SIAM J. OPTIM.
Vol. 13, No. 3, pp. 635–659

GLOBAL CONVERGENCE OF A TRUST-REGION SQP-FILTER ALGORITHM FOR GENERAL NONLINEAR PROGRAMMING*

ROGER FLETCHER[†], NICHOLAS I. M. GOULD[‡], SVEN LEYFFER[†], PHILIPPE L. TOINT[§], AND ANDREAS WÄCHTER[¶]

Abstract. A trust-region SQP-filter algorithm of the type introduced by Fletcher and Leyffer [*Math. Program.*, 91 (2002), pp. 239–269] that decomposes the step into its normal and tangential components allows for an approximate solution of the quadratic subproblem and incorporates the safeguarding tests described in Fletcher, Leyffer, and Toint [*On the Global Convergence of an SLP-Filter Algorithm*, Technical Report 98/13, Department of Mathematics, University of Namur, Namur, Belgium, 1998; *On the Global Convergence of a Filter-SQP Algorithm*, Technical Report 00/15, Department of Mathematics, University of Namur, Namur, Belgium, 2000] is considered. It is proved that, under reasonable conditions and for every possible choice of the starting point, the sequence of iterates has at least one first-order critical accumulation point.

Key words. nonlinear optimization, sequential quadratic programming, filter methods, convergence theory

AMS subject classifications. 90C30, 65K05

PII. S1052623499357258

1. Introduction.

We analyze an algorithm for solving optimization problems where a smooth objective function is to be minimized subject to smooth nonlinear constraints. No convexity assumption is made. More formally, we consider the problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & c_{\mathcal{E}}(x) = 0, \\ & c_{\mathcal{I}}(x) \geq 0, \end{array} \tag{1.1}$$

where f is a twice continuously differentiable real valued function of the variables $x \in \mathbb{R}^n$ and $c_{\mathcal{E}}(x)$ and $c_{\mathcal{I}}(x)$ are twice continuously differentiable functions from $\mathbb{R}^n$ into $\mathbb{R}^m$ and from $\mathbb{R}^n$ into $\mathbb{R}^p$, respectively. Let $c(x)^T = (c_{\mathcal{E}}(x)^T \;\; c_{\mathcal{I}}(x)^T)$.

The class of algorithms that we discuss belongs to the class of trust-region methods and, more specifically, to that of *filter methods* introduced by Fletcher and Leyffer [18], in which the use of a penalty function, a common feature of the large majority of the algorithms for constrained optimization, is replaced by the introduction of a so-called filter.

A global convergence theory for an algorithm of this class is proposed by Fletcher, Leyffer, and Toint in [19], in which the objective function is locally approximated by a linear function, leading, at each iteration, to the (exact) solution of a linear program.

*Received by the editors June 14, 1999; accepted for publication (in revised form) March 12, 2002; published electronically November 6, 2002.
http://www.siam.org/journals/siopt/13-3/35725.html

[†] Department of Mathematics, University of Dundee, Dundee, Scotland (fletcher@maths.dundee.ac.uk, sleyffer@maths.dundee.ac.uk).

[‡] Computational Science and Engineering Department, Rutherford Appleton Laboratory, Chilton, Oxfordshire OX11 0QX, England (gould@rl.ac.uk).

[§] Department of Mathematics, University of Namur, Namur, Belgium (philippe.toint@fundp.ac.be).

[¶] Mathematical Sciences Department, IBM T. J. Watson Research Center, P. O. Box 218, Yorktown Heights, NY 10598. Current address: Department of Chemical Engineering, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213 (andreasw@andrew.cmu.edu).

This algorithm therefore mixes the use of the filter with sequential linear programming (SLP). This approach was generalized by the same authors in [20], where the objective function is approximated by a quadratic model, which results in a sequential quadratic programming (SQP) technique in which each quadratic program must be solved globally. In this paper, we again consider approximating the objective function by a quadratic model, but, at variance with the latter reference, the method discussed here does not require the global solution of the associated nonconvex quadratic programming (QP) subproblem, which is known to be a theoretically difficult process—it is known to be NP hard (see Murty and Kabadi [26]). The algorithm analyzed here also has a different mechanism for deciding on the compatibility of this subproblem and allows for an approximate subproblem solution.

2. A class of trust-region SQP-filter algorithms.

2.1. An approximate SQP framework. SQP methods are iterative. At a given iterate x_k, they implicitly apply Newton's method to solve (a local version of) the first-order necessary optimality conditions by solving the QP subproblem $\mathrm{QP}(x_k)$ given by

$$\begin{array}{ll} \text{minimize} & f_k + \langle g_k, s\rangle + \frac{1}{2}\langle s, H_k s\rangle \\ \text{subject to} & c_{\mathcal{E}}(x_k) + A_{\mathcal{E}}(x_k)s = 0, \\ & c_{\mathcal{I}}(x_k) + A_{\mathcal{I}}(x_k)s \geq 0, \end{array} \tag{2.1}$$

where $f_k = f(x_k)$, $g_k = g(x_k) \overset{\text{def}}{=} \nabla_x f(x_k)$, where $A_{\mathcal{E}}(x_k)$ and $A_{\mathcal{I}}(x_k)$ are the Jacobians of the constraint functions $c_{\mathcal{E}}$ and $c_{\mathcal{I}}$ at x_k, and where H_k is a symmetric matrix. We will not immediately be concerned about how H_k is obtained, but we will return to this point in section 3. Assuming that a suitable matrix H_k can be found, the solution of $\mathrm{QP}(x_k)$ then yields a step s_k. If $s_k = 0$, then x_k is first-order critical for problem (1.1).

Unfortunately, due to the locally convergent nature of Newton's iteration, the step s_k may not always be very good. One possible way to cope with this difficulty is to define an appropriate merit function whose value decreases with the goodness of s_k, which is where penalty functions typically play a role. A trust-region or a linesearch method is then applied to minimize this merit function, ensuring global convergence under reasonable assumptions. However, as one of our objectives is to avoid penalty functions (and the need to update the associated penalty parameter), we instead consider a trust-region approach that will not use any penalty function.[1] In such an approach, the objective function of $\mathrm{QP}(x_k)$ is intended to be only of local interest; that is, we restrict the step s_k in the norm to ensure that $x_k + s_k$ remains in a *trust-region* centered at x_k. In other words, we replace $\mathrm{QP}(x_k)$ by the subproblem $\mathrm{TRQP}(x_k, \Delta_k)$ given by

$$\begin{array}{ll} \text{minimize} & m_k(x_k + s) \\ \text{subject to} & c_{\mathcal{E}}(x_k) + A_{\mathcal{E}}(x_k)s = 0, \\ & c_{\mathcal{I}}(x_k) + A_{\mathcal{I}}(x_k)s \geq 0, \\ \text{and} & \|s\| \leq \Delta_k \end{array} \tag{2.2}$$

for some (positive) value of the *trust-region radius* Δ_k, where we have defined

$$m_k(x_k + s) = f_k + \langle g_k, s\rangle + \frac{1}{2}\langle s, H_k s\rangle \tag{2.3}$$

[1] Recently, Wächter and Biegler [31] have proposed a linesearch variant of the ideas described in this paper.

and where $\|\cdot\|$ denotes the Euclidean norm. This latter choice is purely for ease of exposition. We could equally use a family of iteration-dependent norms $\|\cdot\|_k$, so long as we require that all members of the family are uniformly equivalent to the Euclidean norm. The interested reader may verify that all subsequent developments can be adapted to this more general case by introducing the constants implied by this uniform equivalence wherever needed.

Remarkably, most early SQP algorithms assume that an exact local solution of $\mathrm{QP}(x_k)$ or $\mathrm{TRQP}(x_k, \Delta_k)$ is found, although attempts have been made by Dembo and Tulowitzki [8] and Murray and Prieto [25] to design conditions under which an approximate solution of the subproblem is acceptable. We revisit this issue in what follows, and start by noting that the step s_k may be viewed as the sum of two distinct components, a *normal step* n_k, such that $x_k + n_k$ satisfies the constraints of $\mathrm{TRQP}(x_k, \Delta_k)$, and a *tangential step* t_k, whose purpose is to obtain reduction of the objective function's model while continuing to satisfy those constraints. This framework is therefore similar in spirit to the composite-step SQP methods pioneered by Vardi [30], Byrd, Schnabel, and Shultz [5], and Omojokun [27], and later developed by several authors, including Biegler, Nocedal, and Schmid [1], El-Alem [12, 13], Byrd, Gilbert, and Nocedal [3], Byrd, Hribar, and Nocedal [4], Bielschowsky and Gomes [2], Liu and Yuan [23], and Lalee, Nocedal, and Plantenga [22]. More formally, we write

$$s_k = n_k + t_k \tag{2.4}$$

and assume that

$$c_{\mathcal{E}}(x_k) + A_{\mathcal{E}}(x_k) n_k = 0, \quad c_{\mathcal{I}}(x_k) + A_{\mathcal{I}}(x_k) n_k \geq 0, \tag{2.5}$$

$$\|s_k\| \leq \Delta_k, \tag{2.6}$$

and

$$c_{\mathcal{E}}(x_k) + A_{\mathcal{E}}(x_k) s_k = 0, \quad c_{\mathcal{I}}(x_k) + A_{\mathcal{I}}(x_k) s_k \geq 0. \tag{2.7}$$

