_A(x)^T\)^{-1}\|_2} \geq \frac{\|s_T^N\|_2}{\|J_A(x)^T\|_2} \geq \frac{\|s_T^N\|_2}{\kappa J^3}$.

Using this inequality, norm inequalities, and the fact that $\|s_T^N\|_2 = \sin(\theta)\|s_T\|_2$, we have

(3.33) \quad $\|J_A(x)s_T\|_1 \geq \|J_A(x)s_T\|_2 \geq (\sin(\theta)\|s_T\|_2)/\kappa J$.
Combining this with (3.29) we have
\begin{equation}
(g(x) + σJ(x)^T z)^T s_τ \leq -s_τ^T M s_τ - (κ \sin(θ)/κ_j) s_τ^T s_τ.
\end{equation}
(3.34)

We consider two cases. First suppose that $s_τ^T M s_τ \geq (λ_{min}/8)s_τ^T s_τ$. Then it immediately follows from (3.34) that
\begin{equation}
(g(x) + σJ(x)^T z)^T s_τ \leq -(λ_{min}/8)|s_τ|^2.
\end{equation}
(3.35)

Next, suppose that $s_τ^T M s_τ < (λ_{min}/8)s_τ^T s_τ$. Lemma 3.9 then implies that $0 < \tilde{θ} < θ$, and, therefore, $0 < \sin(\tilde{θ}) < \sin(θ)$. We can then use this fact, (3.34), the Cauchy–Schwarz inequality, the definition of $β_{max}$, and condition 3b to conclude that
\begin{equation}
(g(x) + σJ(x)^T z)^T s_τ \leq |s_τ|^2 M_{ij} - (κ \sin(\tilde{θ}))|s_τ|_{ij}/κ_i
\end{equation}
\begin{equation}
\leq |s_τ|_{ij} (β_{max}|s_τ|_2 - (κ \sin(\tilde{θ}))/κ_i)
\leq -(κ \sin(\tilde{θ})/2κ_i)|s_τ|_{ij}^2 \leq -(κ \sin(\tilde{θ})/2κ_i)|s_τ|^2.
\end{equation}
(3.36)

If we define
\begin{equation}
c_2 = \min \left(\frac{λ_{min}}{8}, \frac{κ \sin(\tilde{θ})}{2κ_i}\right) > 0,
\end{equation}
then it follows from (3.35) and (3.36) that
\begin{equation}
(g(x) + σJ(x)^T z)^T s_τ \leq -c_2|s_τ|^2,
\end{equation}
(3.37)

which completes the proof.

With a little more effort, we can show that the step from the Cauchy step $s^c_k$ to the solution of problem (3.24) is a descent direction for the underlying models. Since the Cauchy step is computed from the predictor step, it is imperative that we choose $B_k$ so that $s^k$ has desirable properties. The results in section 3.1 suggest that we make the following assumption.

**Assumption 3.1.** There exists a number $λ_{max}^B > 0$ such that the sequence of positive-definite matrices $\{B_k\}$ defined in Algorithm 2.1 satisfies
\begin{equation}
s^T B_k s \geq λ_{min}^B s^T s \text{ for all } s \in \mathbb{R}^n \text{ and all } k \geq 0.
\end{equation}
(3.38)

We now show that in the neighborhood of a solution $w^*$, the (unique) solution to
\begin{equation}
(g_k + Ms^c_k)^T s + M^T s = 0 \text{ subject to } c_A(x_k) + J_A(x_k)(s^c_k + s) = 0
\end{equation}
(3.39)

is a descent direction for the underlying model determined by the matrix $M$ (under certain assumptions).

**Lemma 3.11.** Let $w^*$ be a minimizer for problem (NP) that satisfies the LICQ and strict complementarity, and assume that $σ_k > \|y^*\|_{∞}$, that Assumption 3.1 holds, and that $\|B_k\|_{2} \leq b_B$ for some $b_B > 0$. It follows that there exist positive numbers $c_2$ and $ε_4$ such that if iterate $k - 1$ is successful, $w_k \in B_{ε_4}(w^*)$, and $M \in S_{ε_4}$, then problem (3.39) is well defined and the solution $s_k$ satisfies
\begin{equation}
(g_k + Ms^c_k + σJ^T z_k)^T s_k < -c_2|s_k|^2 \text{ for } |z_k| = \begin{cases} 0 & \text{if } i \in \mathcal{V}_k, \\ -1 & \text{if } i \in \mathcal{S}_k, \end{cases}
\end{equation}
(3.40)

where $\mathcal{V}_k = \{i : [c_k + J_k s^c_k]_i < 0\}$ and $\mathcal{S}_k = \{i : [c_k + J_k s^c_k]_i \geq 0\}$.
Lemma \(\text{EQP}\) as discussed in [17, section 2.3.2] and restated on page 2052. The system that arises in place of (3.27) is
\[
\left( \begin{array}{c}
M \\
J_A(x) \\
0
\end{array} \right) \left( \begin{array}{c}
s \\
g(x) - J_A(x)^T y_A + M s_k^{\text{CP}}
\end{array} \right) = - \left( \begin{array}{c}
- q \\
c_A(x) + J_A(x_k) s_k^{\text{CP}}
\end{array} \right),
\]
but since \(\|s_k^{\text{CP}}\|_2 \leq \|s_k\|_2\), we can ensure—by possibly decreasing \(\varepsilon_4\)—that parts 3a and 3b of Lemma 3.10 are once again satisfied. The rest of the proof is identical to Lemma 3.10.

3.3. Local convergence with an (EQP) step. Our first rate-of-convergence result for Algorithm 2.1 assumes that the accelerator step is computed from subproblem (EQP) as discussed in [17, section 2.3.2] and restated on page 2052.

**Theorem 3.12 ((EQP) local convergence result).** Let \(w^* = (x^*, y^*)\) be a minimizer for problem (NP) that satisfies the strong second-order sufficient conditions as given by Definition 3.5. Let Assumption 3.1 hold, and suppose that \(\sigma_k \equiv \sigma_b > \|y^*\|_\infty\)
and \( \|B_k\|_2 \leq b_B \) for some \( b_B > 0 \) and \( \sigma_k > 0 \) and for all \( k \geq 0 \), the accelerator step is computed from subproblem (EQP) with the choice \( H_k = \nabla_{xx}^2 \mathcal{L}(x_k, y_k^*) \), and \( \max J \) fails \( \geq 1 \) in Algorithm 2.1. It follows that there exists a positive number \( \delta \) such that if the accelerator step is computed for every iteration once the first successful iterate of Algorithm 2.1 is contained in \( B_3(w^*) \), 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 \( \nabla_{xx}^2 \mathcal{L}(x, y) \) is Lipschitz continuous in a neighborhood of \( (x^*, y^*) \), then they converge at a Q-quadratic and R-quadratic rate, respectively.

