ON THE CONVERGENCE OF SUCCESSIVE LINEAR-QUADRATIC PROGRAMMING ALGORITHMS*

RICHARD H. BYRD[†], NICHOLAS I. M. GOULD[‡], JORGE NOCEDAL[§], AND RICHARD A. WALTZ[§]

Abstract. The global convergence properties of a class of penalty methods for nonlinear programming are analyzed. These methods include successive linear programming approaches and, more specifically, the successive linear-quadratic programming approach presented by Byrd et al. [*Math. Program.*, 100 (2004), pp. 27–48]. Every iteration requires the solution of two trust-region subproblems involving piecewise linear and quadratic models, respectively. It is shown that, for a fixed penalty parameter, the sequence of iterates approaches stationarity of the penalty function. A procedure for dynamically adjusting the penalty parameter is described, and global convergence results for it are established.

1. Introduction. In this paper we study the global convergence properties of successive linear-quadratic programming (SLQP) algorithms for nonlinear programming. The problem under consideration is

$$\text{(1.1a)} \qquad \underset{x}{\text{minimize}} \ f(x)$$

$$\text{(1.1b)} \qquad \text{subject to} \ h(x) = 0$$

$$\text{(1.1c)} \qquad g(x) \geq 0,$$

where the objective function $f : \mathbb{R}^n \to \mathbb{R}$, and the constraint functions $h : \mathbb{R}^n \to \mathbb{R}^{m_h}$, $g : \mathbb{R}^n \to \mathbb{R}^{m_g}$, are assumed to be continuously differentiable.

The class of algorithms studied in this paper solves (1.1) via the related problem

$$\text{(1.2)} \qquad \underset{x}{\text{minimize}} \quad \phi_\sigma(x),$$

where

$$\text{(1.3)} \qquad \phi_\sigma(x) = f(x) + \sigma\|h(x)\| + \sigma\|g^-(x)\|$$

is an exact penalty function [5, 12] composed of the objective and constraint functions from (1.1). Here $\|\cdot\|$ is a polyhedral norm, $g^-(x)$ is defined componentwise as

$$g_i^-(x) = \min(g_i(x), 0),$$

and $\sigma > 0$ is a parameter that is adaptively chosen so that critical points of (1.1) correspond to those of (1.2). For fixed σ, each iteration of a typical algorithm comprises two phases. In the first (linear) phase, a piecewise linear model of the penalty function ϕ_σ is minimized subject to a trust-region bound. The aim here is to compute a step for which convergence can be guaranteed. The second (quadratic) phase adjusts this step by reducing a quadratic model of the penalty function within a (second) trust-region bound, with the aim of accelerating the convergence of the method. A primary purpose of this article is to establish the global convergence of this class of methods. Once this has been established, it remains to consider methods for adjusting the penalty parameter so as to ensure convergence of the overall algorithm to KKT points for (1.1) or, failing this, critical points of some measure of constraint infeasibility.

This work is motivated by a recently proposed algorithm, described by the authors in [1], and is related to the SLQP algorithm proposed by Fletcher and Sainz de la Maza [8]. In [1] the ℓ_1-norm is used to define the penalty function (1.3). The linear phase utilizes a piecewise linear model of (1.3) at the current iterate x_k,

$$\ell(x_k, d) = f(x_k) + \nabla f(x_k)^T d + \sigma \|h(x_k) + \nabla h(x_k)^T d\| + \sigma \|(g(x_k) + \nabla g(x_k)^T d)^-\|.$$
(1.4)

Defining $\ell_k(d) \stackrel{\text{def}}{=} \ell(x_k, d)$ and imposing an ℓ_∞-norm trust region whose radius is given by the scalar parameter $\Delta_k^{\text{LP}} > 0$, the linear phase consists of solving the (piecewise) linear program (LP)

$$\underset{d}{\text{minimize}} \ \ell_k(d)$$
$$\text{subject to} \ \|d\|_\infty \leq \Delta_k^{\text{LP}},$$

whose solution we denote by d_k^{LP}. A working set $\mathcal{W}_k$ is subsequently defined as the set of constraints that are active at the solution of this problem if these constraints are linearly independent, or otherwise some linearly independent subset of these.

The quadratic phase of the algorithm described in [1] computes a step d_k that makes progress on a piecewise quadratic function

$$(1.5) \qquad\qquad q_k(d) = \ell_k(d) + \tfrac{1}{2} d^T B_k d,$$

subject to a trust-region constraint, where B_k approximates the Hessian of the Lagrangian of the nonlinear program (1.1). The step computation in the quadratic phase is carried out by solving an equality-constrained quadratic programming problem of the form

$$(1.6a) \qquad\qquad \underset{d}{\text{minimize}} \ \tfrac{1}{2} d^T B_k d + (\nabla \phi_\sigma)_k^T d$$

$$(1.6b) \qquad \text{subject to} \ h_i(x_k) + \nabla h_i(x_k)^T d = 0, \quad i \in \mathcal{E} \cap \mathcal{W}_k,$$

$$(1.6c) \qquad\qquad\qquad g_i(x_k) + \nabla g_i(x_k)^T d = 0, \quad i \in \mathcal{I} \cap \mathcal{W}_k,$$

$$(1.6d) \qquad\qquad\qquad \|d\|_2 \leq \Delta_k,$$

where $(\nabla \phi_\sigma)_k$ is the gradient of the part of (1.3) corresponding to the objective function and the violated constraints, and $\mathcal{E}$ and $\mathcal{I}$ denote the sets of equality and inequality constraints, respectively. Notice that in this phase, an ℓ_2-norm trust region is used, and the trust-region parameter Δ_k is distinct from the trust-region parameter Δ_k^{LP} used in the linear phase. The overall step taken by the algorithm is obtained by minimizing q_k along a path formed by d_k^{LP} and d_k, in a manner described in [1].

Our algorithm [1] is distinct from the one proposed by Fletcher and Sainz de la Maza [8] in two important ways. First, the trial step generated by our algorithm is formed from a convex combination of the linear phase step d_k^{LP} and the quadratic phase step d_k, whereas *either* the step d_k or the step d_k^{LP} is taken in [8]. Second, our algorithm imposes a trust-region restriction on the second subproblem, and thus permits the use of second derivatives of the objective function and constraints in the definition of B. The two trust-region radii operate quasi-independently, and the update rules used in [1] will be shown in this paper to offer global convergence guarantees.

The organization of the paper is as follows. In the remainder of this section we discuss the application of SLQP methods to general composite nonsmooth problems and briefly review existing SLQP methods. In section 2 we present an algorithm for the minimization of the penalty function with fixed penalty parameter. We study the global convergence properties of such an algorithm in section 3. Procedures for updating the penalty parameter are studied in section 4. The paper concludes with some final remarks and perspectives.

1.1. The general composite nonsmooth context. It is worth pointing out that problem (1.2) is a nonsmooth problem that is a special case of the more general class of *composite nonsmooth* optimization problems that can be represented as

$$(1.7) \qquad \underset{x}{\text{minimize }} \omega(F(x))$$

for some smooth function $F(x)$ and convex ω. Problem (1.2) has this form if we let

$$(1.8) \qquad F(x) = (f(x), g(x), h(x))$$

and define

$$(1.9) \qquad \omega(F(x)) = f(x) + \sigma\|h(x)\| + \sigma\|g^-(x)\|.$$

Many nondifferentiable approximation problems may also be put in this form.

In this context, the linearized model $\ell(x_k, d)$ in (1.4) corresponds to

$$(1.10) \qquad \omega\left(F(x_k) + F'(x_k)d\right).$$

The strategy described above corresponds to minimizing (1.10) at the current iterate x_k, subject to $\|d\|_\infty \leq \Delta_k^{\mathrm{LP}}$, and using the result to help compute a step making progress on the function

$$(1.11) \qquad \ell_k(d) + \tfrac{1}{2}d^T B_k d.$$

The algorithm described in section 3 applies equivalently to problem (1.7), as does the convergence analysis in section 4.