Of course, this is a strong assumption, since in particular (2.5) or (2.6)/(2.7) may not have a solution. We shall return to this possibility shortly. Given our assumption, there are many ways to compute n_k and t_k. For instance, we could compute n_k from

$$n_k = P_k[x_k] - x_k, \tag{2.8}$$

where P_k is the orthogonal projector onto the feasible set of $\mathrm{QP}(x_k)$. In what follows, we do not make any specific choice for n_k, but we shall make the assumptions that n_k exists when the maximum violation of the nonlinear constraints at the kth iterate $\theta_k \stackrel{\text{def}}{=} \theta(x_k)$, where

$$\theta(x) = \max\left[0, \max_{i \in \mathcal{E}} |c_i(x)|, \max_{i \in \mathcal{I}} -c_i(x)\right] \tag{2.9}$$

is sufficiently small, and that n_k is then reasonably scaled with respect to the values of the constraints. In other words, we assume that

$$n_k \text{ exists and } \|n_k\| \leq \kappa_{\text{usc}} \theta_k \text{ whenever } \theta_k \leq \delta_n \tag{2.10}$$

for some constants $\kappa_{\text{usc}} > 0$ and $\delta_n > 0$. This assumption is also used by Dennis, El-Alem, and Maciel [9] and Dennis and Vicente [11] in the context of problems only involving equality constraints. We can interpret it in terms of the constraint functions themselves and the geometry of the boundary of the feasible set. For instance, if we define the linearized feasible set at x by

$$\mathcal{L}(x) \overset{\text{def}}{=} \{v \in \mathbb{R}^n \mid c_{\mathcal{E}}(x) + A_{\mathcal{E}}(x)(v-x) = 0,\;\; c_{\mathcal{I}}(x) + A_{\mathcal{I}}(x)(v-x) \geq 0\}$$

and assume that, at every limit point x_* of the sequence of iterates, the relative interior of the linearized constraints $\mathrm{ri}\{\mathcal{L}(x_*)\}$ is nonempty and that the active set settles, in that $\mathcal{A}(x_{k_i}) = \mathcal{A}(x_*)$ for sufficiently large k_i with $\lim_i x_{k_i} = x_*$, we know, by applying a continuity argument, that the feasible set of $\mathrm{QP}(x_k)$ is nonempty for such a k, which implies that P_k is well defined and that a normal step n_k of the form (2.8) exists. Furthermore, if the singular values of the Jacobian of constraints active at x_*, $A_{\mathcal{A}(x_*)}(x_*)$, are nonzero, those of $A_{\mathcal{A}(x_*)}(x_k)$ must be bounded away from zero by continuity in a neighborhood of x_*. Since only the constraints active at x_* can be active in a sufficiently small neighborhood of this limit point, this in turn guarantees that (2.10) holds for the normal step

$$-A^T_{\mathcal{A}(x_*)}(x_k)\big[A_{\mathcal{A}(x_*)}(x_k)A^T_{\mathcal{A}(x_*)}(x_k)\big]^{-1}c_{\mathcal{A}(x_*)}(x_k)$$

for all k sufficiently large, provided that the sequence of iterates remains bounded, because this latter assumption ensures that x_k must be arbitrarily close to a least one limit point of the sequence $\{x_k\}$ for such k. Thus we see that (2.10) does not impose conditions on the constraints or the normal step itself that are unduly restrictive.

Having defined the normal step, we are in position to use it if it falls within the trust-region, that is, if $\|n_k\| \leq \Delta_k$. In this case, we write

$$x_k^{\mathrm{N}} = x_k + n_k \tag{2.11}$$

and observe that n_k satisfies the constraints of $\mathrm{TRQP}(x_k, \Delta_k)$ and thus also of $\mathrm{QP}(x_k)$. It is crucial to note, at this stage, that such an n_k may fail to exist because the constraints of $\mathrm{QP}(x_k)$ may be incompatible, in which case P_k is undefined, or because all feasible points for $QP(x_k)$ may lie outside the trust-region.

Let us continue to consider the case where this problem does not arise, and a normal step n_k has been found with $\|n_k\| \leq \Delta_k$. We then have to find a tangential step t_k, starting from x_k^{N} and satisfying (2.6) and (2.7), with the aim of decreasing the value of the objective function. As always in trust-region methods, this is achieved by computing a step that produces a sufficient decrease in m_k, which is to say that we wish $m_k(x_k^{\mathrm{N}}) - m_k(x_k + s_k)$ to be "sufficiently large." Of course, this is only possible if the maximum size of t_k is not too small, which is to say that x_k^{N} is not too close to the trust-region boundary. We formalize this supposition by replacing our condition that $\|n_k\| \leq \Delta_k$ with the stronger requirement that

$$\|n_k\| \leq \kappa_\Delta \Delta_k \min[1, \kappa_\mu \Delta_k^\mu] \tag{2.12}$$

for some $\kappa_\Delta \in (0,1]$, some $\kappa_\mu > 0$, and some $\mu \in (0,1)$. If condition (2.12) does not hold, we assume that the computation of t_k is unlikely to produce a satisfactory decrease in m_k, and proceed just as if the feasible set of $\mathrm{TRQP}(x_k, \Delta_k)$ were empty. If n_k can be computed and (2.12) holds, we shall say that $\mathrm{TRQP}(x_k, \Delta_k)$ is *compatible.* In this case at least a sufficient model decrease seems possible, which we state in the form of a familiar Cauchy-point condition. In order to formalize what we mean, we recall that the feasible set of $\mathrm{QP}(x_k)$ is convex, and we can therefore introduce the measure

$$\chi_k = \left| \min_{\substack{A_{\mathcal{E}}(x_k)t=0 \\ c_{\mathcal{I}}(x_k)+A_{\mathcal{I}}(x_k)(n_k+t)\geq 0 \\ \|t\|\leq 1}} \langle g_k + H_k n_k, t\rangle \right| \tag{2.13}$$

(see Conn et al. [6]), which we will use to deduce first-order criticality for our problem (see Lemma 3.2). Note that this function is zero if the origin is a first-order critical point of the "tangential" problem

$$\begin{array}{ll} \text{minimize} & \langle g_k + H_k n_k, t\rangle + \frac{1}{2}\langle H_k t, t\rangle \\ \text{subject to} & A_{\mathcal{E}}(x_k)t = 0, \\ & c_{\mathcal{I}}(x_k) + A_{\mathcal{I}}(x_k)(n_k + t) \geq 0, \end{array} \tag{2.14}$$

which is, up to the constant term $\frac{1}{2}\langle n_k, H_k n_k\rangle$, equivalent to $\mathrm{QP}(x_k)$ with $s = n_k + t$. Our sufficient decrease condition is then to require that, whenever $\mathrm{TRQP}(x_k, \Delta_k)$ is compatible,

$$m_k(x_k^{\mathrm{N}}) - m_k(x_k^{\mathrm{N}} + t_k) \geq \kappa_{\mathrm{tmd}} \chi_k \min\left[\frac{\chi_k}{\beta_k}, \Delta_k\right] \tag{2.15}$$

for some constant $\kappa_{\mathrm{tmd}} > 0$, where $\beta_k = 1 + \|H_k\|$. We know from Toint [29] and Conn et al. [6] that this condition holds if the model reduction exceeds that which would be obtained at the generalized Cauchy point, that is, the point resulting from a backtracking curvilinear search along the projected gradient path from x_k^{N}, that is,

$$x_k(\alpha) = P_k[x_k^{\mathrm{N}} - \alpha \nabla_x m_k(x_k^{\mathrm{N}})].$$

This technique therefore provides an implementable algorithm for computing a step that satisfies (2.15) (see Gould, Hribar, and Nocedal [21] for an example in the case where $c(x) = c_{\mathcal{E}}(x)$, or Toint [29] and Moré and Toraldo [24] for the case of bound constraints), but, of course, reduction of m_k beyond that imposed by (2.15) is often possible and desirable if fast convergence is sought. Also note that the minimization problem of the right-hand side of (2.13) would reduce to a linear programming problem if we had chosen to use a polyhedral norm in its definition at iteration k.

Let us now return to the case where $\mathrm{TRQP}(x_k, \Delta_k)$ is not compatible, that is, when the feasible set determined by the constraints of $\mathrm{QP}(x_k)$ is empty, or the freedom left to reduce m_k within the trust-region is too small in the sense that (2.12) fails. In this situation, solving $\mathrm{TRQP}(x_k, \Delta_k)$ is most likely pointless, and we must consider an alternative. We base this on the intuitive observation that, if $\theta(x_k)$ is sufficiently small and the true nonlinear constraints are locally compatible, the linearized constraints should also be compatible, since they approximate the nonlinear constraints (locally) correctly. Furthermore, the feasible region for the linearized constraints should then be close enough to x_k for there to be some room to reduce m_k, at least if Δ_k is large enough. If the nonlinear constraints are locally incompatible, we have to find a neighborhood where this is not the case, since the problem (1.1) does not make sense in the current one. We thus rely on a *restoration procedure*, whose aim is to produce a new point $x_k + r_k$ for which $\mathrm{TRQP}(x_k + r_k, \Delta_{k+1})$ is compatible for some $\Delta_{k+1} > 0$—we will actually need another condition which we will discuss shortly.

The idea of the restoration procedure is to (approximately) solve

$$\min_{x\in\mathbb{R}^n} \theta(x), \tag{2.16}$$

perhaps starting from x_k, the current iterate. This is a nonsmooth problem, but we know that there exist methods, possibly of trust-region type (such as that suggested by Yuan [32]), which can be successfully applied to solve it. Thus we will not describe the restoration procedure in detail. Note that we have chosen here to reduce the infinity norm of the constraint violation, but we could equally well consider other norms, such as ℓ_1 or ℓ_2, in which case the methods of Fletcher and Leyffer [17] or of El-Hallabi and Tapia [14] and Dennis, El-Alem, and Williamson [10], respectively, can be considered. Of course, this technique only guarantees convergence to a first-order critical point of the chosen measure of constraint violation, which means that, in fact, the restoration procedure may fail as this critical point may not be feasible for the constraints of (1.1). However, even in this case, the result of the procedure is of interest because it typically produces a local minimizer of $\theta(x)$, or of whatever other measure of constraint violation we choose for the restoration, yielding a point of locally least infeasibility.

There is no easy way to circumvent this drawback, as it is known that finding a feasible point or proving that no such point exists is a global optimization problem and can be as difficult as the optimization problem (1.1) itself. We therefore accept two possible outcomes of the restoration procedure: either the procedure fails in that it does not produce a sequence of iterates converging to feasibility, or a point $x_k + r_k$ is produced such that $\theta(x_k + r_k)$ is as small as we wish. We will shortly see that this is all we need.

2.2. The notion of a filter. Having computed a step $s_k = n_k + t_k$ (or r_k), we still need to decide whether the trial point $x_k + s_k$ (or $x_k + r_k$) is any better than x_k as an approximate solution to our original problem (1.1). We shall use a concept borrowed from multicriteria optimization. We say that a point x_1 *dominates* a point x_2 whenever

$$\theta(x_1) \le \theta(x_2) \;\text{ and }\; f(x_1) \le f(x_2).$$

Thus, if iterate x_k dominates iterate x_j, the latter is of no real interest to us since x_k is at least as good as x_j on account of both feasibility and optimality. All we need to do now is to remember iterates that are not dominated by any other iterates using a structure called a filter. A *filter* is a list $\mathcal{F}$ of pairs of the form (θ_i, f_i) such that either

$$\theta_i < \theta_j \;\text{ or }\; f_i < f_j$$

for $i \neq j$. We thus aim to accept a new iterate x_i only if it is not dominated by any other iterate in the filter. In the vocabulary of multicriteria optimization, this amounts to building elements of the efficient frontier associated with the bicriteria problem of reducing infeasibility and the objective function value.

Figure 2.1 illustrates the concept of a filter by showing the pairs (θ_k, f_k) as black dots in the (θ, f)-space. Each such pair is called the (θ, f)-pair associated with x_k. The lines radiating from each (θ, f)-pair indicate that any iterate whose associated (θ, f)-pair occurs above and to the right of that of a given filter point is dominated by this (θ, f)-pair.

