Global Convergence of a Non-monotone Trust-Region SQP-Filter Algorithm for Nonlinear Programming

by N.I.M. Gould[1] and Ph. L. Toint[2]

Report 02/15 4 February 2003

[1] Rutherford Appleton Laboratory,
Computational Science and Engineering Departement
Chilton, Oxfordshire, England
Email: gould@rl.ac.uk
Current reports available from
http://www.numerical.rl.ac.uk/reports/reports.html

[2] Department of Mathematics, University of Namur,
61, rue de Bruxelles, B-5000 Namur, Belgium,
Email: philippe.toint@fundp.ac.be
Current reports available by anonymous ftp from the directory
"pub/reports" on thales.math.fundp.ac.be

Global Convergence of a Non-monotone Trust-Region Filter Algorithm for Nonlinear Programming

Nick Gould and Philippe L. Toint

4 February 2003

Abstract

A non-monotone variant of the trust-region SQP-filter algorithm analyzed in Fletcher, Gould, Leyffer, Toint and Wächter (2002*a*) is defined, that directly uses the dominated area of the filter as an acceptability criterion for trial points. It is proved that, under reasonable assumptions and for all possible choices of the starting point, the algorithm generates at least a subsequence converging to a first-order critical point.

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* suggested by Fletcher and Leyffer (2002), 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 in Fletcher, Leyffer and Toint (1998), in which the objective function is locally approximated by a linear function, leading, at each iteration, to the (exact) solution of a linear program. This algorithm therefore mixes the use of the filter with sequential linear programming (SLP). Similar results are shown in Fletcher, Leyffer and Toint (2002*b*), where the approximation of the objective function is quadratic, leading to sequential quadratic programming (SQP) methods, but at the expense of finding a global minimizer of the possibly nonconvex quadratic programming subproblem, which is known to be a very difficult task. Convergence of SQP filter methods is also considered in Fletcher et al. (2002a), where the SQP step is decomposed in "normal" and "tangential" components.

The main purpose of this paper, a companion of Fletcher et al. (2002a), is to analyze an algorithm where the use of the filter acceptance criterion for new iterates is relaxed to allow dominated iterates to be accepted in some cases. This is potentially important as it is known that SQP method can generate such iterates in their asymptotic fast convergence phase. The theory developed here therefore provides a possible convergence framework for a filter method with quadratic convergence properties without the need to introduce second-order corrections (as done in Wächter and Biegler, 2001, for instance). An additional feature of the theory presented is that it no longer needs the notion of a "margin" around the filter, a device which is common to all theoretical approaches of the filter method so far.

2 A Non-monotone Filter Algorithm

The algorithm that we are about to describe is iterative and of the Sequential Quadratic Programming (SQP) type. At a given iterate x_k, Newton's method is implicitly applied to solve (a local version of) the first-order necessary optimality conditions by solving the quadratic programming 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 $f_k = f(x_k)$, $g_k = g(x_k) \stackrel{\text{def}}{=} \nabla_x f(x_k)$, 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 a suitable value of 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).

2.1 The composite SQP step

Of course, the step s_k must be computed, typically by solving, possibly approximately, a variant of (2.1). In the trust-region approach, one takes into account the fact that (2.1) only approximates our original problem locally: the step s_k is thus restricted in norm to ensure that $x_k + s_k$ remains in a *trust-region* centred at x_k, where we believe this approximation to be adequate. 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

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

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.

In accordance with the ideas pioneered by Vardi (1985), Byrd, Schnabel and Shultz (1987) and Omojokun (1989), and also with Fletcher et al. (2002*a*), our algorithm decomposes the step s_k into the sum of two distinct components, a *normal step* n_k, such that $x_k + n_k$ satisfies the linear constraints of $\mathrm{TRQP}(x_k, \Delta_k)$ within the trust region, and a *tangential step* t_k, whose purpose is to obtain reduction of the objective function's model while continuing to satisfy those constraints. 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 $\mathrm{QP}(x_k)$. No specific choice for n_k is made, but one instead assumes that n_k exists when the maximum violation of the nonlinear constraints at the k-th iterate $\theta_k \stackrel{\text{def}}{=} \theta(x_k)$, with

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

is sufficiently small, and that n_k is then reasonably scaled with respect to the values of the constraints. In other words, we assume, exactly as in Fletcher et al. (2002*a*), that n_k exists and

$$\|n_k\| \leq \kappa_{\rm usc}\theta_k, \ \text{ whenever } \ \theta_k \leq \delta_n, \tag{2.10}$$

for some constants $\kappa_{\rm usc} > 0$ and $\delta_n > 0$. This assumption is also used by Dennis, El-Alem and Maciel (1997) and Dennis and Vicente (1997) in the context of problemsonly involvong equality constraints. It can be shown not to impose conditions on the constraints or the normal step itself that are unduly restrictive (see Fletcher et al. (2002*a*) for a discussion).

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

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

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 aim to find a tangential step t_k, starting from $x_k^{\rm N}$ and satisfying (2.6) and (2.7), whose objective is to decrease the value of the objective function. This is achieved by computing a step that produces a sufficient decrease in m_k, which is to say that we wish $m_k(x_k^{\rm N}) - m_k(x_k + s_k)$ to be "sufficiently large". Of course, this is only possible if the maximum permitted size of t_k is not too small, which is to say that $x_k^{\rm N}$ is not too close to the trust-region boundary. We formalize this condition by strengthening our requirement that $\|n_k\| \leq \Delta_k$ so that

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

for some $\kappa_\Delta \in (0, 1]$, some $\kappa_\mu > 0$ and some $\mu_k \in [0, 1)$. If condition (2.12) does not hold, we assume, as in Fletcher et al. (2002*a*), 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, $\mathrm{TRQP}(x_k, \Delta_k)$ is said to be *compatible* . In this case, a sufficient model decrease seems possible. We formalize this notion in the form of a familiar Cauchy-point condition, and, recalling that the feasible set of $\mathrm{QP}(x_k)$ is convex, we introduce the first-order criticality 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, Gould, Sartenaer and Toint, 1993). Note that this function is zero if $x_k^{\rm N}$ is a first-order critical point of the linearized "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 $s = n_k + t$. The sufficient decrease condition then consists in assuming that there exists a constant $\kappa_{\rm tmd} > 0$ such that

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

whenever $\mathrm{TRQP}(x_k, \Delta_k)$ is compatible, where $\beta_k = 1 + \|H_k\|$. We know from Toint (1988) and Conn et al. (1993) 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 $x_k^{\rm N}$, that is

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

This technique therefore provides an implementable algorithm for computing a step that satisfies (2.15), but, of course, a further reduction of m_k is often desirable if fast convergence is sought. Also note that the minimization problem of the right-hand side of (2.13) reduces to a linear programming problem if we choose to use a polyhedral norm in its definition at iteration k. However, we recognise that (2.15) may be difficult to verify in practice, since it may be expensive to compute $x_k^{\rm N}$ and P_k when the dimension of the problem is large.