Proof. Set \( \lambda_{\min} = \min(\lambda_{\min}^H/2, \lambda_{\max}^B) \) and \( \beta_{\max} = \max(b_B, \|\nabla_{xx}^2 \mathcal{L}(x^*, y^*)\|_2 + 1) \) in the definition of \( S(x; x^*) \) as given by (3.3), and let \( \beta, \varepsilon_1, \varepsilon_2, \) and \( \mu \) be the positive constants guaranteed by Lemmas 3.6 and 3.7; note that they satisfy \( 0 < \mu \leq \varepsilon_2 \leq \varepsilon_1 \) by construction, so that part (ii) of Lemma 3.6 implies

\[
B_{\mu/2}(w^*) \subseteq B_\mu(w^*) \subseteq B_{\varepsilon_2}(w^*) \subseteq B_{\varepsilon_1}(w^*) \quad \text{and} \quad S_{\mu/2} \subseteq S_{\mu} \subseteq S_{\varepsilon_2} \subseteq S_{\varepsilon_1},
\]

where \( S_\varepsilon \) is defined by (3.3) and (3.4). By possibly decreasing \( \mu \), we can also guarantee that if \( w \) and \( \bar{w} \) are contained in \( B_\mu(w^*) \), then the following conditions are satisfied:

C1. \( \|y - y^*\|_\infty < \sigma_0 - \|y^*\|_\infty \);  
C2. \( \|\nabla_{xx} \mathcal{L}(x, y^*(x))\|_2 \leq \|\nabla_{xx} \mathcal{L}(x^*, y^*)\|_2 + 1 \), where \( y^*(x) \) is any estimate satisfying \( y^*(x) - y^* = O(\|x - x^*\|_2) \);  
C3. \( s^T \nabla_{xx} \mathcal{L}(x, y^*(x)) \geq (\lambda_{\min}^H/2)s^T s \) for all \( s \) satisfying \( J_\lambda^s = 0 \).

C4. Newton's method applied to the function \( F_N \) in (3.1) converges from the point \( w \) to \( w^* \); moreover, the Newton update \( w_+ \) to \( w \) satisfies \( \|w_+ - w^*\|_2 \leq \|w - w^*\|_2 \) (see Dennis and Schnabel [11, Theorem 5.2.1]).

With \( \mu \) defined, we now pick \( \delta_\Delta > 0 \) so that

C5. \( \delta_\Delta \leq \min(\mu/2, \varepsilon_4) \), where \( \varepsilon_4 \) is defined in Lemma 3.11; and  
C6. \( \delta_\Delta \leq \eta_c \Delta_R/2 \), where \( 0 < \Delta_R \leq \Delta_U \) and \( \eta_c \) are used in Algorithm 2.1.

Finally, we choose \( \delta > 0 \) so that

C7. \( \delta \leq \min(\mu/2, \varepsilon_4) \), where \( \varepsilon_4 \) is defined in Lemma 3.11; and  
C8. if \( w \in B_\delta(w^*) \), then the following bound on the KKT equality conditions is satisfied:

\[
\|f_{\text{KKT}}(w)\|_2 \leq \left\| \left( \begin{array}{c} g(x) - J(x)^T y \\ c(x) \cdot y \end{array} \right) \right\|_2 \leq \frac{1}{4\beta} \min(\delta_\Delta, \eta_c \Delta_R).
\]

Now let \( k = 1 \) be the first successful iterate generated by Algorithm 2.1 such that \( w_k \in B_\delta(w^*) \). By construction of Algorithm 2.1 and the fact that the accelerator trust-region scale factor satisfies \( \tau_f \geq 1 \), we have

\[
\Delta_k^L \geq \Delta_R > 0 \quad \text{and} \quad \Delta_k^R \geq \tau_f \Delta_R \geq \Delta_R > 0.
\]

Since (3.42) and C7 imply that \( w_k \in B_{\mu/2}(w^*) \), it follows from C8, Lemma 3.7, and (3.43) that \( J_\lambda(x_k) \) has full row rank, and if \( M \in S_{\mu/2} \), then \( x_k(M) \) correctly identifies the optimal active set and satisfies

\[
\|x_k(M) - x_k\|_\infty \leq 2\beta \|F_M(w_k; w_k)\|_2 \leq \frac{1}{4} \min(\delta_\Delta, \eta_c \Delta_R) \leq \frac{1}{2} \eta_c \min(\Delta_k^L, \Delta_k^R).
\]

We now observe that \( B_k \in S_{\mu/2} \) by construction and is, in fact, positive definite. Furthermore, since C1 implies \( \sigma_k = \sigma_B > \|y_k(B_k)\|_\infty \) and (3.44) implies \( \|x_k(B_k) - x_k\|_\infty \leq (\eta_c/2)\Delta_k^L < \Delta_k^L \), we must have \( w_{k+1}^p = w_k(B_k) \). Thus the solution to the predictor subproblem satisfies \( s_{k+1}^p = x_k(B_k) - x_k \), correctly identifies the optimal
Fig. 3. A depiction of the scenario in Theorem 3.12. The following quantities are displayed: $x_k$ is the current iterate, $s_k^e$ is the predictor step, $x_k^p$ is the predictor point, $s_k^{CP}$ is the Cauchy step, $s_k^A$ is the accelerator step as computed from problem (EQP), $\Delta_k^F$ is the accelerator trust-region radius, $s_T$ is the solution to problem (3.39), $x_k(H_k)$ is the first $n$ components of $w_k(H_k)$, which is the closest minimizer to $w_k$ for problem (3.14) with the choice $M = H_k$, and $F$ denotes the feasible side of the constraint $[c_k + J_k s]_i \geq 0$.

Next we observe that C2 and C3 imply that $H_k \in S(x^*; x^*) \subset S_{\mu/2}$. Therefore, the point $w_k(H_k)$ is well defined, identifies the optimal active set, and is the unique minimizer of problem (3.14) in a neighborhood of $w_k$ for $M = H_k$. Since $J_A(x_k)$ has full row rank, it follows from (3.42), part (iii) of Lemma 3.6, and [1, Lemma 1.27] that subproblem (EQP) has $s_k^A$ as a unique solution. It follows that if $\|x_k(H_k) - (x_k + s_k^e)\|_2 \leq \Delta_k^s$, then $s_k^A = x_k(H_k) - (x_k + s_k^e)$ (see Figure 3). Using the triangle inequality, the definition of $w_k(B_k)$, and (3.44), we have

$$\|x_k(H_k) - (x_k + s_k^e)\|_2 \leq \|x_k(H_k) - x_k\|_2 + \|s_k^e\|_2$$

$$= \|x_k(H_k) - x_k\|_2 + \|x_k(B_k) - x_k\|_2 \leq \eta_k \Delta_k^s \leq \Delta_k^F.$$

Thus, if $s_k^e + s_k^A$ satisfies condition (2.9), then $s_k = s_k^e + s_k^A$, and it follows that $x_k + s_k = x_k(H_k)$ and $y_k^s = y_k(H_k)$. We now show that this is the case. If $s_T \neq 0$, then C5 and Lemma 3.11 show that the vector $s_T$, which satisfies $s_k^{CP} + s_T = x_k(H_k)$, is a descent direction for the model $M_k^\mu$. Therefore, $M_k^\mu(s_k^e + s_k^A) < M_k^\mu(s_k^{CP})$ so that condition (2.9) is satisfied by $s_k^e + s_k^A$. On the other hand, if $s_T = 0$, then it follows that $s_k^e = s_k^{CP}$ and $s_k^A = 0$ so that $s_k^e + s_k^A = s_k^{CP}$ trivially satisfies condition (2.9).