While the idea of not accepting dominated trial points is simple and elegant, it needs to be refined a little in order to provide an efficient algorithmic tool. In

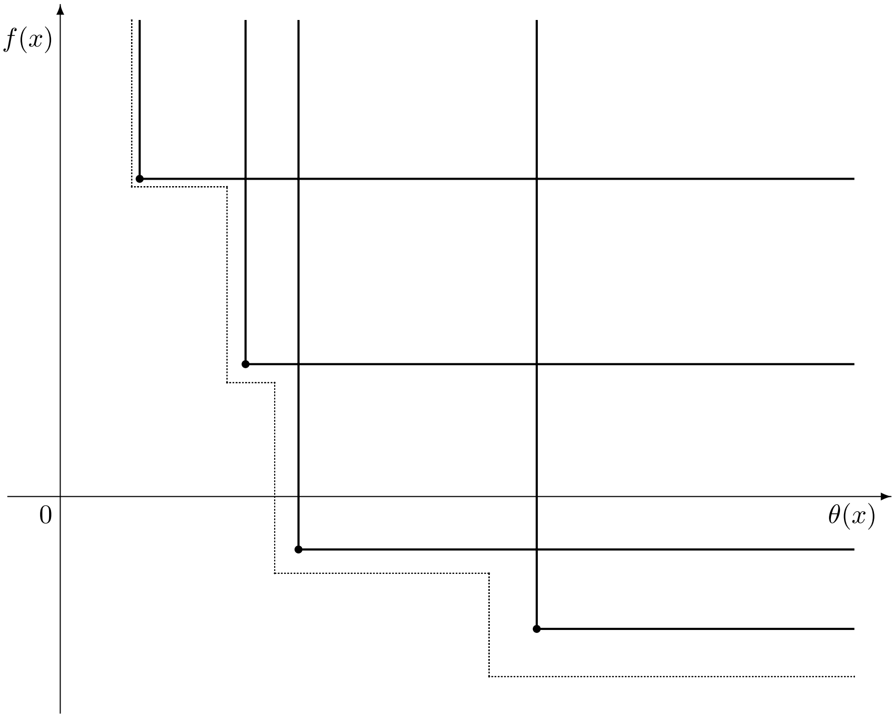

FIG. 2.1. *A filter with four pairs.*

particular, we do not wish to accept $x_k + s_k$ if its (θ, f)-pair is arbitrarily close to that of x_k or that of a point already in the filter. Thus we set a small "margin" around the border of the dominated part of the (θ, f)-space in which we shall also reject trial points. Formally, we say that a point x is *acceptable for the filter* if and only if

$$\theta(x) \leq (1-\gamma_\theta)\theta_j \;\text{ or }\; f(x) \leq f_j - \gamma_\theta \theta_j \;\text{ for all }\; (\theta_j, f_j) \in \mathcal{F} \tag{2.17}$$

for some $\gamma_\theta \in (0,1)$. In Figure 2.1, the set of acceptable points corresponds to the set of (θ, f)-pairs below the thin line. We also say that x is "acceptable for the filter and x_k" if (2.17) holds with $\mathcal{F}$ replaced by $\mathcal{F} \cup (\theta_k, f_k)$. We thus consider moving from x_k to $x_k + s_k$ only if $x_k + s_k$ is acceptable for the filter and x_k.

As the algorithm progresses, we may want to *add a (θ, f)-pair to the filter.* If an iterate x_k is acceptable for $\mathcal{F}$, we do this by adding the pair (θ_k, f_k) to the filter and by removing from it every other pair (θ_j, f_j) such that both

$$\theta_j \geq \theta_k \;\text{ and }\; f_j - \gamma_\theta \theta_j \geq f_k - \gamma_\theta \theta_k. \tag{2.18}$$

Only entries whose *envelope* is dominated by a new entry are thus removed from the filter. As a consequence, the margin of the filter never decreases, and it can be shown that, for all infinite subsequences of points added to the filter, $\lim \theta_{k_i} = 0$ (see Lemma 3.3). We also refer to this operation as "adding x_k to the filter," although, strictly speaking, it is the (θ, f)-pair which is added.

We conclude this section by noting that, if a point x_k is in the filter or is acceptable for the filter, then any other point x such that

$$\theta(x) \leq (1-\gamma_\theta)\theta_k \;\text{ and }\; f(x) \leq f_k - \gamma_\theta \theta_k$$

is also acceptable for the filter and x_k.

2.3. An SQP-filter algorithm. We have now discussed the main ingredients of the class of algorithms we wish to consider, and we are now ready to define it formally as Algorithm 2.1 below. A flowchart of the algorithm is given as an appendix; see Figure A.1.

Algorithm 2.1: SQP-Filter Algorithm.

Step 0: Initialization. Let an initial point x_0, an initial trust-region radius $\Delta_0 > 0$, and an initial symmetric matrix H_0 be given, as well as constants $0 < \gamma_0 < \gamma_1 \leq 1 \leq \gamma_2$, $0 < \eta_1 \leq \eta_2 < 1$, $\gamma_\theta \in (0,1)$, $\kappa_\theta \in (0,1)$, $\kappa_\Delta \in (0,1]$, $\kappa_\mu > 0$, $\mu \in (0,1)$, $\psi > 1/(1+\mu)$, and $\kappa_{\text{tmd}} \in (0,1]$. Compute $f(x_0)$ and $c(x_0)$. Set $\mathcal{F} = \emptyset$ and $k = 0$.

Step 1: Test for optimality. If $\theta_k = \chi_k = 0$, stop.

Step 2: Ensure compatibility. Attempt to compute a step n_k. If TRQP (x_k, Δ_k) is compatible, go to Step 3. Otherwise, include x_k in the filter and compute a restoration step r_k for which TRQP$(x_k + r_k, \Delta_{k+1})$ is compatible for some $\Delta_{k+1} > 0$, and $x_k + r_k$ is acceptable for the filter. If this proves impossible, stop. Otherwise, define $x_{k+1} = x_k + r_k$ and go to Step 7.

Step 3: Determine a trial step. Compute a step t_k and set $s_k = n_k + t_k$.

Step 4: Tests to accept the trial step.

- Evaluate $c(x_k + s_k)$ and $f(x_k + s_k)$.
- If $x_k + s_k$ is not acceptable for the filter and x_k, set $x_{k+1} = x_k$, choose $\Delta_{k+1} \in [\gamma_0 \Delta_k, \gamma_1 \Delta_k]$, set $n_{k+1} = n_k$, increment k by one, and go to Step 2.
- If

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_\theta \theta_k^\psi \tag{2.19}$$

and

$$\rho_k \overset{\text{def}}{=} \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)} < \eta_1, \tag{2.20}$$

again set $x_{k+1} = x_k$, choose $\Delta_{k+1} \in [\gamma_0 \Delta_k, \gamma_1 \Delta_k]$, set $n_{k+1} = n_k$, increment k by one, and go to Step 2.

Step 5: Test to include the current iterate in the filter. If (2.19) fails, include x_k in the filter $\mathcal{F}$.

Step 6: Move to the new iterate. Set $x_{k+1} = x_k + s_k$ and choose Δ_{k+1} such that

$$\Delta_{k+1} \in [\Delta_k, \gamma_2 \Delta_k] \text{ if } \rho_k \geq \eta_2 \text{ and (2.19) holds.}$$

Step 7: Update the Hessian approximation. Determine H_{k+1}. Increment k by one and go to Step 1.

As in Fletcher and Leyffer [18, 17], one may choose $\psi = 2$. (Note that the choice $\psi = 1$ is always possible because $\mu > 0$.) Reasonable values for the constants might then be

$$\gamma_0 = 0.1, \quad \gamma_1 = 0.5, \quad \gamma_2 = 2, \quad \eta_1 = 0.01, \quad \eta_2 = 0.9,$$
$$\gamma_\theta = 10^{-4}, \quad \kappa_\Delta = 0.7, \quad \kappa_\mu = 100, \quad \mu = 0.01, \quad \kappa_\theta = 10^{-4}, \text{ and } \kappa_{\text{tmd}} = 0.01,$$

but it is too early to know if these are even close to the best possible choices.

Observe first that, by construction, every iterate x_k must be acceptable for the filter at the beginning of iteration k, irrespective of the possibility that it might be added to the filter later. Also note that the restoration step r_k cannot be zero, that is, restoration cannot simply entail enlarging the trust-region radius to ensure (2.12), even if n_k exists. This is because x_k is added to the filter before r_k is computed, and $x_k + r_k$ must be acceptable for the filter which now contains x_k. Also note that the restoration procedure cannot be applied on two successive iterations, since the iterate $x_k + r_k$ produced by the first of these iterations leads to a compatible $\mathrm{TRQP}(x_{k+1}, \Delta_{k+1})$ and is acceptable for the filter.

For the restoration procedure in Step 2 to succeed, we have to evaluate whether $\mathrm{TRQP}(x_k + r_k, \Delta_{k+1})$ is compatible for a suitable value of Δ_{k+1}. This requires that a suitable normal step be computed which successfully passes the test (2.12). Of course, once this is achieved, this normal step may be reused at iteration $k+1$. Thus we shall require that the normal step calculated to verify compatibility of $\mathrm{TRQP}(x_k+r_k, \Delta_{k+1})$ should actually be used as n_{k+1}.

As it stands, the algorithm is not specific about how to choose Δ_{k+1} during a restoration iteration. On one hand, there is an advantage to choosing a large Δ_{k+1}, since this allows a large step and, one hopes, good progress. On the other hand, it may be unwise to choose it to be too large, as this may possibly result in a large number of unsuccessful iterations, during which the radius is reduced, before the algorithm can make any progress. A possible choice might be to restart from the radius obtained during the restoration iteration itself, if it uses a trust-region method. Reasonable alternatives would be to use the average radius observed during past successful iterations, or to apply the internal doubling strategy of Byrd, Schnabel, and Shultz [5] to increase the new radius, or even to consider the technique described by Sartenaer [28]. However, we recognize that numerical experience with the algorithm is too limited at this stage to make definite recommendations.

