SIAM J. OPTIM.
Vol. 24, No. 3, pp. 1041–1074

AN INEXACT SEQUENTIAL QUADRATIC OPTIMIZATION ALGORITHM FOR NONLINEAR OPTIMIZATION*

FRANK E. CURTIS†, TRAVIS C. JOHNSON‡, DANIEL P. ROBINSON§, AND ANDREAS WÄCHTER¶

Abstract. We propose a sequential quadratic optimization method for solving nonlinear optimization problems with equality and inequality constraints. The novel feature of the algorithm is that, during each iteration, the primal-dual search direction is allowed to be an inexact solution of a given quadratic optimization subproblem. We present a set of generic, loose conditions that the search direction (i.e., inexact subproblem solution) must satisfy so that global convergence of the algorithm for solving the nonlinear problem is guaranteed. The algorithm can be viewed as a globally convergent inexact Newton-based method. The results of numerical experiments are provided to illustrate the reliability of the proposed numerical method.

Key words. nonlinear optimization, constrained optimization, sequential quadratic optimization, inexact Newton methods, global convergence

AMS subject classifications. 49M05, 49M15, 49M29, 49M37, 65K05, 90C30, 90C55

DOI. 10.1137/130918320

1. Introduction. We propose, analyze, and provide numerical results for a sequential quadratic optimization (SQO, commonly known as SQP) method for solving the following generic nonlinear constrained optimization problem:

$$\text{(NLP)} \qquad \begin{array}{lll} \underset{x}{\text{minimize}} & (\underset{x}{\min}) & f(x) \\ \text{subject to} & (\text{s.t.}) & c(x) = 0,\ \bar{c}(x) \leq 0, \end{array}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $c : \mathbb{R}^n \to \mathbb{R}^m$, and $\bar{c} : \mathbb{R}^n \to \mathbb{R}^{\bar{m}}$ are continuously differentiable. Classical SQO methods (see, e.g., [27, 43, 47]) are characterized by the property that, during each iteration, a primal search direction and updated dual variable values are obtained by solving a quadratic optimization subproblem (QP) that locally approximates (NLP). The novel feature of our proposed inexact SQO (iSQO) algorithm is that these subproblem solutions can be inexact as long as the search direction and updated dual values satisfy one of a few sets of conditions. These conditions, which typically allow a great deal of flexibility for the QP solver, are established so that some amount of inexactness is always allowed (at suboptimal primal-dual points), the

*Received by the editors April 24, 2013; accepted for publication (in revised form) April 11, 2014; published electronically July 17, 2014.
http://www.siam.org/journals/siopt/24-3/91832.html

†Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015 (frank.e.curtis@gmail.com). This author's research was supported by U.S. Department of Energy grant DE-SC0010615 and U.S. National Science Foundation grant DMS-1016291.

‡Department of Engineering Sciences and Applied Mathematics, Northwestern University, Evanston, IL 60208 (traviscj@traviscj.com). This author's research was supported by a grant of the Initiative for Sustainability and Energy at Northwestern University.

§Department of Applied Mathematics and Statistics, Johns Hopkins University, Baltimore, MD 21218 (daniel.p.robinson@gmail.com). This author's research was supported by U.S. National Science Foundation grant DMS-1217153.

¶Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL 60208 (waechter@iems.northwestern.edu). This author's research was supported by U.S. National Science Foundation grant DMS-1216920.

algorithm is well-posed, and global convergence in solving the nonlinear problem is guaranteed under mild assumptions.

Any nonlinear optimization problem with equality and inequality constraints can be formulated as (NLP). If (NLP) is (locally) infeasible, then our algorithm is designed to automatically transition to solving the following feasibility problem, which aims to minimize the ℓ_1-norm violation of the constraints of (NLP):

$$\text{(FP)} \qquad \min_x \; v(x), \quad \text{where} \quad v(x) := \|c(x)\|_1 + \|[\bar{c}(x)]^+\|_1$$

and $[\cdot]^+ := \max\{\cdot, 0\}$ (with the max operator applied elementwise). A point that is stationary for (FP), yet is infeasible with respect to (NLP), is called an infeasible stationary point for (NLP). This feature of our algorithm of converging to the set of first-order stationary points of (FP) when (NLP) is (locally) infeasible is important in any modern optimization algorithm as it guarantees that useful information is provided for a problem even when it involves model and/or data inconsistencies.

Our algorithm may be considered an inexact Newton-based method since, during each iteration, the primal-dual search direction is an inexact solution of a linearized equation corresponding to a nonlinear first-order stationarity equation. See [16, 42] for the foundations of (inexact) Newton methods for solving nonlinear equations, [29, 31, 37, 41] for examples of inexact Newton methods for solving constrained optimization problems, and [6] for a Gauss–Newton strategy that employs a similar linearization technique. We also remark in passing that another class of inexact SQO methods, not in the scope of this paper, are those in which the QPs are formulated using inexact derivative information; see, e.g., [2, 17, 30, 32, 45, 46].

Our approach employs an ℓ_1-norm exact penalty function to drive global convergence. Such a technique has been employed in SQO-type methods for decades; see, e.g., [20, 27, 28, 43] and the more recent methods in [10, 11, 24, 25, 26, 40]. Indeed, the technique has been effectively employed in previous work by some of the authors on inexact Newton methods for equality constrained optimization [7, 8, 14] and inexact interior-point methods for inequality constrained optimization [13, 15]. Our algorithm also has several additional features in common with the algorithm in [5], such as the manner in which up to two QPs are solved during each iteration and that in which the penalty parameter is updated. Note, however, that the method in [5] requires exact QP solutions (in order to ensure rapid infeasibility detection), whereas the central feature of our iSQO method is that the QP solutions may be inexact.

We associate with problems (NLP) and (FP) the Fritz John (FJ) function

$$\mathcal{F}(x, y, \bar{y}, \mu) := \mu f(x) + c(x)^T y + \bar{c}(x)^T \bar{y}$$

and the ℓ_1-norm exact penalty function

$$\phi(x, \mu) := \mu f(x) + v(x).$$

The quantity $\mu \geq 0$ plays the role of both the objective multiplier in the FJ function and the penalty parameter in the penalty function. Optimality conditions for both (NLP) and (FP) can be written in terms of the gradient of the FJ function $\nabla \mathcal{F}$, constraint functions c and $\bar{c}$, and bounds on the dual variables, or more specifically in terms of the primal-dual first-order stationarity residual function

$$\rho(x,y,\bar{y},\mu) := \begin{bmatrix} \mu g(x) + J(x)y + \bar{J}(x)\bar{y} \\ \min\{[c(x)]^+, e-y\} \\ \min\{[c(x)]^-, e+y\} \\ \min\{[\bar{c}(x)]^+, e-\bar{y}\} \\ \min\{[\bar{c}(x)]^-, \quad \bar{y}\} \end{bmatrix},$$

where $g := \nabla f$, $J := \nabla c$, $\bar{J} := \nabla \bar{c}$, $[\cdot]^- := \max\{-\cdot, 0\}$, and e is a vector of ones whose length is determined by the context. (Here, the min and max operators are applied elementwise.) If $\rho(x,y,\bar{y},\mu) = 0$, $v(x) = 0$, and $(y,\bar{y},\mu) \neq 0$, then $(x,y,\bar{y},\mu)$ is an FJ point [33, 38] for problem (NLP). In particular, if $\mu > 0$, then $(x, y/\mu, \bar{y}/\mu)$ is a Karush–Kuhn–Tucker (KKT) point [35, 36] for (NLP). On the other hand, if $\rho(x,y,\bar{y},0) = 0$ and $v(x) > 0$, then $(x,y,\bar{y},0)$ is an FJ point for problem (FP) and x is an infeasible stationary point for (NLP).

The paper is organized as follows. In section 2, we motivate our work by presenting an SQO method in which at most two QPs are solved during each iteration. We then present our new iSQO method. Our approach is modeled after the presented SQO method, but allows inexactness in the subproblem solves, a feature that may allow for significantly reduced computational costs. In section 3, we provide global convergence guarantees for our iSQO method, proving under mild assumptions that the algorithm will converge to KKT points, infeasible stationary points, or feasible points at which the Mangasarian–Fromovitz constraint qualification (MFCQ) fails. The results of numerical experiments illustrating the reliability of our approach are presented in section 4, and concluding remarks are presented in section 5.

Notation. We drop function dependencies once they are defined and use subscripts to denote functions and function values corresponding to iteration numbers; e.g., by f_k we mean $f(x_k)$. Superscripts, on the other hand, are used to denote the element index of a vector; e.g., c^i is the ith constraint function. Unless otherwise specified, $\|\cdot\| := \|\cdot\|_2$. We use e and I to denote a vector of ones and an identity matrix, respectively, where in each case the size is determined by the context. As above, vectors of all zeros are written simply as 0, and, similarly, vectors of all infinite values are written as ∞. Given N-vectors a_1, a_2, and a_3, we use the shorthand $a_1 \in [a_2, a_3]$ to indicate that $a_1^i \in [a_2^i, a_3^i]$ for all $i \in \{1, \ldots, N\}$, and similarly for open-ended intervals. Finally, $A \succeq B$ indicates that $A - B$ is positive semidefinite.

2. Algorithm descriptions.

In this section, we present two algorithms. The purpose of the first algorithm (an SQO method), in which at most two QPs are solved exactly during each iteration, is to outline the algorithmic structure on which the second algorithm (our new iSQO algorithm) is based. By comparing the two algorithms, we highlight the algorithmic features of our iSQO method that are needed to maintain global convergence when the QP solutions are allowed to be inexact.

In both of the algorithms that we present, each iterate has the form

$$\left(x_k, \begin{bmatrix} y'_k \\ \bar{y}'_k \end{bmatrix}, \begin{bmatrix} y''_k \\ \bar{y}''_k \end{bmatrix}, \mu_k\right), \text{ where } \begin{bmatrix} y'_k \\ y''_k \end{bmatrix} \in [-e, e], \; \begin{bmatrix} \bar{y}'_k \\ \bar{y}''_k \end{bmatrix} \in [0, e], \text{ and } \mu_k \in (0, \infty). \tag{2.1}$$

Here, x_k is the primal iterate, $(y'_k, \bar{y}'_k)$ are constraint multipliers for (NLP), $(y''_k, \bar{y}''_k)$ are constraint multipliers for (FP), and μ_k is a penalty parameter. We use separate multipliers for (NLP) and (FP) in order to measure the stationarity error with respect to each problem more accurately than if only one set of multipliers were maintained.

At a given iterate, we define the following piecewise linear model of the penalty function $\phi(\cdot,\mu)$ (i.e., the constraint violation measure v if $\mu = 0$) about x_k:

$$l_k(d,\mu) := \mu(f_k + g_k^T d) + \|c_k + J_k^T d\|_1 + \|[\bar{c}_k + \bar{J}_k^T d]^+\|_1.$$

Given a vector d, we define the reduction in this model as

$$\Delta l_k(d,\mu) := l_k(0,\mu) - l_k(d,\mu) = -\mu g_k^T d + v_k - \|c_k + J_k^T d\|_1 - \|[\bar{c}_k + \bar{J}_k^T d]^+\|_1.$$

Both algorithms require solutions of at most two QPs during each iteration. In particular, we compute $(d_k', y_{k+1}', \bar{y}_{k+1}')$ as a primal-dual solution of the "penalty QP"

$$\text{(PQP)} \qquad \min_d \; -\Delta l_k(d,\mu_k) + \tfrac{1}{2} d^T H_k' d$$

and potentially compute $(d_k'', y_{k+1}'', \bar{y}_{k+1}'')$ as a solution of the "feasibility QP"

$$\text{(FQP)} \qquad \min_d \; -\Delta l_k(d,0) + \tfrac{1}{2} d^T H_k'' d.$$

Here, H_k' is an approximation of the Hessian of $\mathcal{F}$ at $(x_k, y_k', \bar{y}_k', \mu_k)$ and H_k'' is defined similarly corresponding to $(x_k, y_k'', \bar{y}_k'', 0)$. Despite the fact that (PQP) and (FQP) are written with nonsmooth objective functions, they each can be reformulated and solved as the following smooth constrained QP [18] (with $(\mu, H) = (\mu_k, H_k')$ and $(\mu, H) = (0, H_k'')$ for (PQP) and (FQP), respectively):

$$\begin{aligned} &\min_{d,r,s,t} \; \mu g_k^T d - v_k + e^T(r+s) + e^T t + \tfrac{1}{2} d^T H d \\ &\;\text{s.t. } c_k + J_k^T d = r - s, \; \bar{c}_k + \bar{J}_k^T d \le t, \; (r,s,t) \ge 0. \end{aligned} \tag{2.2}$$

In the resulting primal-dual solution—i.e., $(d_k', r_k', s_k', t_k', y_{k+1}', \bar{y}_{k+1}')$ for (PQP) and $(d_k'', r_k'', s_k'', t_k'', y_{k+1}'', \bar{y}_{k+1}'')$ for (FQP)—the multipliers are those corresponding to the (relaxed) linearized equality and linearized inequality constraints. For our purposes, we ignore the artificial variables in the remainder of the algorithm, though we remark that in an exact solution of (PQP) we have

$$r_k' = [c_k + J_k^T d_k']^+, \; s_k' = [c_k + J_k^T d_k']^-, \text{ and } t_k' = [\bar{c}_k + \bar{J}_k^T d_k']^+, \tag{2.3}$$

and similar relationships for the artificial variables for (FQP); e.g., see [5].

To summarize the previous paragraph, our algorithms require solutions of (PQP) and/or (FQP). In practice, these subproblems typically would be solved by employing a QP solver to solve their smooth counterparts, which have the form (2.2). In the exact solutions of these subproblems, the artificial variables are uniquely determined by the primal solutions (i.e., d_k' and d_k'') according to (2.3). Moreover, given any inexact solution, any other values of the artificial variables may be replaced by those in (2.3); i.e., again they may be defined uniquely by the primal solutions. Thus, we may ignore the artificial variables in our algorithm, and may simply refer to subproblems (PQP) and (FQP), which do not involve any artificial variables.

Critical in the descriptions of both algorithms are the model reduction Δl_k as well as the following residual corresponding to subproblems (PQP) and (FQP):

$$\rho_k(d,y,\bar{y},\mu,H) := \begin{bmatrix} \mu g_k + Hd + J_k y + \bar{J}_k \bar{y} \\ \min\{[c_k + J_k^T d]^+, e - y\} \\ \min\{[c_k + J_k^T d]^-, e + y\} \\ \min\{[\bar{c}_k + \bar{J}_k^T d]^+, e - \bar{y}\} \\ \min\{[\bar{c}_k + \bar{J}_k^T d]^-, \quad \bar{y}\} \end{bmatrix}.$$

Observe that if $\rho_k(d'_k, y'_{k+1}, \bar{y}'_{k+1}, \mu_k, H'_k) = 0$, then $(d'_k, y'_{k+1}, \bar{y}'_{k+1})$ is a first-order stationary point for (PQP), and if $\rho_k(d''_k, y''_{k+1}, \bar{y}''_{k+1}, 0, H''_k) = 0$, then $(d''_k, y''_{k+1}, \bar{y}''_{k+1})$ is a first-order stationary point for (FQP). Moreover, we have $\rho_k(0, y, \bar{y}, \mu, H) = \rho(x_k, y, \bar{y}, \mu)$ for any $(y, \bar{y}, \mu, H)$. It is also prudent to note that for any $(d, y, \bar{y}, \mu, H)$, the model reduction $\Delta l_k(d, \mu)$ and residual $\rho_k(d, y, \bar{y}, \mu, H)$ are easily computed with only a few matrix-vector operations.

The algorithms in this section make use of the following user-defined constants, which we define upfront for ease of reference:

$$\{\theta, \xi\} \subset (0, \infty), \quad \{\kappa, \epsilon, \tau, \delta, \gamma, \eta, \lambda, \zeta, \psi\} \subset (0, 1), \quad \text{and} \quad \beta \in (0, \epsilon). \tag{2.4}$$

2.1. An SQO algorithm with exact subproblem solutions. We now present an SQO method that will form the basis for our newly proposed iSQO algorithm. The overall strategy of the SQO method can be summarized as follows. First, if the current iterate corresponds to a KKT point or an infeasible stationary point, then the algorithm terminates; otherwise, a primal-dual solution of (PQP) is computed. If the primal component of this solution yields a reduction in the model of the penalty function that is sufficiently large compared to the current infeasibility measure, then, by choosing this primal-dual search direction, the algorithm predicts that sufficient progress will be made toward both feasibility and optimality. However, if the reduction in the model of the penalty function is not sufficiently large, then the algorithm computes a primal-dual solution of (FQP) in order to obtain a reference for the amount of progress that can be made toward linearized constraint satisfaction from the current iterate. Using this reference direction, the algorithm either determines that the solution produced from (PQP) does in fact predict sufficient progress toward both feasibility and optimality, or that a combination of the primal solutions from (PQP) and (FQP) should be taken as the search direction. In this latter case, the algorithm may determine that the penalty parameter should be reduced to guarantee that the reduction in the model of the penalty function is sufficiently large.

For simplicity in this subsection, we temporarily assume that $H'_k \succeq 2\theta I$ and $H''_k \succeq 2\theta I$ for all $k \geq 0$, which in particular means that we have

$$\tfrac{1}{2} d^T H d \geq \theta \|d\|^2 \quad \text{with} \quad (d, H) = \begin{cases} (d'_k, H'_k) & \text{for (PQP)}, \\ (d''_k, H''_k) & \text{for (FQP)}. \end{cases} \tag{2.5}$$

We do not make this convexity assumption in our iSQO method since, in that algorithm, we include a convexification procedure that will ensure (2.5). However, for our immediate purposes, we simply assume that such a procedure is not required.