If $x_k + s_k$ is a successful step, then $x_{k+1} \leftarrow x_k + s_k$; otherwise, the update $x_{k+1} \leftarrow x_k + s_k$ is still made since $\max_f \text{fails} \geq 1$, but a nonmonotone phase is entered.

In either case, the vector $w_{k+1}$ is the same vector that is obtained by performing one step of Newton’s method on the function $F_N$ (see (3.1)) from the point $(x_k, y_k^s)$ with the understanding that $y_{k+1}$ is formed by “scattering” $y_k^s$ into a zero-vector of length $m$. Since Algorithm 2.1 makes the assignment $w_{k+1} \leftarrow w_k^s$, it follows from C4 that $w_{k+1} \in B_\delta(w^*)$, and so the same argument may be repeated starting from the point $w_{k+1}$; this results in a vector $w_{k+2}$ that has the same properties as $w_{k+1}$ and is, in fact, equivalent to performing one step of Newton’s method on the function...
LOCAL CONVERGENCE AND PRACTICAL ISSUES

Let \( F_N \) from the point \((x_{k+1}, y^e_{k+1})\). The only difference in the argument is that the predictor and accelerator trust-region radii are only guaranteed to be bigger than \( \eta, \Delta_k \), since the predictor trust-region radius may be contracted if the point \( w_{k+1} \) was not successful. However, conditions C1–C8 were chosen to ensure that all the previous estimates still hold. It is shown in [10, section 15.3.2.3] that this process is sufficient for avoiding the Maratos effect, provided the ratio \( r_k \) of actual to predicted decrease in the merit function is defined using the strategy in Algorithm 2.1; therefore, \( w_{k+2} \) will be accepted by the \( \ell_1 \)-merit function. This argument can clearly be repeated so that every remaining step will be accepted. As for rate-of-convergence, we have from [24, Theorem 11.2] and C2 that

\[
\begin{aligned}
(x_{k+1} - x^*) = o \left( \left\| \left( x_k - x^* \right) \right\|_2 \right) = o \left( \left\| x_k - x^* \right\|_2 \right)
\end{aligned}
\]

so that \( \{x_k\} \) and \( \{y_k\} \) converge to \( x^* \) and \( y^* \) Q-superlinearly and R-superlinearly, respectively (see [25, Chapter 9] for a description of Q and R convergence); C2 also shows that \( \{y^e_k\} \) converges to \( y^* \) R-superlinearly. If \( \nabla_x \mathcal{L}(x, y) \) is locally Lipschitz continuous, then a similar argument shows that \( \{x_k\} \) converges to \( x^* \) Q-quadratically and that \( \{y_k\} \) and \( \{y^e_k\} \) converge to \( y^* \) R-quadratically. \( \square \)

**Lemma 3.13.** Let \( w^* = (x^*, y^*) \) be a minimizer for problem (NP) that satisfies the LICQ and strict complementarity, and suppose that \( \sigma_b \geq \|y^*\|_\infty \), that Assumption 3.1 holds, and that \( |B_k|_2 \leq b_2 \) for some \( b_2 > 0 \) and all \( k \geq 0 \). Then there exists a scalar \( \delta > 0 \) such that if iterate \( k - 1 \) is successful and \( w_k \in B_{b_1}(w^*) \), then

\[
\begin{aligned}
y^e_k - y^* = O(\|x_k - x^*\|_2) \quad \text{and} \quad [y^e_k]_2 = 0,
\end{aligned}
\]

where \( y^e_k \) are the multipliers for the predictor subproblem (2.2).

**Proof.** Let \( \delta \) be defined to satisfy conditions C5–C8 of Theorem 3.12. It follows, just as in the proof of Theorem 3.12, that \( s^e_k = x_k(B_k) - x_k \) and that \( s^e_k \) is the unique solution to problem (3.24) with the choice \( \tilde{M} = B_k \). This implies that \( (s^e_k, y^e_k) \) satisfies system (3.7) \( (\pi = y^e_k) \) so that (3.46) follows from (3.6). \( \square \)

### 3.4. Local convergence with an (EIQP) step.

We now consider the rate-of-convergence for Algorithm 2.1 when the accelerator step is computed from subproblem (EIQP) as described in [17, section 2.3.1] and restated on page 2052.

We begin by making two observations. First, since problem (EIQP) is generally a nonconvex inequality constrained QP, we will need to assume that the solution \( s^e_k \) is one of minimal norm; a similar assumption is made by Robinson in [30, section 3]. Although this assumption is not ideal, it is not too offensive within our setting; if we use an active set QP solver with a hot start based on the active set obtained from the predictor step, then the solution to subproblem (EIQP) will ultimately be the same as the solution to subproblem (EQP). Theorem 3.12 validates that this is a good step, and, therefore, if this strategy is used, then the “minimum-norm solution” assumption is not necessary. The second observation is that if the accelerator step is chosen to be one of minimal norm, then the proof of Theorem 3.12 carries over, since (1) the Cauchy step \( s^e_k \) satisfies \( ||s^e_k||_\infty \leq \|s^e_k\|_\infty \); (2) the vector \( x_k(H_k) - s^e_k \) is a solution to subproblem (EIQP); and (3) Lemma 3.11 guarantees that the descent-constraint does not interfere with the step from \( s^e_k \) to \( x_k(H_k) \).

**Theorem 3.14 (EIQP local convergence result).** Let \( w^* = (x^*, y^*) \) be a minimizer for problem (NP) that satisfies the strong second-order sufficient conditions as given by Definition 3.5. Let Assumption 3.1 hold, and assume that \( \sigma_k \equiv \sigma_b > \|y^*\|_\infty \)
and \( \|B_k\|_2 \leq b_T \) for some \( b_T > 0 \) and \( \sigma_k > 0 \) and all \( k \geq 0 \), the accelerator step is computed from subproblem (EIQP) with the choice \( H_k = \nabla_{xx} L(x_k, y_k^*) \), and \( \max \delta_k \geq 1 \) in Algorithm 2.1. It follows that there exists a positive number \( \delta \) 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 is contained in \( B_\delta(w^*) \), 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 \( \nabla_{xx} L(x, y) \) is Lipschitz continuous in a neighborhood of \((x^*, y^*)\), then they converge at a Q-quadratic and R-quadratic rate, respectively.

Proof. The result follows from the proof of Theorem 3.12, the discussion above, and Lemma 3.10.

4. Practical issues concerning \( B_k \) and \( \sigma \). In this section we address two components of Algorithm 2.1 that are important for an efficient implementation. First, we describe a strategy for defining the positive-definite matrix \( B_k \) via a limited-memory BFGS update. Second, we briefly consider a simple strategy for updating the penalty parameter.

4.1. A limited-memory BFGS update. In this section we describe a method for defining the positive-definite matrix needed in the computation of the predictor step (2.1) that is based on the limited-memory BFGS update. We must be cautious, however, since the matrix \( \nabla_{xx} L(x, y) \) is generally indefinite, and, therefore, the traditional update may result in an indefinite matrix [2, 27]. We also note that if the problem dimension is small, then a full BFGS update is practical.