2.2 The restoration procedure

If TRQP(x_k, Δ_k) is not compatible for μ, that is when the feasible set determined by the constraints of 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, we must consider an alternative. Observe 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 neighbourhood where this is not the case, since the problem (1.1) does not make sense in the current one. As in Fletcher et al. (2002*a*), we rely on a *restoration procedure*. The aim of this procedure is to produce a new point $x_k + r_k$ that satisfies two conditions: we require TRQP$(x_k + r_k, \Delta_{k+1})$ to be compatible for some $\Delta_{k+1} > 0$, and also require that $x_k + r_k$ be acceptable, in the sense that we discuss in the Section 2.3.3 (precisely, we require thate either (2.20) or (2.21) holds for such an x_k^+). In what follows, we will denote

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

the set of *restoration* iterations.

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

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

starting from x_k, the current iterate. This is a non-smooth problem, but there exist methods, possibly of trust-region type (such as that suggested by Yuan, 1994), 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 (1998) or of El-Hallabi and Tapia (1995) and Dennis, El-Alem and Williamson (1999) can respectively 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 seems to be 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. One therefore has to 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 desired.

2.3 The filter as a criterion to accept trial points

Unfortunately, because the SQP iteration may only be locally convergent, the step s_k or r_k may not always be very useful. Thus, having computed a step s_k or r_k from our current iterate x_k, we need to decide whether the trial point x_k^+, defined by

$$x_k^+ \stackrel{\text{def}}{=} \begin{cases} x_k + r_k & \text{if } k \in \mathcal{R}, \\ x_k + s_k & \text{otherwise} \end{cases} \tag{2.17}$$

is any better than x_k as an approximate solution to our original problem (1.1). If we decide that this is the case, we say that iteration k is successfull and choose x_k^+ as our next iterate. Let us denote by $\mathcal{S}$ the set of (indices of) all successful iterations, that is

$$\mathcal{S} = \{k \mid x_{k+1} = x_k^+\}.$$

We will discuss the details of precisely when we accept x_k^+ as our next iterate in Section 2.3.3, but note that an important ingredient in the process is the notion of a filter, a notion itself based on that of dominance.

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 unlikely to be of 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 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 $i \neq j$. Fletcher et al. (2002*a*) propose to accept a new trial iterate $x_k + s_k$ only if it is not dominated by any other iterate in the filter and x_k. In the vocabulary of multi-criteria optimization, this amounts to building elements of the efficient frontier associated with the bi-criteria problem of reducing infeasibility and the objective function value. We may describe this concept by associating with each iterate x_k its (θ, f)-pair (θ_k, f_k) and might choose to accept $x_k + s_k$ only if its (θ, f)-pair does not lie, in the two-dimensional space spanned by constraint violation and objective function value, above and on the right of a previously accepted pair. If we define

$$\mathcal{D}(\mathcal{F}) = \{(\theta, f) \mid \theta > \theta_j \text{ and } f > f_j \text{ for some } j \in \mathcal{F}\}, \tag{2.18}$$

the part of the (θ, f)-space that is dominated by the pairs in the filter, this amounts to say that x_k^+ could be accepted if $(\theta(x_k^+), f(x_k^+)) \notin \mathcal{D}(\mathcal{F}_k)$, where $\mathcal{F}_k$ denotes the filter at iteration k.

2.3.1 The contribution of a trial point to the filter

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 effective algorithmic tool. In particular, we may not wish to accept a new point x_k^+ if its (θ, f)-pair

$$(\theta_k^+, f_k^+) \stackrel{\text{def}}{=} (\theta(x_k^+), f(x_k^+))$$

is arbitrarily close to being dominated by another point already in the filter.

Fletcher et al. (2002*a*), as all other theoretical analysis of the filter that we know of, set a small "margin" around the border of $\mathcal{D}(\mathcal{F}_k)$ in which trial points are also rejected. We follow here a different idea and define, for any (θ, f)-pair, an area that represents its contribution to the area of $\mathcal{D}(\mathcal{F}_k)$. For this purpose, we partition the right half-plane $[0, +\infty] \times [-\infty, +\infty]$ into four different regions (see Figure 2.1). If we define $\mathcal{D}(\mathcal{F}_k)^C$ to be the complement of $\mathcal{D}(\mathcal{F}_k)$ in the right half-plane,

$$\theta_{\min}^{\mathcal{F}_k} \stackrel{\text{def}}{=} \min_{j \in \mathcal{F}_k} \theta_j, \qquad \theta_{\max}^{\mathcal{F}_k} \stackrel{\text{def}}{=} \max_{j \in \mathcal{F}_k} \theta_j,$$

and

$$f_{\min}^{\mathcal{F}_k} \stackrel{\text{def}}{=} \min_{j \in \mathcal{F}_k} f_j \qquad f_{\max}^{\mathcal{F}_k} \stackrel{\text{def}}{=} \max_{j \in \mathcal{F}_k} f_j,$$

these four parts are

1. the dominated part of the filter, $\mathcal{D}(\mathcal{F}_k)$.
2. the undominated part of lower left (south-west) corner of the half plane,
$$SW(\mathcal{F}_k) \stackrel{\text{def}}{=} \mathcal{D}(\mathcal{F}_k)^C \cap [0, \theta_{\max}^{\mathcal{F}_k}] \times [-\infty, f_{\max}^{\mathcal{F}_k}],$$
3. the undominated upper left (north-west) corner,
$$NW(\mathcal{F}_k) \stackrel{\text{def}}{=} [0, \theta_{\min}^{\mathcal{F}_k}) \times (f_{\max}^{\mathcal{F}_k}, +\infty],$$
4. the undominated lower right (south-east) corner,
$$SE(\mathcal{F}_k) \stackrel{\text{def}}{=} (\theta_{\max}^{\mathcal{F}_k}, +\infty] \times [-\infty, f_{\min}^{\mathcal{F}_k}).$$