In each iteration of our SQO framework, we compute $(d'_k, y'_{k+1}, \bar{y}'_{k+1})$ satisfying the following "termination test." We use the phrase "termination test" for consistency with the terminology of our iSQO method, in which tests such as this one reveal conditions under which an iterative solver applied to solve a QP may terminate.

Termination Test A
The primal-dual vector $(d'_k, y'_{k+1}, \bar{y}'_{k+1})$ *satisfies*
(2.6) $\qquad \rho_k(d'_k, y'_{k+1}, \bar{y}'_{k+1}, \mu_k, H'_k) = 0.$

Similarly, as previously described, we potentially compute $(d''_k, y''_{k+1}, \bar{y}''_{k+1})$ satisfying the following test.

Termination Test B

The primal-dual vector $(d_k'', y_{k+1}'', \bar{y}_{k+1}'')$ *satisfies*

$$\rho_k(d_k'', y_{k+1}'', \bar{y}_{k+1}'', 0, H_k'') = 0. \tag{2.7}$$

If a solution of (FQP) is not needed, then $(d_k'', y_{k+1}'', \bar{y}_{k+1}'') \leftarrow (0, y_k'', \bar{y}_k'')$ by default.

To compartmentalize this algorithm (and our iSQO method in section 2.2), we state that in each iteration one of a set of possible "scenarios" occurs. Each scenario is defined by a set of conditions that must hold and the resulting updates that will be performed. In particular, in each scenario, the primal search direction will be set as a convex combination of d_k' and d_k''; i.e., for $\tau_k \in [0, 1]$ we set

$$d_k \leftarrow \tau_k d_k' + (1 - \tau_k) d_k''. \tag{2.8}$$

Clearly, if (FQP) is not solved, then (2.8) is well-defined with the default value $d_k'' \leftarrow 0$.

For our SQO framework, we have three scenarios. The first is the following.

Scenario A

Conditions A: *The primal-dual vector* $(d_k', y_{k+1}', \bar{y}_{k+1}')$ *satisfies Termination Test* A *and*

$$\Delta l_k(d_k', \mu_k) \geq \epsilon v_k. \tag{2.9}$$

Updates A: *Set*

$$d_k \leftarrow d_k', \quad \tau_k \leftarrow 1, \quad \textit{and} \quad \mu_{k+1} \leftarrow \mu_k. \tag{2.10}$$

Scenario A represents the simplest case when, for the current value of the penalty parameter, the solution of (PQP) yields a reduction in the model of the penalty function that is large compared to the infeasibility measure; see (2.9). With such a large reduction in the model of the penalty function, the algorithm chooses $d_k \leftarrow d_k'$ as the primal search direction, maintains the current value of the penalty parameter, and avoids solving (FQP). (We set $\tau_k \leftarrow 1$ so that $d_k \leftarrow d_k'$ is consistent with (2.8).)

The second scenario is similar to the first, except it employs the solution of (FQP).

Scenario B

Conditions B: *The primal-dual vectors* $(d_k', y_{k+1}', \bar{y}_{k+1}')$ *and* $(d_k'', y_{k+1}'', \bar{y}_{k+1}'')$ *satisfy Termination Tests* A *and* B, *respectively, and*

$$\Delta l_k(d_k', \mu_k) \geq \epsilon \Delta l_k(d_k'', 0). \tag{2.11}$$

Updates B: *Set quantities as in* (2.10).

With a solution of (FQP) in hand, Scenario B imposes a weaker condition than does Scenario A on the reduction of the model of the penalty function. (The fact that (2.11) is weaker than (2.9) is seen by observing that $\Delta l_k(d, 0) \leq v_k$ for any $d \in \mathbb{R}^n$.) As in Scenario A, with such a large reduction in the model of the penalty function, we set $d_k \leftarrow d_k'$ and maintain the current value of the penalty parameter.

The third scenario occurs whenever the first and second scenarios do not occur.

SCENARIO C	
Conditions C:	*The primal-dual vectors* $(d'_k, y'_{k+1}, \bar{y}'_{k+1})$ *and* $(d''_k, y''_{k+1}, \bar{y}''_{k+1})$ *satisfy Termination Tests* A *and* B*, respectively.*
Updates C:	*Choose* τ_k *as the largest value in* $[0,1]$ *such that* (2.12) $\Delta l_k(\tau_k d'_k + (1-\tau_k)d''_k, 0) \geq \epsilon \Delta l_k(d''_k, 0),$ *then set* d_k *by* (2.8)*, and finally set*

(2.13)
$$\mu_{k+1} \leftarrow \begin{cases} \mu_k & \text{if } \tau_k \geq \tau \text{ and } \Delta l_k(d_k, \mu_k) \geq \beta \Delta l_k(d_k, 0), \\ \delta\mu_k & \text{if } \tau_k < \tau \text{ and } \Delta l_k(d_k, \mu_k) \geq \beta \Delta l_k(d_k, 0), \\ \min\left\{\delta\mu_k, \frac{(1-\beta)\Delta l_k(d_k,0)}{g_k^T d_k + \theta\|d_k\|^2}\right\} & \text{otherwise.} \end{cases}$$

Whenever Scenario C occurs in the algorithm, it has been determined that the solution of (PQP) does not yield a sufficiently large model reduction (as dictated by (2.11)). In such a case, the primal search direction is set as a convex combination of the penalty and feasibility steps in such a way that the resulting reduction in the model of the constraint violation is sufficiently large compared to that obtained by the feasibility step alone. (This condition is reminiscent of conditions imposed in methods that employ "steering" techniques for the penalty parameter; see, e.g., [5, 9, 11, 12, 22].) Then the penalty parameter may be decreased so that the search direction yields a sufficiently large model reduction for the new value of the penalty parameter.

In all of the above scenarios, it can be shown (as in Lemma 3.13) that any nonzero search direction yields a positive reduction in the model of the penalty function for the new value of the penalty parameter, i.e., that $\Delta l_k(d_k, \mu_{k+1}) > 0$, which in turn implies that d_k is a descent direction for $\phi(\cdot, \mu_{k+1})$ at x_k (as in Lemma 3.12). Based on this fact, it is appropriate to update the primal iterate by performing a backtracking Armijo line search to obtain the largest $\alpha_k \in \{\gamma^0, \gamma^1, \gamma^2, \ldots\}$ such that

$$\phi(x_k + \alpha_k d_k, \mu_{k+1}) \leq \phi(x_k, \mu_{k+1}) - \eta\alpha_k \Delta l_k(d_k, \mu_{k+1}). \tag{2.14}$$

As for the dual variables in the following iteration, in the present context we claim that it is appropriate to follow the common SQO strategy of employing those multipliers obtained via the QP solutions. We remark, however, that additional considerations will be made when updating the dual variables in our iSQO algorithm.

A complete description of our SQO framework is presented as Algorithm A. The algorithm terminates finitely with a KKT point or infeasible stationary point if and only if either of the following pairs of conditions are satisfied:

$$\rho(x_k, y'_k, \bar{y}'_k, \mu_k) = 0 \quad \text{and} \quad v_k = 0; \tag{2.15a}$$
$$\rho(x_k, y''_k, \bar{y}''_k, 0) = 0 \quad \text{and} \quad v_k > 0. \tag{2.15b}$$

2.2. An iSQO algorithm with inexact subproblem solutions. We are now prepared to describe the details of our proposed iSQO method. During each iteration, rather than require an exact solution of (PQP), and potentially of (FQP), our iSQO method allows inexactness in the solutions of these subproblems.

The underlying structure of our iSQO algorithm is similar to that of Algorithm A presented in the previous subsection. Indeed, the core strategy of our iSQO method is to attempt to compute an approximate solution of (PQP) that (i) is sufficiently

Algorithm A Sequential quadratic optimizer with exact subproblem solves.

1: Set $k \leftarrow 0$ and choose $(x_k, y'_k, \bar{y}'_k, y''_k, \bar{y}''_k, \mu_k)$ satisfying (2.1).
2: Check for finite termination by performing the following.
  a: If (2.15a) holds, then terminate and return the KKT point $(x_k, y'_k/\mu_k, \bar{y}'_k/\mu_k)$.
  b: If (2.15b) holds, then terminate and return the infeasible stationary point x_k.
3: Compute the exact solution of (PQP) satisfying Termination Test A, and initialize $(d''_k, y''_{k+1}, \bar{y}''_{k+1}) \leftarrow (0, y''_k, \bar{y}''_k)$ by default.
  a: If Conditions A hold, then perform Updates A and go to step 5. // **(Scenario A)**
4: Compute the exact solution of (FQP) satisfying Termination Test B.
  a: If Conditions B hold, then perform Updates B and go to step 5. // **(Scenario B)**
  b: Conditions C hold, so perform Updates C. // **(Scenario C)**
5: Compute α_k as the largest value in $\{\gamma^0, \gamma^1, \gamma^2, \ldots\}$ such that (2.14) is satisfied.
6: Set $x_{k+1} \leftarrow x_k + \alpha_k d_k$ and $k \leftarrow k+1$, then go to step 2.

accurate so as to guarantee global convergence in the primal-dual space (recall Termination Test A) and (ii) yields a reduction in the model of the penalty function that is sufficiently large (recall Scenario A). Moreover, if such an approximate solution to (PQP) does not yield a sufficiently large model reduction, then the strategy of our iSQO method is to compute an approximate solution of (FQP) that (i) is sufficiently accurate so as to guarantee global convergence in the primal-dual space (recall Termination Test B) and (ii) provides a reference for the amount of progress toward linearized constraint satisfaction that can be expected from the current iterate (recall Scenarios B and C).

Issues arise, however, if one were to follow the strategy of Algorithm A without considering certain situations that may occur when inexactness is allowed in the subproblem solutions. For example, one must confront the fact that a given point x_k may be stationary for the penalty function $\phi(\cdot, \mu_k)$. If (PQP) were to be solved exactly at such a point, then the stationarity of x_k would be revealed by an exact solution with $\rho_k(d'_k, y'_{k+1}, \bar{y}'_{k+1}, \mu_k, H'_k) = 0$, where $d'_k = 0$. (Algorithm A is well-defined in this case since it assumes that such an exact solution is computed.) However, in the context of our iSQO method, we avoid situations in which an exact solution of (PQP) is ever required. (Observe that if x_k is stationary for $\phi(\cdot, \mu_k)$, then a solution of (PQP) with $d'_k \neq 0$ may not be a descent direction for the penalty function for a suitable value of the penalty parameter, in which case the line search (2.14) may fail. Such situations also provide motivation for many of the details of our iSQO method.) In order to avoid having to compute exact solutions, we incorporate additional termination tests and scenarios that ensure that inexactness is *always* allowed. It is worthwhile to note that these additional scenarios may result in a null step in the primal space while an update of the dual values and/or penalty parameter is performed.

We remark that due to the additional termination tests and scenarios considered in our iSQO method, it does not entirely reduce to Algorithm A if exact QP solutions are computed. However, Algorithm A still represents the foundation for our iSQO method, so it has been presented as motivation and for reference.

We remark at the outset that our choice of initial point and updating strategy for the feasibility multipliers (see (2.36)) will ensure that, for all $k \geq 0$, we have

$$\|\rho(x_k, y''_k, \bar{y}''_k, 0)\| \leq \|\rho(x_k, 0, 0, 0)\| \leq \|\rho(x_k, 0, 0, 0)\|_1 \leq v_k. \tag{2.16}$$

We also refer the reader to Assumptions 3.2 and 3.3 in section 3.1, under which we illustrate that our termination tests are well-posed. That is, under these assumptions, we show that sufficiently accurate solutions of (PQP) and/or (FQP) will yield primal-dual vectors satisfying an appropriate subset of termination tests.

We consider three termination tests that address the penalty subproblem (PQP). (In each step of our algorithm, we state explicitly which of these three tests is to be considered.) The first outlines the common case when the primal step produces a sufficiently large reduction in the model of the penalty function $\phi(\cdot, \mu_k)$ and corresponds to a sufficiently accurate solution of (PQP).

Termination Test 1

The primal-dual vector $(d'_k, y'_{k+1}, \bar{y}'_{k+1})$ *satisfies*

$$\|\rho_k(d'_k, y'_{k+1}, \bar{y}'_{k+1}, \mu_k, H'_k)\| \leq \kappa \max\{\|\rho(x_k, y'_k, \bar{y}'_k, \mu_k)\|, \|\rho(x_k, y''_k, \bar{y}''_k, 0)\|\}, \tag{2.17}$$

$$y'_{k+1} \in [-e, e], \;\; \bar{y}'_{k+1} \in [0, e], \tag{2.18}$$

and

$$\Delta l_k(d'_k, \mu_k) \geq \theta \|d'_k\|^2 > 0. \tag{2.19}$$

In Termination Test 1, condition (2.17) is reminiscent of conditions commonly employed in inexact Newton methods for solving nonlinear equations; see [16] (and note that a similar condition, namely, (2.28), is imposed for approximate solutions of the feasibility subproblem (FQP)). We remark that with d'_k yielding $\frac{1}{2} {d'_k}^T H'_k d'_k \geq \theta \|d'_k\|^2$—which is ensured by our convexification procedure described later on—the condition (2.19) merely requires that the corresponding objective value of (PQP) is better than that yielded by the zero vector.

The second test is similar to the first, but involves potentially tightened tolerances for the residual, multipliers, and model reduction. This test is enforced before the penalty parameter is allowed to be updated. This will allow us to prove that the penalty parameter will remain bounded away from zero under common assumptions.

Termination Test 2

The primal-dual vector $(d'_k, y'_{k+1}, \bar{y}'_{k+1})$ *satisfies* (2.18), (2.19), *and*

$$\|\rho_k(d'_k, y'_{k+1}, \bar{y}'_{k+1}, \mu_k, H'_k)\| \leq \kappa \|\rho(x_k, y''_k, \bar{y}''_k, 0)\|. \tag{2.20}$$

Furthermore, the following conditions must hold:

(a) *If*

$$\Delta l_k(d'_k, 0) < \epsilon v_k, \tag{2.21}$$

then

$$\left\|(y'_{k+1}, \bar{y}'_{k+1})\right\|_\infty \geq \lambda(\epsilon - \beta). \tag{2.22}$$

(b) *If*

$$\Delta l_k(d'_k, \mu_k) < \beta \Delta l_k(d'_k, 0), \tag{2.23}$$

then

$$\left\|(y'_{k+1}, \bar{y}'_{k+1})\right\|_\infty \geq \lambda \mu_k g_k^T d'_k / v_k. \tag{2.24}$$

We prove that Termination Test 2 is considered only if $\|\rho(x_k, y_k'', \bar{y}_k'', 0)\| > 0$ (see Lemma 3.9), which in turn implies with (2.16) that $v_k > 0$. These facts are used to show that this test is well-posed. Condition (a) in the test is motivated by our convergence analysis; observing the contrapositive of the condition, it requires that if the multipliers are bounded above by $\lambda(\epsilon - \beta) \in (0, 1)$, then the solution of (PQP) must be accurate enough so that the reduction in the model of the constraint violation measure is sufficiently large. Condition (b) is also required by our convergence theory; it enables us to prove that $\mu_k \to 0$ when $v_k \to 0$ only if every limit point of an iterate sequence is an FJ point at which the MFCQ fails; see Lemmas 3.25 and 3.28.

Our third termination test for (PQP) is necessary as there are situations in which Termination Tests 1 and 2 cannot be satisfied. For example, if x_k is stationary for $\phi(\cdot, \mu_k)$, but $\rho(x_k, y_k', \bar{y}_k', \mu_k)$ is nonzero (due to the values of the multiplier estimates), then this test allows an update of the dual solution and/or penalty parameter without requiring a productive step in the primal space.

Termination Test 3

The primal-dual vector $(d_k', y_{k+1}', \bar{y}_{k+1}')$ *satisfies* (2.18) *and*

$$\|\rho_k(0, y_{k+1}', \bar{y}_{k+1}', \mu_k, H_k')\| \leq \kappa \|\rho(x_k, y_k', \bar{y}_k', \mu_k)\|. \tag{2.25}$$

Furthermore, if

$$\|\rho(x_k, y_k', \bar{y}_k', \mu_k)\| < \zeta \|\rho(x_k, y_k'', \bar{y}_k'', 0)\|, \tag{2.26}$$

then

$$\left\|(y_{k+1}', \bar{y}_{k+1}')\right\|_\infty \geq \psi. \tag{2.27}$$

In any scenario in which Termination Test 3 is checked and satisfied, the algorithm subsequently resets $d_k' \leftarrow 0$, which is why the test effectively ignores the value of d_k'. We have in this test that if (2.26) holds, then by (2.16) we have $v_k > 0$. This fact is used to show that the test is well-posed. Motivation for the lower bound (2.27) is similar to that for (2.24), i.e., we use it to prove that $\mu_k \to 0$ when $v_k \to 0$ only if every limit point of an iterate sequence is an FJ point at which the MFCQ fails.

We now define our termination test for the feasibility subproblem (FQP); the test is similar to Termination Test 1 for (PQP). We note that (FQP) is approximately solved only if $(d_k', y_{k+1}', \bar{y}_{k+1}')$ satisfying Termination Test 1 has already been obtained, meaning that it is valid to refer to $\Delta l_k(d_k', \mu_k)$ on the left-hand side of (2.30).