The limited-memory BFGS update uses a fixed number of vectors, say \( l \), to define a positive-definite approximation to \( \nabla_{xx} L(x_k, y_k) \) based on the most recent \( l \) iterations (for more details see [24, 2]). If we define \( d_k = \nabla_x L(x_k + s_k, y_{k+1}) - \nabla_x L(x_k, y_{k+1}) \), then we may write the update as

\[
B_k = B_k^0 + \sum_{i=k-l}^{k-1} (q_i q_i^T - p_i p_i^T),
\]

where \( B_k^0 \) denotes any initial positive-definite approximation to \( \nabla_{xx} L(x_k, y_k) \) and

\[
p_i = \frac{B_i s_i}{(s_i^T B_i s_i)^{1/2}}, \quad q_i = \frac{d_i}{(d_i^T s_i)^{1/2}}, \quad \text{and} \quad B_i = B_k^0 + \sum_{j=k-l}^{i-1} (q_j q_j^T - p_j p_j^T).
\]

Note that in these definitions we have assumed that \( k \geq l-1 \) so that there are \( l \) vectors to use. This formula is relatively simple, but one must be careful. It is tempting to store the vector-pairs \((p_i, q_i)\). However, as (4.2) illustrates, the vector \( p_i \) is defined from \( B_i \) and the matrix \( B_i \) changes from iteration to iteration, since the “oldest” vector-pair \((s_i, d_i)\) is removed from the set of \( l \) vector-pairs. Hence, the vector \( p_i \) must be recomputed at each iteration. The relationships given by (4.2) suggest how this may be done, since

\[
B_i s_i = B_k^0 s_i + \sum_{j=k-l}^{i-1} [(q_j^T s_i) q_j - (p_j^T s_i) p_j].
\]

Algorithm 4.1, which is [24, Procedure 7.6], computes the vector-pair \((p_i, q_i)\) recursively.
ALGORITHM 4.1 COMPUTING THE VECTOR-PAIRS \((p_i, q_i)\).

\[ \text{for } i = k - l, k - l + 1, \ldots, k - 1 \]

\[ q_i \leftarrow d_i / (d_i^T s_i)^{1/2} \]

\[ p_i \leftarrow B_0^k s_i + \sum_{j=k-l}^{i-1} [(q_j^T s_i)(q_j - (p_j^T s_i)p_j] \]

\[ p_i \leftarrow p_i / (s_i^T p_i)^{1/2} \]

end (for)

During the \(k\)th iteration, Algorithm 4.1 computes the values \(q_i\) for \(k - l \leq i \leq k - 1\) and \(q_j^T s_i\) for all \(k - l \leq j \leq i - 1\). However, since \(q_i\) depends only on the data \((d_i, s_i)\), only the value \(q_{k-1}\) and values \(q_j^T s_{k-1}\) \((k - l \leq j \leq k - 2)\) need to be computed (the other quantities should be stored from previous iterations).

Once the vector-pairs \((p_i, q_i)\) have been computed, we set \(B_k = B_k^0 - PP^T + QQ^T\), where we have defined \(P = [p_k - p_{k-1} \ldots p_1]\) and \(Q = [q_k - q_{k-1} \ldots q_1]\). The predictor subproblem (2.2) then becomes

\[
\begin{align*}
\text{minimize } & \quad f_k + g_k^T s + \frac{1}{2} s^T (B_k^0 - PP^T + QQ^T) s + \sigma e^T v \\
\text{subject to } & \quad c_k + J_k s + v \geq 0, \quad v \geq 0, \quad \|s\|_\infty \leq \Delta_k^p.
\end{align*}
\]

If we define the \(2l\) extra variables

\[ w_a = P^T s \quad \text{and} \quad w_b = Q^T s, \]

then problem (4.4) is equivalent to

\[
\begin{align*}
\text{minimize } & \quad f_k + \frac{1}{2} (s^T B_k^0 s - w_a^T w_a + w_b^T w_b) + \sigma e^T v \\
\text{subject to } & \quad c_k + J_k s + v \geq 0, \quad P^T s = w_a, \quad Q^T s = w_b, \quad v \geq 0, \quad \|s\|_\infty \leq \Delta_k^p.
\end{align*}
\]

As a function of \((s, v, w_a, w_b)\), the Hessian associated with subproblem (4.6) is given by

\[
B_k^* = \begin{pmatrix}
B_k^0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & -I & 0 \\
0 & 0 & 0 & I
\end{pmatrix},
\]

which is \textit{not} positive definite. This may seem strange, since problem (4.6) is equivalent to the \textit{strictly convex} QP (4.4) (assuming that the updated matrix was positive definite). However, if the current iterate is feasible for subproblem (4.6), then any step that maintains linear feasibility is guaranteed to be a direction of positive curvature even though \(B_k^*\) is indefinite. To see this, suppose that \((s, v, w_a, w_b)\) is a feasible point so that \(w_a = P^T s\) and \(w_b = Q^T s\). Furthermore, suppose that \(P^T (s + \Delta s) = w_a + \Delta w_a\) and \(Q^T (s + \Delta s) = w_b + \Delta w_b\). Simplification yields \(P^T \Delta s = \Delta w_a\) and \(Q^T \Delta s = \Delta w_b\). It then follows that

\[
(\Delta s, \Delta v, \Delta w_a, \Delta w_b)^T B_k^* (\Delta s, \Delta v, \Delta w_a, \Delta w_b) = \Delta s^T B_k^0 \Delta s - \Delta w_a^T \Delta w_a + \Delta w_b^T \Delta w_b
\]

\[
= \Delta s^T (B_k^0 - PP^T + QQ^T) \Delta s = \Delta s^T B_k \Delta s > 0.
\]
since $B_k$ is positive definite by construction. A great advantage in using subproblem (4.6) is that the Hessian matrix has essentially the same sparsity as $B_k$. In contrast, the Hessian matrix associated with subproblem (4.4) is generally dense, since it uses a sum of rank-1 updates. Note, however, that the 2l extra constraints (4.5) are generally dense; fortunately a limited number of dense constraints can be accommodated easily by modern sparse QP solvers such as QPA and QPB from the GALAHAD library [15].

Until this point we have assumed that the limited-memory BFGS update results in a positive-definite matrix. However, it is well known that this is true if and only if the quantity $d_k^T s_k > 0$, and this is not guaranteed to hold. When the resulting update is not positive definite, then perhaps the simplest strategy is to use the damping technique introduced by Powell [27]. Basically, this approach modifies $d_k$ so that the resulting update is (sufficiently) positive definite.

We have explored other variants of this basic idea in [16, section 2.2].

4.2. Updating the penalty parameter. The updating scheme that we now discuss is based on the simple idea of calculating a sequence of approximate solutions to problem ($\ell_1$-$\sigma$). After each approximate solution is computed, we check the constraint violation and if sufficient improvement is not obtained, then the penalty parameter is increased with the intent of driving the constraint violation to zero. Since the penalty parameter is now allowed to change over a sequence of iterations, we let $\sigma_k$ denote the penalty parameter during the $k$th iterate. We accept the vector-pair $(x_k, \pi_k)$ as an approximate solution for problem ($\ell_1$-$\sigma$) if it satisfies

\[
\begin{align*}
\varepsilon_k^p & \geq \frac{\|g_k + \sigma_k J_k^T \pi_k\|_\infty}{1 + \|g_k\|_\infty}, \\
[\pi_k]_i & = \begin{cases} 
-\varepsilon_k^p \sigma_k, & \text{if } [c_k]_i > \varepsilon_k^p, \\
-1 - \varepsilon_k^p \sigma_k, & \text{if } -\varepsilon_k^p \leq [c_k]_i \leq \varepsilon_k^p, \\
-1 - \varepsilon_k^p, & \text{if } [c_k]_i < -\varepsilon_k^p,
\end{cases}
\end{align*}
\]

where $\varepsilon_k^p$, $\varepsilon_k^d$, and $\varepsilon_k^c$ denote the $k$th primal, dual, and complementary-slackness tolerances, respectively, for problem ($\ell_1$-$\sigma$). These conditions are based on the optimality conditions for an exact minimizer $(x, \pi)$, which are given by $g(x) + \sigma_k J(x)^T \pi = 0$ for $\pi \in \partial \|c(x)\|_1$ (see [12, section 14.3] for more details).