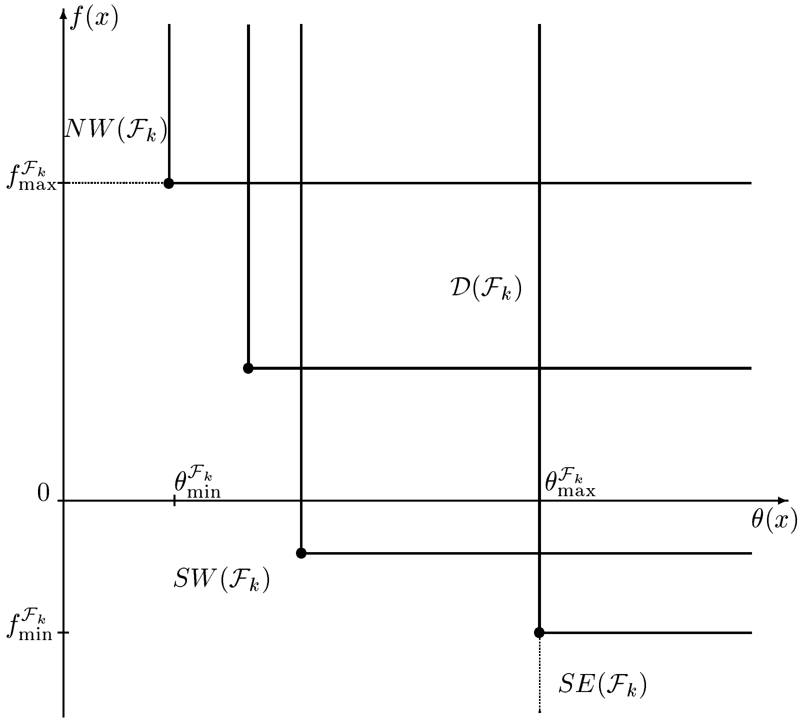

Figure 2.1: The partition of the right half-plane for a filter $\mathcal{F}_k$ containing four (θ, f) pairs.

Consider first a trial iterate x_k^+ with its associated (θ, f)-pair (θ_k^+, f_k^+) with $\theta_k^+ > 0$. If the filter is empty ($\mathcal{F}_k = \emptyset$), then we measure its contribution to the area of the filter by the simple formula

$$\alpha(x_k^+, \mathcal{F}_k) \stackrel{\text{def}}{=} \kappa_{\text{F}}^2$$

for some constant $\kappa_{\text{F}} > 0$. If the filter already contains some past iterates, we measure the contribution of x_k^+ to the area of the filter by

$$\alpha(x_k^+, \mathcal{F}_k) \stackrel{\text{def}}{=} \text{area}\Big(\mathcal{D}(\mathcal{F}_k)^C \cap [\theta_k^+, \theta_{max}^{\mathcal{F}_k} + \kappa_{\text{F}}] \times [f_k^+, f_{\max}^{\mathcal{F}_k} + \kappa_{\text{F}}]\Big) \quad \text{if } (\theta_k^+, f_k^+) \in SW(\mathcal{F}_k)),$$

by

$$\alpha(x_k^+, \mathcal{F}_k) \stackrel{\text{def}}{=} \kappa_{\text{F}}(\theta_{min}^{\mathcal{F}_k} - \theta_k^+) \quad \text{if } (\theta_k^+, f_k^+) \in NW(\mathcal{F}_k)),$$

by

$$\alpha(x_k^+, \mathcal{F}_k) \stackrel{\text{def}}{=} \kappa_{\text{F}}(f_{min}^{\mathcal{F}_k} - f_k^+) \quad \text{if } (\theta_k^+, f_k^+) \in SE(\mathcal{F}_k)),$$

and by

$$\alpha(x_k^+, \mathcal{F}_k) \stackrel{\text{def}}{=} -\text{area}\Big(\mathcal{D}(\mathcal{F}_k) \cap [\theta_k^+ - \theta_{\min}^{\mathcal{P}_k}] \times [f_k^+ - f_{\min}^{\mathcal{P}_k}]\Big), \quad \text{if } (\theta_k^+, f_k^+) \in \mathcal{D}(\mathcal{F}_k)),$$

where

$$\mathcal{P}_k \stackrel{\text{def}}{=} \{(\theta_j, f_j) \in \mathcal{F}_k \mid \theta_j < \theta_k^+ \text{ and } f_j < f_k^+\},$$

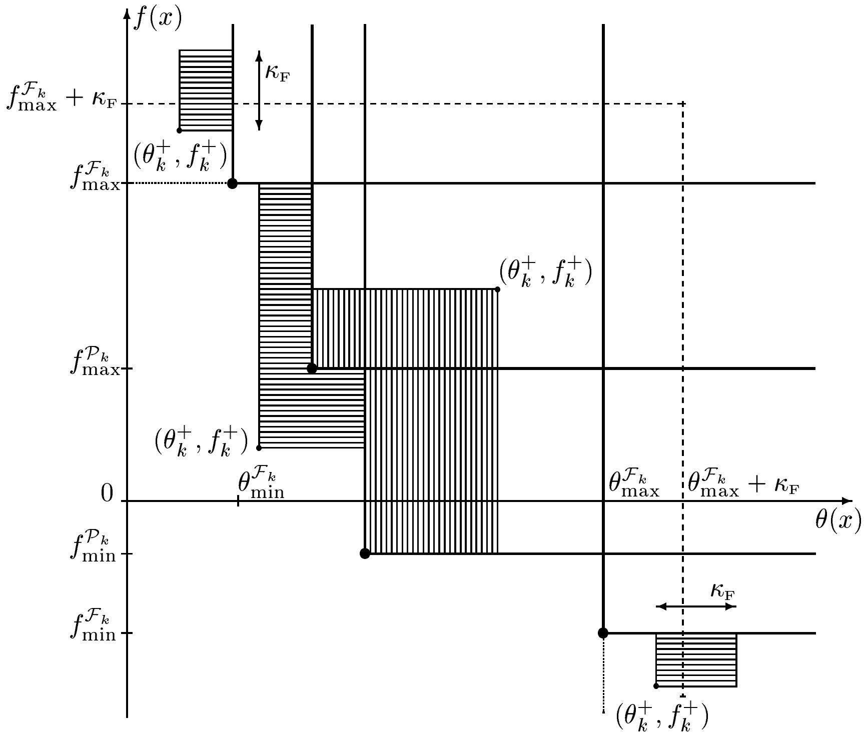

Figure 2.2: The contributions of four (θ_k^+, f_k^+) pairs (in $\mathcal{D}(\mathcal{F}_k)$, $SW(\mathcal{F}_k)$, $NW(\mathcal{F}_k)$ and $SE(\mathcal{F}_k)$) to the area of the filter. Horizontal stripes indicate a positive contribution and vertical stripes a negative one.