Termination Test 4

The primal-dual vector $(d_k'', y_{k+1}'', \bar{y}_{k+1}'')$ *satisfies*

$$\|\rho_k(d_k'', y_{k+1}'', \bar{y}_{k+1}'', 0, H_k'')\| \leq \kappa \|\rho(x_k, y_k'', \bar{y}_k'', 0)\|, \tag{2.28}$$

$$y_{k+1}'' \in [-e, e], \;\; \bar{y}_{k+1}'' \in [0, e], \tag{2.29}$$

and

$$\max\{\Delta l_k(d_k', \mu_k), \Delta l_k(d_k'', 0)\} \geq \theta \|d_k''\|^2. \tag{2.30}$$

We are now prepared to describe the six scenarios that may occur in our iSQO method. Three of the scenarios, namely, Scenarios 2–4, mimic Scenarios A–C, respectively, in Algorithm A. The remaining scenarios are motivated by our goal to provide global convergence guarantees given that we only require (inexact) QP solutions satisfying (subsets of) the above termination tests.

The first scenario considers the case when the algorithm arrives at a stationary point for the penalty function—with multipliers such that $\rho(x_k, y'_k, \bar{y}'_k, \mu_k) = 0$—that is infeasible for (NLP).

Scenario 1

Conditions 1: *The primal-dual residual satisfies* $\rho(x_k, y'_k, \bar{y}'_k, \mu_k) = 0$ *and the infeasibility measure satisfies* $v_k > 0$.

Updates 1: *Set*

$$d_k \leftarrow d'_k \leftarrow d''_k \leftarrow 0, \quad \tau_k \leftarrow 1, \quad \mu_{k+1} \leftarrow \delta\mu_k,$$

and $(y'_{k+1}, \bar{y}'_{k+1}) \leftarrow (y'_k, \bar{y}'_k)$, *then set*

$$(y''_{k+1}, \bar{y}''_{k+1}) \leftarrow \begin{cases} (y'_k, \bar{y}'_k) & \text{if } \|\rho(x_k, y'_k, \bar{y}'_k, 0)\| \leq \|\rho(x_k, y''_k, \bar{y}''_k, 0)\|, \\ (y''_k, \bar{y}''_k) & \text{otherwise.} \end{cases} \tag{2.31}$$

We claim that explicit consideration of Scenario 1, which is expected to occur only rarely in practice, is not necessary in Algorithm A. Indeed, at such a first-order stationary point for the penalty function (and under the assumption, made in section 2.1, that (2.5) holds), Algorithm A would also compute a null step in the primal space and reduce the penalty parameter, as is done here. However, we consider the scenario explicitly in order to avoid requiring an exact solution of (PQP). In fact, our consideration of this scenario does not even require an inexact solution of either (PQP) or (FQP), so it may be considered before either subproblem is approximately solved. Motivation for (2.31) is that $(y'_k, \bar{y}'_k)$ may actually be better multipliers for (FQP) than $(y''_k, \bar{y}''_k)$ for the new (reduced) value of the penalty parameter set in Updates 1. This update is required in our analysis to show that the first-order stationarity residual for the feasibility problem converges to zero.

The second scenario represents a more common case when an inexact solution of (PQP) is computed that yields a productive step in the primal-dual space. As in the case of Scenario A in Algorithm A, a benefit of this scenario is that it can be considered without having to compute an approximate solution of (FQP).

Scenario 2

Conditions 2: *The primal-dual vector* $(d'_k, y'_{k+1}, \bar{y}'_{k+1})$ *satisfies Termination Test* 1 *and* (2.9) *holds.*

Updates 2: *Set quantities as in* (2.10).

The third scenario is similar to the second, but exploits an inexact solution of (FQP) to relax the requirement on the penalty model reduction; recall Scenario B.

Scenario 3

Conditions 3: *The primal-dual vectors* $(d'_k, y'_{k+1}, \bar{y}'_{k+1})$ *and* $(d''_k, y''_{k+1}, \bar{y}''_{k+1})$ *satisfy Termination Tests* 1 *and* 4, *respectively, and* (2.11) *holds.*

Updates 3: *Set quantities as in* (2.10).

The fourth scenario represents a case when a productive direction in the primal space has been computed, but an update of the penalty parameter may be required to yield a penalty model reduction that is sufficient; recall Scenario C. This scenario requires a primal-dual vector satisfying Termination Test 2, which is more restrictive than Termination Test 1, the test employed in Scenarios 2 and 3.

Scenario 4	
Conditions 4:	*The primal-dual vectors* $(d_k', y_{k+1}', \bar{y}_{k+1}')$ *and* $(d_k'', y_{k+1}'', \bar{y}_{k+1}'')$ *satisfy Termination Tests* 2 *and* 4, *respectively.*
Updates 4:	*Choose* τ_k *as the largest value in* $[0,1]$ *such that* (2.12) *holds, then set* d_k *by* (2.8) *and* μ_{k+1} *by* (2.13).

A few remarks are pertinent with respect to Scenario 4. In particular, the scenario is only considered when Termination Tests 2 and 4 hold—in which case Termination Test 1 also clearly holds—but (2.11) is not satisfied; i.e., it is only considered when Scenario 3 does not occur. The satisfaction of (2.19) and the violation of (2.11) imply

$$0 < \theta\|d_k'\|^2 \le \Delta l_k(d_k', \mu_k) < \Delta l_k(d_k'', 0) \quad \text{with} \quad d_k'' \neq 0, \tag{2.32}$$

which along with (2.30) and (2.12), respectively, means that

$$\Delta l_k(d_k'', 0) \ge \theta\|d_k''\|^2 \quad \text{and} \quad d_k \neq 0. \tag{2.33}$$

The fact that (2.32) and (2.33) both hold in Scenario 4 is critical in our analysis.

The last two scenarios concern cases when a productive step in the primal space has not been obtained, yet a productive step in the dual space is available. These scenarios may occur whenever x_k is (nearly) stationary for the penalty function $\phi(\cdot, \mu_k)$.

Scenario 5

Conditions 5: *The primal-dual vector* $(d_k', y_{k+1}', \bar{y}_{k+1}')$ *satisfies Termination Test* 3, *but* (2.26) *fails to hold.*

Updates 5: *Set*

$$d_k \leftarrow d_k' \leftarrow d_k'' \leftarrow 0, \quad \tau_k \leftarrow 1, \quad \mu_{k+1} \leftarrow \mu_k, \tag{2.34}$$

and then set

$$(y_{k+1}'', \bar{y}_{k+1}'') \leftarrow \begin{cases} (0,0) & \text{if } \|\rho(x_k, 0, 0, 0)\| \le \|\rho(x_k, y_k'', \bar{y}_k'', 0)\|, \\ (y_k'', \bar{y}_k'') & \text{otherwise.} \end{cases} \tag{2.35}$$

Scenario 6	
Conditions 6:	*The primal-dual vector* $(d_k', y_{k+1}', \bar{y}_{k+1}')$ *satisfies Termination Test* 3 *and* (2.26) *holds.*
Updates 6:	*Set quantities as in Updates* 1.

As evidenced by the absence of scenarios similar to Scenarios 5 and 6 in Algorithm A, we claim that they would not need to be considered if (PQP) and (FQP) were to be solved exactly. However, they are required in order to have a well-posed and globally convergent algorithm when inexact QP solutions are allowed. The scenarios distinguish between two cases depending on the relationship between the residuals $\rho(x_k, y_k'', \bar{y}_k'', 0)$ and $\rho(x_k, y_k', \bar{y}_k', \mu_k)$; see (2.26). Motivation for (2.35) is that, when (2.26) fails to hold, we have an indication that the infeasibility measure v may be vanishing, in which case zero multipliers may be better than $(y_k'', \bar{y}_k'')$ with respect to the stationarity residual for (FQP). This update is required in our analysis.

As in Algorithm A, after the search direction computation, we perform a backtracking line search to obtain the largest $\alpha_k \in \{\gamma^0, \gamma^1, \gamma^2, \ldots\}$ such that (2.14) holds. If $d_k = 0$ (which is true in Scenarios 1, 5, and 6), then (2.14) is trivially satisfied by

$\alpha_k = 1$. We also perform a final update of the feasibility multipliers:

$$(y''_{k+1}, \bar{y}''_{k+1}) \leftarrow \begin{cases} (0,0) & \text{if } \|\rho(x_{k+1},0,0,0)\| \leq \|\rho(x_{k+1}, y''_{k+1}, \bar{y}''_{k+1}, 0)\|, \\ (y''_{k+1}, \bar{y}''_{k+1}) & \text{otherwise.} \end{cases} \tag{2.36}$$

This update and our choice of initial point ensure that (2.16) holds for all $k \geq 0$.

The framework given in Algorithm 1 is one that may be used to approximately solve either (PQP) or (FQP) until a termination test is satisfied. Importantly, this algorithm includes a convexification procedure for the given Hessian approximation, which ensures that at the conclusion of a run we have that (2.5) holds (though not necessarily that the Hessian approximation is positive semidefinite). Since the formulations of (PQP) and (FQP) differ only by the choices of penalty parameter and Hessian approximation, we specify these as the signifying inputs to the algorithm.

Algorithm 1 Quadratic optimizer for solving (PQP) or (FQP).

1: Input $(\mu, H) \leftarrow (\mu_k, H'_k)$ for (PQP) or $(\mu, H) \leftarrow (0, H''_k)$ for (FQP).
2: Choose an initial solution estimate $(d_\circ, y_\circ, \bar{y}_\circ)$.
3: Using $(d_\circ, y_\circ, \bar{y}_\circ)$ as an initial estimate, call a QP solver to solve

$$\min_d \; -\Delta l_k(d,\mu) + \tfrac{1}{2} d^T H d, \tag{2.37}$$

obtaining an improved solution estimate $(d, y, \bar{y})$ satisfying $y \in [-e, e]$ and $\bar{y} \in [0, e]$.
4: If $\frac{1}{2} d^T H d < \theta \|d\|^2$, then set $H \leftarrow H + \xi I$ and go to step 3.
5: If a termination test (specified by Algorithm 2) holds, then return $(d, y, \bar{y})$ and H.
6: Set $(d_\circ, y_\circ, \bar{y}_\circ) \leftarrow (d, y, \bar{y})$ and go to step 3.

Our complete iSQO algorithm is presented as Algorithm 2. For simplicity, we state that the Hessian approximations H'_k and H''_k are initialized during each iteration, though in practice each matrix need only be initialized if it is used.

We have written Algorithms 1 and 2 as separate routines so as to highlight details required in the subproblem solver that should be considered separately from details of the nonlinear problem solver. However, in an efficient implementation of our framework, our intention is that these algorithms would be fully integrated. In particular, in various steps of Algorithm 2, the algorithm requires an approximate solution of a QP satisfying one of a set of possible termination tests. In an efficient implementation of our framework, these tests would be checked as the termination conditions within the QP solver itself. In this manner, the QP solver iterates until the needs of the (outer) algorithm are met. These comments also apply to the Hessian modification strategy in Algorithm 1; i.e., in an efficient implementation, these modifications would be performed in the QP solver itself to avoid any computational expense that may be caused by completely restarting the solver.

3. Convergence analysis.

In this section, we analyze the convergence properties of Algorithm 2 when Algorithm 1 is employed as the QP solver framework. We first prove that each iteration of the algorithm is well-posed, and then prove that the algorithm is globally convergent to the set of first-order stationary points for (NLP), or at least that of (FP).

It is worthwhile to remind the reader that while updated multiplier estimates for (FP) are computed as part of each scenario, these quantities may also be updated at the end of iteration k by (2.36). Similarly, while the Hessian approximations

Algorithm 2 Sequential quadratic optimizer with inexact subproblem solves.

1: Set $k \leftarrow 0$ and $(x_k, y'_k, \bar{y}'_k, y''_k, \bar{y}''_k, \mu_k)$ satisfying (2.1) and (2.16).
2: Check for finite termination by performing the following.
  a: If (2.15a) holds, then terminate and return the KKT point $(x_k, y'_k/\mu_k, \bar{y}'_k/\mu_k)$.
  b: If (2.15b) holds, then terminate and return the infeasible stationary point x_k.
3: Initialize the symmetric Hessian approximations H'_k and H''_k.
4: Check for a trivial iteration by performing the following.
  a: If Conditions 1 hold, then perform Updates 1 and go to step 8. // **(Scenario 1)**
5: Use Algorithm 1 to compute an approximate solution of (PQP) satisfying Termination Test 1 or 3, and initialize $(d''_k, y''_{k+1}, \bar{y}''_{k+1}) \leftarrow (0, y''_k, \bar{y}''_k)$ by default.
  a: If Conditions 2 hold, then perform Updates 2 and go to step 8. // **(Scenario 2)**
6: If Termination Test 1 holds, then use Algorithm 1 to compute an approximate solution of (FQP) satisfying Termination Test 4. In any case, do the following.
  a: If Conditions 3 hold, then perform Updates 3 and go to step 8. // **(Scenario 3)**
  b: If Conditions 5 hold, then perform Updates 5 and go to step 8. // **(Scenario 5)**
7: Use Algorithm 1 to compute an approximate solution of (PQP) satisfying Termination Test 2 or 3. If Termination Test 1 holds, then use Algorithm 1 to (re)compute an approximate solution of (FQP) satisfying Termination Test 4. In any case, do the following.
  a: If Conditions 3 hold, then perform Updates 3 and go to step 8. // **(Scenario 3)**
  b: If Conditions 4 hold, then perform Updates 4 and go to step 8. // **(Scenario 4)**
  c: If Conditions 5 hold, then perform Updates 5 and go to step 8. // **(Scenario 5)**
  d: Conditions 6 hold, so perform Updates 6. // **(Scenario 6)**
8: Compute α_k as the largest value in $\{\gamma^0, \gamma^1, \gamma^2, \ldots\}$ such that (2.14) is satisfied.
9: Set $x_{k+1} \leftarrow x_k + \alpha_k d_k$, $(y''_{k+1}, \bar{y}''_{k+1})$ by (2.36), and $k \leftarrow k+1$, then go to step 2.

are initialized at the start of each iteration, these matrices may be updated via the modification strategy in Algorithm 1. Hence, for clarity in our analysis, we specify the following about our notation: by $(y'_k, \bar{y}'_k)$ and $(y''_k, \bar{y}''_k)$ (with subscript k), we are referring to the multiplier estimates for (NLP) and (FP), respectively, that are available at the start of iteration k; unless otherwise specified, by $(y'_{k+1}, \bar{y}'_{k+1})$ and $(y''_{k+1}, \bar{y}''_{k+1})$ (with subscript $k+1$) we are referring to the multiplier estimates obtained at the end of iteration k; and by H'_k and H''_k, we are referring to the values of these matrices obtained at the end of iteration k.

3.1. Well-posedness. We show that either Algorithm 2 will terminate finitely, or it will produce an infinite sequence of iterates satisfying (2.1). This well-posedness property of Algorithm 2 is proved under the following assumption.

ASSUMPTION 3.1. *The functions f, c, and $\bar{c}$ are continuously differentiable in an open convex set Ω containing the sequences $\{x_k\}$ and $\{x_k + d_k\}$.*

We also require the following assumptions about the QP solver employed in Algorithm 1 to solve subproblems (PQP) and (FQP). We state these assumptions and then discuss their implications vis-à-vis our termination tests in a series of lemmas.

ASSUMPTION 3.2. *Suppose that with $\mu \in (0, \infty)$ and a fixed H, Algorithm* 1 *repeatedly executes step* 3*. Then the following hold:*

(a) *If x_k is stationary for $\phi(\cdot, \mu)$, then the executions of step* 3 *will eventually produce $y \in [-e, e]$ and $\bar{y} \in [0, e]$ with $\rho_k(0, y, \bar{y}, \mu, H)$ arbitrarily small.*
(b) *If x_k is not stationary for $\phi(\cdot, \mu)$, then the executions of step* 3 *will eventually produce d, $y \in [-e, e]$, and $\bar{y} \in [0, e]$ with $\rho_k(d, y, \bar{y}, \mu, H)$ arbitrarily small and $\Delta l_k(d, \mu) \geq \frac{1}{2} d^T H d$.*

ASSUMPTION 3.3. *Suppose that with $\mu = 0$ and a fixed H, Algorithm* 1 *repeatedly executes step* 3*. Then, with $\Delta l_k(d'_k, \mu_k) > 0$, the executions of step* 3 *will eventually* *produce* d, $y \in [-e, e]$, *and* $\bar{y} \in [0, e]$ *with* $\rho_k(d, y, \bar{y}, 0, H)$ *arbitrarily small and either* $\Delta l_k(d, 0) \geq \frac{1}{2} d^T H d$ *or* $\Delta l_k(d'_k, \mu_k) \geq \frac{1}{2} d^T H d$.

We remark that Assumptions 3.2 and 3.3 only concern situations in which Algorithm 1 repeatedly executes step 3 while H remains fixed. By the construction of the algorithm, if an execution of step 3 ever yields d with $\frac{1}{2} d^T H d < \theta \|d\|^2$, then H will be modified. Hence, these assumptions do not apply until H remains fixed during a run of Algorithm 1, which is in fact guaranteed to occur after a finite number of executions of step 3; see Lemma 3.4 below. We also remark that the last inequality in Assumption 3.2(b) merely requires that d yield an objective value for (2.37) (with $\mu \in (0, \infty)$) that is at least as good as that yielded by the zero vector, which is a reasonable assumption for many QP solvers. Similarly, the last inequalities in Assumption 3.3 require that d yields an objective value for (2.37) (with $\mu = 0$) that is at least as good as that yielded by the zero vector (which is reasonable when x_k is not stationary for $\phi(\cdot, 0) = v(\cdot)$), or that the inner product $\frac{1}{2} d^T H d$ is less than $\Delta l_k(d'_k, \mu_k)$ (which is reasonable when x_k is stationary for v since then there exists a stationary point for (FQP) with $d = 0$).