1.2. Existing SLQP algorithms. To the best of our knowledge, the earliest SLQP method was proposed by Fletcher and Sainz de la Maza [8], based on ideas in [6, 13]. The method is described in terms of general composite nonsmooth optimization problems of the form (1.7). At the iterate x_k, a linearized approximation of the form (1.10) is minimized within a given trust region. A solution to this problem, d_k^{LP}, is then used to assess the suitability of a trial step d_k, obtained without regard to the trust region and by whatever means is appropriate. If a finite number of different attempts to find a suitable d_k have failed, the choice $d_k = d_k^{\mathrm{LP}}$ is tried, and if this too fails, x_{k+1} is left at x_k and the trust-region radius reduced. Fletcher and Sainz de la Maza suggest using the subdifferential structure of ω predicted by $\ell_k(d_k^{\mathrm{LP}})$ as one means of finding d_k. Specifically, the minimizer of the (locally) smooth part of the quadratic model q_k is minimized subject to the linearized (locally) nonsmooth part being unchanged. This "equality-constrained" quadratic program (EQP) is invariably a far simpler problem than trying to minimize q_k. Importantly, Fletcher and Sainz de la Maza show that, under reasonable nondegeneracy and second-order conditions, the "active" subdifferential structure of $\ell_k(d_k^{\mathrm{LP}})$ ultimately predicts that of $\omega(F)$ at limit points of $\{x_k\}$, and thus that the EQP leads to fast asymptotic convergence.

A more recent SLQP method due to Chin [2] and Chin and Fletcher [3] is aimed specifically at the nonlinear programming problem (1.1). Rather than using the nonsmooth penalty function (1.3) to force convergence, Chin and Fletcher use a nonlinear programming "filter" [7] to do so. A succession of steps are allowed at each iteration, in which unbounded quadratic programming steps of various forms are given precedence over linear programming ones. Nevertheless, as with the methods in [1] and [8], the linear programming subproblem

$$
\begin{aligned}
\underset{d}{\text{minimize}} \quad & d^T \nabla f(x_k) \\
\text{subject to} \quad & h(x_k) + \nabla h(x_k)^T d = 0, \\
& g(x_k) + \nabla g(x_k)^T d \geq 0, \\
& \|d\|_\infty \leq \Delta_k
\end{aligned}
$$

(1.12)

is central and drives the convergence of the method. In particular, if d_k^{LP} is a solution[1] of (1.12), and if more complicated steps are unacceptable for the filter, the method reverts to a Cauchy step along d_k^{LP}. The trust-region radius will only be reduced as a last resort.

While this is undoubtedly an SLQP method, it is once again a trust region on the linear programming component that is used to force convergence. There appears to be no control of any quadratic programming component, and thus no precaution to guard against large or unbounded QP steps.

Most recently, Waltz [14] and Gate [9] suggested the idea of using a second trust region to control the EQP phase of SLQP methods. Waltz's method forms the basis of that described in [1] and analyzed here. Gate's method is an extension of the Chin–Fletcher filter approach, and although there is no formal analysis, this method appears to perform well in his numerical tests.

It should be noted that the theory of nonsmooth optimization developed by Yuan [15, 16] cannot be applied to the algorithm considered here and in [1, 14] because in these algorithms the two trust regions influence each other, whereas Yuan assumes that a single trust region is used. The analysis presented here is significantly different from that in the literature due to the effects caused by the interactions between the

―――――――――

[1]If (1.12) has no solution, a "restoration" phase [3] is entered.

two trust regions. In addition, we establish new results about update procedures for the penalty parameter.

2. An SLQP algorithm. Our first goal is to propose and analyze an algorithm for minimizing the penalty function ϕ_σ, given by (1.3), for a fixed value of σ. Notice that this analysis presupposes that the penalty parameter σ has been fixed at a sufficiently large value such that critical points of (1.1) correspond to those of (1.2), but we will delay a discussions of suitable mechanisms to ensure that this is so until section 4.

As noted earlier, the algorithm consists of two phases based, respectively, on piecewise linear and piecewise quadratic models at the current estimate x_k of the minimizer. The first phase minimizes the piecewise linear model $\ell_k(d)$, given by (1.4). The second phase is based on an appropriate piecewise quadratic model $q_k(d)$ of the form (1.5) that includes a second-order term to account for curvature. For the linear model, we will use a trust region of the form $\|\cdot\|_{\mathrm{LP}} \leq \Delta^{\mathrm{LP}}$ for some (polyhedral) norm $\|\cdot\|_{\mathrm{LP}}$, while for the quadratic model it will be $\|\cdot\| \leq \Delta$. Since all norms are equivalent in $\mathbb{R}^n$, there is a constant $\gamma \geq 1$ such that

$$(2.1) \qquad\qquad\qquad \|d\| \leq \gamma \|d\|_{\mathrm{LP}}$$

for all $d \in \mathbb{R}^n$.

We now define our Algorithm 2.1 for minimizing the penalty function (1.3) for a fixed value of σ. Throughout this section we omit the subscript and refer to our penalty function simply as ϕ in the case where σ is fixed.

Step 1 of Algorithm 2.1 aims to find the largest reduction in the linearized model within its trust region—we refer to this as the *linearized* problem—and attach the suffix LP to quantities associated with it. The intentions here are twofold.

First, the aim is to identify constraints whose inclusion in the working set for an EQP results in progress in the overall minimization. Ideally, near the solution these will correspond to active constraints at the solution. This is not the issue under consideration here, but it does have some ramifications on the design of our algorithm since we hope that our algorithm class is broad enough to permit correct identification of the active constraint set at the solution.

Second, the direction given by d_k^{LP} is also used to define the Cauchy step d_k^{C}, which, as in many trust-region methods, is used to guarantee convergence to a critical point. This is because the value of the LP solution provides a measure of nearness to optimality, and the Cauchy step is a step that provides corresponding improvement on the quadratic model. Condition (2.5) ensures that d_k^{C} is short enough that the quadratic model value is related to the LP model. The descent properties of the Cauchy step are what drive the bulk of our convergence theory; thus we ensure in step 2b that the step actually taken, d_k, shares these descent properties. Note that the Cauchy step d_k^{C} satisfies the conditions of step 2b, but the intention is to find a better step by solving a problem of the form (1.6).

Steps 3 and 4 are standard trust-region acceptance rules [4]. The ratio ρ_k of the actual to the predicted reduction of ϕ is used as a step acceptance criterion. If this ratio is negative, or close to zero, the step is rejected and the overall trust-region radius reduced. Otherwise the step will be accepted and, if ρ_k is close to one, the radius may be enlarged. We say that iteration k is *successful* if $\rho_k \geq \rho_u$. It is *very successful* if $\rho_k \geq \rho_s$.

ALGORITHM 2.1: MINIMIZATION ALGORITHM FOR $\phi(\mathbf{x})$

Initial data: x_0, $\Delta_0 > 0$, $\Delta_0^{\mathrm{LP}} > 0$, $0 < \rho_u \leq \rho_s < 1$, $0 < \kappa_l \leq \kappa_u < 1$, $\eta > 0$, $0 < \tau < 1$, and $\theta > 0$.

For $k = 0, 1, \ldots$, until a stopping test is satisfied, perform the following steps.

1.   Compute a solution d_k^{LP} to

$$\underset{\|d\|_{\mathrm{LP}} \leq \Delta_k^{\mathrm{LP}}}{\text{minimize}} \ \ell_k(d).$$

2a. Cauchy step. Compute $\alpha_k \leq 1$ as the first member of the sequence $\{\tau^i \min(1, \Delta_k/\|d_k^{\mathrm{LP}}\|)\}_{i=0,1,\ldots}$ for which

(2.2) $$\phi(x_k) - q_k(\alpha_k d_k^{\mathrm{LP}}) \geq \eta \left[\phi(x_k) - \ell_k(\alpha_k d_k^{\mathrm{LP}})\right].$$

Set $d_k^{\mathrm{C}} = \alpha_k d_k^{\mathrm{LP}}$.

2b. Compute d_k so that $\|d_k\| \leq \Delta_k$ and

$$q_k(d_k) \leq q_k(d_k^{\mathrm{C}}).$$

3.   Compute

$$\rho_k = \frac{\phi(x_k) - \phi(x_k + d_k)}{\phi(x_k) - q_k(d_k)}.$$

4a. If $\rho_k \geq \rho_s$, choose

$$\Delta_{k+1} \geq \Delta_k.$$

Otherwise set

(2.3) $$\Delta_{k+1} \in [\kappa_l \|d_k\|, \kappa_u \Delta_k].$$

4b. If $\rho_k \geq \rho_u$, set

$$x_{k+1} = x_k + d_k.$$

Otherwise set

$$x_{k+1} = x_k.$$

5. LP trust-region update.

If $\rho_k \geq \rho_u$, pick $\Delta_{k+1}^{\mathrm{LP}}$ so that the following two conditions hold:

(2.4)   (i) $\Delta_{k+1}^{\mathrm{LP}} \geq \|d_k^{\mathrm{C}}\|_{\mathrm{LP}}$,
(2.5)   (ii) $\Delta_{k+1}^{\mathrm{LP}} \leq \Delta_k^{\mathrm{LP}}$ if $\alpha_k < 1$.