(the set of filter pairs that dominate (θ_k^+, f_k^+)), and

$$\theta_{\min}^{\mathcal{P}_k} \stackrel{\text{def}}{=} \min_{j \in \mathcal{P}_k} \theta_j, \qquad \theta_{\max}^{\mathcal{P}_k} \stackrel{\text{def}}{=} \max_{j \in \mathcal{P}_k} \theta_j.$$

Figure 2.2 illustrate the corresponding areas in the filter for four possible (θ_k^+, f_k^+) pairs (in $\mathcal{D}(\mathcal{F}_k)$, $SW(\mathcal{F}_k)$, $NW(\mathcal{F}_k)$ and $SE(\mathcal{F}_k)$) to the area of the filter. Horizontally dashed surfaces indicate a positive contribution and vertically dashed ones a negative contribution. Note that $\alpha(x, \mathcal{F})$ is a continuous function of $(\theta(x), f(x))$, and thus of x, for a given filter $\mathcal{F}$. Furthermore, $\alpha(x, \mathcal{F})$ is identically zero if $(\theta(x), f(x))$ is on the boundary of the dominated region $\mathcal{D}(\mathcal{F})$. Also note that, although seemingly complicated, the value of $\alpha(x, \mathcal{F})$ is not difficult to compute, since its calculation requires, in the worst case, us to consider all the points currently in the filter only once.

2.3.2 Updating the filter

The procedure to update the filter for a particular (θ, f) pair is extremely simple. If $(\theta_k, f_k) = (\theta(x_k), f(x_k))$ does not belong to $\mathcal{D}(\mathcal{F}_k)$ (i.e. if x_k is not dominated), then

$$\mathcal{F}_{k+1} \leftarrow \mathcal{F}_k \cup (\theta_k, f_k),$$

while if $(\theta_k, f_k) \in \mathcal{D}(\mathcal{F}_k)$ (if x_k is dominated),

$$\mathcal{F}_{k+1} \leftarrow (\mathcal{F}_k \setminus \mathcal{P}_k) \cup (\theta_{\min}^{\mathcal{P}_k}, f_k) \cup (\theta_k, f_{\min}^{\mathcal{P}_k}),$$

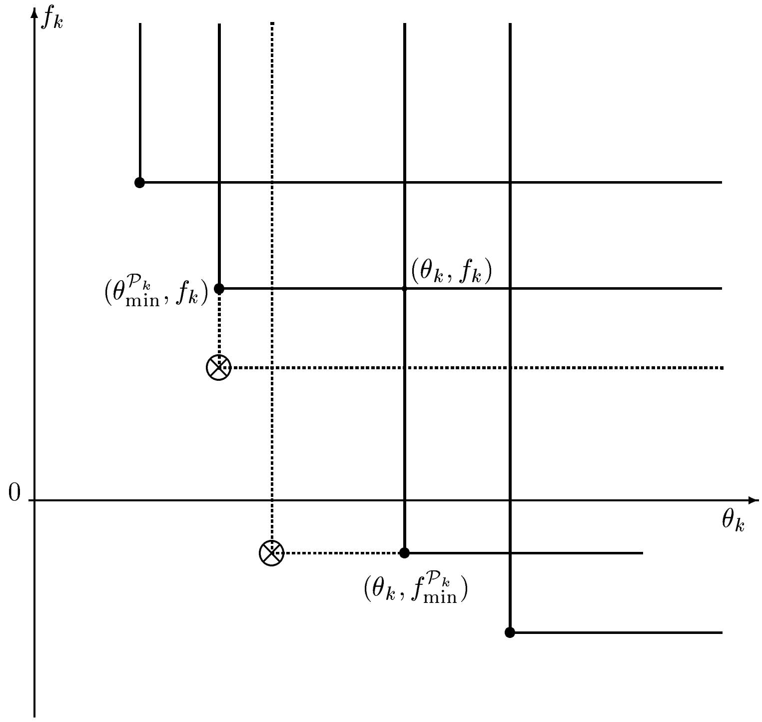

Figure 2.3: The filter $\mathcal{F}_{k+1}$ after including the dominated pair (θ_k, f_k) into $\mathcal{F}_k$.

where P_k is now the subset of pairs in $\mathcal{F}_k$ that dominate (θ_k, f_k). This last situation is illustrated by Figure 2.3, which shows the filter resulting from the operation of including the pair (θ_k, f_k) belonging to $\mathcal{D}(\mathcal{F}_k)$ (that associated with the vertically shaded "increment" in the filter area of Figure 2.2) in the filter. The two points in $\mathcal{P}_k$ that have been removed are marked with crossed circles and their associated dominated orthants are indicated by dotted lines.

Observe that it may happen that the number of points in the filter decreases when the set of dominating points $\mathcal{P}_k$ contains more than two filter pairs. Moreover, the pair for which the filter is updated is not always itself included in the filter (as shown in Figure 2.3).

2.3.3 Acceptability of potential iterates

We now return to the question of deciding whether or not a trial point x_k^+ is *acceptable* for the filter. We will insist that this is a necessary condition for the iteration k to be *successful* in the sense that $x_{k+1} = x_k^+$, i.e. the algorithm changes its current iterate to the trial point. Note that all restoration iterations are successful ($\mathcal{R} \subseteq \mathcal{S}$). Note also that (except for x_0) all iterates are produced by successful iterations : if we consider an iterate x_k, there must exists a *predecessor iteration* of index $p(k) \in \mathcal{S}$ such that

$$x_{p(k)}^+ = x_{p(k)+1} = x_k. \tag{2.19}$$

Observe that we do not always have that $p(k) = k - 1$ since not all iterations need being successful.

A monotone version of our method (rather similar to that developped in Fletcher et al., 2002*a*, but using $\alpha(x, \mathcal{F})$ rather that a margin around the filter) would be to accept x_k^+ whenever this trial point results in an sufficient increase in the dominated area of the filter, i.e. $\mathcal{D}(\mathcal{F}_k)$. This is to say that x_k^+ would be acceptable for the filter whenever

$$\alpha_k \geq \gamma_{\mathcal{F}} (\theta_k^+)^2, \tag{2.20}$$

where $\alpha_k \stackrel{\text{def}}{=} \alpha(x_k^+, \mathcal{F}_k)$ and where $\gamma_{\mathcal{F}} \in (0,1)$ is a constant. The non-monotone version that we analyze below replaces this condition by the weaker requirement that

$$\sum_{\substack{j=r(k)+1 \\ j \in \mathcal{U}}}^{k} \alpha_{p(j)} + \alpha_k \geq \gamma_{\mathcal{F}} \left[\sum_{\substack{j=r(k)+1 \\ j \in \mathcal{U}}}^{k} \theta_j^2 + (\theta_k^+)^2 \right] \tag{2.21}$$

where $\alpha_q \stackrel{\text{def}}{=} \alpha(x_q^+, \mathcal{F}_q)$ (and thus $\alpha_{p(q)} = \alpha(x_q, \mathcal{F}_{p(q)})$), where

$$\mathcal{U} = \{k \mid x_k \text{ the filter is updated for} (\theta_k, f_k)\},$$

and where $r(k) \leq k$ is some past *reference iteration* such that $r(k) \in \mathcal{U}$. Note that condition (2.21) may equivalently be written in the more symmetric form

$$\sum_{\substack{j=r(k)+1 \\ j \in \mathcal{U}}}^{k} \alpha_{p(j)} + \alpha_k \geq \gamma_{\mathcal{F}} \left[\sum_{\substack{j=r(k)+1 \\ j \in \mathcal{U}}}^{k} (\theta_{p(j)}^+)^2 + (\theta_k^+)^2 \right]$$

because of (2.19).