The role of condition (2.19) may be interpreted as follows. If this condition fails, then one may think that the constraint violation is significant and that one should aim to improve on this situation in the future by inserting the current point into the filter. Fletcher, Leyffer, and Toint [19] use the term of "θ-step" in such circumstances to indicate that the main preoccupation is to improve feasibility. On the other hand, if condition (2.19) holds, then the reduction in the objective function predicted by the model is more significant than the current constraint violation, and it is thus appealing to let the algorithm behave as if it were unconstrained. Fletcher and Leyffer [18] use the term "f-step" to denote the step generated, in order to reflect the dominant role of the objective function f. In this case, it is important that the predicted decrease in the model be realized by the actual decrease in the function, which is why we also require that (2.20) not hold. In particular, if the iterate x_k is feasible, then (2.10) implies that $x_k = x_k^{\mathrm{N}}$, and we obtain that

$$\kappa_\theta \theta_k^\psi = 0 \le m_k(x_k^{\mathrm{N}}) - m_k(x_k + s_k) = m_k(x_k) - m_k(x_k + s_k). \tag{2.21}$$

As a consequence, the filter mechanism is irrelevant if all iterates are feasible, and the algorithm reduces to a classical unconstrained trust-region method. Another consequence of (2.21) is that *no feasible iterate is ever included in the filter*, which is crucial in allowing finite termination of the restoration procedure. Indeed, if the restoration procedure is required at iteration k of the filter algorithm and produces a sequence of points $\{x_{k,j}\}$ converging to feasibility, there must be an iterate $x_{k,j}$ for which

$$\theta_{k,j} \stackrel{\text{def}}{=} \theta(x_{k,j}) \leq \min\left[(1-\gamma_\theta)\theta_k^{\min}, \frac{\kappa_\Delta}{\kappa_{\text{usc}}}\Delta_{k+1}\min[1,\kappa_\mu\Delta_{k+1}^\mu]\right]$$

for any given $\Delta_{k+1} > 0$, where

$$\theta_k^{\min} = \min_{i\in\mathcal{Z},\, i\leq k} \theta_i > 0$$

and

$$\mathcal{Z} = \{k \mid x_k \text{ is added to the filter}\}.$$

Moreover, $\theta_{k,j}$ must eventually be small enough to ensure, using our assumption on the normal step, the existence of a normal step $n_{k,j}$ from $x_{k,j}$. In other words, the restoration iteration must eventually find an iterate $x_{k,j}$ which is acceptable for the filter and for which the normal step exists and satisfies (2.12), i.e., an iterate x_j which is both acceptable and compatible. As a consequence, the restoration procedure will terminate in a finite number of steps, and the filter algorithm may then proceed. Note that the restoration step may not terminate in a finite number of iterations if we do not assume the existence of the normal step when the constraint violation is small enough, even if this violation converges to zero (see Fletcher, Leyffer, and Toint [19], for an example).

Notice also that (2.19) ensures that the denominator of ρ_k in (2.20) will be strictly positive whenever θ_k is. If $\theta_k = 0$, then $x_k = x_k^{\text{N}}$, and the denominator of (2.20) will be strictly positive unless x_k is a first-order critical point because of (2.15).

The reader may have observed that Step 6 allows a relatively wide choice of the new trust-region radius Δ_{k+1}. While the stated conditions appear to be sufficient for the theory developed below, one must obviously be more specific in practice. For instance, one may wish to distinguish, at this point in the algorithm, the cases where (2.19) fails or holds. If (2.19) fails, the main effect of the current iteration is not to reduce the model (which makes the value of ρ_k essentially irrelevant), but rather to reduce the constraint violation (which is taken care of by inserting the current iterate into the filter at Step 5). In this case, Step 6 imposes no further restriction on Δ_{k+1}. In practice, it may be reasonable not to reduce the trust-region radius, because this might cause too small steps towards feasibility or an unnecessary restoration phase. However, there is no compelling reason to increase the radius either, given the compatibility of TRQP(x_k,Δ_k). A reasonable strategy might then be to choose $\Delta_{k+1} = \Delta_k$. If, on the other hand, (2.19) holds, the emphasis of the iteration is then on reducing the objective function, a case akin to unconstrained minimization. Thus a more detailed rule of the type

$$\Delta_{k+1} \in \begin{cases} [\gamma_1\Delta_k, \gamma_2\Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\Delta_k, \gamma_2\Delta_k] & \text{if } \rho_k \geq \eta_2 \end{cases}$$

seems reasonable in these circumstances.

Finally, we recognize that (2.15) may be difficult to verify in practice, since it may be expensive to compute x_k^{N} and P_k when the dimension of the problem is large.

3. Convergence to first-order critical points. We now prove that our algorithm generates a globally convergent sequence of iterates. In the following analysis, we concentrate on the case in which the restoration iteration always succeeds. If this is not the case, then it usually follows that the restoration phase has converged to an approximate solution of the feasibility problem (2.16) and we can conclude that (1.1) is locally inconsistent. For the purpose of our analysis, we shall consider

$$\mathcal{S} = \{k \mid x_{k+1} = x_k + s_k\},$$

the set of (indices of) successful iterations, and

$$\mathcal{R} = \{k \mid n_k \text{ does not satisfy (2.10)} \quad \text{or} \quad \|n_k\| > \kappa_\Delta \Delta_k \min[1, \kappa_\mu \Delta_k^\mu]\},$$

the set of *restoration* iterations. In order to obtain our global convergence result, we will use the following assumptions.

AS1. f and the constraint functions $c_{\mathcal{E}}$ and $c_{\mathcal{I}}$ are twice continuously differentiable.

AS2. There exists $\kappa_{\text{umh}} > 1$ such that

$$\|H_k\| \le \kappa_{\text{umh}} - 1 \text{ for all } k.$$

AS3. The iterates $\{x_k\}$ remain in a closed, bounded domain $X \subset \mathbb{R}^n$.

If, for example, H_k is chosen as the Hessian of the Lagrangian function

$$\ell(x, y) = f(x) + \langle y_{\mathcal{E}}, c_{\mathcal{E}}(x)\rangle + \langle y_{\mathcal{I}}, c_{\mathcal{I}}(x)\rangle$$

at x_k, in that

$$H_k = \nabla_{xx} f(x_k) + \sum_{i \in \mathcal{E} \cup \mathcal{I}} [y_k]_i \nabla_{xx} c_i(x_k), \tag{3.1}$$

where $[y_k]_i$ denotes the ith component of the vector of Lagrange multipliers $y_k^T = (y_{\mathcal{E},k}^T \; y_{\mathcal{I},k}^T)$, then we see from AS1 and AS3 that AS2 is satisfied when these multipliers remain bounded. The same is true if the Hessian matrices in (3.1) are replaced by bounded approximations.

A first immediate consequence of AS1–AS3 is that there exists a constant $\kappa_{\text{ubh}} > 1$ such that, for all k,

$$|f(x_k + s_k) - m_k(x_k + s_k)| \le \kappa_{\text{ubh}} \Delta_k^2. \tag{3.2}$$

A proof of this property, based on Taylor expansion, may be found, for instance, in Toint [29]. A second important consequence of our assumptions is that AS1 and AS3 together directly ensure that, for all k,

$$f^{\min} \le f(x_k) \text{ and } 0 \le \theta_k \le \theta^{\max} \tag{3.3}$$

for some constants $f^{\min}$ and $\theta^{\max} > 0$. Thus the part of the (θ, f)-space in which the (θ, f)-pairs associated with the filter iterates lie is restricted to the rectangle

$$\mathcal{A}_0 = [0, \theta^{\max}] \times [f^{\min}, \infty].$$

We also note the following simple consequence of (2.10) and AS3.

Lemma 3.1. *Suppose that Algorithm* 2.1 *is applied to problem* (1.1). *Suppose also that* (2.10) *and* AS3 *hold and that*

$$\theta_k \le \delta_n.$$

Then there exists a constant $\kappa_{\rm lsc} > 0$ *independent of* k *such that*

$$\kappa_{lsc}\theta_k \leq \|n_k\|. \tag{3.4}$$

Proof. First define

$$\mathcal{V}_k \stackrel{\rm def}{=} \{j \in \mathcal{E} \mid \theta_k = |c_j(x_k)|\} \bigcup \{j \in \mathcal{I} \mid \theta_k = -c_j(x_k)\},$$

which is the subset of most-violated constraints. From the definitions of θ_k in (2.9) and of the normal step in (2.5) we obtain, using the Cauchy–Schwarz inequality, that

$$\theta_k \leq |\langle \nabla_x c_j(x_k), n_k\rangle| \leq \|\nabla_x c_j(x_k)\| \, \|n_k\| \tag{3.5}$$

for all $j \in \mathcal{V}_k$. But AS3 ensures that there exists a constant $\kappa_{\rm lsc} > 0$ such that

$$\max_{j\in\mathcal{E}\cup\mathcal{I}} \max_{x\in X} \|\nabla_x c_j(x)\| \stackrel{\rm def}{=} \frac{1}{\kappa_{\rm lsc}}.$$

We then obtain the desired conclusion by substituting this bound into (3.5). □

Our assumptions and the definition of χ_k in (2.13) ensure that θ_k and χ_k can be used (together) to measure criticality for problem (1.1).

LEMMA 3.2. *Suppose that Algorithm* 2.1 *is applied to problem* (1.1) *and that finite termination does not occur. Suppose also that* AS1 *and* AS3 *hold, and that there exists a subsequence* $\{k_i\}$ *such that, for any* i, $k_i \notin \mathcal{R}$ *with*

$$\lim_{i\to\infty} \chi_{k_i} = 0 \;\; and \;\; \lim_{i\to\infty} \theta_{k_i} = 0. \tag{3.6}$$

Then every limit point of the subsequence $\{x_{k_i}\}$ *is a first-order critical point for problem* (1.1).

Proof. Consider x_*, a limit point of the sequence $\{x_{k_i}\}$, whose existence is ensured by AS3, and assume that $\{k_\ell\} \subseteq \{k_i\}$ is the index set of a subsequence such that $\{x_{k_\ell}\}$ converges to x_*. The fact that $k_\ell \notin \mathcal{R}$ implies that n_{k_ℓ} satisfies (2.10) for sufficiently large ℓ and converges to zero, because $\{\theta_{k_\ell}\}$ converges to zero and the second part of this condition. As a consequence, we deduce from (2.11) that $\{x^{\rm N}_{k_\ell}\}$ also converges to x_*. Since the minimization problem occurring in the definition of χ_{k_ℓ} (in (2.13)) is convex, we then obtain from classical perturbation theory (see, for instance, Fiacco [15, pp. 14–17], AS1, and the first part of (3.6) that

$$\left| \min_{\substack{A_{\mathcal{E}}(x_*)t=0\\ c_{\mathcal{I}}(x_*)+A_{\mathcal{I}}(x_*)t\geq 0\\ \|t\|\leq 1}} \langle g_*, t\rangle \right| = 0.$$

This in turn guarantees that x_* is first-order critical for problem (1.1). □

We start our analysis by examining what happens when an infinite number of iterates (that is, their (θ, f)-pairs) are added to the filter.