Otherwise pick

(2.6) $\Delta_{k+1}^{\mathrm{LP}} \in [\min(\theta\|d_k\|_{\mathrm{LP}}, \Delta_k^{\mathrm{LP}}), \Delta_k^{\mathrm{LP}}]$.

Step 5 gives the conditions imposed on the radius for the linear model. In [1] a specific strategy is described that tries to relate Δ^{LP} to the expected step length so as to promote selection of a good active set. However, in this algorithm framework we only specify the characteristics such a strategy must have in order to guarantee global convergence. In the case of a successful step, we impose a limit on how much Δ^{LP} may be reduced, and allow increase only if the full LP step was taken. If the step d_k was not successful we allow for the possibility of decreasing Δ^{LP}, as Δ_k was decreased in step 4a.[2]

3. Convergence results for a fixed penalty function. In this section, we investigate the global convergence properties of Algorithm 2.1. In order to proceed, we need to make the following assumptions on the problem and the algorithm:

P1. The functions f, g, and h in (1.1) are Lipschitz continuous and have Lipschitz continuous derivatives over a bounded convex set whose interior contains the closure of the iterates $\{x_k\}$ generated by Algorithm 2.1.

P2. The sequence of Hessian matrices $\{B_k\}$ in (1.5) is bounded; thus there exists a constant $\beta > 0$ such that $|d^T B_k d| \leq \beta \|d\|^2$ for all k and all $d \in \mathbb{R}^n$.

Assumption P2 is made to simplify the analysis; see [15] for an analysis of a composite nonsmooth optimization algorithm in which B_k is computed by quasi-Newton updating. (As pointed out in section 1.1, both Algorithm 2.1 and the analysis in this section apply also to the case where $\phi(x) = \omega(F(x))$, with ℓ_k and q_k given by (1.10) and (1.11). In this case assumption P1 requires Lipschitz continuity of F, F', and ω.)

Under assumption P1 it follows immediately that $\phi(x)$ and $\ell_k(d)$ are Lipschitz continuous, and in particular that

$$(3.1) \qquad |\ell_k(d) - \ell_k(0)| \leq \lambda \|d\|_{\mathrm{LP}}$$

for some Lipschitz constant $\lambda > 0$.

The goal of our analysis is to prove that Algorithm 2.1 will find a critical point of ϕ. To do so, we follow Yuan [15] and define

$$(3.2) \qquad \Psi(x, \Delta) = \ell(x, 0) - \min_{\|d\| \leq \Delta} \ell(x, d),$$

which is the optimal decrease in the "linear" model $\ell(x, d)$ for a radius of size Δ. We can characterize criticality of ϕ using Ψ.

DEFINITION 3.1. $x_* \in \mathbb{R}^n$ *is a critical point (or stationary point) of ϕ if* $\Psi(x_*, 1) = 0$.

For future reference we note that, from assumption P2 and the subsequent convexity of $\ell(x, \cdot)$, we have in general

$$(3.3) \qquad \ell(x, 0) - \ell(x, \alpha d) \geq \alpha[\ell(x, 0) - \ell(x, d)],$$

and more specifically

$$(3.4) \qquad \phi(x_k) - \ell_k(\alpha d) \geq \alpha[\phi(x_k) - \ell_k(d)]$$

for any $\alpha \in [0, 1]$.

We now establish a number of intermediate lemmas leading up to our main global convergence result. Our first result provides bounds on the achievable reduction in

---

[2]The upper bound of one on α_k in (2.5) is used for simplicity. However, this bound can be generalized.

the linearized model for a radius of size Δ relative to that achieved with a radius of 1. From now on we use the following notation.

Notation. The solution d^{LP} of

$$(3.5) \qquad \min_{\|d\|_{\mathrm{LP}} \leq \Delta} \ell(x, d)$$

will also be denoted as d_Δ to emphasize its dependence on Δ. In particular, d_1 denotes the solution of (3.5) when $\Delta = 1$.

LEMMA 3.1. *Suppose that assumptions* P1 *and* P2 *hold. Then*

$$(3.6) \qquad \max(\Delta, 1)\Psi(x_k, 1) \geq \Psi(x_k, \Delta) \geq \min(\Delta, 1)\Psi(x_k, 1)$$

for any scalar $\Delta > 0$.

Proof. Since d_Δ is a solution of (3.5),

$$\Psi(x_k, \Delta) = \ell(x_k, 0) - \ell(x_k, d_\Delta).$$

There are two cases to consider. First consider the case $\Delta \leq 1$. Since $\|d_\Delta\|_{\mathrm{LP}} \leq 1$, the definition (3.2) implies that

$$\Psi(x_k, 1) \geq \ell(x_k, 0) - \ell(x_k, d_\Delta) = \Psi(x_k, \Delta),$$

which gives the left inequality of (3.6) in this case.

To get the right inequality, we need to show that $\Psi(x_k, \Delta) \geq \Delta\Psi(x_k, 1)$. By definition of d_Δ we have that $\|\Delta d_1\|_{\mathrm{LP}} \leq \Delta$, and so by (3.2) and (3.3),

$$\begin{aligned}
\Psi(x_k, \Delta) &\geq \ell(x_k, 0) - \ell(x_k, \Delta d_1) \\
&\geq \Delta(\ell(x_k, 0) - \ell(x_k, d_1)) \\
&= \Delta\Psi(x_k, 1).
\end{aligned}$$

This gives us (3.6) when $\Delta \leq 1$. In the case $\Delta \geq 1$, we need to establish

$$\Delta\Psi(x_k, 1) \geq \Psi(x_k, \Delta) \geq \Psi(x_k, 1),$$

but these inequalities follow immediately by making the above two-case argument with the values Δ and 1 interchanged. □

Lemma 3.1 essentially states that $\Psi(x, \cdot)$ is concave and monotonically increasing.

We shall also need the following result, which states that at a noncritical point of ϕ, the trust-region bound for the linearized problem, $\|d_\Delta\|_{\mathrm{LP}} \leq \Delta$, is active whenever the radius Δ is small enough. For brevity, let $\Psi_k(\Delta) \stackrel{\mathrm{def}}{=} \Psi(x_k, \Delta)$.

LEMMA 3.2. *Suppose that assumptions* P1 *and* P2 *hold (and thus that there is a Lipschitz constant* λ *for which* (3.1) *holds) and that* $\Psi_k(1) \neq 0$. *Then if* d_Δ *is a solution of* (3.5) *when* $x = x_k$,

$$(3.7) \qquad \|d_\Delta\|_{LP} \geq \min\left(\Delta, \frac{\Psi_k(1)}{\lambda}\right).$$

Proof. As before, let d_1 denote a solution of (3.5) when $x = x_k$ and $\Delta = 1$. Suppose that $\|d_\Delta\|_{\mathrm{LP}} < \Psi_k(1)/\lambda$. Then (3.1) gives that

$$(3.8) \qquad \ell_k(d_\Delta) \geq \ell_k(0) - \lambda\|d_\Delta\|_{\mathrm{LP}} > \ell_k(0) - \Psi_k(1) = \ell_k(d_1).$$

If $\Delta \geq 1$, this contradicts our definition of d_Δ as a solution of (3.5), so we must have $\|d_\Delta\|_{\mathrm{LP}} \geq \Psi_k(1)/\lambda$ and thus (3.7) in this case. If $\Delta < 1$, then (3.8) and the convexity of ℓ_k imply that ℓ_k is strictly decreasing along a line from d_Δ to d_1 (at least initially). Therefore, since d_Δ minimizes ℓ_k, it cannot lie in the strict interior of the trust region $\|d\|_{\mathrm{LP}} \leq \Delta$, and hence $\|d_\Delta\|_{\mathrm{LP}} = \Delta$. $\quad\square$

The next result provides a lower bound on the achievable reduction in the piecewise quadratic model in terms of the stepsize, the trust-region radius for the linearized problem, and our criticality measure. At this point, recall that we use d_k^{LP} to refer to the solution of the linear subproblem (3.5) solved in step 1 of Algorithm 2.1.