In practice, we define estimates $\pi_k = -y_k^c / \sigma_k$, where $y_k^c$ is the Lagrange multiplier vector for problem (2.2). Provided the sequence $\{y_k^c\}$ converges to a Lagrange multiplier vector for the elastic version of problem ($\ell_1$-$\sigma$), this strategy will eventually produce a vector-pair $(x_k, \pi_k)$ satisfying (4.8). We note that alternatives, such as defining $\pi_k = -y_k^c / \sigma_k$ for accelerator multipliers $y_k^a$ or defining $\pi_k$ as a solution of the optimization problem

\[
\begin{align*}
\min_{\pi \in \mathbb{R}^m} & \frac{1}{2} \|g_k + \sigma_k J_k^T \pi\|_2^2 \\
\text{subject to } & \pi \text{ satisfying (4.8b)},
\end{align*}
\]

may be used.

Algorithm 4.2 provides the pseudocode for updating the penalty parameter as well as the additional parameter initiations that must be made.
Algorithm 4.2 Updating $\sigma$ based on approximate critical points to problem ($\ell_1$-$\sigma$).

begin (additions to preamble of Algorithm 2.1)
Choose $\sigma_0 > 0$, $\eta_0 > 0$, $0 < \varepsilon_c < 1$, $0 < \varepsilon^p_0 < \varepsilon_c \eta_0$, and $1 < \sigma_c$.
Set $\varepsilon^p_0 = \varepsilon^C_0 = \varepsilon^D_0$.
end (additions to preamble of Algorithm 2.1)

if $(x_k, \pi_k)$ satisfies condition (4.8), then
if $c(x_k) \geq -\eta_k e$, then [an approximate critical point]
$\eta_{k+1} \leftarrow \eta_k$ [successful]
$\varepsilon^p_{k+1} \leftarrow \varepsilon_c \eta_{k+1}$ [decrease $\eta_k$]
$\varepsilon^D_{k+1} \leftarrow \varepsilon^p_{k+1}$, $\varepsilon^C_{k+1} \leftarrow \varepsilon^D_{k+1}$
$\sigma_{k+1} \leftarrow \sigma_k$ [ensure that $\varepsilon^p_k$ is less than $\eta_k$]

else [unsuccessful]
$\eta_{k+1} \leftarrow \eta_k$
$\varepsilon^p_{k+1} \leftarrow \varepsilon^p_k$
$\varepsilon^D_{k+1} \leftarrow \varepsilon^D_k$, $\varepsilon^C_{k+1} \leftarrow \varepsilon^D_k$
$\sigma_{k+1} \leftarrow \sigma e \sigma_k$ [increase $\sigma_k$]

end if
else [not an approximate critical point]
$\eta_{k+1} \leftarrow \eta_k$, $\varepsilon^p_{k+1} \leftarrow \varepsilon^p_k$, $\varepsilon^D_{k+1} \leftarrow \varepsilon^D_k$, $\varepsilon^C_{k+1} \leftarrow \varepsilon^D_k$, $\sigma_{k+1} \leftarrow \sigma_k$

end if

For simplicity, we have defined $\varepsilon^p_k = \varepsilon^C_k = \varepsilon^D_k$. However, all that is required is that $\lim_{k \to \infty} \varepsilon^p_k = \lim_{k \to \infty} \varepsilon^D_k = \lim_{k \to \infty} \varepsilon^C_k = 0$.

For numerical considerations, it is generally undesirable to let the penalty parameter grow “too large.” However, there are two situations in which the penalty parameter should converge to infinity. The first is when the problem contains infeasible constraints. Detecting this situation is difficult and is equivalent to showing that the global solution of

\[(4.10) \quad \text{minimize} \quad ||[c(x)]^-||_1\]

is strictly positive (see [4] for some recent work on this topic). The second situation occurs when the iterates converge to a critical point of problem (4.10) for which $||[c(x)]^-||_1 > 0$. This undesirable situation may occur for all penalty methods, but it is rarely encountered in practice. Barring these two situations and under reasonable assumptions, Theorem 4.2 below shows that the penalty parameter remains uniformly bounded and that we can expect to generate an approximate solution to problem (NP) in a finite number of iterations. We use the following definition.

Definition 4.1. A point $x$ is a first-order critical point for problem (4.10) if it satisfies

\[(4.11) \quad J(x)^T y = 0\]

for some $y \in \partial ||[c(x)]^-||_1$.

For given primal, dual, and complementary-slackness tolerances $\tau_p$, $\tau_d$, and $\tau_c$, respectively, we say that a vector-pair $(x_k, y_k)$ is an approximate solution to problem
(NP) if it satisfies

\[
\frac{\|g_k - J_k^T y_k\|_\infty}{1 + \|g_k\|_\infty} \leq \tau_d,
\]

\[
c_k \geq -\tau_p \epsilon,
\]

\[
y_k \geq -\tau_\epsilon \epsilon,
\]

\[
\max(|c_k|, |y_k|) \leq \tau_\epsilon \epsilon,
\]  

where condition (4.12d) should be interpreted componentwise.

**Theorem 4.2.** Let the assumptions for global convergence hold [17, Theorem 4.3], and let \( \{x_k\} \) be the sequence of iterates generated by Algorithm 2.1 with penalty parameter update given by Algorithm 4.2. Assume that at all limit points \( x_\ast \) of \( \{x_k\} \) the Jacobian of active constraints has full row rank and if \( x_\ast \) is a first-order critical point for problem (4.10), then \( \|c(x_\ast)\|_1 = 0 \). Then

(i) the penalty parameter remains uniformly bounded; and 

(ii) if \( \tau_p, \tau_d, \) and \( \tau_\epsilon \) denote positive primal, dual, and complementary-slackness tolerances, respectively, for problem (NP), then the algorithm described in this theorem terminates in a finite number of iterations with an approximate solution to problem (NP) as given by (4.12), where \( y_k \equiv -\sigma_k \pi_k \) and \( (x_k, \pi_k) \) is an approximate solution to (4.1) as given by (4.8) for the value \( \sigma_k \).

**Proof.** The proof of these statements is a relatively straightforward exercise. For a detailed proof see [16, Theorem 3.3].