LEMMA 3.3. *Suppose that Algorithm* 2.1 *is applied to problem* (1.1) *and that finite termination does not occur. Suppose that* AS1 *and* AS3 *hold and that* $|\mathcal{Z}| = \infty$. *Then*

$$\lim_{\substack{k\to\infty\\ k\in\mathcal{Z}}} \theta_k = 0.$$

Proof. Suppose, for the purpose of obtaining a contradiction, that there exists an infinite subsequence $\{k_i\} \subseteq \mathcal{Z}$ such that

$$\theta_{k_i} \geq \epsilon \tag{3.7}$$

for all i and for some $\epsilon > 0$. At each iteration k_i, the (θ, f)-pair associated with x_{k_i}, that is (θ_{k_i}, f_{k_i}), is added to the filter. This means that no other (θ, f)-pair can be added to the filter at a later stage within the square

$$[\theta_{k_i} - \gamma_\theta \epsilon, \theta_{k_i}] \times [f_{k_i} - \gamma_\theta \epsilon, f_{k_i}]$$

or with the intersection of this square with $\mathcal{A}_0$. Note that this holds, even if (θ_{k_i}, f_{k_i}) is later removed from the filter, since the rule for removing entries, (2.18), ensures that the envelope never shrinks. Now observe that the area of each of these squares is $\gamma_\theta^2 \epsilon^2$. As a consequence, the set $\mathcal{A}_0 \cap \{(\theta, f) | f \leq \kappa_f\}$ is completely covered by at most a finite number of such squares, for any choice of $\kappa_f \geq f^{\min}$. Since the pairs (θ_{k_i}, f_{k_i}) keep on being added to the filter, this implies that f_{k_i} tends to infinity when i tends to infinity. Let us assume, without loss of generality, that $f_{k_{i+1}} \geq f_{k_i}$ for all i sufficiently large. But (2.17) and (3.7) then imply that

$$\theta_{k_{i+1}} \leq (1 - \gamma_\theta)\theta_{k_i} \leq \theta_{k_i} - \gamma_\theta \epsilon,$$

and therefore that θ_{k_i} converges to zero, which contradicts (3.7). Hence this latter assumption is impossssible and the conclusion follows. □

We next examine the size of the constraint violation before and after an iteration where restoration did not occur.

LEMMA 3.4. *Suppose that Algorithm* 2.1 *is applied to problem* (1.1) *and that finite termination does not occur. Suppose also that* AS1 *and* AS3 *hold, that* $k \notin \mathcal{R}$, *and that* n_k *satisfies* (3.4). *Then*

$$\theta_k \leq \kappa_{\text{ubt}} \Delta_k^{1+\mu} \tag{3.8}$$

and

$$\theta(x_k + s_k) \leq \kappa_{\text{ubt}} \Delta_k^2 \tag{3.9}$$

for some constant $\kappa_{\text{ubt}} \geq 0$.

Proof. Since $k \notin \mathcal{R}$, we have from (3.4) and (2.12) that

$$\kappa_{\text{lsc}} \theta_k \leq \|n_k\| \leq \kappa_\Delta \kappa_\mu \Delta_k^{1+\mu}, \tag{3.10}$$

which gives (3.8). Now, the ith constraint function at $x_k + s_k$ can be expressed as

$$c_i(x_k + s_k) = c_i(x_k) + \langle e_i, A_k s_k \rangle + \frac{1}{2} \langle s_k, \nabla_{xx} c_i(\xi_k) s_k \rangle$$

for $i \in \mathcal{E} \cup \mathcal{I}$, where we have used AS1 and the mean-value theorem and where ξ_k belongs to the segment $[x_k, x_k + s_k]$. Using AS3, we may bound the Hessian of the constraint functions, and we obtain from (2.7), the Cauchy–Schwarz inequality, and (2.6) that

$$|c_i(x_k + s_k)| \leq \frac{1}{2} \max_{x \in X} \|\nabla_{xx} c_i(x)\| \, \|s_k\|^2 \leq \kappa_1 \Delta_k^2$$

if $i \in \mathcal{E}$, or

$$-c_i(x_k + s_k) \le \frac{1}{2} \max_{x \in X} \|\nabla_{xx} c_i(x)\| \, \|s_k\|^2 \le \kappa_1 \Delta_k^2$$

if $i \in \mathcal{I}$, where we have defined

$$\kappa_1 \overset{\rm def}{=} \frac{1}{2} \max_{i \in \mathcal{E} \cup \mathcal{I}} \max_{x \in X} \|\nabla_{xx} c_i(x)\|.$$

This gives the desired bound with

$$\kappa_{\rm ubt} = \max[\kappa_1, \kappa_\Delta \kappa_\mu / \kappa_{\rm lsc}]. \qquad \square$$

We next assess the model decrease when the trust-region radius is sufficiently small.

Lemma 3.5. *Suppose that Algorithm* 2.1 *is applied to problem* (1.1) *and that finite termination does not occur. Suppose also that* AS1–AS3, (2.12), *and* (2.15) *hold, that* $k \notin \mathcal{R}$, *that*

$$\chi_k \ge \epsilon \tag{3.11}$$

for some $\epsilon > 0$, *and that*

$$\Delta_k \le \min\left[\frac{\epsilon}{\kappa_{umh}}, \left(2\frac{\kappa_{ubg}}{\kappa_{umh}\kappa_\Delta\kappa_\mu}\right)^{\frac{1}{1+\mu}}, \left(\frac{\kappa_{tmd}\epsilon}{4\kappa_{ubg}\kappa_\Delta\kappa_\mu}\right)^{\frac{1}{\mu}}\right] \overset{\rm def}{=} \delta_m, \tag{3.12}$$

where $\kappa_{ubg} \overset{\rm def}{=} \max_{x \in X} \|\nabla_x f(x)\|$. *Then*

$$m_k(x_k) - m_k(x_k + s_k) \ge \frac{1}{2}\kappa_{tmd}\epsilon\Delta_k.$$

Proof. We first note that, by (2.15), AS2, (3.11), and (3.12),

$$m_k(x_k^{\rm N}) - m_k(x_k + s_k) \ge \kappa_{\rm tmd}\chi_k \min\left[\frac{\chi_k}{\kappa_{\rm umh}}, \Delta_k\right] \ge \kappa_{\rm tmd}\epsilon\Delta_k. \tag{3.13}$$

Now

$$m_k(x_k^{\rm N}) = m_k(x_k) + \langle g_k, n_k\rangle + \frac{1}{2}\langle n_k, H_k n_k\rangle,$$

and therefore, using the Cauchy–Schwarz inequality, AS2, (2.12), and (3.12),

$$\begin{aligned}
|m_k(x_k) - m_k(x_k^{\rm N})| &\le \|n_k\| \, \|g_k\| + \tfrac{1}{2}\|H_k\| \, \|n_k\|^2 \\
&\le \kappa_{\rm ubg}\|n_k\| + \tfrac{1}{2}\kappa_{\rm umh}\|n_k\|^2 \\
&\le \kappa_{\rm ubg}\kappa_\Delta\kappa_\mu\Delta_k^{1+\mu} + \tfrac{1}{2}\kappa_{\rm umh}\kappa_\Delta^2\kappa_\mu^2\Delta_k^{2(1+\mu)} \\
&\le 2\kappa_{\rm ubg}\kappa_\Delta\kappa_\mu\Delta_k^{1+\mu} \\
&\le \tfrac{1}{2}\kappa_{\rm tmd}\epsilon\Delta_k.
\end{aligned}$$

We thus conclude from this last inequality and (3.13) that the desired conclusion holds. $\square$

We continue our analysis by showing, as the reader has grown to expect, that iterations have to be very successful when the trust-region radius is sufficiently small.

Lemma 3.6. *Suppose that Algorithm* 2.1 *is applied to problem* (1.1) *and that finite termination does not occur. Suppose also that* AS1–AS3*,* (2.15)*, and* (3.11) *hold, that* $k \notin \mathcal{R}$*, and that*

$$\Delta_k \le \min\left[\delta_m, \frac{(1-\eta_2)\kappa_{tmd}\epsilon}{2\kappa_{ubh}}\right] \overset{\rm def}{=} \delta_\rho. \tag{3.14}$$

Then

$$\rho_k \ge \eta_2.$$

Proof. Using the definition of ρ_k in (2.20), (3.2), Lemma 3.5, and (3.14), we find that

$$|\rho_k - 1| = \frac{|f(x_k+s_k) - m_k(x_k+s_k)|}{|m_k(x_k) - m_k(x_k+s_k)|} \le \frac{\kappa_{\rm ubh}\Delta_k^2}{\frac{1}{2}\kappa_{\rm tmd}\epsilon\Delta_k} \le 1-\eta_2,$$

from which the conclusion immediately follows. □

Note that this proof could easily be extended if the definition of ρ_k in (2.20) were altered to be of the form

$$\rho_k \overset{\rm def}{=} \frac{f(x_k) - f(x_k+s_k) + \Theta_k}{m_k(x_k) - m_k(x_k+s_k)}, \tag{3.15}$$

provided that Θ_k is bounded above by a multiple of Δ_k^2. We will comment in section 4 why such a modification might be of interest (see also section 10.4.3 of Conn, Gould, and Toint [7]).

Now, we also show that the test (2.19) will always be satisfied when the trust-region radius is sufficiently small.

Lemma 3.7. *Suppose that Algorithm* 2.1 *is applied to problem* (1.1) *and that finite termination does not occur. Suppose also that* AS1–AS3*,* (2.12)*,* (2.15)*, and* (3.11) *hold, that* $k \notin \mathcal{R}$*, that* n_k *satisfies* (3.4)*, and that*

$$\Delta_k \le \min\left[\delta_m, \left(\frac{\kappa_{tmd}\epsilon}{2\kappa_\theta\kappa_{ubt}^\psi}\right)^{\frac{1}{\psi(1+\mu)-1}}\right] \overset{\rm def}{=} \delta_f. \tag{3.16}$$

Then

$$m_k(x_k) - m_k(x_k+s_k) \ge \kappa_\theta\theta_k^\psi.$$

Proof. This directly results from the inequalities

$$\kappa_\theta\theta_k^\psi \le \kappa_\theta\kappa_{\rm ubt}^\psi\Delta_k^{\psi(1+\mu)} \le \frac{1}{2}\kappa_{\rm tmd}\epsilon\Delta_k \le m_k(x_k) - m_k(x_k+s_k),$$

where we have successively used Lemma 3.4, (3.16), and Lemma 3.5. □

We may also guarantee a decrease in the objective function, large enough to ensure that the trial point is acceptable with respect to the (θ, f)-pair associated with x_k, so long as the constraint violation is itself sufficiently small.