LEMMA 3.3. *Suppose that assumptions* P1 *and* P2 *hold. Then the model decrease satisfies*

$$\phi(x_k) - q_k(d_k) \geq \phi(x_k) - q_k(d_k^C) \geq \eta\alpha_k\Psi_k(\Delta_k^{LP}) \geq \eta\alpha_k\min(\Delta_k^{LP}, 1)\Psi_k(1).$$

Proof. The first inequality follows directly from the requirement in step 2b of Algorithm 2.1. To prove the second, note that inequality (3.4) and the requirement in step 2a give that

$$\begin{aligned}
\phi(x_k) - q_k(d_k^{\mathrm{C}}) &= \phi(x_k) - q_k(\alpha_k d_k^{\mathrm{LP}}) &\geq \eta\left[\phi(x_k) - \ell_k(\alpha_k d_k^{\mathrm{LP}})\right] \\
&\geq \eta\alpha_k\left[\phi(x_k) - \ell_k(d_k^{\mathrm{LP}})\right] &= \eta\alpha_k\Psi_k(\Delta_k^{\mathrm{LP}}).
\end{aligned}$$

The third inequality follows immediately from Lemma 3.1. $\quad\square$

Next, we establish an intuitive bound on the error introduced when using our quadratic approximation to ϕ.

LEMMA 3.4. *Suppose that assumptions* P1 *and* P2 *hold. Then*

$$|q_k(d_k) - \phi(x_k + d_k)| \leq M\|d_k\|^2$$

for some positive constant M.

Proof. As pointed out in section 1.1, the function ϕ can be expressed as $\phi(x) = \omega(F(x))$, where F and ω are defined as in (1.8) and (1.9). It follows from assumption P1 that F has a Lipschitz continuous derivative with constant λ^{F}, which implies that

$$\|F(x_k + d_k) - F(x_k) - F'(x_k)d_k\| \leq \lambda^{\mathrm{F}}\|d_k\|^2.$$

Since the function ω is Lipschitz continuous with some constant λ^ω, this inequality, together with assumption P2, implies that

$$\begin{aligned}
|q_k(d_k) - \phi(x_k + d_k)| &= |\omega(F(x_k) + F'(x_k)d_k) + \tfrac{1}{2}d_k^T B_k d_k - \omega(F(x_k + d_k))| \\
&\leq \lambda^\omega\|F(x_k + d_k) - F(x_k) - F'(x_k)d_k\| + \tfrac{1}{2}\beta\|d_k\|^2 \\
&\leq (\lambda^\omega\lambda^{\mathrm{F}} + \tfrac{1}{2}\beta)\|d_k\|^2 \\
&= M\|d_k\|^2,
\end{aligned}$$

where $M = \lambda^\omega\lambda^{\mathrm{F}} + \tfrac{1}{2}\beta$. $\quad\square$

The following technical result essentially says that either the Cauchy step is on the boundary of one of our trust regions or it has a lower bound proportional to the optimality criterion.

LEMMA 3.5. *Suppose that assumptions* P1 *and* P2 *hold. Then at any iteration of Algorithm* 2.1

$$(3.9) \quad \alpha_k\Delta_k^{LP} \geq \|d_k^C\|_{LP} \geq \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{LP}, \frac{\Psi_k(1)}{\lambda}, \min\left(1, \frac{1}{\Delta_k^{LP}}\right)\frac{2(1-\eta)\tau\Psi_k(1)}{\beta\gamma^2}\right).$$

Proof. The first inequality in (3.9) follows immediately, since

$$\|d_k^{\mathrm{C}}\|_{\mathrm{LP}} = \alpha_k\|d_k^{\mathrm{LP}}\|_{\mathrm{LP}} \le \alpha_k\Delta_k^{\mathrm{LP}}.$$

To establish the second inequality, suppose first that the decrease condition (2.2) in step 2a of Algorithm 2.1 is immediately satisfied for $\alpha_k = \min(1, \Delta_k/\|d_k^{\mathrm{LP}}\|)$. Then, using (2.1) and Lemma 3.2,

$$\|d_k^{\mathrm{C}}\|_{\mathrm{LP}} = \|\alpha_k d_k^{\mathrm{LP}}\|_{\mathrm{LP}} = \min\left(\frac{\Delta_k}{\|d_k^{\mathrm{LP}}\|}, 1\right)\|d_k^{\mathrm{LP}}\|_{\mathrm{LP}}$$

(3.10)
$$\ge \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\mathrm{LP}}, \frac{\Psi_k(1)}{\lambda}\right),$$

which gives the first three terms in (3.9). On the other hand, if $\alpha_k < \min(1, \Delta_k/\|d_k^{\mathrm{LP}}\|)$, then the decrease condition (2.2) must have been violated for α_k/τ, and so

$$\phi(x_k) - q_k(\alpha_k d_k^{\mathrm{LP}}/\tau) = \phi(x_k) - \ell_k(\alpha_k d_k^{\mathrm{LP}}/\tau) - \tfrac{1}{2}(\alpha_k/\tau)^2(d_k^{\mathrm{LP}})^T B_k d_k^{\mathrm{LP}}$$

(3.11)
$$\le \eta\left[\phi(x_k) - \ell_k(\alpha_k d_k^{\mathrm{LP}}/\tau)\right].$$

Now using assumption P2, (2.1), (3.4), and Lemma 3.1, this inequality implies that

$$\tfrac{1}{2}(\alpha_k/\tau)^2(d_k^{\mathrm{LP}})^T B_k d_k^{\mathrm{LP}} \ge (1-\eta)\left[\phi(x_k) - \ell_k(\alpha_k d_k^{\mathrm{LP}}/\tau)\right]$$
$$\tfrac{1}{2}(\alpha_k/\tau)^2\beta\gamma^2\|d_k^{\mathrm{LP}}\|_{\mathrm{LP}}^2 \ge (1-\eta)(\alpha_k/\tau)\Psi_k(\Delta_k^{\mathrm{LP}})$$
$$\tfrac{1}{2}(\alpha_k/\tau)\beta\gamma^2\|d_k^{\mathrm{LP}}\|_{\mathrm{LP}}\,\Delta_k^{\mathrm{LP}} \ge (1-\eta)\min(\Delta_k^{\mathrm{LP}}, 1)\Psi_k(1)$$

(3.12)
$$\alpha_k\|d_k^{\mathrm{LP}}\|_{\mathrm{LP}} \ge \frac{2(1-\eta)\tau}{\beta\gamma^2}\min\left(1, \frac{1}{\Delta_k^{\mathrm{LP}}}\right)\Psi_k(1).$$

Since $\alpha_k d_k^{\mathrm{LP}} = d_k^{\mathrm{C}}$, this inequality combined with (3.10) gives the second inequality in (3.9).   □

Our next result is crucial. It provides lower bounds on both the trust-region radius Δ_k and the length of the Cauchy step at a noncritical iterate in the case where the trust-region radius for the linearized problem stays bounded.

LEMMA 3.6. *Suppose Algorithm 2.1 is applied to the problem (1.2) and that assumptions* P1 *and* P2 *hold. Suppose that $\{\Delta_k^{LP}\}$ is bounded above and that $\Psi_k(1) \ge \delta > 0$ for all k. Then there exists a constant $\Delta_{min} > 0$ such that*

(3.13)
$$\Delta_k \ge \Delta_{min} \quad and \quad \alpha_k\Delta_k^{LP} \ge \frac{\Delta_{min}}{\gamma}$$

for all k.