We close this section by mentioning two potential drawbacks associated with using Algorithm 4.2. First, if the initial penalty parameter is substantially smaller than the threshold value required to guarantee convergence [10, Theorem 14.5.1], then Algorithm 4.2 may be laborious, since it is based on computing a sequence of approximate minimizers of the merit function. We also note that when the penalty parameter is too small, the merit function may not even have a well-defined minimizer [6, Example 1]. Second, even if the merit function does have a well-defined minimizer, there may not exist a strictly decreasing path that connects a poor initial point \( x_0 \) to this minimizer [6, Example 2]. A possible way of avoiding these situations is to dynamically update the penalty parameter based on linear infeasibility. The so-called steering method is based on this idea and has been studied by Byrd et al. [5] and Byrd, Nocedal, and Waltz [6]. Their algorithm is composed of essentially two stages that we now briefly describe using our notation. If we denote the current penalty parameter by \( \sigma_c \), then the first stage is to compute a step \( s_\infty \) that locally minimizes the linearized constraint violation; this can be viewed as essentially solving the predictor subproblem with penalty parameter \( \sigma = \infty \). The second stage is to compute a predictor step \( s_k^p \) and a new penalty parameter \( \sigma_N \) that satisfy the following conditions: (i) the decrease in the linearized constraint violation obtained from \( s_k^p \) must be at least a fixed multiple of the decrease obtained from \( s_\infty \); and (ii) the decrease in the faithful model must respect the progress made by \( s_k^p \) on the linearized infeasibility by satisfying

\[
\Delta M_k^N(s_k^p) \geq \varepsilon_d \sigma_N (||c_k||_1 - ||c_k + J_k s_k^p||_1),
\]

where the constant \( \varepsilon_d \) satisfies \( 0 < \varepsilon_d < 1 \) (note that \( \Delta M_k^N(s_k) \) depends on \( \sigma_N \), although the notation does not make this explicit). The authors present three compelling examples that elucidate the strengths of this approach. For this approach to be beneficial, however, the additional cost must be offset by the “superior” values for the penalty parameter. This dynamic strategy is used in a sequential linear
quadratic programming method that is part of the KNITRO software package [33], and the authors report results that are superior to static penalty updating strategies.

We take the stance that both approaches should be available to the user. If a reasonable estimate of the size of the entire sequence of computed multiplier estimates is known in advance, then steering is likely to be less efficient because of the potential overhead associated with the method. However, since this is usually not the case, we generally recommend steering.

5. Numerical results. Preliminary testing of Algorithm 2.1 was performed on the Hock–Schittkowski (HS) [19] test problems. The HS test suite is comprised of generally small and dense problems that are very useful during early stages of code development; the small size of the problems allows for relatively careful inspection of each problem. We note that problem HS87 has been removed from the test set, since the objective function is not continuous.

To be precise, we tested three variants of Algorithm 2.1—they differ in how we compute the accelerator step and update the penalty parameter. In the first variant we computed the accelerator step from the inequality constrained subproblem (EIQP) on page 2052 and updated the penalty parameter by using “steering” as briefly described at the end of section 4.2. In the second variant we computed the accelerator step from the equality constrained subproblem (EQP) on page 2052 and again used steering to update the penalty parameter. Finally, in the third variant we computed the accelerator step from the inequality constrained subproblem (EIQP) and updated the penalty parameter by using Algorithm 4.2.

Since the problems in the test set are of small dimension, we chose to update the positive-definite matrix $B_k$ in the predictor subproblem (2.2) by using the BFGS update. To perform this update, we used the vectors $s_k$ and $d_k = \nabla x_L(x_k + s_k, y_k + 1) - \nabla x_L(x_k, y_k + 1)$. If these vectors did not result in a sufficiently positive-definite update, then we used the damping technique introduced by Powell [27]. For simplicity, we chose $B_0 = I$.

In all cases, we chose $H_k \equiv \nabla^2_{xx} L(x_k, y^p_k)$ during the computation of the accelerator and Cauchy step, where $y^p_k$ is the multiplier vector from the predictor subproblem. We solved both the (convex) quadratic program (2.2) and the (generally indefinite) quadratic program (EQP) using the GALAHAD [15] package QPC, which is a “crossover” QP solver. In the first phase, QPC calls the GALAHAD interior-point QP solver QPB [9] to compute an approximate solution and an estimate of the optimal active set. In the second phase, QPC calls the GALAHAD active set QP solver QPA [18] to “refine” the approximate solution from the first phase. To solve the equality constrained QP (EQP) we used the GALAHAD package EQP, which has been designed to solve problems of precisely this form. We should mention that most of the GALAHAD packages, including the QP solvers mentioned above, use the sparse solvers MA48 and MA57 from [20] to handle the required systems. The modular design of all the GALAHAD packages makes it easy to call these subroutines as needed.

The following parameters were used in all cases: primal/dual/complementarity slackness tolerances $\tau_p = \tau_d = \tau_c = 1.0e^{-5}$, successful/very successful tolerances $\eta_s = 0.01$ and $\eta_{0s} = 0.7$, maximum predictor trust-region radius $\Delta_p = 1000$, trust-region “reset” radius $\Delta_R = 1.0e^{-4}$, accelerator trust-region scale factor $\tau_f = 4.0$, number of nonmonotone steps allowed $\text{max fails} = 1$, and trust-region contraction and expansion factors $\eta_c = 0.1$ and $\eta_e = 5.0$. We used an initial penalty parameter of $\sigma = 1.0$ for the first and second strategies, and we used an initial penalty parameter of $\sigma = 9.0$ for the third strategy. The larger (seemingly arbitrary) initial penalty
Table 1

We record the number of function and gradient evaluations required and note that the difference between these values indicates the number of unsuccessful steps attempted.

<table>
<thead>
<tr>
<th>Prob.</th>
<th>#fc</th>
<th>#gJ</th>
<th>#fc</th>
<th>#gJ</th>
<th>#fc</th>
<th>#gJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>HS1</td>
<td>46</td>
<td>25</td>
<td>39</td>
<td>24</td>
<td>46</td>
<td>25</td>
</tr>
<tr>
<td>HS2</td>
<td>11</td>
<td>8</td>
<td>11</td>
<td>8</td>
<td>11</td>
<td>8</td>
</tr>
<tr>
<td>HS3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>HS4</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>HS5</td>
<td>10</td>
<td>6</td>
<td>9</td>
<td>6</td>
<td>10</td>
<td>6</td>
</tr>
<tr>
<td>HS6</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>HS7</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>8</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>HS8</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>HS9</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>HS10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>HS11</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>HS12</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>6</td>
<td>14</td>
<td>11</td>
</tr>
<tr>
<td>HS13</td>
<td>15</td>
<td>12</td>
<td>29</td>
<td>27</td>
<td>56</td>
<td>54</td>
</tr>
<tr>
<td>HS14</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>HS15</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>HS16</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>HS17</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>HS18</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>HS19</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>HS20</td>
<td>8</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>16</td>
<td>15</td>
</tr>
<tr>
<td>HS21</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>HS22</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>HS23</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>HS24</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>HS25</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>HS26</td>
<td>17</td>
<td>17</td>
<td>16</td>
<td>16</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>HS27</td>
<td>12</td>
<td>11</td>
<td>14</td>
<td>13</td>
<td>13</td>
<td>12</td>
</tr>
<tr>
<td>HS28</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>HS29</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>31</td>
<td>23</td>
</tr>
</tbody>
</table>

parameter for the third variant was chosen based on performance and seems to be related to the less dynamic nature of the update as compared with steering.