Lemma 3.8. *Suppose that Algorithm* 2.1 *is applied to problem* (1.1) *and that finite termination does not occur. Suppose also that* AS1–AS3*,* (2.15)*,* (3.11)*, and* (3.14) *hold, that* $k \notin \mathcal{R}$*, that* n_k *satisfies* (3.4)*, and that*

$$\theta_k \le \kappa_{ubt}^{-\frac{1}{\mu}}\left(\frac{\eta_2\kappa_{tmd}\epsilon}{2\gamma_\theta}\right)^{\frac{1+\mu}{\mu}} \overset{\rm def}{=} \delta_\theta. \tag{3.17}$$

Then

$$f(x_k + s_k) \leq f(x_k) - \gamma_\theta \theta_k.$$

Proof. Applying Lemmas 3.4–3.6—which is possible because of (3.11), (3.14), $k \notin \mathcal{R}$, and the fact that n_k satisfies (3.4)—and (3.17), we obtain that

$$\begin{array}{rcl} f(x_k) - f(x_k + s_k) & \geq & \eta_2[m_k(x_k) - m_k(x_k + s_k)] \\ & \geq & \frac{1}{2}\eta_2 \kappa_{\mathrm{tmd}} \epsilon \Delta_k \\ & \geq & \frac{1}{2}\eta_2 \kappa_{\mathrm{tmd}} \epsilon \left(\frac{\theta_k}{\kappa_{\mathrm{ubt}}}\right)^{\frac{1}{1+\mu}} \\ & \geq & \gamma_\theta \theta_k, \end{array}$$

and the desired inequality follows. □

We now establish that if the trust-region radius and the constraint violation are both small at a noncritical iterate x_k, TRQP(x_k, Δ_k) must be compatible.

LEMMA 3.9. *Suppose that Algorithm* 2.1 *is applied to problem* (1.1) *and that finite termination does not occur. Suppose also that* AS1–AS3, (2.10), *and* (3.11) *hold, that* (2.15) *holds for $k \notin \mathcal{R}$, and that*

$$\Delta_k \leq \min\left[\gamma_0 \delta_\rho, \left(\frac{1}{\kappa_\mu}\right)^{\frac{1}{\mu}}, \left(\frac{\gamma_0^2(1-\gamma_\theta)\kappa_\Delta \kappa_\mu}{\kappa_{usc}\kappa_{ubt}}\right)^{\frac{1}{1-\mu}}\right] \stackrel{\mathrm{def}}{=} \delta_{\mathcal{R}}. \tag{3.18}$$

Suppose furthermore that $k > 0$ and that

$$\theta_k \leq \min[\delta_\theta, \delta_n]. \tag{3.19}$$

Then $k \notin \mathcal{R}$.

Proof. Because $\theta_k \leq \delta_n$, we know from (2.10) and Lemma 3.1 that n_k satisfies (2.10) and (3.4). Moreover, since $\theta_k \leq \delta_\theta$, we have that (3.17) also holds. Assume, for the purpose of deriving a contradiction, that $k \in \mathcal{R}$, that is,

$$\|n_k\| > \kappa_\Delta \kappa_\mu \Delta_k^{1+\mu}, \tag{3.20}$$

where we have used (2.12) and the fact that $\kappa_\mu \Delta_k^\mu \leq 1$ because of (3.18). In this case, the mechanism of the algorithm then ensures that $k - 1 \notin \mathcal{R}$. Now assume that iteration $k-1$ is unsuccessful. Because of Lemmas 3.6 and 3.8, which hold at iteration $k - 1 \notin \mathcal{R}$ because of (3.18), the fact that $\theta_k = \theta_{k-1}$, (2.10), and (3.17), we obtain that

$$\rho_{k-1} \geq \eta_2 \text{ and } f(x_{k-1} + s_{k-1}) \leq f(x_{k-1}) - \gamma_\theta \theta_{k-1}. \tag{3.21}$$

Hence, given that x_{k-1} is acceptable for the filter at the beginning of iteration $k - 1$, if this iteration is unsuccessful, it must be because $x_{k-1} + s_{k-1}$ is not acceptable for the filter and x_{k-1}, which in turn can happen only if

$$\theta(x_{k-1} + s_{k-1}) > (1 - \gamma_\theta)\theta_{k-1} = (1 - \gamma_\theta)\theta_k$$

because of (3.21) (see the last paragraph of section 2.2). But Lemma 3.4 and the mechanism of the algorithm then imply that

$$(1 - \gamma_\theta)\theta_k \leq \kappa_{\mathrm{ubt}} \Delta_{k-1}^2 \leq \frac{\kappa_{\mathrm{ubt}}}{\gamma_0^2} \Delta_k^2.$$

Combining this last bound with (3.20) and (2.10), we deduce that

$$\kappa_\Delta \kappa_\mu \Delta_k^{1+\mu} < \|n_k\| \le \kappa_{\rm usc}\theta_k \le \frac{\kappa_{\rm usc}\kappa_{\rm ubt}}{\gamma_0^2(1-\gamma_\theta)}\Delta_k^2$$

and hence that

$$\Delta_k^{1-\mu} > \frac{\gamma_0^2(1-\gamma_\theta)\kappa_\Delta\kappa_\mu}{\kappa_{\rm usc}\kappa_{\rm ubt}}.$$

Since this last inequality contradicts (3.18), our assumption that iteration $k-1$ is unsuccessful must be false. Thus iteration $k-1$ is successful and $\theta_k = \theta(x_{k-1}+s_{k-1})$. We then obtain from (3.20), (2.10), and (3.9) that

$$\kappa_\Delta \kappa_\mu \Delta_k^{1+\mu} < \|n_k\| \le \kappa_{\rm usc}\theta_k \le \kappa_{\rm usc}\kappa_{\rm ubt}\Delta_{k-1}^2 \le \frac{\kappa_{\rm usc}\kappa_{\rm ubt}}{\gamma_0^2}\Delta_k^2,$$

which is again impossible because of (3.18) and because $(1-\gamma_\theta) < 1$. Hence our initial assumption (3.20) must be false, which yields the desired conclusion. □

We now distinguish two mutually exclusive cases. For the first, we consider what happens if there is an infinite subsequence of iterates belonging to the filter.

LEMMA 3.10. *Suppose that Algorithm* 2.1 *is applied to problem* (1.1) *and that finite termination does not occur. Suppose also that* AS1–AS3, (2.10) *hold, and* (2.15) *holds for $k \notin \mathcal{R}$. Suppose furthermore that $|\mathcal{Z}| = \infty$. Then there exists a subsequence $\{k_j\} \subseteq \mathcal{Z}$ such that*

$$\lim_{j\to\infty} \theta_{k_j} = 0 \tag{3.22}$$

and

$$\lim_{j\to\infty} \chi_{k_j} = 0. \tag{3.23}$$

Proof. Let $\{k_i\}$ be any infinite subsequence of $\mathcal{Z}$. We observe that (3.22) follows from Lemma 3.3. Suppose now that

$$\chi_{k_i} \ge \epsilon_2 > 0 \tag{3.24}$$

for all i and some $\epsilon_2 > 0$. Suppose furthermore that there exists $\epsilon_3 > 0$ such that, for all $i \ge i_0$,

$$\Delta_{k_i} \ge \epsilon_3. \tag{3.25}$$

Observe first that (3.22) and (2.10) ensure that

$$\lim_{i\to\infty} \|n_{k_i}\| = 0. \tag{3.26}$$

Thus (3.25) ensures that (2.12) holds for sufficiently large i and thus $k_i \notin \mathcal{R}$ for such i. Now, as we noted in the proof of Lemma 3.5,

$$|m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i}^{\rm N})| \le \kappa_{\rm ubg}\|n_{k_i}\| + \frac{1}{2}\kappa_{\rm umh}\|n_{k_i}\|^2,$$

which in turn, with (3.26), yields that

$$\lim_{i\to\infty} [m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i}^{\rm N})] = 0. \tag{3.27}$$

We also deduce from (2.15) and AS2 that

$$m_{k_i}(x_{k_i}^{\mathrm{N}}) - m_{k_i}(x_{k_i} + s_{k_i}) \geq \kappa_{\mathrm{tmd}}\epsilon_2 \min\left[\frac{\epsilon_2}{\kappa_{\mathrm{umh}}}, \epsilon_3\right] \stackrel{\mathrm{def}}{=} \delta > 0. \tag{3.28}$$

We now decompose the model decrease in its normal and tangential components, that is,

$$m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i} + s_{k_i}) = m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i}^{\mathrm{N}}) + m_{k_i}(x_{k_i}^{\mathrm{N}}) - m_{k_i}(x_{k_i} + s_{k_i}).$$

Substituting (3.27) and (3.28) into this decomposition, we find that

$$\liminf_{i\to\infty}[m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i} + s_{k_i})] \geq \delta > 0. \tag{3.29}$$

We now observe that, because x_{k_i} is added to the filter at iteration k_i, we know from the mechanism of the algorithm that either iteration $k_i \in \mathcal{R}$ or (2.19) must fail. Since we have already shown that $k_i \notin \mathcal{R}$, (2.19) must fail for i sufficiently large, that is,

$$m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i} + s_{k_i}) < \kappa_\theta \theta_{k_i}^{\psi}. \tag{3.30}$$

Combining this bound with (3.29), we find that θ_{k_i} is bounded away from zero for i sufficiently large, which is impossible in view of (3.22). We therefore deduce that (3.25) cannot hold and obtain that there is a subsequence $\{k_\ell\} \subseteq \{k_i\}$ for which

$$\lim_{\ell\to\infty} \Delta_{k_\ell} = 0.$$

We now restrict our attention to the tail of this subsequence, that is, to the set of indices $k_\ell > 0$ that are large enough to ensure that (3.16), (3.17), and (3.18) hold, which is possible by definition of the subsequence and because of (3.22). For these indices, we may therefore apply Lemma 3.9 and deduce that iteration $k_\ell \notin \mathcal{R}$ for ℓ sufficiently large. Hence, as above, (3.30) must hold for ℓ sufficiently large. However, we may also apply Lemma 3.7, which contradicts (3.30), and therefore (3.24) cannot hold, yielding the desired result. □

Thus, if an infinite subsequence of iterates is added to the filter, Lemma 3.2 ensures that there exists a limit point which is a first-order critical point. Our remaining analysis then naturally concentrates on the possibility that there may be no such infinite subsequence. In this case, no further iterates are added to the filter for k sufficiently large. In particular, this means that the number of restoration iterations, $|\mathcal{R}|$, must be finite. In what follows, we assume that $k_0 \geq 0$ is the last iteration for which x_{k_0-1} is added to the filter.