Proof. By assumption, there exists $\Delta_{\max} \ge 1$ such that

(3.14)
$$\Delta_k^{\mathrm{LP}} \le \Delta_{\max} \quad \text{for all } k.$$

This inequality, the assumption $\Psi_k(1) \ge \delta$, and Lemma 3.5 imply

(3.15)
$$\|d_k^{\mathrm{C}}\|_{\mathrm{LP}} \ge \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\mathrm{LP}}, \Delta_{\mathrm{crit}}\right),$$

where

(3.16)
$$\Delta_{\mathrm{crit}} = \min\left(\frac{1}{\lambda}, \frac{2(1-\eta)\tau}{\beta\gamma^2\Delta_{\max}}\right)\delta.$$

If the iteration is successful ($\rho_k \geq \rho_u$), the rule (2.4) for choosing Δ_k^{LP} in step 5 of the algorithm ensures that $\Delta_{k+1}^{\mathrm{LP}} \geq \|d_k^{\mathrm{C}}\|_{\mathrm{LP}}$ and therefore

$$(3.17) \qquad \Delta_{k+1}^{\mathrm{LP}} \geq \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\mathrm{LP}}, \Delta_{\mathrm{crit}}\right).$$

Let us now consider the case in which the iteration is unsuccessful. Using Lemma 3.3 and (3.15), we have that

$$\phi(x_k) - q_k(d_k) \geq \phi(x_k) - q_k(d_k^{\mathrm{C}}) \geq \eta\alpha_k \min(\Delta_k^{\mathrm{LP}}, 1)\delta = \eta\alpha_k\Delta_k^{\mathrm{LP}} \min\left(\frac{1}{\Delta_k^{\mathrm{LP}}}, 1\right)\delta$$

$$\geq \frac{\eta\delta}{\Delta_{\max}}\alpha_k\Delta_k^{\mathrm{LP}} \geq \frac{\eta\delta}{\Delta_{\max}} \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\mathrm{LP}}, \Delta_{\mathrm{crit}}\right).$$

(3.18)

From Lemma 3.4 and (3.18) we have that

$$(3.19) \qquad 1 - \rho_k \leq \frac{|\phi(x_k + d_k) - q_k(d_k)|}{\phi(x_k) - q_k(d_k)} \leq \frac{M\|d_k\|^2 \Delta_{\max}}{\eta\delta \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\mathrm{LP}}, \Delta_{\mathrm{crit}}\right)}.$$

This implies that $\|d_k\|$ and $(1 - \rho_k)$ are related by the inequality

$$(3.20) \qquad \|d_k\|^2 \geq \frac{(1 - \rho_k)\eta\delta}{M\Delta_{\max}} \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\mathrm{LP}}, \Delta_{\mathrm{crit}}\right)$$

at each step. Now, since the iteration is unsuccessful, $\rho_k < \rho_u$ and $1 - \rho_k > 1 - \rho_u$, which, using (2.1) and (3.20), implies

$$\theta^2\|d_k\|_{\mathrm{LP}}^2 \geq \frac{\theta^2}{\gamma^2}\|d_k\|^2 \geq \theta^2\frac{(1 - \rho_u)\eta\delta}{\gamma^2 M\Delta_{\max}} \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\mathrm{LP}}, \Delta_{\mathrm{crit}}\right)$$

$$\geq \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\mathrm{LP}}, \Delta_{\mathrm{crit}}, \frac{(1 - \rho_u)\eta\theta^2\delta}{\gamma^2 M\Delta_{\max}}\right)^2.$$

Using this fact and the lower bound in (2.6), we have that, if the step is unsuccessful,

$$(3.21) \qquad \Delta_{k+1}^{\mathrm{LP}} \geq \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\mathrm{LP}}, \Delta_{\mathrm{crit}}, \frac{(1 - \rho_u)\eta\theta^2\delta}{\gamma^2 M\Delta_{\max}}\right).$$

Since the right side of (3.21) is clearly less than or equal to the right side of (3.17), which holds when the step is accepted, then (3.21) must hold at each iteration.

We can consider Δ_k in a similar fashion. If Δ_k was decreased because $\rho_k < \rho_s$, then $1 - \rho_k > 1 - \rho_s$ and (3.20) implies

$$\frac{\kappa_l^2}{\gamma^2}\|d_k\|^2 \geq \frac{(1 - \rho_s)\eta\kappa_l^2\delta}{\gamma^2 M\Delta_{\max}} \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\mathrm{LP}}, \Delta_{\mathrm{crit}}\right)$$

$$\geq \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\mathrm{LP}}, \Delta_{\mathrm{crit}}, \frac{(1 - \rho_s)\eta\kappa_l^2\delta}{\gamma^2 M\Delta_{\max}}\right)^2.$$

Together with (2.3) this implies

$$(3.22) \qquad \frac{\Delta_{k+1}}{\gamma} \geq \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\mathrm{LP}}, \Delta_{\mathrm{crit}}, \frac{(1 - \rho_s)\eta\kappa_l^2\delta}{\gamma^2 M\Delta_{\max}}\right).$$

Since Δ_k is not reduced when $\rho_k \geq \rho_s$, (3.22) must then hold at each iteration.

Now we can combine the recursions (3.21) and (3.22) to yield

$$(3.23) \quad \min\left(\frac{\Delta_{k+1}}{\gamma}, \Delta_{k+1}^{\mathrm{LP}}\right) \geq \min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\mathrm{LP}}, \Delta_{\mathrm{crit}}, \frac{(1-\rho_s)\eta\kappa_l^2\delta}{\gamma^2 M \Delta_{\max}}, \frac{(1-\rho_u)\eta\theta^2\delta}{\gamma^2 M \Delta_{\max}}\right),$$

which holds at every iteration. Applying this recursion over the entire sequence implies that for all k

$$\min\left(\frac{\Delta_k}{\gamma}, \Delta_k^{\mathrm{LP}}\right) \geq \min\left(\frac{\Delta_0}{\gamma}, \Delta_0^{\mathrm{LP}}, \Delta_{\mathrm{crit}}, \frac{(1-\rho_s)\eta\kappa_l^2\delta}{\gamma^2 M \Delta_{\max}}, \frac{(1-\rho_u)\eta\theta^2\delta}{\gamma^2 M \Delta_{\max}}\right)$$
$$\equiv \Delta_{\mathrm{low}}.$$

Thus we can conclude that $\Delta_k \geq \Delta_{\min} \equiv \gamma\Delta_{\mathrm{low}}$ for all k. It then follows from (3.15) and the fact that $\Delta_{\mathrm{low}} \leq \Delta_{\mathrm{crit}}$ that

$$\alpha_k \Delta_k^{\mathrm{LP}} \geq \|d_k^{\mathrm{C}}\|_{\mathrm{LP}} \geq \Delta_{\mathrm{low}} = \frac{\Delta_{\min}}{\gamma}$$

for all k.   □

This immediately enables us to deduce that if the algorithm is unable to make progress, it must be because it has reached a critical point.

COROLLARY 3.7. *Suppose that assumptions* P1 *and* P2 *hold and that there are only a finite number of iterations for which* $\rho_k \geq \rho_u$. *Then* $x_k = x_*$ *for all sufficiently large* k, *and* x_* *is a critical point of* $\phi(x)$.

Proof. Step 4 of the algorithm ensures that if there are only a finite number of (successful) iterations for which $\rho_k \geq \rho_u$, then $x_k = x_*$ for all $k > k_0$ for some $k_0 \geq 0$. Moreover, $\Psi_k(1) = \Psi_{k_0}(1)$ for all $k \geq k_0$. Furthermore, as $\rho_k < \rho_u$ for all $k \geq k_0$, the update rules for the trust regions imply that Δ_k converges to zero and Δ_k^{LP} is bounded above for all k. But then $\Psi_k(1) = 0$ for all $k \geq k_0$, since otherwise Lemma 3.6 contradicts the fact that Δ_k converge to zero. It thus follows from Definition 3.1 that x_* is a critical point of ϕ.   □

Finally we are able to state our main global convergence result.

THEOREM 3.8. *Suppose Algorithm* 2.1 *is applied to problem* (1.2) *and that assumptions* P1 *and* P2 *hold. If the sequence* $\{\phi(x_k)\}$ *is bounded below, then either*

$$\Psi_l(1) = 0 \ \text{ for some } \ l \geq 0$$

or

$$\liminf_{k\to\infty} \Psi_k(1) = 0.$$

Proof. If there are only a finite number of successful iterations, the first of the stated possibilities follows immediately from Corollary 3.7. Otherwise, there is an infinite subsequence $\mathcal{K}$ of successful iterations. This means that $\rho_k \geq \rho_u$, and $\{\phi(x_k)\}$ is bounded from below for all $k \in \mathcal{K}$.

The proof now proceeds by contradiction. Assume there is a constant δ such that $\Psi_k(1) \geq \delta > 0$ for all $k \in \mathcal{K}$. We will consider separately the two cases: when the LP trust-region radius $\{\Delta_k^{\mathrm{LP}}\}$ is bounded above and when $\{\Delta_k^{\mathrm{LP}}\}$ is unbounded.