Our first result relates to the strategy for modifying H in step 4 of Algorithm 1.

LEMMA 3.4. *During a run of Algorithm* 1*, the matrix* H *will be modified in step* 4 *only a finite number of times. Hence, after a finite number of executions of step* 3*, all subsequent executions in the run will yield* $\frac{1}{2} d^T H d \geq \theta \|d\|^2$.

Proof. If Algorithm 1 does not terminate prior, then after a finite number of modifications of H, it will satisfy $H \succeq 2\theta I$, after which point the condition in step 4 will never be satisfied and no further modifications will be triggered. ∎

We now prove the following result for the case when Algorithm 1 is employed to solve (PQP) when the current iterate is a stationary point for the penalty function.

LEMMA 3.5. *Suppose that* x_k *is stationary for* $\phi(\cdot, \mu_k)$*, but* $\|\rho(x_k, y'_k, \bar{y}'_k, \mu_k)\| > 0$. *Then if Algorithm* 1 *is employed to solve* (PQP)*, Termination Test* 3 *will be satisfied after a finite number of executions of step* 3.

Proof. By Lemma 3.4, we have that after a finite number of executions of step 3, H will remain fixed. Hence, without loss of generality, we may assume that all executions of step 3 have $(\mu, H) = (\mu_k, H'_k)$. Then, under Assumption 3.2, repeated executions of step 3 will eventually produce $y \in [-e, e]$ and $\bar{y} \in [0, e]$ with $\rho_k(0, y, \bar{y}, \mu_k, H'_k)$ arbitrarily small. Since $\|\rho(x_k, y'_k, \bar{y}'_k, \mu_k)\| > 0$, it follows that $(y'_{k+1}, \bar{y}'_{k+1}) = (y, \bar{y})$ will satisfy (2.18) and (2.25) after a finite number of such executions. Moreover, if (2.26) holds, then by (2.16) we have $v_k > 0$. Hence, $\rho_k(0, y, \bar{y}, \mu_k, H'_k) \to 0$ implies $\|(y, \bar{y})\|_\infty \to 1$, from which we conclude that with $\rho_k(0, y, \bar{y}, \mu_k, H'_k)$ sufficiently small, $(y'_{k+1}, \bar{y}'_{k+1}) = (y, \bar{y})$ will satisfy (2.27). ∎

Similar results follow when Algorithm 1 is employed to solve (PQP) when the current iterate is not stationary for the penalty function. However, before considering that case, we prove the following lemma related to first-order stationary points of (2.37).

LEMMA 3.6. *For any iteration* k *and constant* $\sigma > 0$*, there exists* $\bar{\sigma} > 0$ *such that if* $\|\rho_k(d, y, \bar{y}, \mu, H)\| \leq \bar{\sigma}$ *and* $\Delta l_k(d, \mu) \geq \theta \|d\|^2$*, then*

$$\Delta l_k(d, \mu) \geq \Delta l_k(d, 0) - \|(y, \bar{y})\|_\infty v_k + d^T H d - \sigma.$$

Proof. First note that the inequality $\Delta l_k(d, \mu) \geq \theta \|d\|^2$ implies that d is bounded since $\Delta l_k(\cdot, \mu)$ is globally Lipschitz continuous for any given k and $\mu \geq 0$. Now consider an arbitrary $\sigma > 0$. If for some $\bar{\sigma} > 0$ we have $\|\rho_k(d, y, \bar{y}, \mu, H)\| \leq \bar{\sigma}$, then by the boundedness of d it follows that for some $C > 0$ independent of $\bar{\sigma}$ we have

$$-\mu g_k^T d - d^T H d - (J_k y + \bar{J}_k \bar{y})^T d \geq -C\bar{\sigma}$$
$$\text{and} \quad (c_k + J_k^T d)^T y + (\bar{c}_k + \bar{J}_k^T d)^T \bar{y} \geq -C\bar{\sigma}.$$

We then obtain by the definition of Δl_k and the Cauchy–Schwarz inequality that

$$\begin{aligned}\Delta l_k(d,\mu) - d^T H d &= \Delta l_k(d,0) - \mu g_k^T d - d^T H d \\ &\geq \Delta l_k(d,0) + (J_k y + \bar{J}_k \bar{y})^T d - C\bar{\sigma} \\ &= \Delta l_k(d,0) + (c_k + J_k^T d)^T y - c_k^T y + (\bar{c}_k + \bar{J}_k^T d)^T \bar{y} - \bar{c}_k^T \bar{y} - C\bar{\sigma} \\ &\geq \Delta l_k(d,0) - \|(y,\bar{y})\|_\infty v_k - 2C\bar{\sigma}.\end{aligned}$$

The result follows by choosing $\bar{\sigma}$ sufficiently small such that $2C\bar{\sigma} \leq \sigma$. ∎

We now consider the employment of Algorithm 1 when x_k is not stationary for the penalty function. As will be seen in the proof of Lemma 3.9, Algorithm 1 will only be employed to find an inexact solution that satisfies Termination Test 2 if the residual for the feasibility problem is nonzero.

LEMMA 3.7. *Suppose that x_k is not stationary for $\phi(\cdot, \mu_k)$. Then if Algorithm* 1 *is employed to solve* (PQP)*, Termination Test* 1 *will be satisfied after a finite number of executions of step* 3*. Moreover, if $\|\rho(x_k, y_k'', \bar{y}_k'', 0)\| > 0$, then Termination Test* 2 *will also be satisfied after a finite number of such executions.*

Proof. By Lemma 3.4, we have that after a finite number of executions of step 3, H will remain fixed. Hence, without loss of generality, we may assume that all executions of step 3 have $(\mu, H) = (\mu_k, H_k')$, and that all values of d computed in step 3 satisfy $\frac{1}{2} d^T H d \geq \theta \|d\|^2$. Since x_k is not stationary for $\phi(\cdot, \mu_k)$, it follows that $\|\rho(x_k, y_k', \bar{y}_k', \mu_k)\| > 0$ and that $d \neq 0$ in any first-order stationary point $(d, y, \bar{y})$ of (2.37). Hence, under Assumption 3.2, we have that after a finite number of executions of step 3, the vector $(d_k', y_{k+1}', \bar{y}_{k+1}') = (d, y, \bar{y})$ will satisfy (2.18), (2.17), and (2.19).

Now suppose that $\|\rho(x_k, y_k'', \bar{y}_k'', 0)\| > 0$. Then, by the same argument as above, we have that after a finite number of executions of step 3, the vector $(d_k', y_{k+1}', \bar{y}_{k+1}') = (d, y, \bar{y})$ will satisfy (2.18), (2.19), and (2.20). Moreover, note that if a first-order stationary point $(d, y, \bar{y})$ of (2.37) (with $\rho_k(d, y, \bar{y}, \mu_k, H_k') = 0$) has $\|(y, \bar{y})\|_\infty < \lambda(\epsilon - \beta) \in (0,1)$, then it also has $\Delta l_k(d, 0) = v_k$. Hence, (2.21) will imply (2.22) when $\|\rho_k(d_k', y_{k+1}', \bar{y}_{k+1}', \mu_k, H_k')\|$ is sufficiently small. Now, since $\|\rho(x_k, y_k'', \bar{y}_k'', 0)\| > 0$, we have from (2.16) that $v_k > 0$. Moreover, from Lemma 3.6, we have that for any constant $\sigma > 0$, there exists $\bar{\sigma} > 0$ such that the inequalities $\|\rho_k(d, y, \bar{y}, \mu_k, H_k')\| \leq \bar{\sigma}$ and $\Delta l_k(d, \mu_k) \geq \theta \|d\|^2 > 0$ imply

$$\mu_k g_k^T d \leq \|(y, \bar{y})\|_\infty v_k - d^T H_k' d + \sigma < \|(y, \bar{y})\|_\infty v_k + \sigma.$$

Consequently, when $\|\rho_k(d_k', y_{k+1}', \bar{y}_{k+1}', \mu_k, H_k')\|$ is sufficiently small, (2.24) will hold (regardless of whether or not (2.23) is satisfied). ∎

We now prove a similar result for when Algorithm 1 is employed to solve (FQP). As can be seen in Algorithm 2 and the proof of Lemma 3.9, this occurs only when Termination Test 1 is satisfied (which requires $\Delta l_k(d_k', \mu_k) > 0$) and $\|\rho(x_k, y_k'', \bar{y}_k'', 0)\| > 0$.

LEMMA 3.8. *Suppose that $\Delta l_k(d_k', \mu_k) > 0$ and $\|\rho(x_k, y_k'', \bar{y}_k'', 0)\| > 0$. Then if Algorithm* 1 *is employed to solve* (FQP)*, Termination Test* 4 *will be satisfied after a finite number of executions of step* 3.

Proof. By Lemma 3.4, we have that after a finite number of executions of step 3, H will remain fixed. Hence, without loss of generality, we may assume that all executions of step 3 have $(\mu, H) = (0, H''_k)$, and that all values of d computed in step 3 satisfy $\frac{1}{2}d^T H d \geq \theta\|d\|^2$. Under Assumption 3.3, repeated executions of step 3 will eventually produce d, $y \in [-e, e]$, and $\bar{y} \in [0, e]$ with $\rho_k(d, y, \bar{y}, 0, H''_k)$ arbitrarily small, meaning that $(d''_k, y''_{k+1}, \bar{y}''_{k+1}) = (d, y, \bar{y})$ will satisfy (2.29), (2.28), and (2.30) after a finite number of such executions. ∎

Now that we have established that Algorithm 1 will terminate finitely in a variety of situations of interest, we prove the following lemma showing that Algorithm 1 will always terminate finitely in the context of Algorithm 2.

LEMMA 3.9. *Algorithm* 1 *terminates finitely whenever it is called by Algorithm* 2.

Proof. Consider the call to Algorithm 1 to solve (PQP) in step 5 of Algorithm 2. If (2.15a) holds, or if Conditions 1 hold, then Algorithm 2 would have terminated in step 2, or at least would have skipped step 5. Thus, we may assume that (2.15a) and Conditions 1 do not hold, meaning that $\|\rho(x_k, y'_k, \bar{y}'_k, \mu_k)\| > 0$. If x_k is stationary for $\phi(\cdot, \mu_k)$, then by Lemma 3.5 we have that Algorithm 1 will terminate finitely with $(d'_k, y'_{k+1}, \bar{y}'_{k+1})$ satisfying Termination Test 3. (Termination Test 1 cannot be satisfied when x_k is stationary for $\phi(\cdot, \mu_k)$ due to the strict inequality in (2.19).) Similarly, if x_k is not stationary for $\phi(\cdot, \mu_k)$, then by Lemma 3.7 we have that Algorithm 1 will terminate finitely with $(d'_k, y'_{k+1}, \bar{y}'_{k+1})$ satisfying Termination Test 1 and/or 3.

Next, consider the call to Algorithm 1 to approximately solve (FQP) in step 6 of Algorithm 2, which occurs only if Termination Test 1 holds (and so $\Delta l_k(d'_k, \mu_k) > 0$). If $v_k = 0$, then the satisfaction of (2.19) implies the satisfaction of (2.9). Consequently, Conditions 2 would have been satisfied in step 5 of Algorithm 2, which would have caused the algorithm to skip step 6. Thus, we may assume $v_k > 0$, which in turn means $\|\rho(x_k, y''_k, \bar{y}''_k, 0)\| > 0$, or else Algorithm 2 would have terminated in step 2. It then follows from Lemma 3.8 that Algorithm 1 will terminate finitely with $(d''_k, y''_{k+1}, \bar{y}''_{k+1})$ satisfying Termination Test 4.

Finally, consider the calls to Algorithm 1 in step 7 of Algorithm 2. We claim that we must have $\|\rho(x_k, y''_k, \bar{y}''_k, 0)\| > 0$ in this step. Indeed, if $\|\rho(x_k, y''_k, \bar{y}''_k, 0)\| = 0$, then we must have $v_k = 0$, or else Algorithm 1 would have terminated in step 2 since (2.15b) would have been satisfied. Moreover, since $v_k = 0$, if Termination Test 1 was satisfied in step 5, then Conditions 2 would have been satisfied and Algorithm 1 would have skipped to step 8. Consequently, we may assume that Termination Test 3, but not Termination Test 1, held in step 5. However, since Termination Test 3 held after step 5 and $\|\rho(x_k, y''_k, \bar{y}''_k, 0)\| = 0$, it follows that Conditions 5 would have held in step 6, meaning that Algorithm 2 would have skipped to step 8. Overall, we have shown that we must have $\|\rho(x_k, y''_k, \bar{y}''_k, 0)\| > 0$ in step 7. Consequently, by Lemmas 3.7 and 3.8, we conclude that Algorithm 1 will terminate finitely with $(d'_k, y'_{k+1}, \bar{y}'_{k+1})$ satisfying Termination Test 2 and/or 3, and then if Termination Test 1 holds, it will terminate finitely with $(d''_k, y''_{k+1}, \bar{y}''_{k+1})$ satisfying Termination Test 4. ∎

Our next lemma shows that one of our proposed scenarios will occur.

LEMMA 3.10. *If Algorithm* 2 *does not terminate in step* 2*, then Scenario* 1, 2, 3, 4, 5*, or* 6 *will occur.*

Proof. If Algorithm 2 does not terminate in step 2 and Conditions 1 hold, then Scenario 1 occurs. Otherwise, without loss of generality, we may assume that Algorithm 2 reaches step 7, in which case it follows from Lemma 3.9 that a primal-dual vector satisfying either Termination Test 2 or 3 will be computed. In particular, if Termination Test 2 holds, then Termination Test 1 also holds, and Algorithm 2 will proceed to compute a primal-dual vector satisfying Termination Test 4. Consequently, it follows that in step 7 either Termination Tests 1, 2, and 4 hold, or at least Termination Test 3 holds. In the former case when Termination Tests 1, 2, and 4 are satisfied, then at least Conditions 4 hold, in which case Scenario 4 (if not Scenario 3) would occur. Otherwise, when Termination Test 3 is satisfied, it is clear that either Conditions 5 or 6 hold, in which case Scenario 5 or 6, respectively, would occur. ∎

The major consequence of the previous lemma is that if Algorithm 2 does not terminate in iteration k, then exactly one scenario will occur. For ease of exposition in the remainder of our analysis, we define

$$K_i := \{k \mid \text{Scenario } i \text{ occurs in iteration } k\}.$$

We now show that the sequence of penalty parameters will be positive.

LEMMA 3.11. *For all k, it follows that $\mu_{k+1} \in (0, \mu_k]$.*

Proof. Note that $\mu_{k+1} \leftarrow \mu_k$ for $k \in K_2 \cup K_3 \cup K_5$. Moreover, for $k \in K_1 \cup K_6$, we have $\mu_{k+1} \leftarrow \delta\mu_k$. Thus, we need only show that $\mu_{k+1} \in (0, \mu_k]$ for $k \in K_4$.

Consider $k \in K_4$. By the definition of Δl_k, we have

$$\Delta l_k(d_k, \mu_k) \geq \beta \Delta l_k(d_k, 0) \iff \mu_k g_k^T d_k \leq (1-\beta)\Delta l_k(d_k, 0). \tag{3.1}$$

Consequently, it follows from (2.13) that $\mu_{k+1} \in \{\delta\mu_k, \mu_k\} > 0$ unless we find

$$\mu_k g_k^T d_k > (1-\beta)\Delta l_k(d_k, 0) \geq (1-\beta)\epsilon\theta\|d_k''\|^2,$$

where the latter inequality follows from (2.12) and (2.33). This immediately implies

$$g_k^T d_k > 0 \quad \text{and} \quad \|d_k\| > 0. \tag{3.2}$$

In such cases, we set μ_{k+1} by (2.13) where

$$\frac{(1-\beta)\Delta l_k(d_k, 0)}{g_k^T d_k + \theta\|d_k\|^2} \geq \frac{(1-\beta)\epsilon\theta\|d_k''\|^2}{g_k^T d_k + \theta\|d_k\|^2}.$$

The relationships in (2.32) and the inequalities in (3.2), respectively, imply that the numerator and denominator of the right-hand side of this expression is positive, meaning that μ_{k+1} set by (2.13) is both positive and less than or equal to μ_k. ∎

Our next goal is to prove that the backtracking line search in Algorithm 2 is well-posed. This requires the following result, which states that $-\Delta l_k(\cdot, \mu)$ can be used as a surrogate for the directional derivative of $\phi(\cdot, \mu)$ at x_k, call it $D\phi(\cdot; x_k, \mu)$; for a proof, see, e.g., [3, 4]. This fact will be used in the proof of the subsequent lemma to show that Algorithm 2 produces a direction of strict descent for $\phi(\cdot, \mu_{k+1})$ from x_k as long as $\Delta l_k(d_k, \mu_{k+1}) > 0$.

LEMMA 3.12. *At any iterate x_k and for any $\mu \geq 0$ and $d \in \mathbb{R}^n$, it follows that*

$$D\phi(d; x_k, \mu) \leq -\Delta l_k(d, \mu).$$

Thus, if $\Delta l_k(d, \mu) > 0$, then d is a direction of strict descent for $\phi(\cdot, \mu)$ from x_k.

We now provide a nonnegative lower bound for the model reduction corresponding to the new value of the penalty parameter, and consequently show that the backtracking line search in Algorithm 2 is well-posed.