The reader may notice that condition (2.21) is reminiscent of the condition for non-monotone trust-region algorithms developed in Toint (1997) (see also Chapter 10.1 of Conn, Gould and Toint, 2000). It requires that the average contribution to the filter area of the last points included in the filter and x_k^+ together to be globally (sufficiently) positive, but makes it possible to accept x_k^+ even though it may be dominated (i.e. lie in $\mathcal{D}(\mathcal{F}_k)$).

However, if x_k^+ provides a clear monotonic improvement, in the sense that (2.20) holds, we are also prepared to accept it. Thus, x_k^+ *will be called acceptable at iteration* k *if either (2.20) or (2.21) holds.* We will denote

$$\mathcal{A} \stackrel{\text{def}}{=} \{k \in \mathcal{S} \mid (2.21) \text{ holds}\} \tag{2.22}$$

Observe also that we could replace θ_k^2 by $\min[\theta_k^\sigma, \kappa_\theta]$ in (2.20) and (2.21), where σ and κ_θ are strictly positive constants. This variant may be more numerically sensible, and does not affect the theory developed below.

2.4 The non-monotone Algorithm

We are now ready to define our algorithm formally as Algorithm 2.1. A flow-chart of the algorithm is given as an appendix.

Algorithm 2.1: Non-monotone 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 $0 < \gamma_0 < \gamma_1 \leq 1 \leq \gamma_2$, $0 < \eta_1 \leq \eta_2 < 1$, $\gamma_{\mathcal{F}} \in (0,1)$, $\kappa_\theta \in (0,1)$, $\kappa_\Delta \in (0,1]$, $\kappa_\mu > 0$, $\mu \in (0,1)$, $\psi > 1/(1+\mu)$ 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, update the filter for (θ_k, f_k) 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 $x_k^+ = x_k + r_k$ is acceptable. If this proves impossible, stop. Otherwise, set $x_{k+1} = x_k^+$ and go to Step 7.

Step 3: Determine a trial step. Compute a step t_k, set $x_k^+ = x_k + n_k + t_k$, and evaluate $c(x_k^+)$ and $f(x_k^+)$.

Step 4: Test acceptability of the trial point. If x_k^+ is not acceptable, again set $x_{k+1} = x_k$, choose $\Delta_{k+1} \in [\gamma_0\Delta_k, \gamma_1\Delta_k]$, set $n_{k+1} = n_k$, and go to Step 7. If

$$m_k(x_k) - m_k(x_k^+) < \kappa_\theta \theta_k^\psi, \tag{2.23}$$

then update the filter for (θ_k, f_k) and go to Step 6.

Step 5: Test predicted vs. achieved reduction. If

$$\rho_k \stackrel{\text{def}}{=} \frac{f(x_k) - f(x_k^+)}{m_k(x_k) - m_k(x_k^+)} < \eta_1, \tag{2.24}$$

set $x_{k+1} = x_k$, choose $\Delta_{k+1} \in [\gamma_0\Delta_k, \gamma_1\Delta_k]$, set $n_{k+1} = n_k$ and go to Step 7.

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

$$\Delta_{k+1} \in [\Delta_k, \gamma_2\Delta_k] \text{ if } \rho_k \geq \eta_2 \text{ and (2.23) fails.}$$

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

As in Fletcher and Leyffer (2002, 1998), 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

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

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

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 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}. 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.

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, 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 et al. (1987) to increase the new radius, or even to consider the technique described by Sartenaer (1997). However, we recognize that numerical experience with the algorithm is too limited at this stage to make definite recommendations.

The role of condition (2.23) may be interpreted as follows. If this condition holds, 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 in the filter. Fletcher et al. (1998) use the term of "θ-step" in such circumstances, to indicate that the main preoccupation is to reduce constraint violation. On the other hand, if condition (2.23) fails, 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 (2002) use the term of "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 is realized by the actual decrease in the function, which is why we then perform the test (2.24). In particular, if the iterate x_k is feasible, then (2.10) implies that $x_k = x_k^{\mathrm{N}}$ and we obtain that

$$\kappa_\theta \theta_k^\psi = 0 \leq m_k(x_k^{\mathrm{N}}) - m_k(x_k^+) = m_k(x_k) - m_k(x_k^+). \tag{2.25}$$

As a consequence, the filter mechanism is irrelevant if all iterates are feasible, and the algorithm reduces to a traditional unconstrained trust-region method. Another consequence of (2.25) is that *no feasible iterate is ever included in the filter*, which is crucial in allowing finite termination of the restoration procedure, as explained in Fletcher et al. (2002*a*). Note that the argument may fail and a 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, 1998, for an example).

Notice also that (2.23) ensures that the denominator of ρ_k in (2.24) will be strictly positive whenever θ_k is. If $\theta_k = 0$, then $x_k = x_k^{\rm N}$, and the denominator of (2.24) 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 (2.23) fails or holds. If (2.23) holds, 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 in the filter at Step 4). 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, (2.23) fails, 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, observe that the mechanism of the algorithm imposes that

$$\mathcal{U} \subseteq \mathcal{S}, \tag{2.26}$$

i.e. that iterates are included in the filter only at successful iterations.

3 Convergence to First-Order Critical Points

We now prove that our non-monotone algorithm generates a globally convergent sequence of iterates. In the following analysis, we concentrate on the case that 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.

In order to obtain our global convergence result, we will use the assumptions

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

AS2: there exists $\kappa_{\rm umh} > 1$ such that

$$\|H_k\| \leq \kappa_{\rm umh} - 1 \ \text{ for all } \ k,$$

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 i-th 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 $\kappa_{\rm ubh} > 1$ such that, for all k,

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

A proof of this property, based on Taylor expansion, may be found, for instance, in Toint (1988) or on p. 133 of Conn et al. (2000). A second important consequence of our assumptions is that AS1 and AS3 together directly ensure that, for all k,

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

for some constants $f^{\min} \le f^{\max}$ 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 $[0, \theta^{\max}] \times [f^{\min}, \infty]$.