Case 1. If $\{\Delta_k^{\mathrm{LP}}\}$ is bounded above, it follows from Lemma 3.6 that $\Delta_k \geq \Delta_{\min} > 0$.

For our infinite subsequence $\mathcal{K}$ of successful iterations, Lemmas 3.3 and 3.6 give

$$
\begin{aligned}
\phi(x_k) - \phi(x_{k+1}) &\geq \rho_u(\phi(x_k) - q_k(d_k)) \\
&\geq \rho_u \eta \alpha_k \min(\Delta_k^{\mathrm{LP}}, 1)\delta \\
&\geq \rho_u \eta \alpha_k \Delta_k^{\mathrm{LP}} \min(1, 1/\Delta_k^{\mathrm{LP}})\delta \\
&\geq \rho_u \eta \Delta_{\min}\delta/(\gamma\Delta_{\max}) > 0,
\end{aligned}
$$

for all $k \in \mathcal{K}$, where $\Delta_{\max} > 1$ is the upper bound for Δ_k^{LP}. But then summing this inequality over all $k \in \mathcal{K}$ contradicts the fact that the sequence $\{\phi(x_k)\}$ is bounded from below. Thus Case 1 does not occur.

Case 2. Suppose that the LP trust-region radius $\{\Delta_k^{\mathrm{LP}}\}$ is unbounded. Then, since the radius is only increased in step 5 of Algorithm 2.1 when $\alpha_k \geq 1$, there is an infinite sequence $\mathcal{K}$ such that $\Delta_k^{\mathrm{LP}} > 1$, $\alpha_k \geq 1$, and $\rho_k \geq \rho_u$ for all $k \in \mathcal{K}$. Then from Lemma 3.3 we have

$$
\begin{aligned}
\phi(x_k) - \phi(x_{k+1}) &\geq \rho_u(\phi(x_k) - q_k(d_k)) \\
&\geq \rho_u \eta \alpha_k \min(\Delta_k^{\mathrm{LP}}, 1)\Psi_k(1) \\
&\geq \rho_u \eta \Psi_k(1) \\
&\geq \rho_u \eta \delta,
\end{aligned}
$$

for all $k \in \mathcal{K}$. This again contradicts the assumption that $\{\phi(x)\}$ is bounded from below, and Case 2 cannot occur.

Cases 1 and 2 therefore imply that the assumption $\Psi_k(1) \geq \delta > 0$ for all k must be false, which proves the desired result

$$
\liminf_{k \to \infty} \Psi_k(1) = 0. \qquad \square
$$

This result guarantees that if $\phi(x)$ is bounded below, the criticality criterion $\Psi_k(1)$ eventually becomes arbitrarily small. This implies that if the sequence $\{x_k\}$ is bounded, there exists an accumulation point of Algorithm 2.1 which is a critical point for (1.2).

4. A penalty method for nonlinear programming. We now discuss how to automatically adjust the penalty parameter σ as our algorithm proceeds so as to encourage convergence to a critical point of (1.1). We will make use of the following definitions.

We let

$$
(4.1) \qquad v(x) = \|h(x)\| + \|g^-(x)\|
$$

be a measure of constraint violation, so that the penalty function (1.3) can be written as

$$
(4.2) \qquad \phi_\sigma(x) = f(x) + \sigma v(x).
$$

We define a (piecewise) linear model of the constraint violation by

$$
(4.3) \qquad \ell^v(x, d) = \|h(x) + \nabla h(x)^T d\| + \|(g(x) + \nabla g(x)^T d)^-\|.
$$

We can therefore write the model (1.4) of the penalty function as

$$
(4.4) \qquad \ell^{\phi_\sigma}(x, d) = f(x) + \nabla f(x)^T d + \sigma \ell^v(x, d).
$$

Since the penalty parameter σ will now vary, we write the measure of criticality (3.2) for the penalty function as

$$(4.5) \qquad \Psi_\sigma(x, \Delta) = \ell^{\phi_\sigma}(x, 0) - \min_{\|d\| \le \Delta} \ell^{\phi_\sigma}(x, d).$$

Definition 3.1 states that $x_* \in \mathbb{R}^n$ is a critical point of the penalty function ϕ_σ if $\Psi_\sigma(x_*, 1) = 0$. Criticality of the measure of constraint violation $v(x)$ will be measured by the function

$$(4.6) \qquad \theta(x, \Delta) = \ell^v(x, 0) - \min_{\|d\| \le \Delta} \ell^v(x, d).$$

DEFINITION 4.1. $x_* \in \mathbb{R}^n$ is a critical point of the infeasibility measure $v(x)$ if $\theta(x_*, 1) = 0$.

It is well known [10] that the penalty function (4.2) is exact in the sense that, for sufficiently large values of σ, strict local minimizers of the nonlinear program (1.1) that satisfy the Mangasarian–Fromovitz constraint qualification (MFCQ) are minimizers of ϕ_σ. We are also interested in the converse result, given that our algorithm minimizes the penalty function.

THEOREM 4.1. If x_* is a critical point of ϕ_σ for some σ and is feasible for (1.1), then x_* is a KKT point of the nonlinear program (1.1). If x_* is infeasible and is a critical point of ϕ_σ for all sufficiently large σ, then x_* is an infeasible critical point of $v(x)$.

Proof. At a feasible critical point x_* of ϕ_σ, the vector $d = 0$ minimizes $\ell^{\phi_\sigma}(x_*, d)$, which implies that $d = 0$ is an optimal feasible solution of the linear program

$$(4.7) \qquad \begin{aligned} \underset{d}{\text{minimize}} \quad & d^T \nabla f(x_*) \\ \text{subject to} \quad & h(x_*) + \nabla h_i(x_*)^T d = 0, \\ & g(x_*) + \nabla g_i(x_*)^T d \ge 0. \end{aligned}$$

Since the constraints of (4.7) are linear, the KKT conditions for (4.7) are satisfied, and the KKT conditions for (4.7) are identical to the KKT conditions for (1.1).

Suppose that x_* is infeasible and assume, by way of contradiction, that $\theta(x_*, 1) > 0$. Then by (4.6), there exists $\|d^*\| \le 1$ such that $\ell^v(x_*, 0) - \ell^v(x_*, d^*) > 0$, and therefore for any σ large enough

$$-\nabla f(x_*)^T d^* + \sigma \left(\ell^v(x_*, 0) - \ell^v(x_*, d^*) \right) > 0$$

or

$$\ell^{\phi_\sigma}(x_*, 0) - \ell^{\phi_\sigma}(x_*, d^*) > 0.$$

By (4.4) this implies that $\Psi_\sigma(x_*, 1) > 0$ for arbitrarily large σ, contradicting the assumption that x_* is a critical point for ϕ_σ for all large σ. This contradiction implies that if x_* is infeasible, then $\theta(x_*, 1) = 0$.   □

4.1. Penalty update procedure. Our penalty parameter strategy is based on our belief that it is as important to try to decrease the violation $v(x)$ as it is to aim for criticality of $\phi_\sigma(x)$. Since we cannot be sure that there is a (locally) feasible point for the constraints (1.1c)–(1.1b), we might instead measure the quality of the current violation in terms of its criticality, $\theta(x, \Delta)$. Thus we contend it is reasonable to ask that the current value of the penalty parameter σ always satisfies

$$\Psi_\sigma(x_k, 1) \ge \xi \sigma \theta(x_k, 1)$$

for some predefined constant $\xi \in (0,1)$, and to increase the current value if this inequality fails. Hence we cannot consider our iterate to be near a critical point for $\phi_\sigma(x)$ unless it is near a critical point of $v(x)$.

The use of the criticality measure $\Psi_\sigma(x_k, 1)$ requires the solution of an LP with radius 1. Since the algorithm computes the quantity $\Psi_\sigma(x_k, \Delta_k)$ at every iteration, we would instead prefer to use this quantity to estimate criticality, thereby avoiding the extra computational cost of solving a second LP. As we show below, this is possible so long as Δ_k lies within a preset interval $[\delta_{\min}, \delta_{\max}]$. If Δ_k is not in this interval, we will use $\Psi_\sigma(x_k, \delta_k)$ to measure criticality, where δ_k is the closest value to Δ_k in $[\delta_{\min}, \delta_{\max}]$.