LEMMA 3.13. *For all $k \notin K_4$ we have*

$$\Delta l_k(d_k, \mu_{k+1}) \geq \theta\|d_k\|^2,$$

and for all $k \in K_4$ we have

$$\Delta l_k(d_k, \mu_{k+1}) \geq \beta\Delta l_k(d_k, 0) \geq \beta\epsilon\theta\|d_k''\|^2 > 0.$$

Consequently, for all k we have $\alpha_k > 0$.

Proof. The first statement in the lemma is trivial if $k \in K_1 \cup K_5 \cup K_6$ since for all such k we set $d_k \leftarrow 0$. For $k \in K_2 \cup K_3$, we have from (2.19) and the facts that $d_k \leftarrow d'_k$ and $\mu_{k+1} \leftarrow \mu_k$ that

$$\Delta l_k(d_k, \mu_{k+1}) \geq \theta \|d_k\|^2 > 0.$$

Finally, consider $k \in K_4$, where by (2.33) we have $d_k \neq 0$. We proceed by considering the three possibilities in (2.13). If $\tau_k \geq \tau$ and $\Delta l_k(d_k, \mu_k) \geq \beta \Delta l_k(d_k, 0)$, then we set $\mu_{k+1} \leftarrow \mu_k$ and by (2.12), (2.32), and (2.33) have

$$\Delta l_k(d_k, \mu_{k+1}) = \Delta l_k(d_k, \mu_k) \geq \beta \Delta l_k(d_k, 0) \geq \beta \epsilon \theta \|d''_k\|^2 > 0.$$

Otherwise, if $\tau_k < \tau$ and $\Delta l_k(d_k, \mu_k) \geq \beta \Delta l_k(d_k, 0)$, then since we have $\mu_k g_k^T d_k \leq (1-\beta)\Delta l_k(d_k, 0)$ (recall (3.1)) and since (2.12) and (2.32) imply $\Delta l_k(d_k, 0) > 0$, we also have $\delta \mu_k g_k^T d_k \leq (1-\beta)\Delta l_k(d_k, 0)$. Thus, after setting $\mu_{k+1} \leftarrow \delta \mu_k$ by (2.13), we have from (2.12), (2.32), and (2.33) (and recalling (3.1)) that

$$\Delta l_k(d_k, \mu_{k+1}) \geq \beta \Delta l_k(d_k, 0) \geq \beta \epsilon \theta \|d''_k\|^2 > 0.$$

Finally, if $\Delta l_k(d_k, \mu_k) < \beta \Delta l_k(d_k, 0)$, then after setting $\mu_{k+1} < \mu_k$ by (2.13), we have from the fact that $g_k^T d_k > 0$ (recall (3.2)) and (2.12), (2.13), (2.32), and (2.33) that

$$\begin{aligned}\Delta l_k(d_k, \mu_{k+1}) &= -\mu_{k+1} g_k^T d_k + \Delta l_k(d_k, 0) \\ &\geq -\left(\frac{(1-\beta)\Delta l_k(d_k, 0)}{g_k^T d_k + \theta \|d_k\|^2}\right) g_k^T d_k + \Delta l_k(d_k, 0) \\ &\geq \beta \Delta l_k(d_k, 0) \geq \beta \epsilon \theta \|d''_k\|^2 > 0.\end{aligned}$$

The final statement in the lemma follows since for $k \in K_1 \cup K_5 \cup K_6$ we set $\alpha_k \leftarrow 1$, and for $k \in K_2 \cup K_3 \cup K_4$ we have $\Delta l_k(d_k, \mu_{k+1}) > 0$; in the latter case Lemma 3.12 implies that step 8 of Algorithm 2 yields $\alpha_k > 0$. ∎

We now have the following theorem about the well-posedness of Algorithm 2.

THEOREM 3.14. *One of the following holds:*

(a) *Algorithm* 2 *terminates with a KKT point or an infeasible stationary point satisfying a condition in* (2.15).

(b) *Algorithm* 2 *generates an infinite sequence of iterates satisfying* (2.1).

Proof. If, during iteration k, Algorithm 2 does not terminate in step 2, then it follows from Lemmas 3.9 and 3.13 that each call to Algorithm 1 and the backtracking line search will terminate finitely, which in turn implies that all steps in iteration k will terminate finitely. Moreover, it follows from our updating strategies for the multiplier estimates and Lemma 3.11 that (2.1) will hold at the start of the next iteration. Indeed, by induction, (2.1) will hold at the start of all subsequent iterations. ∎

3.2. Global convergence. Under the assumption that Algorithm 2 does not terminate finitely—and so, by Theorem 3.14, produces an infinite sequence of iterations satisfying (2.1)—we prove that appropriate measures of stationarity for problems (NLP) and (FP) (see (2.15)) converge to zero. Overall, we prove that Algorithm 2 possesses meaningful global convergence guarantees.

We make the following assumptions for our analysis in this section.

ASSUMPTION 3.15. *The functions f, c, and $\bar{c}$ are continuously differentiable in an open convex set Ω containing the sequences $\{x_k\}$ and $\{x_k + d_k\}$. Moreover, in Ω, the functions and their first derivatives are bounded and Lipschitz continuous.*

ASSUMPTION 3.16. *The Hessian matrices H'_k and H''_k—including their initial values and those returned from Algorithm* 1*—are bounded in norm.*

Assumption 3.15 represents a strengthening of Assumption 3.1. Moreover, we continue to make Assumptions 3.2 and 3.3 so that all of the results in section 3.1 apply.

Our first lemma in this section shows that the search directions are bounded.

LEMMA 3.17. *The sequences $\{d'_k\}$, $\{d''_k\}$, and $\{d_k\}$ are bounded in norm.*

Proof. Consider $\{d'_k\}$. For $k \in K_1 \cup K_5 \cup K_6$, Algorithm 2 sets $d'_k \leftarrow 0$. Otherwise, Termination Test 1 or 2 holds, so by (2.19) we have

$$\Delta l_k(d'_k, \mu_k) \geq \theta \|d'_k\|^2.$$

Since all quantities (other than d'_k) in the piecewise linear function on the left-hand side of this inequality are uniformly bounded by Assumption 3.15, and since $\mu_k \in (0, \mu_0]$ for all k by Lemma 3.11, it follows that d'_k is uniformly bounded in norm.

Now consider $\{d''_k\}$. For $k \in K_1 \cup K_2 \cup K_5 \cup K_6$, Algorithm 2 sets $d''_k \leftarrow 0$. Otherwise, Termination Test 4 holds, so by (2.30) we have

$$\max\{\Delta l_k(d'_k, \mu_k), \Delta l_k(d''_k, 0)\} \geq \theta \|d''_k\|^2.$$

The fact that d''_k is uniformly bounded in norm follows due to reasoning (for d'_k) similar to that in the previous paragraph.

Finally, since for all k Algorithm 2 sets d_k by (2.8) (i.e., as a convex combination of d'_k and d''_k), it follows from above that d_k is uniformly bounded in norm. ◻

We now provide a lower bound on the step-sizes that is more precise than that given by Lemma 3.13.

LEMMA 3.18. *For all k, the step-size satisfies $\alpha_k \geq \omega \Delta l_k(d_k, \mu_{k+1})$ for some constant $\omega > 0$ independent of k.*

Proof. The result is trivial for $k \in K_1 \cup K_5 \cup K_6$ since for all such k Algorithm 2 sets $d_k \leftarrow 0$ and $\alpha_k \leftarrow 1$. It remains to consider $k \in K_2 \cup K_3 \cup K_4$ where from (2.10), (2.19), and (2.33) we have that $d_k \neq 0$.

Let $\bar{\alpha}$ be a step-size for which (2.14) is not satisfied, i.e.,

$$\phi(x_k + \bar{\alpha} d_k, \mu_{k+1}) - \phi(x_k, \mu_{k+1}) > -\eta \bar{\alpha} \Delta l_k(d_k, \mu_{k+1}).$$

Using Assumption 3.15, Taylor's theorem, and the convexity of $\|\cdot\|_1$, we also have

$$\begin{aligned}
&\phi(x_k + \bar{\alpha} d_k, \mu_{k+1}) - \phi(x_k, \mu_{k+1}) \\
&= \mu_{k+1}(f(x_k + \bar{\alpha} d_k) - f_k) + v(x_k + \bar{\alpha} d_k) - v_k \\
&\leq \bar{\alpha} \mu_{k+1} g_k^T d_k + \bar{\alpha}(\|c_k + J_k^T d_k\|_1 + \|[\bar{c}_k + \bar{J}_k^T d_k]^+\|_1) + (1 - \bar{\alpha}) v_k - v_k + \bar{\alpha}^2 C \|d_k\|^2 \\
&= -\bar{\alpha} \Delta l_k(d_k, \mu_{k+1}) + \bar{\alpha}^2 C \|d_k\|^2
\end{aligned}$$

for some $C > 0$ independent of k. Combining the last two inequalities, we have

$$\bar{\alpha} C \|d_k\|^2 > (1 - \eta) \Delta l_k(d_k, \mu_{k+1}),$$

which implies that the line search yields $\alpha_k \geq \gamma(1 - \eta) \Delta l_k(d_k, \mu_{k+1}) / (C \|d_k\|^2)$. The result then follows since $\{d_k\}$ is bounded by Lemma 3.17. ◻

Our next goal is to prove that, in the limit, the sequence of reductions in the model of the penalty function and of the constraint violation measure converge to zero. For this, it will be convenient to work with the shifted penalty function

$$\varphi(x, \mu) := \mu(f(x) - \underline{f}) + v(x),$$

where $\underline{f}$ is the infimum of f over the smallest convex set containing $\{x_k\}$ whose existence follows under Assumption 3.15. The function φ satisfies an important monotonicity property proved in the following lemma.

LEMMA 3.19. *For all k,*

$$\varphi(x_{k+1},\mu_{k+2}) \le \varphi(x_k,\mu_{k+1}) - \eta\alpha_k\Delta l_k(d_k,\mu_{k+1}),$$

implying that $\{\varphi(x_k,\mu_{k+1})\}$ decreases monotonically.

Proof. According to the line search condition (2.14), we have

$$\varphi(x_{k+1},\mu_{k+1}) \le \varphi(x_k,\mu_{k+1}) - \eta\alpha_k\Delta l_k(d_k,\mu_{k+1}).$$

This inequality implies

$$\varphi(x_{k+1},\mu_{k+2}) \le \varphi(x_k,\mu_{k+1}) - (\mu_{k+1}-\mu_{k+2})(f_{k+1}-\underline{f}) - \eta\alpha_k\Delta l_k(d_k,\mu_{k+1}).$$

The result follows since $\Delta l_k(d_k,\mu_{k+1}) \ge 0$ (by Lemma 3.13), $\{\mu_k\}$ is monotonically decreasing, and $f_k \ge \underline{f}$ for all k. ∎

We now prove that the model reductions and search directions converge to zero.

LEMMA 3.20. *The following limits hold:*

$$0 = \lim_{k\to\infty}\Delta l_k(d_k,\mu_{k+1}) = \lim_{k\to\infty}\|d_k''\| = \lim_{k\to\infty}\|d_k'\| = \lim_{k\to\infty}\|d_k\| = \lim_{k\to\infty}\Delta l_k(d_k,0). \tag{3.3}$$

Proof. By Lemmas 3.18 and 3.19, if there exists an infinite subsequence of iterations with $\Delta l_k(d_k,\mu_{k+1}) \ge C$ for some constant $C > 0$, then we must have $\varphi(x_k,\mu_{k+1}) \to -\infty$. However, that contradicts the fact that φ is bounded below by zero. Hence, we must have that $\Delta l_k(d_k,\mu_{k+1}) \to 0$.

Now consider $\{d_k''\}$. For $k \in K_1\cup K_2\cup K_5\cup K_6$, we have $d_k'' \leftarrow 0$. Otherwise, for $k \in K_3$ we have from (2.11), (2.30), and the facts that $d_k \leftarrow d_k'$ and $\mu_{k+1} \leftarrow \mu_k$ that

$$\Delta l_k(d_k,\mu_{k+1}) = \Delta l_k(d_k',\mu_k) \ge \begin{cases}\theta\|d_k''\|^2 & \text{if } \Delta l_k(d_k',\mu_k) \ge \Delta l_k(d_k'',0),\\ \epsilon\Delta l_k(d_k'',0) \ge \epsilon\theta\|d_k''\|^2 & \text{otherwise,}\end{cases}$$

and similarly for $k \in K_4$ we have from Lemma 3.13 that

$$\Delta l_k(d_k,\mu_{k+1}) \ge \beta\epsilon\theta\|d_k''\|^2.$$

Since $\Delta l_k(d_k,\mu_{k+1}) \to 0$, it follows from these last two expressions that $d_k'' \to 0$.

Now consider $\{d_k'\}$. For $k \in K_1\cup K_5\cup K_6$, we have $d_k' \leftarrow 0$. Otherwise, for $k \in K_2\cup K_3$, we have from (2.19) and the facts that $d_k \leftarrow d_k'$ and $\mu_{k+1} \leftarrow \mu_k$ that

$$\Delta l_k(d_k,\mu_{k+1}) \ge \theta\|d_k'\|^2.$$

Finally, for $k \in K_4$ we have from (2.32) that

$$\Delta l_k(d_k'',0) \ge \theta\|d_k'\|^2.$$

Since $\Delta l_k(d_k,\mu_{k+1}) \to 0$ and $d_k'' \to 0$, it follows that $d_k' \to 0$.

The last two limits in (3.3) follow from (2.8) and since $d_k'' \to 0$ and $d_k' \to 0$. ∎

We now present a useful lemma.

LEMMA 3.21. *Let $\{r_k\}$, $\{e_k\}$, and $\{\bar r_k\}$ be infinite sequences of nonnegative real numbers, and let $\mathcal{K}$ be an infinite subsequence of iteration numbers such that*

$$e_k \to 0, \quad \bar r_k \to 0, \quad \textit{and} \quad r_{k+1} \le \begin{cases}\kappa r_k + e_k & \textit{for } k \in \mathcal{K},\\ \max\{r_k,\bar r_k\} & \textit{for } k \notin \mathcal{K}.\end{cases} \tag{3.4}$$

Then $r_k \to 0$.

Proof. Let $C > 0$ be an arbitrary constant. Since $e_k \to 0$ and $\bar{r}_k \to 0$, there exists $k_1 \geq 0$ such that for all $k \geq k_1$ we have $e_k \leq (1-\kappa)C/2$ and $\bar{r}_k \leq C$. If for $k \geq k_1$ with $k \in \mathcal{K}$ we have $r_k > C$, then since $\kappa \in (0,1)$ the inequality in (3.4) yields

$$\begin{aligned} r_{k+1} &\leq \kappa r_k + \tfrac{1-\kappa}{2}C \\ &= (\kappa - 1)r_k + r_k + \tfrac{1-\kappa}{2}C \\ &< (\kappa - 1)C + r_k + \tfrac{1-\kappa}{2}C \\ &= r_k - \tfrac{1-\kappa}{2}C. \end{aligned}$$

Hence, $r_k - r_{k+1} \geq (1-\kappa)C/2$ for all $k \geq k_1$ with $k \in \mathcal{K}$ and $r_k > C$. This, along with the facts that $\mathcal{K}$ is infinite, $r_{k+1} \leq \max\{r_k, \bar{r}_k\}$ for $k \notin \mathcal{K}$, and $\bar{r}_k \to 0$, means that for some $k_2 \geq k_1$ we find $r_{k_2} \leq C$. If for $k = k_2$ we have $k \notin \mathcal{K}$, then by (3.4) we have $r_{k+1} \leq \max\{r_k, \bar{r}_k\} \leq C$, and otherwise (i.e., when $k \in \mathcal{K}$) we similarly have

$$r_{k+1} \leq \kappa C + \tfrac{1-\kappa}{2}C = \tfrac{\kappa+1}{2}C \leq C.$$

By induction, $r_k \leq C$ for all $k \geq k_2$. The result follows since $C > 0$ was arbitrary. ∎

We now prove that the sequence of residuals for (FP) converges to zero. (At this point, we remind the reader about the manner in which we refer to the multiplier estimates involved during iteration k; see the discussion at the beginning of section 3.)

LEMMA 3.22. *The following limit holds:*

$$\lim_{k\to\infty} \|\rho(x_k, y_k'', \bar{y}_k'', 0)\| = 0. \tag{3.5}$$

Proof. We consider two cases depending on the nature of the set of iterations in which $k \in K_1 \cup K_5 \cup K_6$.

Case 1. Suppose $k \in K_1 \cup K_5 \cup K_6$ for all sufficiently large k, in which case we can assume without loss of generality that $k \in K_1 \cup K_5 \cup K_6$ for all $k \geq 0$. It follows that $x_{k+1} \leftarrow x_k$ for all $k \geq 0$, and from Updates 1, 5, and 6 (in particular, from (2.31) and (2.35)) and (2.36), we have that $\{\|\rho(x_k, y_k'', \bar{y}_k'', 0)\|\}$ decreases monotonically. We proceed by distinguishing whether or not $K_1 \cup K_6$ is finite.

If $K_1 \cup K_6$ is finite, then there exists $k_1 \geq 0$ such that $k \in K_5$ for all $k \geq k_1$. Since Conditions 5 hold for all $k \geq k_1$, it follows from Termination Test 3 that (2.25) holds for all $k \geq k_1$, yielding $\rho(x_k, y_k', \bar{y}_k', \mu_k) \to 0$. Consequently, since we also have that (2.26) does not hold for all $k \geq k_1$, we have (3.5).