Lemma 3.11. *Suppose that Algorithm* 2.1 *is applied to problem* (1.1)*, that finite termination does not occur, and that* $|\mathcal{Z}| < \infty$*. Suppose also that* AS1–AS3 *and* (2.10) *hold and that* (2.15) *holds for* $k \notin \mathcal{R}$*. Then we have that*

$$\lim_{k\to\infty} \theta_k = 0. \tag{3.31}$$

Furthermore, n_k *satisfies* (3.4) *for all* $k \geq k_0$ *sufficiently large.*

Proof. Consider any successful iterate with $k \geq k_0$. Since x_k is not added to the filter, it follows from the mechanism of the algorithm that $\rho_k \geq \eta_1$ holds and thus that

$$f(x_k) - f(x_{k+1}) \geq \eta_1[m_k(x_k) - m_k(x_k + s_k)] \geq \eta_1\kappa_\theta\theta_k^{\psi} \geq 0. \tag{3.32}$$

Thus the objective function does not increase for all successful iterations with $k \geq k_0$. But AS1 and AS3 imply (3.3), and therefore we must have, from the first part of this statement, that

$$\lim_{\substack{k \in \mathcal{S} \\ k \to \infty}} f(x_k) - f(x_{k+1}) = 0. \tag{3.33}$$

The limit (3.31) then immediately follows from (3.32) and the fact that $\theta_j = \theta_k$ for all unsuccessful iterations j that immediately follow the successful iteration k, if any. The last conclusion then results from (2.10) and Lemma 3.1. □

We now show that the trust-region radius cannot become arbitrarily small if the (asymptotically feasible) iterates stay away from first-order critical points.

LEMMA 3.12. *Suppose that Algorithm* 2.1 *is applied to problem* (1.1)*, that finite termination does not occur, and that* $|\mathcal{Z}| < \infty$*. Suppose also that* AS1–AS3 *hold and* (2.15) *holds for* $k \notin \mathcal{R}$*. Suppose furthermore that* (3.11) *hold for all* $k \geq k_0$*. Then there exists a* $\Delta_{\min} > 0$ *such that*

$$\Delta_k \geq \Delta_{\min}$$

for all k*.*

Proof. Suppose that $k_1 \geq k_0$ is chosen sufficiently large to ensure that (3.19) holds and that n_k satisfies (2.10) for all $k \geq k_1$, which is possible because of Lemma 3.11. Suppose also, for the purpose of obtaining a contradiction, that iteration j is the first iteration following iteration k_1 for which

$$\Delta_j \leq \gamma_0 \min\left[\delta_\rho, \sqrt{\frac{(1-\gamma_\theta)\theta^{\mathrm{F}}}{\kappa_{\mathrm{ubt}}}}, \Delta_{k_1}\right] \stackrel{\mathrm{def}}{=} \gamma_0 \delta_s, \tag{3.34}$$

where

$$\theta^{\mathrm{F}} \stackrel{\mathrm{def}}{=} \min_{i \in \mathcal{Z}} \theta_i$$

is the smallest constraint violation appearing in the filter. Note also that the inequality $\Delta_j \leq \gamma_0 \Delta_{k_1}$, which is implied by (3.34), ensures that $j \geq k_1 + 1$ and hence that $j - 1 \geq k_1$ and thus that $j - 1 \notin \mathcal{R}$. Then the mechanism of the algorithm and (3.34) imply that

$$\Delta_{j-1} \leq \frac{1}{\gamma_0}\Delta_j \leq \delta_s, \tag{3.35}$$

and Lemma 3.6, which is applicable because (3.34) and (3.35) together imply (3.14) with k replaced by $j-1$, then ensures that

$$\rho_{j-1} \geq \eta_2. \tag{3.36}$$

Furthermore, since n_{j-1} satisfies (2.10), Lemma 3.1 implies that we can apply Lemma 3.4. This, together with (3.34) and (3.35), gives that

$$\theta(x_{j-1} + s_{j-1}) \leq \kappa_{\mathrm{ubt}} \Delta_{j-1}^2 \leq (1-\gamma_\theta)\theta^{\mathrm{F}}. \tag{3.37}$$

We may also apply Lemma 3.8 because (3.34) and (3.35) ensure that (3.14) holds and because (3.17) also holds for $j - 1 \geq k_1$. Hence we deduce that

$$f(x_{j-1} + s_{j-1}) \leq f(x_{j-1}) - \gamma_\theta \theta_{j-1}.$$

This last relation and (3.37) ensure that $x_{j-1} + s_{j-1}$ is acceptable for the filter and x_{j-1}. Combining this conclusion with (3.36) and the mechanism of the algorithm, we obtain that $\Delta_j \geq \Delta_{j-1}$. As a consequence, and since (2.19) also holds at iteration $j-1$, iteration j cannot be the first iteration following k_1 for which (3.34) holds. This contradiction shows that $\Delta_k \geq \gamma_0 \delta_s$ for all $k > k_1$, and the desired result follows if we define

$$\Delta_{\min} = \min[\Delta_0, \ldots, \Delta_{k_1}, \gamma_0 \delta_s]. \qquad \square$$

We may now analyze the convergence of χ_k itself.

LEMMA 3.13. *Suppose that Algorithm* 2.1 *is applied to problem* (1.1)*, that finite termination does not occur, and that* $|\mathcal{Z}| < \infty$*. Suppose also that* AS1–AS3*,* (2.10) *hold, and* (2.15) *holds for* $k \notin \mathcal{R}$*. Then*

$$\liminf_{k \to \infty} \chi_k = 0. \tag{3.38}$$

Proof. We start by observing that Lemma 3.11 implies that the second conclusion of (2.10) holds for k sufficiently large. Moreover, as in Lemma 3.11, we obtain (3.32) and therefore (3.33) for each $k \in \mathcal{S}$, $k \geq k_0$. Suppose now, for the purpose of obtaining a contradiction, that (3.11) holds, and notice that

$$m_k(x_k) - m_k(x_k + s_k) = m_k(x_k) - m_k(x_k^{\mathrm{N}}) + m_k(x_k^{\mathrm{N}}) - m_k(x_k + s_k). \tag{3.39}$$

Moreover, note, as in Lemma 3.5, that

$$|m_k(x_k) - m_k(x_k^{\mathrm{N}})| \leq \kappa_{\mathrm{ubg}} \|n_k\| + \kappa_{\mathrm{umh}} \|n_k\|^2,$$

which in turn yields that

$$\lim_{k \to \infty} [m_k(x_k) - m_k(x_k^{\mathrm{N}})] = 0$$

because of Lemma 3.11 and the second conclusion of (2.10). This limit, together with (3.32), (3.33), and (3.39), then gives that

$$\lim_{\substack{k \to \infty \\ k \in \mathcal{S}}} [m_k(x_k^{\mathrm{N}}) - m_k(x_k + s_k)] = 0. \tag{3.40}$$

But (2.15), (3.11), AS2, and Lemma 3.12 together imply that for all $k \geq k_0$

$$\begin{aligned} m_k(x_k^{\mathrm{N}}) - m_k(x_k + s_k) &\geq \kappa_{\mathrm{tmd}} \chi_k \min\left[\frac{\chi_k}{\beta_k}, \Delta_k\right] \\ &\geq \kappa_{\mathrm{tmd}} \epsilon \min\left[\frac{\epsilon}{\kappa_{\mathrm{umh}}}, \Delta_{\min}\right], \end{aligned} \tag{3.41}$$

immediately giving a contradiction with (3.40). Hence (3.11) cannot hold and the desired result follows. $\square$

We may summarize all of the above in our main global convergence result.

THEOREM 3.14. *Suppose that Algorithm* 2.1 *is applied to problem* (1.1) *and that finite termination does not occur. Suppose also that* AS1*,* (2.10)*,* AS3*, and* AS2 *hold, and that* (2.15) *holds for* $k \notin \mathcal{R}$*. Let* $\{x_k\}$ *be the sequence of iterates produced by the algorithm. Then either the restoration procedure terminates unsuccessfully by converging to an infeasible first-order critical point of problem* (2.16)*, or there is a subsequence* $\{k_j\}$ *for which*

$$\lim_{j \to \infty} x_{k_j} = x_*$$

and x_* is a first-order critical point for problem (1.1).

Proof. Suppose that the restoration iteration always terminates successfully. From AS3, Lemmas 3.10, 3.11, and 3.13, we obtain that, for some subsequence $\{k_j\}$,

$$\lim_{j\to\infty} \theta_{k_j} = \lim_{j\to\infty} \chi_{k_j} = 0. \tag{3.42}$$

The conclusion then follows from Lemma 3.2. □

Can we dispense with AS3 to obtain this result? First, this assumption ensures that the objective remains bounded below and the constraint violation remains bounded above (see (3.3)). This is crucial for the rest of the analysis because the convergence of the iterates to feasibility depends on this fact. Thus, if AS3 does not hold, we have to verify that (3.3) holds for other reasons. The second part of this statement may be ensured quite simply by initializing the filter to $(\theta^{\max}, -\infty)$, for some $\theta^{\max} > \theta_0$, in Step 0 of the algorithm. This has the effect of putting an upper bound on the infeasibility of all iterates, which may be useful in practice. However, this does not prevent the objective function from being unbounded below in

$$\mathcal{C}(\theta^{\max}) = \{x \in \mathbb{R}^n \mid \theta(x) \le \theta^{\max}\},$$

and we cannot exclude the possibility that a sequence of infeasible iterates might both continue to improve the value of the objective function and satisfy (2.19). If $\mathcal{C}(\theta^{\max})$ is bounded, AS3 is most certainly satisfied. If this is not the case, we could assume that

$$f^{\min} \le f(x) \;\text{ and }\; 0 \le \theta(x) \le \theta^{\max} \;\text{ for }\; x \in \mathcal{C}(\theta^{\max}) \tag{3.43}$$

for some value of $f^{\min}$ and simply monitor that the values $f(x_k)$ are reasonable—in view of the problem being solved—as the algorithm proceeds. To summarize, we may replace AS1 and AS3 by the following assumption.

AS4. The functions f and c are twice continuously differentiable on an open set containing $\mathcal{C}(\theta^{\max})$, their first and second derivatives are uniformly bounded on $\mathcal{C}(\theta^{\max})$, and (3.43) holds.

The reader should note that AS4 no longer ensures the existence of a limit point, but only that (3.42) holds for some subsequence $\{k_j\}$. Furthermore, the comments following the statement of (2.10) no longer apply if limit points at infinity are allowed.

4. Conclusion and perspectives. We have introduced a trust-region SQP-filter algorithm for general nonlinear programming and have shown this algorithm to be globally convergent to first-order critical points. The proposed algorithm differs from that discussed by Fletcher and Leyffer [18], notably because it uses a decomposition of the step in its normal and tangential components and imposes some restrictions on the length of the former. However, preliminary numerical experiments indicate that its practical performance is similar to that reported in [18]. Since the performance of the latter is excellent, the theory developed in this paper provides the reassurance that filter algorithms also have reasonable convergence properties, which then makes these methods very attractive.