Based on this strategy, the set of permissible penalty parameters at an iterate x_k, with trust-region radius Δ_k, is defined as

$$\Omega(x_k, \delta_k) \stackrel{\text{def}}{=} \{\sigma \mid \Psi_\sigma(x_k, \delta_k) \geq \xi \sigma \theta(x_k, \delta_k)\}.$$

Of course, computation of the quantity $\theta(x_k, \delta_k)$ also involves the solution of an additional LP, but once x_k is near the feasible region, linearized feasibility is likely to be attainable inside the trust region so that $\theta(x_k, \delta_k) = v(x_k)$, and the stronger but easier-to-check condition, $\Psi_\sigma(x_k, \delta_k) \geq \xi \sigma v(x_k)$, will often hold.

We now describe an algorithm for solving the nonlinear programming problem (1.1) in which the penalty parameter is updated at every iteration. It makes use of Algorithm 2.1 to generate steps.

ALGORITHM 4.1: PENALTY METHOD FOR SOLVING (1.1)

Initial data: x_1, σ_0. Set the initial parameters of Algorithm 2.1 as well as $\epsilon > 0$, $0 < \xi < 1$, and $0 < \delta_{\min} \leq \delta_{\max}$.

For $k = 1, 2, \ldots$, until a stopping test for (1.1) is satisfied, perform the following steps.
1.   Let $\delta_k = \text{mid}(\delta_{\min}, \Delta_k^{\text{LP}}, \delta_{\max})$.
         If $\sigma_{k-1} \in \Omega(x_k, \delta_k)$,
             set $\sigma_k = \sigma_{k-1}$.
         Else,
             choose any $\sigma_k \in \Omega(x_k, \delta_k)$ for which $\sigma_k \geq \sigma_{k-1} + \epsilon$.
2.   Perform steps 1–5 of Algorithm 2.1.

As was our stated intention, the penalty update strategy in step 1 allows us to (re-)use quantities computed at Δ_k whenever $\Delta_k \in [\delta_{\min}, \delta_{\max}]$. It also ensures that

$$(4.8) \qquad\qquad \Psi_{\sigma_k}(x_k, \delta_k) \geq \xi \sigma_k \theta(x_k, \delta_k)$$

is satisfied at each iteration, and that if σ is increased, it is because $\sigma_{k-1} \notin \Omega(x_k, \delta_k)$; i.e.,

$$(4.9) \qquad\qquad \Psi_{\sigma_{k-1}}(x_k, \delta_k) < \xi \sigma_{k-1} \theta(x_k, \delta_k).$$

It is always possible to find a point in $\Omega(x_k, \delta_k)$ for any $\xi < 1$ so that step 1 of Algorithm 4.1 is well defined. To see this, note that definitions (4.1)–(4.6) imply that

$$(4.10) \qquad \Psi_\sigma(x_k, \delta_k) = \sigma v(x_k) - \min_{\|d\| \leq \delta_k} \left(\nabla f(x_k)^T d + \sigma \ell^v(x_k, d) \right)$$

$$\geq \sigma v(x_k) - \|\nabla f(x_k)\|\delta_k - \sigma \min_{\|d\|\leq \delta_k} \ell^v(x_k, d)$$

$$= -\|\nabla f(x_k)\|\delta_k + \sigma \left(v(x_k) - \min_{\|d\|\leq \delta_k} \ell^v(x_k, d) \right)$$

(4.11) $$= -\|\nabla f(x_k)\|\delta_k + \sigma\theta(x_k, \delta),$$

and thus that $\sigma \in \Omega(x_k, \delta_k)$ for all

(4.12) $$\sigma \geq \frac{\|\nabla f(x_k)\|\delta_k}{(1-\xi)\theta(x_k, \delta_k)}.$$

Notice, however, that it is highly likely that this value will grow without bound if x_k approaches feasibility, so the simpler expedient of always setting σ_k to ensure (4.12) is not to be recommended.

4.2. Penalty method analysis. We begin by recasting the inequality (4.8) in terms of $\Psi_\sigma(x_k, 1)$ instead of $\Psi_\sigma(x_k, \delta_k)$. To do this we recall Lemma 3.1 and note that since the function $\theta(x, \delta)$, like $\Psi(x, \delta)$, is monotonically increasing and concave in δ, the same arguments as used in the proof of Lemma 3.1 imply that

(4.13) $$\min(\delta_k, 1)\theta(x_k, 1) \leq \theta(x_k, \delta_k) \leq \max(\delta_k, 1)\theta(x_k, 1).$$

This then implies the following bound.

LEMMA 4.2. *The values σ_k generated by Algorithm 4.1 satisfy*

(4.14) $$\Psi_{\sigma_k}(x_k, 1) \geq \xi \min\left(\delta_{\min}, \frac{1}{\delta_{\max}} \right) \sigma_k \theta(x_k, 1).$$

Proof. Using (3.6) followed by (4.8) followed by (4.13) yields

$$\Psi_{\sigma_k}(x_k, 1) \geq \frac{\Psi_{\sigma_k}(x_k, \delta_k)}{\max(\delta_k, 1)} \geq \frac{\xi\sigma_k\theta(x_k, \delta_k)}{\max(\delta_k, 1)} \geq \xi\sigma_k \frac{\min(\delta_k, 1)}{\max(\delta_k, 1)}\theta(x_k, 1),$$

which gives (4.14), since $\delta_k \in [\delta_{\min}, \delta_{\max}]$.   □

We now present two convergence results for Algorithm 4.1 that rely heavily on the convergence properties of Algorithm 2.1. We first consider the case in which the penalty parameter is updated only a finite number of times.

THEOREM 4.3. *Suppose Algorithm 4.1 applied to (1.1) generates a bounded sequence of iterates and that assumptions* P1 *and* P2 *hold. If $\{\sigma_k\}$ is bounded, then there is a cluster point x_* of the sequence $\{x_k\}$ which is either a KKT point of the nonlinear program (1.1) or a critical point of v.*

Proof. Since $\{\sigma_k\}$ is bounded, it follows from step 1 of Algorithm 4.1 that $\sigma_k = \sigma$ is constant for all large k. Algorithm 4.1 therefore reduces to Algorithm 2.1, i.e., to the minimization of a single penalty function. By Theorem 3.8, if $\{\phi_{\sigma_k}(x_k)\}$ is bounded below, there is a limit point x_* of the sequence of iterates $\{x_k\}$ such that

(4.15) $$\Psi_\sigma(x_*, 1) = 0.$$

If x_* is infeasible, since there is a subsequence $\{x_l\}$ with $\Psi_\sigma(x_l, 1) \to 0$ and since (4.14) holds at each iteration, we must have that $\theta(x_l, 1) \to 0$. Then, since $\theta(\cdot, 1)$ is continuous, $\theta(x_*, 1) = 0$. Therefore, x_* is an infeasible critical point.

If x_* is feasible, i.e., $v(x_*) = 0$, then it follows immediately from Theorem 4.1 that x_* is a KKT point for the nonlinear program (1.1).   □

Our final result describes possible outcomes when the penalty parameter is unbounded.

THEOREM 4.4. *Suppose that Algorithm 4.1 generates a bounded sequence of iterates $\{x_k\}$ and that $\{\sigma_k\} \to \infty$. Then either*

(i) *the sequence $\{x_k\}$ is not asymptotically feasible (i.e., $v(x_k) \not\to 0$), in which case there is an infeasible cluster point x_* that satisfies $\theta(x_*, 1) = 0$; or*

(ii) *the sequence $\{x_k\}$ is feasible in the sense that $v(x_k) \to 0$. In this case, either* (a) *there is a cluster point of $\{x_k\}$ that satisfies the KKT conditions, or* (b) *there is a feasible cluster point of $\{x_k\}$ at which MFCQ is violated.*

Proof. Consider the sequence of iterates at which the penalty parameter is increased. For each k in this subsequence, condition (4.9) holds, and thus we have

$$\Psi_{\sigma_{k-1}}(x_k, \delta_k) < \xi \sigma_{k-1} \theta(x_k, \delta_k).$$

Now (4.11) holds here, so

$$\Psi_{\sigma_{k-1}}(x_k, \delta_k) \geq -\|\nabla f(x_k)\|\delta_k + \sigma_{k-1}\theta(x_k, \delta_k)$$

and thus, using (4.13),

$$(4.16) \quad (1 - \xi)\sigma_{k-1} \leq \frac{\|\nabla f(x_k)\|\delta_k}{\theta(x_k, \delta_k)} \leq \frac{\|\nabla f(x_k)\|\delta_k}{\theta(x_k, 1)\min(1, \delta_k)} \leq \frac{\|\nabla f(x_k)\|\delta_{\max}}{\theta(x_k, 1)}.$$

But as $\{\sigma_k\}$, and consequently $\{\sigma_{k-1}\}$, is assumed unbounded and $\{\nabla f(x_k)\}$ is bounded, it follows that, for that subsequence of $\{x_k\}$ for which σ was increased, we have $\theta(x_k, 1) \to 0$.