Now suppose that $K_1 \cup K_6$ is infinite. For $k \in K_1 \cup K_6$, Algorithm 2 updates $\mu_{k+1} \leftarrow \delta\mu_k$, so the fact that $K_1 \cup K_6$ is infinite implies that $\mu_k \to 0$. In particular, if K_1 is infinite, then it follows that

$$\lim_{k\in K_1} \rho(x_k, y_k', \bar{y}_k', 0) = 0,$$

which, along with (2.31) and the fact that $\{\|\rho(x_k, y_k'', \bar{y}_k'', 0)\|\}$ decreases monotonically, implies that (3.5) holds.

It remains to consider the case when $K_1 \cup K_6$ is infinite, but K_1 is finite, or in other words the case when K_6 is infinite and $k \in K_5 \cup K_6$ for all large k. For the purpose of deriving a contradiction, suppose that (3.5) does not hold, i.e., that

$$\lim_{k\to\infty} \|\rho(x_k, y_k'', \bar{y}_k'', 0)\| =: \bar{\rho}_1 > 0. \tag{3.6}$$

(The limit in (3.6) exists due to the Monotone Convergence Theorem.) Define $\zeta_1 := (\zeta+1)/2 \in (\zeta, 1)$ and let $K \subseteq K_6$ be the subset of iterations in which

$$\|\rho(x_k, y_k', \bar{y}_k', \delta\mu_k)\| \geq \zeta_1 \|\rho(x_k, y_k'', \bar{y}_k'', 0)\|. \tag{3.7}$$

If K is infinite, then for some $C_1 > 0$ we have from (3.7) and for all $k \in K$ that

$$\bar{\rho}_1 \leq \|\rho(x_k, y_k'', \bar{y}_k'', 0)\| \leq \tfrac{1}{\zeta_1}\|\rho(x_k, y_k', \bar{y}_k', \delta\mu_k)\| \leq \tfrac{1}{\zeta_1}\|\rho(x_k, y_k', \bar{y}_k', \mu_k)\| + C_1\mu_k. \tag{3.8}$$

Therefore, since $\mu_k \to 0$, there exists $\bar{\rho}_2 > 0$ such that for large $k \in K$ we have $\|\rho(x_k, y_k', \bar{y}_k', \mu_k)\| \geq \bar{\rho}_2$. For such k we further have from (3.8) and the fact that Conditions 6 require that (2.26) hold that

$$\tfrac{1}{\zeta}\|\rho(x_k, y_k', \bar{y}_k', \mu_k)\| < \|\rho(x_k, y_k'', \bar{y}_k'', 0)\| \leq \tfrac{1}{\zeta_1}\|\rho(x_k, y_k', \bar{y}_k', \mu_k)\| + C_1\mu_k.$$

Rearranging terms yields that for sufficiently large $k \in K$ we have

$$0 < \frac{1}{\zeta} - \frac{1}{\zeta_1} < \frac{C_1\mu_k}{\|\rho(x_k, y_k', \bar{y}_k', \mu_k)\|} \leq \frac{C_1\mu_k}{\bar{\rho}_2}.$$

Recalling $\mu_k \to 0$, this constitutes a contradiction to the supposition that K is infinite.

Finally, continuing with the supposition that $\bar{\rho}_1$ exists as in (3.6), consider the case when K is finite, i.e., when $k \in K_5 \cup (K_6 \backslash K)$ for all large k. By definition, inequality (3.7) is violated for $k \in K_6 \backslash K$. Moreover, for $k \in K_6 \backslash K$, Algorithm 2 sets $\mu_{k+1} \leftarrow \delta\mu_k$ and $(y_{k+1}', \bar{y}_{k+1}') \leftarrow (y_k', \bar{y}_k')$, so we have for some constant $C_2 > 0$ that

$$\begin{aligned}\|\rho(x_{k+1}, y_{k+1}', \bar{y}_{k+1}', 0)\| &\leq \|\rho(x_k, y_{k+1}', \bar{y}_{k+1}', \mu_{k+1})\| + C_2\mu_{k+1} \\ &< \zeta_1\|\rho(x_k, y_k'', \bar{y}_k'', 0)\| + C_2\mu_{k+1}.\end{aligned}$$

Since $\zeta_1 \in (0, 1)$ and $\mu_k \to 0$, it follows from above, from (3.6), and from the monotonicity of $\|\rho(x_k, y_k'', \bar{y}_k'', 0)\|$ that for all large $k \in K_6 \backslash K$ we have during Scenario 6 that

$$\|\rho(x_{k+1}, y_{k+1}', \bar{y}_{k+1}', 0)\| \leq \bar{\rho}_1 \leq \|\rho(x_{k+1}, y_{k+1}'', \bar{y}_{k+1}'', 0)\|. \tag{3.9}$$

Hence, the update (2.31) will set $(y_{k+1}'', \bar{y}_{k+1}'') \leftarrow (y_k', \bar{y}_k') = (y_{k+1}', \bar{y}_{k+1}')$ for all sufficiently large $k \in K_6 \backslash K$, and consequently $\|\rho(x_{k+1}, y_{k+1}'', \bar{y}_{k+1}'', 0)\| = \bar{\rho}_1$ will hold for all sufficiently large $k \in K_6 \backslash K$. (Note here that the update (2.36) can only decrease the value of $\|\rho(x_{k+1}, y_{k+1}'', \bar{y}_{k+1}'', 0)\|$.) Moreover, since $\{\|\rho(x_k, y_k'', \bar{y}_k'', 0)\|\}$ is monotonically decreasing, we have the stronger conclusion that

$$\|\rho(x_k, y_k'', \bar{y}_k'', 0)\| = \bar{\rho}_1 \quad \text{for all large } k, \tag{3.10}$$

and since $\mu_k \to 0$, we also have that $\|\rho(x_{k+1}, y_{k+1}'', \bar{y}_{k+1}'', \mu_{k+1})\| \to \bar{\rho}_1$. Now, since $(1+\zeta_1)/2 \in (0, 1)$, it follows that for sufficiently large $k \in K_6 \backslash K$ we have

$$\bar{\rho}_1 \leq \tfrac{2}{1+\zeta_1}\|\rho(x_{k+1}, y_{k+1}'', \bar{y}_{k+1}'', \mu_{k+1})\| = \tfrac{2}{1+\zeta_1}\|\rho(x_{k+1}, y_k', \bar{y}_k', \mu_{k+1})\|,$$

where the last equality follows since the update (2.31) will set $(y_{k+1}'', \bar{y}_{k+1}'') \leftarrow (y_k', \bar{y}_k')$ for all sufficiently large $k \in K_6 \backslash K$ (which followed above due to (3.9)). Since (3.7) does not hold for such k, we find that for sufficiently large $k \in K_6 \backslash K$ we have

$$\bar{\rho}_1 \leq \tfrac{2\zeta_1}{1+\zeta_1}\|\rho(x_k, y_k'', \bar{y}_k'', 0)\| = \tfrac{2\zeta_1}{1+\zeta_1}\bar{\rho}_1,$$

which is a contradiction since $2\zeta_1/(1+\zeta_1) < 1$. Hence, the supposition that there exists $\bar{\rho}_1 > 0$ satisfying (3.6) cannot be true, and as a result we have shown (3.5).

Case 2. Suppose that $K_2 \cup K_3 \cup K_4$ is infinite. From (2.16) and Taylor's theorem, it follows that for some $C_3 > 0$, at the start of iteration $k+1$ with $k \in K_2$, we have

$$\|\rho(x_{k+1}, y''_{k+1}, \bar{y}''_{k+1}, 0)\| \le v_{k+1} \le v_k + C_3\|d_k\|. \tag{3.11}$$

Moreover, if K_2 is infinite, then $\lim_{k\in K_2} v_k = 0$. (To see this, note that for $k \in K_2$ we have $\mu_{k+1} = \mu_k$ and $d_k = d'_k$, and hence from (2.9) it follows that $\Delta l_k(d_k, \mu_{k+1}) = \Delta l_k(d'_k, \mu_k) \ge \epsilon v_k$. Lemma 3.20 then yields $\lim_{k\in K_2} v_k = 0$.) Now consider the start of iteration $k+1$ such that $k \in K_3 \cup K_4$. It follows from Taylor's theorem and the boundedness of $(y''_{k+1}, \bar{y}''_{k+1})$ due to (2.1) that for some $\{C_4, C_5\} \in (0,\infty)$ we have

$$\begin{aligned}
\|\rho(x_{k+1}, y''_{k+1}, \bar{y}''_{k+1}, 0)\| &= \left\| \begin{bmatrix} J_{k+1}y''_{k+1} + \bar{J}_{k+1}\bar{y}''_{k+1} \\ \min\{[c_{k+1}]^+, e - y''_{k+1}\} \\ \min\{[c_{k+1}]^-, e + y''_{k+1}\} \\ \min\{[\bar{c}_{k+1}]^+, e - \bar{y}''_{k+1}\} \\ \min\{[\bar{c}_{k+1}]^-, \quad \bar{y}''_{k+1}\} \end{bmatrix} \right\| \\
&\le \left\| \begin{bmatrix} J_k y''_{k+1} + \bar{J}_k \bar{y}''_{k+1} \\ \min\{[c_k]^+, e - y''_{k+1}\} \\ \min\{[c_k]^-, e + y''_{k+1}\} \\ \min\{[\bar{c}_k]^+, e - \bar{y}''_{k+1}\} \\ \min\{[\bar{c}_k]^-, \quad \bar{y}''_{k+1}\} \end{bmatrix} \right\| + C_4\|d_k\| \\
&= \left\| \begin{bmatrix} H''_k d''_k - H''_k d''_k + J_k y''_{k+1} + \bar{J}_k \bar{y}''_{k+1} \\ \min\{[c_k + J_k^T d''_k - J_k^T d''_k]^+, e - y''_{k+1}\} \\ \min\{[c_k + J_k^T d''_k - J_k^T d''_k]^-, e + y''_{k+1}\} \\ \min\{[\bar{c}_k + \bar{J}_k^T d''_k - \bar{J}_k^T d''_k]^+, e - \bar{y}''_{k+1}\} \\ \min\{[\bar{c}_k + \bar{J}_k^T d''_k - \bar{J}_k^T d''_k]^-, \quad \bar{y}''_{k+1}\} \end{bmatrix} \right\| + C_4\|d_k\| \\
&\le \|\rho_k(d''_k, y''_{k+1}, \bar{y}''_{k+1}, 0, H''_k)\| + C_4\|d_k\| + C_5\|d''_k\|.
\end{aligned} \tag{3.12}$$

It then follows from Termination Test 4 that for $k \in K_3 \cup K_4$ we have

$$\|\rho_k(d''_k, y''_{k+1}, \bar{y}''_{k+1}, 0, H''_k)\| \le \kappa\|\rho(x_k, y''_k, \bar{y}''_k, 0)\|, \tag{3.13}$$

which together with (3.12) yields

$$\|\rho(x_{k+1}, y''_{k+1}, \bar{y}''_{k+1}, 0)\| \le \kappa\|\rho(x_k, y''_k, \bar{y}''_k, 0)\| + C_4\|d_k\| + C_5\|d''_k\|. \tag{3.14}$$

(Note that the final multiplier update (2.36) can only decrease the left-hand side, so the above inequality holds both before and after this update is applied.) Finally, note that from (2.31), (2.35), and (2.36), we have at the beginning of iteration $k+1$ with $k \in K_1 \cup K_5 \cup K_6$ that

$$\|\rho(x_{k+1}, y''_{k+1}, \bar{y}''_{k+1}, 0)\| \le \|\rho(x_k, y''_k, \bar{y}''_k, 0)\|. \tag{3.15}$$

Define $r_k := \|\rho(x_k, y''_k, \bar{y}''_k, 0)\|$ for all $k \ge 0$, $e_k := v_k + C_3\|d_k\|$ for $k \in K_2$, $e_k := C_4\|d_k\| + C_5\|d''_k\|$ for $k \in K_3 \cup K_4$, $\bar{r}_k := 0$ for all $k \ge 0$, and $\mathcal{K} := K_2 \cup K_3 \cup K_4$. Since $d_k \to 0$ and $d''_k \to 0$ follow from Lemma 3.20 and $\lim_{k\in K_2} v_k = 0$, it follows from (3.11), (3.14), and (3.15) that Lemma 3.21 implies (3.5).

The result follows from the analyses of these two cases. $\square$

A similar result follows for the residual for the penalty problem.

LEMMA 3.23. *The following limit holds:*

$$\lim_{k\to\infty} \|\rho(x_k, y_k', \bar{y}_k', \mu_k)\| = 0. \tag{3.16}$$

Proof. We prove the result by considering two cases.

Case 1. Suppose that $k \in K_1 \cup K_6$ for all large k. In order to derive a contradiction, suppose that there exists an infinite $K \subseteq K_1 \cup K_6$ such that for some $C_1 > 0$ we have $\|\rho(x_k, y_k', \bar{y}_k', \mu_k)\| \geq C_1$ for all $k \in K$. Under Conditions 1, it follows that in fact $K \cap K_1 = \emptyset$, so under Conditions 6 we have for all $k \in K$ that

$$C_1 \leq \|\rho(x_k, y_k', \bar{y}_k', \mu_k)\| < \zeta\|\rho(x_k, y_k'', \bar{y}_k'', 0)\|.$$

However, this contradicts Lemma 3.22, which means that (3.16) must hold.

Case 2. Suppose $K_2 \cup K_3 \cup K_4 \cup K_5$ is infinite. From the definition of the residual ρ, we have for some $\{C_2, C_3, C_4\} \subset (0, \infty)$ that

$$\begin{aligned}
&\|\rho(x_{k+1}, y_{k+1}', \bar{y}_{k+1}', \mu_{k+1})\| \\
&= \left\| \begin{bmatrix} \mu_{k+1} g_{k+1} + J_{k+1} y_{k+1}' + \bar{J}_{k+1} \bar{y}_{k+1}' \\ \min\{[c_{k+1}]^+, e - y_{k+1}'\} \\ \min\{[c_{k+1}]^-, e + y_{k+1}'\} \\ \min\{[\bar{c}_{k+1}]^+, e - \bar{y}_{k+1}'\} \\ \min\{[\bar{c}_{k+1}]^-, \quad \bar{y}_{k+1}'\} \end{bmatrix} \right\| \\
&\leq \left\| \begin{bmatrix} \mu_{k+1} g_k + J_k y_{k+1}' + \bar{J}_k \bar{y}_{k+1}' \\ \min\{[c_k]^+, e - y_{k+1}'\} \\ \min\{[c_k]^-, e + y_{k+1}'\} \\ \min\{[\bar{c}_k]^+, e - \bar{y}_{k+1}'\} \\ \min\{[\bar{c}_k]^-, \quad \bar{y}_{k+1}'\} \end{bmatrix} \right\| + C_2\|d_k\| \\
&= \left\| \begin{bmatrix} \mu_{k+1} g_k + H_k' d_k' - H_k' d_k' + J_k y_{k+1}' + \bar{J}_k \bar{y}_{k+1}' \\ \min\{[c_k + J_k^T d_k' - J_k^T d_k']^+, e - y_{k+1}'\} \\ \min\{[c_k + J_k^T d_k' - J_k^T d_k']^-, e + y_{k+1}'\} \\ \min\{[\bar{c}_k + \bar{J}_k^T d_k' - \bar{J}_k^T d_k']^+, e - \bar{y}_{k+1}'\} \\ \min\{[\bar{c}_k + \bar{J}_k^T d_k' - \bar{J}_k^T d_k']^-, \quad \bar{y}_{k+1}'\} \end{bmatrix} \right\| + C_2\|d_k\| \\
&\leq \|\rho_k(d_k', y_{k+1}', \bar{y}_{k+1}', \mu_{k+1}, H_k')\| + C_2\|d_k\| + C_3\|d_k'\| \\
&\leq \|\rho_k(d_k', y_{k+1}', \bar{y}_{k+1}', \mu_k, H_k')\| + C_2\|d_k\| + C_3\|d_k'\| + C_4(\mu_k - \mu_{k+1})\|g_k\|,
\end{aligned} \tag{3.17}$$

where the first inequality follows from Taylor's theorem and the boundedness of $(y_{k+1}', \bar{y}_{k+1}')$ due to (2.1). For $k \in K_2 \cup K_3 \cup K_4$, Termination Test 1 and/or 2 holds, so we have from (2.17) and/or (2.20) that

$$\|\rho_k(d_k', y_{k+1}', \bar{y}_{k+1}', \mu_k, H_k')\| \leq \kappa\|\rho(x_k, y_k', \bar{y}_k', \mu_k)\| + \kappa\|\rho(x_k, y_k'', \bar{y}_k'', 0)\|. \tag{3.18}$$

Similarly, for $k \in K_5$, we have from Termination Test 3 (specifically, (2.25) and the update $d_k' \leftarrow 0$) that

$$\|\rho_k(d_k', y_{k+1}', \bar{y}_{k+1}', \mu_k, H_k')\| \leq \kappa\|\rho(x_k, y_k', \bar{y}_k', \mu_k)\|. \tag{3.19}$$

Lastly, for $k \in K_1 \cup K_6$, we have along with (2.26) that

$$\|\rho_k(d_k', y_{k+1}', \bar{y}_{k+1}', \mu_k, H_k')\| = \|\rho(x_k, y_k', \bar{y}_k', \mu_k)\| < \zeta\|\rho(x_k, y_k'', \bar{y}_k'', 0)\|.$$

By (3.17), (3.18), (3.19), the fact that $\rho(x_k, y''_k, \bar{y}''_k, 0) \to 0$ by Lemma 3.22, the facts that $d_k \to 0$ and $d'_k \to 0$ by Lemma 3.20, the fact that $(\mu_k - \mu_{k+1}) \to 0$ by the monotonicity and nonnegativity of $\{\mu_k\}$, and the fact that $\{g_k\}$ is bounded under Assumption 3.15, we find that with $r_k := \|\rho(x_k, y'_k, \bar{y}'_k, \mu_k)\|$, $e_k := \kappa\|\rho(x_k, y''_k, \bar{y}''_k, 0)\| + C_2\|d_k\| + C_3\|d'_k\| + C_4(\mu_k - \mu_{k+1})\|g_k\|$, $\bar{r}_k := \zeta\|\rho(x_k, y''_k, \bar{y}''_k, 0)\| + C_2\|d_k\| + C_3\|d'_k\| + C_4(\mu_k - \mu_{k+1})\|g_k\|$, and $\mathcal{K} := K_2 \cup K_3 \cup K_4 \cup K_5$, Lemma 3.21 yields (3.16).