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

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

Proof. See Lemma 3.1 of Fletcher et al. (2002*a*). □

Our assumptions and the definition of χ_k in (2.13) also 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 $\{k_i\} \not\subseteq \mathcal{R}$ such that

$$\lim_{i\to\infty} \chi_{k_i} = 0 \text{ and } \lim_{i\to\infty} \theta_{k_i} = 0. \tag{3.5}$$

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

Proof. See Lemma 3.2 of Fletcher et al. (2002*a*). □

We start our analysis by examining the impact of our non-monotone acceptance criteria (2.20) and (2.21). Once a trial point as accepted as a new iterate, it must be because it provide some improvement, compared to either a past reference iterate (using (2.21)), or to the previous iterate (using (2.20)). We formalize this notion by saying that iterate $x_k = x_{p(k)+1}$ improves on iterate $x_{i(k)}$, where

$$i(k) = r(p(k)) \text{ if } p(k) \in \mathcal{A},$$

that is if x_k is accepted at iteration $p(k)$ using (2.21), and

$$i(k) = p(k) \text{ if } p(k) \notin \mathcal{A}, \tag{3.6}$$

that is if x_k is accepted at iteration $p(k)$ using (2.20). Now consider any iterate x_k. This iterate improved on $x_{i(k)}$, which was itself accepted because it improved on $x_{i(i(k))}$, and so on, back to the stage where x_0 is reached by this backwards referencing process. Hence we may construct, for each k, a *chain* of successful iterations indexed by $\mathcal{C}_k = \{\ell_1, \ell_1, \ell_2, \ldots, \ell_q\}$ such that

$$\ell_1 = 0, \quad \ell_q = k \text{ and } x_{\ell_j} = x_{i(\ell_j+1)} \text{ for } j = 1, \ldots, q-1.$$

We start by proving the following useful lemma.

Lemma 3.3 Suppose that Algorithm 2.1 is applied to problem (1.1). Then, for each k,

$$\text{area}(\mathcal{D}(\mathcal{F}_k)) \geq \gamma_{\mathcal{F}} \sum_{\substack{i=0 \\ j\in\mathcal{U}}}^{k-1} \theta_i^2$$

Proof. Consider now the backward referencing chain from iteration $k-1$, $\mathcal{C}_{k-1}$, and any ℓ_j ($j > 0$) in this chain. Observe that, if $p(\ell_j) \in \mathcal{A}$, then (2.21) implies that $i(\ell_j) = r(p(\ell_j)) = \ell_{j-1}$ and that

$$\sum_{\substack{i=\ell_{j-1}+1 \\ i\in\mathcal{U}}}^{\ell_j} \alpha_{p(j)} \geq \gamma_{\mathcal{F}} \sum_{\substack{i=\ell_{j-1}+1 \\ i\in\mathcal{U}}}^{\ell_j} \theta_i^2 \tag{3.7}$$

If now $p(\ell_j) \notin \mathcal{A}$, then $\ell_{j-1} = p(\ell_j)$ and thus

$$\{\ell_{j-1}+1,\ldots,\ell_j\} \cap \mathcal{U} \subseteq \{\ell_{j-1}+1,\ldots,\ell_j\} \cap S = \{\ell_j\},$$

where we have used (2.26). Moreover, (2.20) then implies that $\alpha_{p(\ell_j)} \geq \gamma_{\mathcal{F}}\theta_{\ell_j}^2$, so that (3.7) holds again in this case. Combining these inequalities with the inequality

$$\text{area}(\mathcal{D}(\mathcal{F}_k)) \geq \sum_{\substack{i=0 \\ i\in\mathcal{U}}}^{k-1} \alpha_{p(i)} = \sum_{j=0}^{q} \left[\sum_{\substack{i=\ell_{j-1}+1 \\ i\in\mathcal{U}}}^{\ell_j} \alpha_{p(i)} \right]$$

then gives the desired result. □

We now consider what happens when the filter is updated an infinite number of times.

Lemma 3.4 Suppose that Algorithm 2.1 is applied to problem (1.1). Suppose also that AS1 and AS3 hold and that $|\mathcal{U}| = \infty$. Then

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

Proof. Suppose, for the purpose of obtaining a contradiction, that there exists an infinite subsequence $\{k_i\} \subseteq \mathcal{U}$ such that $\theta_{k_i} \geq \epsilon$ for all i and for some $\epsilon > 0$. Applying now Lemma 3.3, we deduce that

$$\text{area}(\mathcal{D}(\mathcal{F}_{k_i+1})) \geq i\gamma_{\mathcal{F}}\epsilon^2.$$

However, (3.3) implies that, for any k, area$(\mathcal{D}(\mathcal{F}_k))$ is bounded above by a constant $\kappa_{\rm F}^{\max} \geq 0$ independent of k. Hence we obtain that

$$i \leq \frac{\kappa_{\rm F}^{\max}}{\gamma_{\mathcal{F}}\epsilon^2},$$

and i must also be finite. This contradicts the fact that the subsequence $\{k_i\}$ is infinite. Hence this latter assumption is impossible and the conclusion follows. □

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

Lemma 3.5 Suppose that Algorithm 2.1 is applied to problem (1.1), that AS1 and AS3 hold, that $k \notin \mathcal{R}$ and that n_k satisfies (3.4). Then

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

and

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

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

Proof. See Lemma 3.4 of Fletcher et al. (2002*a*). □

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

Lemma 3.6 Suppose that Algorithm 2.1 is applied to problem (1.1). Suppose also that AS1–AS3, (2.12) and (2.15) hold, that $k \notin \mathcal{R}$, that

$$\chi_k \geq \epsilon, \tag{3.10}$$

for some $\epsilon > 0$, and that

$$\Delta_k \leq \min\left[\frac{\epsilon}{\kappa_{\rm umh}}, \left(2\frac{\kappa_{\rm ubg}}{\kappa_{\rm umh}\kappa_\Delta\kappa_\mu}\right)^{\frac{1}{1+\mu}}, \left(\frac{\kappa_{\rm tmd}\epsilon}{4\kappa_{\rm ubg}\kappa_\Delta\kappa_\mu}\right)^{\frac{1}{\mu}}\right] \stackrel{\rm def}{=} \delta_m, \tag{3.11}$$