If $\limsup v(x_k) > 0$, then, since the sequence $\{x_k\}$ is bounded and $\theta(x_k, 1) \to 0$, there is a limit point with $v(\hat{x}) > 0$ and $\theta(\hat{x}, 1) = 0$; i.e., $\hat{x}$ is an infeasible stationary point of $v(x)$. This implies (i) in that case.

On the other hand, if $\lim v(x_k) = 0$, then there is a cluster point $\hat{x}$ with $v(\hat{x}) = 0$. If $\hat{x}$ satisfies MFCQ, then $\nabla h(\hat{x})$ has full rank and there is a direction $\|d^M\| < \delta_{\min}$ such that

$$\nabla h(\hat{x})^T d^M = 0 = -h(\hat{x}) \text{ and } \nabla g(\hat{x})^T d^M + g(\hat{x}) > 0.$$

Suppose by way of contradiction that $\hat{x}$ is not a KKT point. Then there is a first-order feasible descent direction $\|d^F\| < \delta_{\min}$ such that

$$\nabla h(\hat{x})^T d^F = 0 = -h(\hat{x}), \quad \nabla g(\hat{x})^T d^F + g(\hat{x}) \geq 0, \quad \text{and} \quad \nabla f(\hat{x})^T d^F < 0.$$

Clearly, there is a convex combination $\hat{d} = (1 - \alpha)d^F + \alpha d^M$, with $\alpha \in (0, 1)$, such that

$$(4.17) \qquad\qquad \nabla h(\hat{x})^T \hat{d} + h(\hat{x}) = 0,$$
$$\nabla g(\hat{x})^T \hat{d} + g(\hat{x}) > 0, \quad \text{and} \quad \nabla f(\hat{x})^T \hat{d} < 0.$$

Now since $\nabla h(\hat{x})$ has full rank, for any x sufficiently near $\hat{x}$ there is a unique vector $d(x)$ of the form

$$(4.18) \qquad\qquad d(x) = \hat{d} + \nabla h(\hat{x})u(x)$$

for some $u(x) \in R^m$, which (nonuniquely) solves

$$(4.19) \qquad\qquad h(x) + \nabla h(x)^T d(x) = 0.$$

To see this, note that (4.18)–(4.19) imply

$$(4.20) \qquad \left[h(x) + \nabla h(x)^T \hat{d} \right] + \nabla h(x)^T \nabla h(\hat{x}) u(x) = 0.$$

Since h is smooth, this equation shows that $u(x)$ is uniquely defined in a neighborhood of $\hat{x}$ and varies continuously with x—and so does $d(x)$. Furthermore, by (4.17) the term in square brackets in (4.20) can be made arbitrarily small if x is close to $\hat{x}$, and hence $d(x)$ is arbitrarily close to $\hat{d}$.

Using these facts we have that $d(x)$ satisfies

$$(4.21) \qquad\qquad \nabla g(x)^T d(x) + g(x) > 0,$$
$$(4.22) \qquad\qquad \nabla f(x)^T d(x) < 0$$

for x sufficiently near $\hat{x}$.

Now note that since $\|d(x)\| < \delta_{\min}$, we have by (4.3), (4.19), and (4.21) that $\ell^v(x, d(x)) = 0$. By the nonnegativity of $\ell^v(x, d(x))$ and the definition (4.6), this implies that $\theta(x) = \ell^v(x, 0) = v(x)$. In addition, since $\nabla f(x)^T d(x) < 0$, we have from (4.10) that

$$\Psi_\sigma(x, \delta) > \sigma v(x) = \sigma \theta(x, \delta)$$

for any $\delta \geq \delta_{\min}$. Therefore, for any iterate x_k sufficiently near $\hat{x}$, $\sigma \in \Omega(x_k, \delta_k)$ for all $\sigma \geq 0$. As a result, for this subsequence of iterates, σ is never updated in a neighborhood of $\hat{x}$.

This argument applies to any feasible limit point that satisfies MFCQ. Therefore, it is not possible for all such points to have a descent direction, for otherwise the penalty parameter would be updated only a finite number of times, contradicting the assumption that $\sigma_k \to \infty$. In other words, we cannot have that all limit points satisfy MFCQ and are not KKT points. This proves (ii).   ☐

Thus we are able to embed a relatively simple penalty-parameter update scheme *within* Algorithm 2.1 and derive useful convergence results. Another possibility which could be tried is to update the penalty parameter as needed once a globally convergent method has approximately minimized ϕ_σ with the current (fixed) σ. Rules to achieve this are known [4, 11], but we are concerned that this may prove to be inefficient, particularly when an inappropriate initial σ is specified.

In the current version of the exact penalty method SLIQUE [1], the penalty parameter is updated by a procedure that requires $\sigma_k \in \Omega(x_k, \delta_k)$ at each iteration, as well as some further conditions. Therefore, Theorem 4.3 essentially holds for SLIQUE.[3] However, because of the additional conditions on σ_k, it is not clear whether a result like Theorem 4.4 can be proved for SLIQUE.

5. Conclusions and perspectives. In this paper we have proposed a trust-region algorithm for nonlinear optimization that uses a combination of linear and quadratic model steps and has separate, quasi-autonomous trust regions to control these. At least one subsequence generated by the algorithm is shown to be globally convergent to a critical point of the problem under modest assumptions. Our framework for trust-region radius updates is deliberately general. This is because we wished it to apply in the case of the current implementation of our evolving nonlinear programming code SLIQUE [1] as well as to cover its future evolution.

---

[3]This is true of the current implementation, but the description in [1] differs in some minor details.

We have not considered the ultimate convergence rate of the algorithm, nor its ability to identify the optimal active constraints in a finite number of iterations (these two aspects are most likely closely linked [8]), although we have strong numerical evidence to suggest that the latter does occur and that the convergence rate may thereafter be made to be superlinear. The study of these and other issues is ongoing.

Acknowledgment. The authors are grateful to two anonymous referees for their helpful comments on this paper.

REFERENCES

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

[2] C. M. Chin, *A New Trust Region Based SLP-Filter Algorithm Which Uses EQP Active-Set Strategy*, Ph.D. thesis, University of Dundee, Scotland, 2001.

[3] C. M. Chin and R. Fletcher, *On the global convergence of an SLP-filter algorithm that takes EQP steps*, Math. Program., 96 (2003), pp. 161–177.

[4] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Trust Region Methods*, MPS-SIAM Ser. Optim. 1, SIAM, Philadelphia, 2000.

[5] A. R. Conn and T. Pietrzykowski, *A penalty function method converging directly to a constrained optimum*, SIAM J. Numer. Anal., 14 (1977), pp. 348–375.

[6] R. Fletcher, *Non-differential optimization*, Chapter 14 in Practical Methods of Optimization: Constrained Optimization, Vol. 2, John Wiley and Sons, Chichester, New York, 1981.

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

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

[9] R. W. Gate, *Development of Algorithms for Solving Large Optimization Problems*, Ph.D. thesis, University of Dundee, Scotland, 2004.

[10] S. P. Han and O. L. Mangasarian, *Exact penalty functions in nonlinear programming*, Math. Program., 17 (1979), pp. 251–269.

[11] D. Q. Mayne and E. Polak, *Feasible directions algorithms for optimisation problems with equality and inequality constraints*, Math. Program., 11 (1976), pp. 67–80.

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

[13] E. Sainz de la Maza, *Nonlinear Programming Algorithms Based on ℓ_1 Linear Programming and Reduced Hessian Approximation*, Ph.D. thesis, University of Dundee, Scotland, 1987.

[14] R. A. Waltz, *Algorithms for Large-Scale Nonlinear Optimization*, Ph.D. thesis, Northwestern University, Evanston, IL, 2002.

[15] Y. Yuan, *Conditions for convergence of trust region algorithms for non-smooth optimization*, Math. Program., 31 (1985), pp. 220–228.

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