The result follows from the analyses of these two cases. ∎

The next lemmas describe situations when the penalty parameter vanishes. A result similar to the first was also proved in [5].

LEMMA 3.24. *If $\mu_k \to 0$, then either all limit points of $\{x_k\}$ are feasible for* (NLP) *or all are infeasible for* (NLP).

Proof. In order to derive a contradiction, suppose that there exist infinite subsequences K_* and $K_\times$ such that $\{x_k\}_{k\in K_*} \to x_*$ with $v(x_*) = 0$ and $\{x_k\}_{k\in K_\times} \to x_\times$ with $v(x_\times) = C_1$ for some $C_1 > 0$. Since $\mu_k \to 0$, we have by the boundedness of $\{f(x_k)\}$ under Assumption 3.15 that there exists $k_* \geq 0$ such that for all $k \in K_*$ with $k \geq k_*$ we have $\mu_{k+1}(f(x_k) - \underline{f}) < C_1/4$ and $v(x_k) < C_1/4$, meaning that $\varphi(x_k, \mu_{k+1}) < C_1/2$. (Recall that $\underline{f}$ is the infimum of f over the smallest convex set containing $\{x_k\}$.) On the other hand, we also have $\mu_{k+1}(f(x_k) - \underline{f}) \geq 0$ for all $k \geq 0$ and that there exists $k_\times \geq 0$ such that for all $k \in K_\times$ with $k \geq k_\times$ we have $v(x_k) \geq C_1/2$, meaning that $\varphi(x_k, \mu_{k+1}) \geq C_1/2$. This is a contradiction since by Lemma 3.19 we have that $\{\varphi(x_k, \mu_{k+1})\}$ is monotonically decreasing. Thus, the set of limit points of $\{x_k\}$ cannot include points that are feasible for (NLP) and points that are infeasible. ∎

LEMMA 3.25. *If $\mu_k \to 0$ and all limit points of $\{x_k\}$ are feasible for* (NLP)*, then, with $K_\mu := \{k \mid \mu_{k+1} < \mu_k\}$, all limit points of $\{x_k\}_{k\in K_\mu}$ are FJ points.*

Proof. Let $K_* \subseteq K_\mu \subseteq K_1 \cup K_4 \cup K_6$ be an infinite subsequence such that $\{x_k\}_{k\in K_*} \to x_*$ for some limit point x_* of $\{x_k\}_{k\in K_\mu}$. We first show that the sequence $\{(y'_{k+1}, \bar{y}'_{k+1})\}_{k\in K_*}$ has a limit point $(y_*, \bar{y}_*) \neq 0$. We consider three cases.

Case 1. Suppose that $K_* \cap K_1$ is infinite. Since $\rho(x_k, y'_k, \bar{y}'_k, \mu_k) = 0$ and $v_k > 0$ for $k \in K_* \cap K_1$, it follows from the definition of $\rho(\cdot)$ that $\|(y'_{k+1}, \bar{y}'_{k+1})\|_\infty = 1$ for all such k, which in turn means that $\{(y'_{k+1}, \bar{y}'_{k+1})\}_{k\in K_*\cap K_1}$ has a limit point $(y_*, \bar{y}_*) \neq 0$.

Case 2. Suppose that $K_* \cap K_4$ is infinite. Then we claim that $\|(y'_{k+1}, \bar{y}'_{k+1})\|_\infty \geq \lambda(\epsilon - \beta) \in (0, 1)$ for all large $k \in K_* \cap K_4$. Indeed, in order to derive a contradiction, suppose that there exists an infinite subsequence $K \subseteq K_* \cap K_4$ such that for $k \in K$ we have $\|(y'_{k+1}, \bar{y}'_{k+1})\|_\infty < \lambda(\epsilon - \beta)$. Then, from (2.21) and (2.22), we have $\Delta l_k(d'_k, 0) \geq \epsilon v_k \geq \epsilon \Delta l_k(d''_k, 0)$, meaning that $\tau_k \leftarrow 1$, $d'_k \leftarrow d_k$, and $\Delta l_k(d_k, 0) \geq \epsilon v_k$. We also claim that $\Delta l_k(d_k, \mu_k) \geq \beta \Delta l_k(d_k, 0)$. (Otherwise, note that from (2.23), (2.24), and the inequality $\|(y'_{k+1}, \bar{y}'_{k+1})\|_\infty < \lambda(\epsilon - \beta)$, we have that

$$\Delta l_k(d_k, \mu_k) = \Delta l_k(d_k, 0) - \mu_k g_k^T d_k \geq \epsilon v_k - \|(y'_{k+1}, \bar{y}'_{k+1})\|_\infty v_k/\lambda \geq \beta \Delta l_k(d_k, 0),$$

which is a contradiction.) Consequently, it follows from (2.13) that $\mu_{k+1} \leftarrow \mu_k$, contradicting the fact that $k \in K_\mu$. Hence, $\|(y'_{k+1}, \bar{y}'_{k+1})\|_\infty \geq \lambda(\epsilon - \beta)$ for all $k \in K_*$. Since $K_* \cap K_4$ is infinite and $0 < \lambda(\epsilon - \beta) \leq \|(y'_{k+1}, \bar{y}'_{k+1})\|_\infty \leq 1$ for all large $k \in K_* \cap K_4$, it follows that $\{(y'_{k+1}, \bar{y}'_{k+1})\}_{k\in K_*\cap K_4}$ has a limit point $(y_*, \bar{y}_*) \neq 0$.

Case 3. Suppose that for all large $k \in K_*$, we have $k \in K_6$. Since, for all such k, Termination Test 3 holds and (2.26) is satisfied, it follows that $1 \geq \|(y'_{k+1}, \bar{y}'_{k+1})\|_\infty \geq \psi > 0$ for all such k. Hence, $\{(y'_{k+1}, \bar{y}'_{k+1})\}_{k\in K_*\cap K_6}$ has a limit point $(y_*, \bar{y}_*) \neq 0$.

Overall, we have shown that $\{(y'_{k+1}, \bar{y}'_{k+1})\}_{k\in K_*}$ has a limit point $(y_*, \bar{y}_*) \neq 0$.

Then, since $\rho(x_k, y'_{k+1}, \bar{y}'_{k+1}, \mu_k) \to 0$ by Lemma 3.23 with $\mu_k \to 0$, it follows by Assumption 3.15 that $(x_*, y_*, \bar{y}_*, 0)$ is an FJ point. The result follows since the limit point x_* of $\{x_k\}_{k \in K_\mu}$ was chosen arbitrarily. ◻

The following definitions are needed for our next lemma.

DEFINITION 3.26. *The set of active inequality constraints of* (NLP) *at* x *is*

$$\mathcal{A}(x) := \{i : \bar{c}^i(x) = 0\}.$$

DEFINITION 3.27. *A point* x *that is feasible for* (NLP) *satisfies the Mangasarian–Fromovitz constraint qualification (MFCQ) for* (NLP) *if* $J(x)$ *has full column rank and there exists* $d \in \mathbb{R}^n$ *such that*

$$c(x) + J(x)^T d = 0 \quad \textit{and} \quad \bar{c}(x) + \bar{J}(x)^T d < 0.$$

We now prove that at certain first-order stationary points, the MFCQ fails to hold.

LEMMA 3.28. *Suppose that* $\rho(x_*, y_*, \bar{y}_*, 0) = 0$, *where* x_* *is feasible for* (NLP) *and* $(y_*, \bar{y}_*) \neq 0$. *Then the MFCQ fails at* x_*.

Proof. Suppose that x_* is feasible for (NLP). Under the conditions of the lemma, it follows that $\bar{y}^i_* = 0$ for all $i \notin \mathcal{A}_* := \mathcal{A}(x_*)$, which implies that

$$0 = J_* y_* + \bar{J}_* \bar{y}_* = J_* y_* + \bar{J}_*^{\mathcal{A}_*} \bar{y}_*^{\mathcal{A}_*}, \tag{3.20}$$

where $J_* := J(x_*)$ and $\bar{J}_* := \bar{J}(x_*)$, and $\bar{J}_*^{\mathcal{A}_*}$ and $\bar{y}_*^{\mathcal{A}_*}$ denote the columns of $\bar{J}_*$ and entries of $\bar{y}_*$, respectively, corresponding to $\mathcal{A}_*$. In order to derive a contradiction to the result of the lemma, suppose that the MFCQ holds at x_* so that there exists d such that $d^T J_* = 0$ and $d^T \bar{J}_*^{\mathcal{A}_*} < 0$. It then follows from (3.20) that

$$0 = d^T J_* y_* + d^T \bar{J}_*^{\mathcal{A}_*} \bar{y}_*^{\mathcal{A}_*} = d^T \bar{J}_*^{\mathcal{A}_*} \bar{y}_*^{\mathcal{A}_*}, \tag{3.21}$$

and since $d^T \bar{J}_*^{\mathcal{A}_*} < 0$ and $\bar{y}_*^{\mathcal{A}_*} \geq 0$, we may conclude that $\bar{y}_*^{\mathcal{A}_*} = 0$. Thus, from (3.20) and the fact that under the MFCQ the columns of J_* are linearly independent, we have $y_* = 0$. Overall, we have shown that $(y_*, \bar{y}_*) = 0$, but that contradicts the condition of the lemma that $(y_*, \bar{y}_*) \neq 0$. Hence, the MFCQ must fail at x_*. ◻

We now state our main theorem of this section.

THEOREM 3.29. *One of the following holds:*

(a) $\mu_k = \underline{\mu}$ *for some* $\underline{\mu} > 0$ *for all large* k *and either every limit point* x_* *of* $\{x_k\}$ *corresponds to a KKT point or is an infeasible stationary point;*

(b) $\mu_k \to 0$ *and every limit point* x_* *of* $\{x_k\}$ *is an infeasible stationary point; or*

(c) $\mu_k \to 0$, *all limit points of* $\{x_k\}$ *are feasible for* (NLP), *and, with* $K_\mu := \{k : \mu_{k+1} < \mu_k\}$, *every limit point* x_* *of* $\{x_k\}_{k \in K_\mu}$ *corresponds to an FJ point at which the MFCQ fails.*

Proof. Since if $\mu_{k+1} < \mu_k$, then $\mu_{k+1} \leq \delta \mu_k$, it follows that either $\mu_k \to 0$ or $\mu_k = \underline{\mu}$ for some $\underline{\mu} > 0$ for all large k. If $\mu_k = \underline{\mu} > 0$ for all large k, then the fact that either every limit point of $\{x_k\}$ corresponds to a KKT point or every limit point is an infeasible stationary point follows from Lemmas 3.22 and 3.23. On the other hand, if $\mu_k \to 0$, then (b) or (c) occurs due to Lemmas 3.22, 3.23, 3.24, 3.25, and 3.28. ◻

We close our analysis with the following corollary of Theorem 3.29.

COROLLARY 3.30. *If* $\{x_k\}$ *is bounded and every limit point of this sequence is a feasible point at which the MFCQ holds, then* $\mu_k = \underline{\mu}$ *for some* $\underline{\mu} > 0$ *for all large* k *and every limit point of* $\{x_k\}$ *corresponds to a KKT point.*

Proof. Since every limit point of $\{x_k\}$ is feasible, only situation (a) or (c) in Theorem 3.29 could occur. Suppose that situation (c) holds. Then $\mu_k \to 0$, i.e., K_μ is infinite, and since $\{x_k\}$ is bounded, this implies that $\{x_k\}_{k\in K_\mu}$ must have limit points. By the conditions of situation (c) in Theorem 3.29, this leads to a contradiction of the supposition that the MFCQ holds at all (feasible) limit points. Thus, only situation (a) can occur, and the result follows. ∎

4. Numerical experiments. In this section, we describe a preliminary implementation of Algorithm 2 and show, when it is employed to solve a set of standard test problems, that the use of inexact subproblem solutions does not substantially degrade the reliability or performance (in terms of iterations required to find a stationary point) of the algorithm. These results suggest that with the reduced per-iteration computational costs that may be gained by allowing inexactness in the subproblem solutions, there may be an overall reduction in computational costs by employing our strategies in a more sophisticated implementation of our algorithm.

4.1. Implementation and experimental setup. An implementation, hereafter referred to as **iSQO**, was created in MATLAB. A critical component of the implementation is the employed QP solver. Algorithm 1, which describes our procedure for computing inexact QP solutions that are used in Algorithm 2, postulates that the QP (2.37) is handled by an iterative QP solver that can provide a sequence of solutions for (2.37) such that Assumptions 3.2 and 3.3 are satisfied. In particular, it postulates that the employed QP solver is able to provide as accurate a solution to (2.37) as required by our termination tests while reliably handling potential nonconvexity of the QP. Since there does not currently exist a QP solver that satisfies these requirements and provides an interface to readily incorporate our termination tests (so that the computed solutions are not *more* accurate than necessary), for the purposes of our preliminary experiments we have generated inexact solutions by perturbing the exact solutions provided by a state-of-the-art nonconvex QP solver. To this end, we used **bqpd** [19], an implementation of a reliable primal active-set QP solver capable of handling indefinite Hessian matrices. (Our technique for perturbing the exact solutions obtained from **bqpd** is described in detail below.)

At the beginning of the optimization routine in **iSQO**, the objective and constraint functions of the problem statement were scaled by the strategy described in [44, section 3.8]. This helped avoid numerical difficulties due to badly scaled problem formulations. We used the bisection routine outlined in [5] to determine the convex combination value τ_k satisfying (2.12). Our implemented algorithm terminated when any of the following counterparts of (2.15) held for given $\{\epsilon_{tol}, \epsilon_\mu\} \subset (0,\infty)$:

$$\|\rho(x_k, y_k', \bar{y}_k', \mu_k)\| \le \epsilon_{tol} \quad \text{and} \quad v_k \le \epsilon_{tol}; \tag{4.1a}$$

$$\|\rho(x_k, y_k'', \bar{y}_k'', 0)\| = 0 \quad \text{and} \quad v_k > 0; \tag{4.1b}$$

$$\|\rho(x_k, y_k'', \bar{y}_k'', 0)\| \le \epsilon_{tol} \quad \text{and} \quad v_k > \epsilon_{tol} \quad \text{and} \quad \mu_k \le \epsilon_\mu. \tag{4.1c}$$

Equation (4.1a) (resp., (4.1b) and (4.1c)) corresponds to an "optimal solution found" (resp., "infeasible stationary point found") exit status. Observe that we could have chosen to terminate **iSQO** with the "infeasible stationary point found" status if the first two conditions in (4.1c) held, regardless of the value for μ_k. However, in our experience, this would increase the likelihood that the code would terminate with this status, even though it may eventually terminate with the (more desirable) "optimal solution found" status if it were allowed to continue. Hence, we added to (4.1c) the restriction that $\mu_k \leq \epsilon_\mu$, allowing the algorithm to continue to search for an optimal solution unless the penalty parameter had been reduced to a relatively small value.

Table 1 specifies the values of all user-defined constants defined in (2.4), as well as the tolerances described above for finite termination (i.e., ϵ_{tol} and ϵ_μ) and the maximum number of iterations, denoted by $k_{\max}$. Note that in the experiments described in the following subsections, we consider the stated various values of κ.

Table 1
Parameter values used in our **iSQO** *algorithm implementation.*

Constant	Value	Constant	Value	Constant	Value
κ	0.01, 0.1, 0.5	ζ	10	λ, ψ, ϵ	0.1
δ	0.2	τ	10^{-3}	β	0.01
ξ	cf. Alg. IC [44]	γ	0.5	η, θ, ϵ_μ	10^{-8}
ϵ_{tol}	10^{-6}	$k_{\max}$	1000		

As mentioned above, for our purposes in presenting the results of a preliminary implementation of our approach, we use **bqpd** to solve each subproblem exactly and then induce inexactness for (2.37) via a random perturbation of each solution. In particular, in order to induce inexactness in the subproblem solutions for both (PQP) and (FQP), we took the following approach: For a given set of termination tests, we perturbed an exact subproblem solution $(d^*, y^*, \bar{y}^*)$, using uniform random vectors $u_\ell \in \mathcal{U}(-1,1)^\ell$, where $\ell \in \{n, m, \bar{m}\}$, by finding the first element in the sequence $j = 0, 1, 2, \ldots$ such that

$$d := d^* + 0.5^j u_n, \qquad y := y^* + 0.5^j u_m, \qquad \text{and} \qquad \bar{y} := \bar{y}^* + 0.5^j u_{\bar{m}} \tag{4.2}$$

satisfied at least one of the given tests. Using $(d^*, y^*, \bar{y}^*)$ without such a perturbation yields a variant of Algorithm 2 with exact subproblem solutions, results for which we present as a means of comparison with our **iSQO** routine.