We are aware, however, that the convergence study is not complete, as we have not discussed local convergence properties. As it is possible to exhibit examples where the SQP step increases *both* the objective function *and* the constraint violation,[2] it is

[2] Such an example is provided by Figure 9.3.1 in Fletcher [16], taking the case $\beta = \frac{1}{4}$. Any feasible point close to the origin illustrates the effect.

very likely that such a study will have to introduce second-order corrections (see [16, section 14.4]) to ensure that the Maratos effect does not take place and that a fast (quadratic) rate of convergence can be achieved. Moreover, convergence to second-order critical points also remains, for now, an open question. In this context, the alternative definition of ρ_k presented in (3.15) is also likely to play a role if we choose H_k according to (3.1). In this case, we might choose

$$\Theta_k = \sum_{i \in \mathcal{E} \cup \mathcal{I}} [y_k]_i \langle s_k, \nabla_{xx} c_i(x_k) s_k \rangle$$

in order to ensure that the denominator of the fraction defining ρ_k is a correct model of its numerator not only up to first-order, but also up to second-order. These questions are the subject of ongoing work.

Appendix.

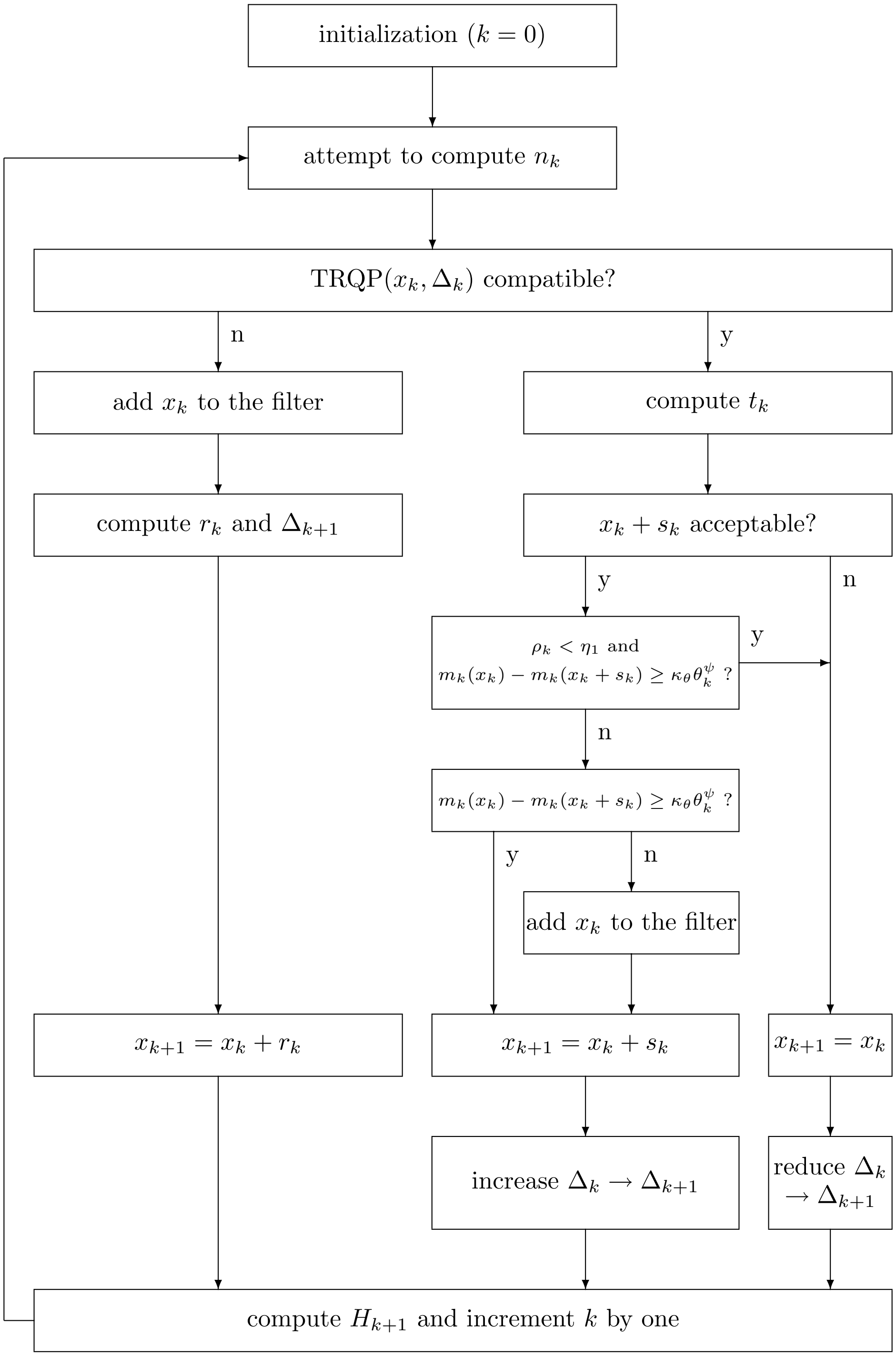
initialization ($k = 0$)
attempt to compute n_k
TRQP(x_k, Δ_k) compatible?
n
y
add x_k to the filter
compute t_k
compute r_k and Δ_{k+1}
$x_k + s_k$ acceptable?
y
n
$\rho_k < \eta_1$ and
$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_\theta \theta_k^\psi$?
y
n
$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_\theta \theta_k^\psi$?
y
n
add x_k to the filter
$x_{k+1} = x_k + r_k$
$x_{k+1} = x_k + s_k$
$x_{k+1} = x_k$
increase $\Delta_k \to \Delta_{k+1}$
reduce Δ_k
$\to \Delta_{k+1}$
compute H_{k+1} and increment k by one

FIG. A.1. *Flowchart of Algorithm* 2.1.

REFERENCES

[1] L. T. Biegler, J. Nocedal, and C. Schmid, *A reduced Hessian method for large-scale constrained optimization*, SIAM J. Optim., 5 (1995), pp. 314–347.

[2] R. H. Bielschowsky and F. A. M. Gomes, *Dynamical Control of Infeasibility in Nonlinearly Constrained Optimization*, presentation at the Optimization '98 Conference, University of Coimbra, Portugal, 1998.

[3] R. H. Byrd, J. C. Gilbert, and J. Nocedal, *A trust region method based on interior point techniques for nonlinear programming*, Math. Program., 89 (2000), pp. 149–186.

[4] R. H. Byrd, M. E. Hribar, and J. Nocedal, *An interior point algorithm for large-scale nonlinear programming*, SIAM J. Optim., 9 (2000), pp. 877–900.

[5] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, *A trust region algorithm for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 24 (1987), pp. 1152–1170.

[6] A. R. Conn, N. Gould, A. Sartenaer, and Ph. L. Toint, *Global convergence of a class of trust region algorithms for optimization using inexact projections on convex constraints*, SIAM J. Optim., 3 (1993), pp. 164–221.

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

[8] R. S. Dembo and U. Tulowitzki, *On the Minimization of Quadratic Functions Subject to Box Constraints*, Working paper, Series B, 71, School of Organization and Management, Yale University, New Haven, CT, 1983.

[9] J. E. Dennis, Jr., M. El-Alem, and M. C. Maciel, *A global convergence theory for general trust-region based algorithms for equality constrained optimization*, SIAM J. Optim., 7 (1997), pp. 177–207.

[10] J. E. Dennis, Jr., M. El-Alem, and K. Williamson, *A trust-region approach to nonlinear systems of equalities and inequalities*, SIAM J. Optim., 9 (1999), pp. 291–315.

[11] J. E. Dennis and L. N. Vicente, *On the convergence theory of trust-region-based algorithms for equality-constrained optimization*, SIAM J. Optim., 7 (1997), pp. 927–950.

[12] M. El-Alem, *Global convergence without the assumption of linear independence for a trust-region algorithm for constrained optimization*, J. Optim. Theory Appl., 87 (1995), pp. 563–577.

[13] M. El-Alem, *A global convergence theory for Dennis, El-Alem, and Maciel's class of trust-region algorithms for constrained optimization without assuming regularity*, SIAM J. Optim., 9 (1999), pp. 965–990.

[14] M. El-Hallabi and R. A. Tapia, *An Inexact Trust-Region Feasible-Point Algorithm for Nonlinear Systems of Equalities and Inequalities*, Technical Report TR95-09, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1995.

[15] A. V. Fiacco, *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, Academic Press, London, 1983.

[16] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., J. Wiley and Sons, Chichester, England, 1987.

[17] R. Fletcher and S. Leyffer, *User Manual for FilterSQP*, Numerical Analysis Report NA/181, Department of Mathematics, University of Dundee, Dundee, Scotland, 1998.

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

[19] R. Fletcher, S. Leyffer, and P. L. Toint, *On the Global Convergence of an SLP-Filter Algorithm*, Technical Report 98/13, Department of Mathematics, University of Namur, Namur, Belgium, 1998.

[20] R. Fletcher, S. Leyffer, and P. L. Toint, *On the Global Convergence of a Filter-SQP Algorithm*, Technical Report 00/15, Department of Mathematics, University of Namur, Namur, Belgium, 2000.

[21] N. I. M. Gould, M. E. Hribar, and J. Nocedal, *On the solution of equality constrained quadratic programming problems arising in optimization*, SIAM J. Sci. Comput., 23 (2001), pp. 1376–1395.

[22] M. Lalee, J. Nocedal, and T. D. Plantenga, *On the implementation of an algorithm for large-scale equality constrained optimization*, SIAM J. Optim., 8 (1998), pp. 682–706.

[23] X. Liu and Y. Yuan, *A Robust Trust-Region Algorithm for Solving General Nonlinear Programming Problems*, presentation at the International Conference on Nonlinear Programming and Variational Inequalities, Hong Kong, 1998.

[24] J. J. Moré and G. Toraldo, *On the solution of large quadratic programming problems with bound constraints*, SIAM J. Optim., 1 (1991), pp. 93–113.

[25] W. Murray and F. J. Prieto, *A sequential quadratic programming algorithm using an in-*

complete solution of the subproblem, SIAM J. Optim., 5 (1995), pp. 590–640.

[26] K. G. Murty and S. N. Kabadi, *Some NP-complete problems in quadratic and nonlinear programming*, Math. Programming, 39 (1987), pp. 117–129.

[27] E. O. Omojokun, *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints*, Ph.D. thesis, University of Colorado, Boulder, CO, 1989.

[28] A. Sartenaer, *Automatic determination of an initial trust region in nonlinear programming*, SIAM J. Sci. Comput., 18 (1997), pp. 1788–1803.

[29] P. L. Toint, *Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space*, IMA J. Numer. Anal., 8 (1988), pp. 231–252.

[30] A. Vardi, *A trust region algorithm for equality constrained minimization: Convergence properties and implementation*, SIAM J. Numer. Anal., 22 (1985), pp. 575–591.

[31] A. Wächter and L. T. Biegler, *Global and Local Convergence of Line Search Filter Methods for Nonlinear Programming*, Technical Report CAPD B-01-09, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA, 2001.

[32] Y. Yuan, *Trust region algorithms for nonlinear programming*, in Computational Mathematics in China, Contemp. Math. 163, Z. C. Shi, ed., AMS, Providence, RI, 1994, pp. 205–225.