Tables 1, 2, 3, and 4 give our preliminary numerical results for these three strategies; column EIQP-steer corresponds to the first strategy, column EQP-steer corresponds to the second strategy, and column EIQP-seq corresponds to the third strategy. For each strategy we have recorded the number of function evaluations #fc and the number of gradient evaluations #gJ. Note that if the quantity #fc - #gJ is positive, then its value represents the number of unsuccessful iterations, i.e., the number of times that the trust-region radii were necessarily decreased in order to obtain good agreement between the faithful model $M_k^q$ and the merit function $\phi$. An F indicates that more than 500 evaluations were required, and an FQP indicates that the QP solver failed.
We record the number of function and gradient evaluations required and note that the difference between these values indicates the number of unsuccessful steps attempted; an F indicates that more than 500 evaluations were required.

<table>
<thead>
<tr>
<th>Prob.</th>
<th>EIQP-steer #fc</th>
<th>#gJ</th>
<th>EQP-steer #fc</th>
<th>#gJ</th>
<th>EIQP-seq #fc</th>
<th>#gJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>HS30</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>HS31</td>
<td>8</td>
<td>8</td>
<td>9</td>
<td>7</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>HS32</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>HS33</td>
<td>5</td>
<td>5</td>
<td>14</td>
<td>11</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>HS34</td>
<td>9</td>
<td>8</td>
<td>9</td>
<td>8</td>
<td>34</td>
<td>28</td>
</tr>
<tr>
<td>HS35</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>HS36</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>HS37</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>HS38</td>
<td>84</td>
<td>38</td>
<td>65</td>
<td>46</td>
<td>84</td>
<td>38</td>
</tr>
<tr>
<td>HS39</td>
<td>12</td>
<td>11</td>
<td>13</td>
<td>11</td>
<td>12</td>
<td>11</td>
</tr>
<tr>
<td>HS40</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>HS41</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>HS42</td>
<td>6</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>HS43</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>12</td>
<td>10</td>
</tr>
<tr>
<td>HS44</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>HS45</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>HS46</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>HS47</td>
<td>15</td>
<td>15</td>
<td>20</td>
<td>18</td>
<td>19</td>
<td>17</td>
</tr>
<tr>
<td>HS48</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>HS49</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>HS50</td>
<td>9</td>
<td>9</td>
<td>6</td>
<td>6</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>HS51</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>HS52</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>HS53</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>HS54</td>
<td>9</td>
<td>9</td>
<td>14</td>
<td>14</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td>HS55</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>HS56</td>
<td>110</td>
<td>82</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>HS57</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>HS59</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>18</td>
<td>15</td>
</tr>
</tbody>
</table>

The strictly convex predictor subproblem combined with either accelerator subproblem (EIQP) or (EQP) typically generates iterates that rapidly converge to a solution; the nonmonotone approach avoids the Maratos effect. The results also indicate that using steering to update the penalty parameter generally performs better than the method discussed in section 4.2; this agrees with [5, 6]. In particular, steering was essential in solving HS93 for otherwise the merit function converged to minus infinity, the constraints blew up, and the method failed. In a less clear manner, the update to the penalty parameter is important in solving HS56. When the update in section 4.2 was used, the merit function again converged to minus infinity; the same occurred when the accelerator step was computed from subproblem (EQP) and steering was used.
Table 3
We record the number of function and gradient evaluations required and note that the difference between these values indicates the number of unsuccessful steps attempted.

<table>
<thead>
<tr>
<th>Prob.</th>
<th>EIQP-steer</th>
<th>EQP-steer</th>
<th>EIQP-seq</th>
</tr>
</thead>
<tbody>
<tr>
<td>$HS_{60}$</td>
<td>7 7 7 7 7 7</td>
<td>7 7 7 7 7 7</td>
<td>7 7 7 7 7 7</td>
</tr>
<tr>
<td>$HS_{61}$</td>
<td>5 5 5 5 14 11</td>
<td>6 6 6 6 22 16</td>
<td>4 4 4 4 4 4</td>
</tr>
<tr>
<td>$HS_{62}$</td>
<td>12 6 8 7 12 6</td>
<td>7 7 10 10 14 14</td>
<td>7 7 10 10 14 14</td>
</tr>
<tr>
<td>$HS_{63}$</td>
<td>15 15 20 17 23 23</td>
<td>6 6 6 6 22 16</td>
<td>7 7 10 10 14 14</td>
</tr>
<tr>
<td>$HS_{64}$</td>
<td>6 6 6 6 22 16</td>
<td>4 4 4 4 4 4</td>
<td>7 7 10 10 14 14</td>
</tr>
<tr>
<td>$HS_{65}$</td>
<td>4 4 4 4 4 4</td>
<td>7 7 10 10 14 14</td>
<td>7 7 10 10 14 14</td>
</tr>
<tr>
<td>$HS_{66}$</td>
<td>27 20 38 27 20 16</td>
<td>27 20 38 27 20 16</td>
<td>7 7 10 10 14 14</td>
</tr>
<tr>
<td>$HS_{67}$</td>
<td>24 19 42 30 45 34</td>
<td>24 19 42 30 45 34</td>
<td>7 7 10 10 14 14</td>
</tr>
<tr>
<td>$HS_{68}$</td>
<td>21 17 41 33 21 17</td>
<td>21 17 41 33 21 17</td>
<td>7 7 10 10 14 14</td>
</tr>
<tr>
<td>$HS_{69}$</td>
<td>5 5 5 5 5 5</td>
<td>5 5 5 5 5 5</td>
<td>5 5 5 5 5 5</td>
</tr>
<tr>
<td>$HS_{70}$</td>
<td>16 15 15 14 16 15</td>
<td>16 15 15 14 16 15</td>
<td>5 5 5 5 5 5</td>
</tr>
<tr>
<td>$HS_{71}$</td>
<td>3 3 3 3 3 3</td>
<td>3 3 3 3 3 3</td>
<td>3 3 3 3 3 3</td>
</tr>
<tr>
<td>$HS_{72}$</td>
<td>8 8 8 8 8 8</td>
<td>8 8 8 8 8 8</td>
<td>8 8 8 8 8 8</td>
</tr>
<tr>
<td>$HS_{73}$</td>
<td>8 8 8 8 8 8</td>
<td>8 8 8 8 8 8</td>
<td>8 8 8 8 8 8</td>
</tr>
<tr>
<td>$HS_{74}$</td>
<td>2 2 4 4 2 2</td>
<td>2 2 4 4 2 2</td>
<td>2 2 4 4 2 2</td>
</tr>
<tr>
<td>$HS_{75}$</td>
<td>12 12 12 12 12 12</td>
<td>12 12 12 12 12 12</td>
<td>2 2 4 4 2 2</td>
</tr>
<tr>
<td>$HS_{76}$</td>
<td>4 4 4 4 4 4</td>
<td>4 4 4 4 4 4</td>
<td>4 4 4 4 4 4</td>
</tr>
<tr>
<td>$HS_{77}$</td>
<td>5 5 5 5 5 5</td>
<td>5 5 5 5 5 5</td>
<td>5 5 5 5 5 5</td>
</tr>
<tr>
<td>$HS_{78}$</td>
<td>6 6 6 6 6 6</td>
<td>6 6 6 6 6 6</td>
<td>6 6 6 6 6 6</td>
</tr>
<tr>
<td>$HS_{79}$</td>
<td>6 6 6 6 6 6</td>
<td>6 6 6 6 6 6</td>
<td>6 6 6 6 6 6</td>
</tr>
<tr>
<td>$HS_{80}$</td>
<td>4 4 4 4 4 4</td>
<td>4 4 4 4 4 4</td>
<td>4 4 4 4 4 4</td>
</tr>
<tr>
<td>$HS_{81}$</td>
<td>8 8 13 12 21 20</td>
<td>8 8 13 12 21 20</td>
<td>4 4 4 4 4 4</td>
</tr>
<tr>
<td>$HS_{82}$</td>
<td>20 18 25 21 43 41</td>
<td>20 18 25 21 43 41</td>
<td>4 4 4 4 4 4</td>
</tr>
<tr>
<td>$HS_{83}$</td>
<td>23 21 64 46 45 43</td>
<td>23 21 64 46 45 43</td>
<td>4 4 4 4 4 4</td>
</tr>
</tbody>
</table>