The test suite employed in our experiments comprises all 309 CUTE [23] problems in AMPL [21] format provided in [1] such that (a) the problem includes at least one free variable and one general (nonbound) constraint, (b) the numbers of variables and constraints sum to not more than 20,000, and (c) **bqpd** did not fail during the run of any of the four algorithm variants described in the following subsection.[1] We disabled the AMPL presolve feature to maintain the idiosyncrasies of each formulation.

4.2. Numerical results. Table 2 compares exit status counts from **iSQO** when using exact and inexact subproblem solutions. Here, it is evident that the use of inexact subproblem solutions did not have a significant impact on the (approximately 96%) success rate, i.e., the percentage of problems that yield an "optimal solution found" or "infeasible stationary point found" exit status.

Next, we assess the level of inaccuracy of the subproblem solutions computed in **iSQO** in order to illustrate that relatively inexact solutions are indeed employed in the algorithm. Given an iterate satisfying (2.1) and a given subproblem solution, we calculated the *residual ratio* as

$$\kappa_I := \frac{\|\rho_k(d, y, \bar{y}, \mu_k, H'_k)\|}{\|\rho(x_k, y'_k, \bar{y}'_k, \mu_k)\|} \quad \text{or} \quad \kappa_I := \frac{\|\rho_k(d, y, \bar{y}, 0, H''_k)\|}{\|\rho(x_k, y''_k, \bar{y}''_k, 0)\|}, \tag{4.3}$$

[1]The only exceptions were the problems `dallasm` and `dallass`, which were excluded as AMPL function evaluation errors were encountered. Overall, our test set includes over 59 problems whose sum of the numbers of variables and constraints is over 100, where, of those, 19 problems have a sum of the numbers of variables and constraints that is over 1,000.

TABLE 2
iSQO *exit status counts when using exact and inexact subproblem solutions.*

	Exact	Inexact		
		$\kappa = 0.01$	$\kappa = 0.1$	$\kappa = 0.5$
Optimal solution found	289	291	293	293
Infeasible stationary point found	8	7	6	6
Iteration limit reached	12	11	10	10

for the penalty or feasibility subproblems, respectively. A small κ_I value indicates a very accurate solve. (Here, for comparison purposes, we remark that the exact solutions returned from **bqpd** typically yielded κ_I values on the order of 10^{-16}.)

Given the jth instance in our set, we denote by $\kappa_I(j)$ the minimum of all κ_I values observed for a subproblem solution during the execution of Algorithm 2. Note that each iteration included as many as two κ_I values, one for the penalty subproblem and one for the feasibility subproblem. We also use $\bar{\kappa}_I(j)$ to denote the geometric average of these values for the jth instance. Table 3 lists the number of NLPs for which $\kappa_I(j)$ and $\bar{\kappa}_I(j)$ fall into different intervals (excluding a few problems that were solved at the initial point once an improved dual solution was computed). We also include the geometric averages $\kappa_{I,\text{mean}}$ and $\bar{\kappa}_{I,\text{mean}}$ of $\kappa_I(j)$ and $\bar{\kappa}_I(j)$, respectively, as a cumulative measure of these values when one considers the entire test set.

It is evident from Table 3 that the termination tests permitted a substantial amount of inexactness in the subproblem solutions. In particular, the distribution of $\kappa_I(j)$ shows that for a majority of the problems, $\kappa_I(j)$ was within two orders of magnitude of κ. The average behavior is even more encouraging as it shows that typical κ_I values were within one order of magnitude of κ in all but one case. We also observe that $\bar{\kappa}_I(j) \geq \kappa$ for a minority of the problems, indicating the acceptance of inexact subproblem solutions that yield relatively large residuals. Together, these observations indicate that the accepted inexact subproblem solutions were significantly different from the exact subproblem solutions.

TABLE 3
Comparison of number of successfully solved NLPs with $\kappa_I(j)$ or $\bar{\kappa}_I(j)$ in the specified range.

min	κ	$\bar{\kappa}_{I,\text{min}}$	$[0-10^{-8})$	$[10^{-8}-10^{-6})$	$[10^{-6}-10^{-4})$	$[10^{-4}-10^{-3})$	$[10^{-3}-0.01)$	$[0.01-0.1)$	$[0.1-0.5)$	$[0.5-1)$	$[1-\infty)$
$\kappa_I(j)$	0.01	`3.8e-03`	0	1	9	6	277	1	0	0	0
	0.1	`3.0e-02`	0	0	1	10	30	254	0	0	0
	0.5	`9.0e-02`	0	0	2	4	24	77	188	0	0
mean	κ	$\bar{\kappa}_{I,\text{mean}}$									
$\bar{\kappa}_I(j)$	0.01	`7.4e-03`	0	0	0	0	270	24	0	0	0
	0.1	`7.0e-02`	0	0	0	0	0	280	15	0	0
	0.5	`3.5e-01`	0	0	0	0	0	1	282	12	0

We now show that the use of inexact subproblem solutions did not lead to a substantial increase in iterations in **iSQO**. Following [39], we compare the iteration counts of solvers A and B on problem j with the logarithmic *outperforming factor*

$$r_{AB}^j := -\log_2(\text{iter}_A^j/\text{iter}_B^j). \tag{4.4}$$

For example, the value $r^j_{AB} = 3$ means that solver A required $\frac{1}{2^3}$ of the iterations required by solver B. Figure 1 shows r^j_{AB} with A (resp., B) representing the inexact algorithm with $\kappa = 0.01$ (resp., exact algorithm) for all instances successfully solved by both solvers with more than three iterations. It is not surprising that exact subproblem solutions often led to fewer iterations, but it is encouraging that for all but 13 problems, the number of iterations were within a factor of two, and typically the factors were much smaller. In fact, for almost 100 problems, the numbers of iterations were exactly the same, and there are also a few instances where (surprisingly) the inexact algorithm required many fewer iterations. We also remark that by measuring problem size by the sum of the numbers of variables and constraints, the outperforming factors appear independent from problem size; see the plot on the left-hand side of Figure 1. This suggests that inexact subproblem solutions—if controlled appropriately, such as by the termination tests in our algorithm—might not lead to a significant increase in iteration counts if one were to consider even larger problems.

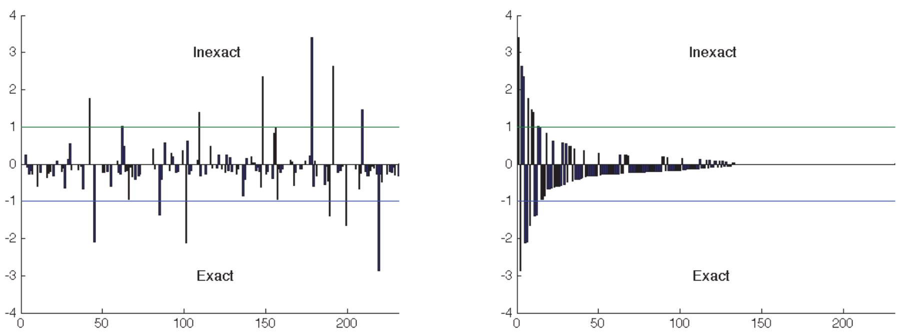

FIG. 1. *Performance of* **iSQO** *with inexact and exact QP solutions measured by* r^j_{AB}. *The horizontal lines at* -1 *and* 1 *indicate a difference in iterations by a factor of* 2, *and the direction of each vertical bar indicates whether the algorithm with inexact (up) or exact (down) QP solutions required fewer iterations. On the left, the instances are ordered by increasing sum of numbers of variables and constraints, and on the right the instances are ordered by decreasing* $|r^j_{AB}|$.

In summary, our numerical experiments demonstrate that our proposed algorithm exhibits a promising level of reliability (in terms of successful terminations) and performance (in terms of iteration counts), and that these results can be obtained without accurate subproblem solutions.

5. Conclusion. In this paper, we have proposed an inexact sequential quadratic optimization (iSQO) method for solving nonlinear constrained optimization problems. The novel feature of the algorithm is a set of loose conditions that the primal-dual search directions must satisfy, which allow for the use of inexact subproblem solutions obtained via any QP solver that satisfies a mild set of assumptions. We have proved that the algorithm is well-posed in that some amount of inexactness is allowed whenever the QP solver is initiated. We have also proved that the algorithm is globally convergent to the set of first-order stationary points of the nonlinear optimization problem (NLP), or at least that of the corresponding feasibility problem (FP). In particular, if the algorithm avoids infeasible stationary points and all (feasible) limit points satisfy the MFCQ, then we have shown that all limit points of the algorithm are KKT points for (NLP). Our numerical experiments illustrate that the algorithm is as reliable as an algorithm that computes exact QP solutions during every iteration, often at the expense of only a modest number of additional iterations. These results suggest that with the computational benefits that may be gained by terminating a QP solver early, the algorithm can offer an overall reduction in computational costs compared to an algorithm that employs exact QP solutions.

One class of problems for which the proposed computational benefits of our algorithm may be realized is that of real-time optimal control problems where approximate solutions of each "new" QP may be computed by repeated hot-starts of an "older" QP [34]. In such a setting, the size of problem (NLP) may be moderate, but savings in computational costs can be achieved from the fact that vector products with the constraint Jacobian are significantly cheaper than computing the entire Jacobian when the constraints involve the integration of differential equations. We also anticipate that the approach in this paper can be beneficial for large-scale problems for which QP solvers that can generate relatively good inexact solutions quickly can be employed in a more computationally efficient implementation of our algorithm.

Acknowledgments. The authors would like to thank the associate editor and the two anonymous referees, whose comments and suggestions helped to improve the presentation of the paper.

REFERENCES

[1] H. Y. Benson, *Cute Models*, http://www.orfe.princeton.edu/~rvdb/ampl/nlmodels/cute/index.html.

[2] H. G. Bock, M. Diehl, P. Kühl, E. Kostina, J. P. Schiöder, and L. Wirsching, *Numerical methods for efficient and fast nonlinear model predictive control*, in Assessment and Future Directions of Nonlinear Model Predictive Control, R. Findeisen, F. Allgöwer, and L. T. Biegler, eds., Lecture Notes in Control and Inform. Sci. 358, Springer, Berlin, Heidelberg, 2007, pp. 163–179.

[3] J. V. Burke, *Descent methods for composite nondifferentiable optimization problems*, Math. Programming, 33 (1985), pp. 260–279.

[4] J. V. Burke, *Second order necessary and sufficient conditions for convex composite NDO*, Math. Programming, 38 (1987), pp. 287–302.

[5] J. V. Burke, F. E. Curtis, and H. Wang, *A Sequential Quadratic Optimization Algorithm with Fast Infeasibility Detection*, Tech. Report 12T-012, COR@L Laboratory, Department of ISE, Lehigh University, 2012.

[6] J. V. Burke and M. C. Ferris, *A Gauss-Newton method for convex composite optimization*, Math. Programming, 71 (1995), pp. 179–194.

[7] R. H. Byrd, F. E. Curtis, and J. Nocedal, *An inexact SQP method for equality constrained optimization*, SIAM J. Optim., 19 (2008), pp. 351–369.

[8] R. H. Byrd, F. E. Curtis, and J. Nocedal, *An inexact Newton method for nonconvex equality constrained optimization*, Math. Program., 122 (2010), pp. 273–299.

[9] R. H. Byrd, F. E. Curtis, and J. Nocedal, *Infeasibility detection and SQP methods for nonlinear optimization*, SIAM J. Optim., 20 (2010), pp. 2281–2299.

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

[11] R. H. Byrd, G. López-Calva, and J. Nocedal, *A line search exact penalty method using steering rules*, Math. Program., 133 (2012), pp. 39–73

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

[13] F. E. Curtis, J. Huber, O. Schenk, and A. Wächter, *A note on the implementation of an interior-point algorithm for nonlinear optimization with inexact step computations*, Math. Program., 136 (2012), pp. 209–227.

[14] F. E. Curtis, J. Nocedal, and A. Wächter, *A matrix-free algorithm for equality constrained optimization problems with rank deficient Jacobians*, SIAM J. Optim., 20 (2009), pp. 1224–1249.

[15] F. E. Curtis, O. Schenk, and A. Wächter, *An interior-point algorithm for large-scale nonlinear optimization with inexact step computations*, SIAM J. Sci. Comput., 32 (2010), pp. 3447–3475.

[16] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.
[17] M. Diehl, A. Walther, H. G. Bock, and E. Kostina, *An adjoint-based SQP algorithm with quasi-Newton Jacobian updates for inequality constrained optimization*, Optim. Methods Softw., 25 (2010), pp. 531–552.
[18] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., Wiley-Interscience, New York, 1987.
[19] R. Fletcher, *Stable reduced Hessian updates for indefinite quadratic programming*, Math. Program., 87 (2000), pp. 251–264.
[20] R. Fletcher and E. Sainz de la Maza, *Nonlinear programming and nonsmooth optimization by successive linear programming*, Math. Programming, 43 (1989), pp. 235–256.
[21] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, 2nd ed., Brooks/Cole, Pacific Grove, CA, 2002.
[22] N. I. M. Gould, Y. Loh, and D. P. Robinson, *A filter method with unified step computation for nonlinear optimization*, SIAM J. Optim., 24 (2014), pp. 175–209
[23] N. I. M. Gould, D. Orban, and Ph. L. Toint, *Cuter and SIFDEC: A constrained and unconstrained testing environment, revisited*, ACM Trans. Math. Software, 29 (2003), pp. 373–394.
[24] N. I. M. Gould and D. P. Robinson, *A second derivative SQP method: Global convergence*, SIAM J. Optim., 20 (2010), pp. 2023–2048.
[25] N. I. M. Gould and D. P. Robinson, *A second derivative SQP method: Local convergence and practical issues*, SIAM J. Optim., 20 (2010), pp. 2049–2079.
[26] N. I. M. Gould and D. P. Robinson, *A second-derivative SQP method with a "trust-region-free" predictor step*, IMA J. Numer. Anal., 32 (2012), pp. 580–601.
[27] S. P. Han, *A globally convergent method for nonlinear programming*, J. Optim. Theory Appl., 22 (1977), pp. 297–309.
[28] S. P. Han and O. L. Mangasarian, *Exact penalty functions in nonlinear programming*, Math. Programming, 17 (1979), pp. 251–269.
[29] M. Heinkenschloss and L. N. Vicente, *Analysis of inexact trust-region SQP algorithms*, SIAM J. Optim., 12 (2001), pp. 283–302.
[30] B. Houska and M. Diehl, *A quadratically convergent inexact SQP method for optimal control of differential algebraic equations*, Optimal Control Appl. Methods, 34 (2013), pp. 396–414.
[31] A. F. Izmailov and M. V. Solodov, *A truncated SQP method based on inexact interior-point solutions of subproblems*, SIAM J. Optim., 20 (2010), pp. 2584–2613.
[32] H. Jäger and E. W. Sachs, *Global convergence of inexact reduced SQP methods*, Optim. Methods Softw., 7 (1997), pp. 83–110.
[33] F. John, *Extremum problems with inequalities as side conditions*, in Studies and Essays, Courant Anniversary Volume, K. O. Friedrichs, O. E. Neugebauer, and J. J. Stoker, eds., Wiley (Interscience), New York, 1948, pp. 187–204.
[34] T. C. Johnson, C. Kirches, and A Wächter, *An Active-Set Quadratic Programming Method Based on Sequential Hot-Starts*, Tech. report, 2013; available online from http://www.optimization-online.org/DB_HTML/2013/10/4084.html.
[35] W. Karush, *Minima of Functions of Several Variables with Inequalities as Side Constraints*, Master's thesis, Department of Mathematics, University of Chicago, Chicago, IL, 1939.
[36] H. W. Kuhn and A. W. Tucker, *Nonlinear programming*, in Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, J. Neyman, ed., University of California Press, Berkeley, CA, 1951, pp. 481–492.
[37] F. Leibfritz and E. W. Sachs, *Inexact SQP interior point methods and large scale optimal control problems*, SIAM J. Control Optim., 38 (1999), pp. 272–293.
[38] O. L. Mangasarian and S. Fromovitz, *The Fritz John necessary optimality conditions in the presence of equality and inequality constraints*, J. Math. Anal. Appl., 17 (1967), pp. 37–47.
[39] J. L. Morales, *A numerical study of limited memory BFGS methods*, Appl. Math. Lett., 15 (2002), pp. 481–487.
[40] J. L. Morales, J. Nocedal, and Y. Wu, *A sequential quadratic programming algorithm with an additional equality constrained phase*, IMA J. Numer. Anal., 32 (2012), pp. 553–579.
[41] W. Murray and F. J. Prieto, *A sequential quadratic programming algorithm using an incomplete solution of the subproblem*, SIAM J. Optim., 5 (1995), pp. 590–640.
[42] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Classics Appl. Math. 30, SIAM, Philadelphia, PA, 2000.
[43] M. J. D. Powell, *A fast algorithm for nonlinearly constrained optimization calculations*, in Numerical Analysis, Lecture Notes in Math. 630, Springer, Berlin, Heidelberg, 1978, pp. 144–157.
[44] A. Wächter and L. T. Biegler, *On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming*, Math. Program., 106 (2006), pp. 25–57.

[45] A. Walther, *A first-order convergence analysis of trust-region methods with inexact Jacobians*, SIAM J. Optim., 19 (2008), pp. 307–325.

[46] A. Walther, S. R. R. Vetukuri, and L. T. Biegler, *A first-order convergence analysis of trust-region methods with inexact Jacobians and inequality constraints*, Optim. Methods Softw., 27 (2012), pp. 373–389.

[47] R. B. Wilson, *A Simplicial Algorithm for Concave Programming*, Ph.D. thesis, Graduate School of Business Administration, Harvard University, Cambridge, MA, 1963.