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

$$m_k(x_k) - m_k(x_k^+) \geq \tfrac{1}{2}\kappa_{\rm tmd}\epsilon\Delta_k.$$

Proof. See Lemma 3.5 of Fletcher et al. (2002*a*). □

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 3.7 Suppose that Algorithm 2.1 is applied to problem (1.1). Suppose also that AS1–AS3, (2.15) and (3.10) hold, that $k \notin \mathcal{R}$, and that

$$\Delta_k \leq \min\left[\delta_m, \frac{(1-\eta_2)\kappa_{\rm tmd}\epsilon}{2\kappa_{\rm ubh}}\right] \stackrel{\rm def}{=} \delta_\rho. \tag{3.12}$$

Then

$$\rho_k \geq \eta_2.$$

Proof. See Lemma 3.6 of Fletcher et al. (2002*a*). □

Note that this proof could easily be extended if the definition of ρ_k in (2.24) were altered to be of the form

$$\rho_k \stackrel{\text{def}}{=} \frac{f(x_k) - f(x_k^+) + \Theta_k}{m_k(x_k) - m_k(x_k^+)} \tag{3.13}$$

provided 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 et al., 2000).

Now, we also show that the test (2.23) will always fail when the trust-region radius is sufficiently small.

Lemma 3.8 Suppose that Algorithm 2.1 is applied to problem (1.1). Suppose also that AS1–AS3, (2.12), (2.15) and (3.10) hold, that $k \notin \mathcal{R}$, that n_k satisfies (3.4), and that

$$\Delta_k \leq \min\left[\delta_m, \left(\frac{\kappa_{\rm tmd}\epsilon}{2\kappa_\theta \kappa_{\rm ubt}^{\psi}}\right)^{\frac{1}{\psi(1+\mu)-1}}\right] \stackrel{\text{def}}{=} \delta_f. \tag{3.14}$$

Then

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

Proof. This directly results from the inequalities

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

where we successively used Lemma 3.5, (3.14) and Lemma 3.6. □

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 3.9 Suppose that Algorithm 2.1 is applied to problem (1.1). Suppose also that AS1–AS3, (2.15), (3.10) and (3.12) hold, that $k \notin \mathcal{R}$, that n_k satisfies (3.4), and that

$$\theta_k \leq \kappa_{\rm ubt}^{-\frac{1}{\mu}} \left(\frac{\eta_2 \kappa_{\rm tmd}\epsilon}{2\sqrt{\gamma_{\mathcal{F}}}}\right)^{\frac{1+\mu}{\mu}} \stackrel{\text{def}}{=} \delta_\theta. \tag{3.15}$$

Then

$$f(x_k^+) \leq f(x_k) - \sqrt{\gamma_{\mathcal{F}}}\,\theta_k.$$

Proof. Applying Lemmas 3.5–3.7—which is possible because of (3.10), (3.12),

$k \notin \mathcal{R}$ and n_k satisfies (3.4)—and (3.15), we obtain that

$$\begin{aligned} f(x_k) - f(x_k^+) &\geq \eta_2[m_k(x_k) - m_k(x_k^+)] \\ &\geq \tfrac{1}{2}\eta_2 \kappa_{\rm tmd} \epsilon \Delta_k \\ &\geq \tfrac{1}{2}\eta_2 \kappa_{\rm tmd} \epsilon \left(\frac{\theta_k}{\kappa_{\rm ubt}}\right)^{\frac{1}{1+\mu}} \\ &\geq \sqrt{\gamma_{\mathcal{F}}}\theta_k \end{aligned}$$

and the desired inequality follows. □

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

Lemma 3.10 Suppose that Algorithm 2.1 is applied to problem (1.1). Suppose also that AS1–AS3, (2.10) and (3.10) hold, that (2.15) holds for $k \notin \mathcal{R}$, and that

$$\Delta_k \leq \min\left[\gamma_0 \delta_\rho, \left(\frac{1}{\kappa_\mu}\right)^{\frac{1}{\mu}}, \left(\frac{\gamma_0^2(1-\sqrt{\gamma_{\mathcal{F}}})\kappa_\Delta \kappa_\mu}{\kappa_{\rm usc}\kappa_{\rm ubt}}\right)^{\frac{1}{1-\mu}}\right] \stackrel{\rm def}{=} \delta_{\mathcal{R}}. \tag{3.16}$$

Suppose furthermore that

$$\theta_k \leq \min[\delta_\theta, \delta_n]. \tag{3.17}$$

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 (3.15) also holds. Assume, for the purpose of deriving a contradiction, that $k \in \mathcal{R}$, that is

$$\|n_k\| > \kappa_\Delta \kappa_\mu \Delta_k^{1+\mu}, \tag{3.18}$$

where we have used (2.12) and the fact that $\kappa_\mu \Delta_k^\mu \leq 1$ because of (3.16). 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 3.7 and 3.9, which hold at iteration $k-1 \notin \mathcal{R}$ because of (3.16), the fact that $\theta_k = \theta_{k-1}$, (2.10), and (3.15), we obtain that

$$\rho_{k-1} \geq \eta_2 \ \text{ and } \ f(x_{k-1}^+) \leq f(x_{k-1}) - \sqrt{\gamma_{\mathcal{F}}}\theta_{k-1}. \tag{3.19}$$

Hence, if iteration $k-1$ is unsuccessful and this must be because x_{k-1}^+ is not acceptable for the filter. However, if we have that

$$\theta_{k-1}^+ \leq (1-\sqrt{\gamma_{\mathcal{F}}})\theta_{k-1}, \tag{3.20}$$

then, using the second part of (3.19) and the fact that $(\theta_{k-1}, f_{k-1}) \in SW(\mathcal{F}_{k-1})$,

$$\alpha(x_{k-1}^+, \mathcal{F}_{k-1}) \geq [f(x_{k-1}) - f(x_{k-1}^+)][\theta_{k-1} - \theta_{k-1}^+] \geq \gamma_{\mathcal{F}}\theta_{k-1}^2 \geq \gamma_{\mathcal{F}}[\theta_{k-1}^+]^2,$$

and x_{k-1}^+ is acceptable for the filter because of (2.20). Since this is not the case, (3.20) cannot hold and we must have that

$$\theta_{k-1}^+ > (1 - \sqrt{\gamma_{\mathcal{F}}})\theta_{k-1} = (1 - \sqrt{\gamma_{\mathcal{F}}})\theta_k.$$

But Lemma 3.5 and the mechanism of the algorithm then imply that

$$(1 - \sqrt{\gamma_{\mathcal{F}}})\theta_k < \kappa_{\rm ubt}\Delta_{k-1}^2 < \frac{\kappa_{\rm ubt}}{\gamma_0^2}\Delta_k^2.$$