6. Conclusions. In [17], we proved global convergence of a second derivative SQP method for minimizing the $\ell_1$-penalty function for a fixed value of the penalty parameter. The main purpose of this paper was to study the local convergence properties of a nonmonotone variant of that algorithm. In section 3 we gave two local convergence results—the first applies when the accelerator step is computed from an equality constrained subproblem (the so-called SEQP approach), and the second applies when the accelerator step is computed from an inequality constrained subproblem (the so-called SIQP approach). Both results show superlinear convergence of the iterates to a solution satisfying the strong second-order sufficiency conditions; under slightly stronger assumptions on the second derivatives, the convergence is quadratic.

Algorithm 2.1 requires the definition of a positive-definite matrix that approximates the Hessian of the Lagrangian. In section 4.1 we discussed a strategy for defining these matrices based on limited-memory BFGS updating. In particular, we showed how the resultant dense predictor step QP could be transformed into an equiv-
We record the number of function and gradient evaluations required and note that the difference between these values indicates the number of unsuccessful steps attempted; an F indicates that more than 500 evaluations were required, and an FQP indicates that the QP solver QPC failed.

<table>
<thead>
<tr>
<th>Prob.</th>
<th>#fc</th>
<th>#gJ</th>
<th>#fc</th>
<th>#gJ</th>
<th>#fc</th>
<th>#gJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>HS90</td>
<td>44</td>
<td>33</td>
<td>40</td>
<td>30</td>
<td>42</td>
<td>34</td>
</tr>
<tr>
<td>HS91</td>
<td>43</td>
<td>32</td>
<td>32</td>
<td>25</td>
<td>73</td>
<td>61</td>
</tr>
<tr>
<td>HS92</td>
<td>35</td>
<td>25</td>
<td>34</td>
<td>27</td>
<td>64</td>
<td>37</td>
</tr>
<tr>
<td>HS93</td>
<td>18</td>
<td>15</td>
<td>6</td>
<td>6</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>HS95</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>HS96</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>HS97</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>HS98</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>HS99</td>
<td>5</td>
<td>5</td>
<td>FQP</td>
<td>FQP</td>
<td>75</td>
<td>34</td>
</tr>
<tr>
<td>HS100</td>
<td>10</td>
<td>9</td>
<td>13</td>
<td>10</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>HS101</td>
<td>34</td>
<td>27</td>
<td>68</td>
<td>40</td>
<td>67</td>
<td>52</td>
</tr>
<tr>
<td>HS102</td>
<td>28</td>
<td>21</td>
<td>55</td>
<td>34</td>
<td>44</td>
<td>39</td>
</tr>
<tr>
<td>HS103</td>
<td>26</td>
<td>20</td>
<td>27</td>
<td>20</td>
<td>79</td>
<td>62</td>
</tr>
<tr>
<td>HS104</td>
<td>17</td>
<td>14</td>
<td>19</td>
<td>14</td>
<td>14</td>
<td>11</td>
</tr>
<tr>
<td>HS105</td>
<td>21</td>
<td>14</td>
<td>31</td>
<td>23</td>
<td>21</td>
<td>14</td>
</tr>
<tr>
<td>HS106</td>
<td>103</td>
<td>101</td>
<td>122</td>
<td>66</td>
<td>103</td>
<td>101</td>
</tr>
<tr>
<td>HS107</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>HS108</td>
<td>12</td>
<td>9</td>
<td>12</td>
<td>10</td>
<td>242</td>
<td>177</td>
</tr>
<tr>
<td>HS109</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>HS110</td>
<td>9</td>
<td>5</td>
<td>8</td>
<td>6</td>
<td>9</td>
<td>5</td>
</tr>
<tr>
<td>HS111</td>
<td>28</td>
<td>23</td>
<td>41</td>
<td>31</td>
<td>23</td>
<td>19</td>
</tr>
<tr>
<td>HS112</td>
<td>11</td>
<td>11</td>
<td>50</td>
<td>50</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>HS113</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>HS114</td>
<td>142</td>
<td>142</td>
<td>13</td>
<td>13</td>
<td>136</td>
<td>134</td>
</tr>
<tr>
<td>HS116</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>HS117</td>
<td>10</td>
<td>10</td>
<td>11</td>
<td>11</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>HS118</td>
<td>3</td>
<td>3</td>
<td>12</td>
<td>12</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>HS119</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>8</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

In section 4.2 we gave details on a simple strategy for updating the penalty parameter based on minimizing the \( \ell_1 \)-penalty function over a sequence of increasing values of the penalty parameter. Although the basic idea is certainly not new [8, 28, 21, 32, 26, 34, 3, 22], the details of our very simple strategy have not been published to our knowledge.

In section 5 we gave preliminary numerical results for the HS test problems. Our first set of results were based on using “steering” [5, 6] to update the penalty parameter and computing the accelerator step from the inequality constrained subproblem (EIQP) on page 2052. Although this requires “solving” a potentially indefinite QP, which is generally perceived as a bad idea, the results are quite good. Our second set of results also used steering to update the penalty parameter, but instead...
computed the accelerator step from the equality constrained subproblem (EQP) on page 2052. Our last set of results was based on solving subproblem (EIQP) for the accelerator step, but updating the penalty parameter by monitoring the norm of the constraint violation over a sequence of approximate minimizers of the merit function (see section 4.2). We stress that these results are preliminary and that they are not intended to compare the SIQP approach with the SEQP approach, but rather to show that both approaches have the potential to be successful in practice.

During essentially simultaneous work, Morales, Nocedal, and Wu [23] have developed a similar $\ell_1$-SQP line-search algorithm. Roughly, they compute a predictor step (without a trust-region constraint) followed by an accelerator step defined as the solution to problem (EQP). They then reduce the $\ell_1$-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. First, Algorithm 2.1 is based on trust-region methodology, while their algorithm is based on line-search philosophy. Second, 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. Third, 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 subproblem (EQP) is used to compute an accelerator step. Following the rejection of a trial step, Morales, Nocedal, and Wu 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.

Acknowledgments. The authors thank an anonymous referee and the associate editor for their helpful comments on this paper.

REFERENCES