Combining this last bound with (3.18) and (2.10), we deduce that

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

and hence that

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

Since this last inequality contradicts (3.16), our assumption that iteration $k-1$ is unsuccessful must be false. Thus iteration $k-1$ is successful and $\theta_k = \theta_{k-1}^+$. We then obtain from (3.18), (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 (3.16) and because $(1 - \sqrt{\gamma_{\mathcal{F}}}) < 1$. Hence our initial assumption (3.18) must be false, which yields the desired conclusion. □

We continue to follow Fletcher et al. (2002*a*) and 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 3.11 Suppose that Algorithm 2.1 is applied to problem (1.1). Suppose also that AS1–AS3, (2.10) hold and (2.15) holds for $k \notin \mathcal{R}$. Suppose furthermore that $|\mathcal{U}| = \infty$. Then there exists a subsequence $\{k_j\} \subseteq \mathcal{U}$ such that

$$\lim_{j\to\infty} \theta_{k_j} = 0 \tag{3.21}$$

and

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

Proof. Let $\{k_i\}$ be any infinite subsequence of $\mathcal{U}$. We observe that (3.21) follows from Lemma 3.4. Suppose now that

$$\chi_{k_i} \ge \epsilon_2 > 0 \tag{3.23}$$

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

$$\Delta_{k_i} \ge \epsilon_3. \tag{3.24}$$

Observe first that (3.21) and (2.10) ensure that

$$\lim_{i\to\infty} \|n_{k_i}\| = 0. \tag{3.25}$$

Thus (3.24) 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 3.6,

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

which in turn, with (3.25), yields that

$$\lim_{i\to\infty} [m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i}^{\rm N})] = 0. \tag{3.26}$$

We also deduce from (2.15) and AS2 that

$$m_{k_i}(x_{k_i}^{\rm N}) - m_{k_i}(x_{k_i}^{+}) \ge \kappa_{\rm tmd}\epsilon_2 \min\left[\frac{\epsilon_2}{\kappa_{\rm umh}}, \epsilon_3\right] \stackrel{\rm def}{=} \delta > 0. \tag{3.27}$$

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

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

Substituting (3.26) and (3.27) into this decomposition, we find that

$$\liminf_{i\to\infty} [m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i}^{+})] \ge \delta > 0. \tag{3.28}$$

We now observe that, because $k_i \in \mathcal{U} \setminus \mathcal{R}$, we know from the mechanism of the algorithm that (2.23) must hold, that is

$$m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i}^{+}) < \kappa_\theta \theta_{k_i}^{\psi}. \tag{3.29}$$

Combining this bound with (3.28), we find that θ_{k_i} is bounded away from zero for i sufficiently large, which is impossible in view of (3.21). We therefore deduce that (3.24) 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_ℓ that are large enough to ensure that (3.14), (3.15) and (3.16) hold, which is possible by definition of the subsequence and because of (3.21). For these indices, we may therefore apply Lemma 3.10, and deduce that iteration $k_\ell \notin \mathcal{R}$ for ℓ sufficiently large. Hence, as above, (3.29) must hold for ℓ sufficiently large. However, we may also apply Lemma 3.8, which contradicts (3.29), and therefore (3.23) cannot hold, yielding the desired result. □

Thus, if the filter is updated at an infinite subsequence of iterates, 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, the filter is unchanged 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 at which the filter was updated.

Lemma 3.12 Suppose that Algorithm 2.1 is applied to problem (1.1), that finite termination does not occur and that $|\mathcal{U}| < \infty$. Suppose also that AS1–AS3, (2.10) hold and that (2.15) holds for $k \notin \mathcal{R}$. Then we have that

$$\lim_{k\to\infty} \theta_k = 0. \tag{3.30}$$

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 the filter is not updated at iteration k, it follows from the mechanism of the algorithm that $\rho_k \geq \eta_1$ holds and thus that

$$f(x_k) - f(x_{k+1}) \geq \eta_1[m_k(x_k) - m_k(x_k^+)] \geq \eta_1 \kappa_\theta \theta_k^\psi \geq 0. \tag{3.31}$$

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

$$\lim_{\substack{k\in\mathcal{S}\\ k\to\infty}} f(x_k) - f(x_{k+1}) = 0. \tag{3.32}$$

The limit (3.30) then immediately follows from (3.31) 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. $\square$

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 3.13 Suppose that Algorithm 2.1 is applied to problem (1.1), that finite termination does not occur and that $|\mathcal{U}| < \infty$. Suppose also that AS1–AS3 hold and (2.15) holds for $k \notin \mathcal{R}$. Suppose furthermore that (3.10) 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 (3.17) holds and that n_k satisfies (2.10) for all $k \geq k_1$, which is possible because of

Lemma 3.12. Suppose also, for the purpose of obtaining a contradiction, that iteration j is the first iteration following iteration k_1 for which

$$\Delta_j \leq \gamma_0 \min\left[\delta_\rho, \sqrt{\frac{(1-\sqrt{\gamma_{\mathcal{F}}})\theta^{\mathrm{F}}}{\kappa_{\mathrm{ubt}}}}, \Delta_{k_1}\right] \stackrel{\mathrm{def}}{=} \gamma_0 \delta_s, \tag{3.33}$$

where

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

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 (3.33), 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 (3.33) imply that

$$\Delta_{j-1} \leq \frac{1}{\gamma_0} \Delta_j \leq \delta_s \tag{3.34}$$

and Lemma 3.7, which is applicable because (3.33) and (3.34) together imply (3.12) with k replaced by $j-1$, then ensures that

$$\rho_{j-1} \geq \eta_2. \tag{3.35}$$

Furthermore, since n_{j-1} satisfies (2.10), Lemma 3.1 implies that we can apply Lemma 3.5. This together with (3.33) and (3.34), gives that

$$\theta_{j-1}^{+} \leq \kappa_{\mathrm{ubt}} \Delta_{j-1}^2 \leq (1-\sqrt{\gamma_{\mathcal{F}}})\theta^{\mathrm{F}}. \tag{3.36}$$

We may also apply Lemma 3.9 because (3.33) and (3.34) ensure that (3.12) holds and because (3.15) also holds for $j-1 \geq k_1$. Hence we deduce that

$$f(x_{j-1}^{+}) \leq f(x_{j-1}) - \sqrt{\gamma_{\mathcal{F}}}\theta_{j-1}.$$

This last relation and (3.36) ensure that x_{j-1}^{+} is acceptable for the filter. Combining this conclusion with (3.35) and the mechanism of the algorithm, we obtain that $\Delta_j \geq \Delta_{j-1}$. As a consequence, and since (2.23) also fails at iteration $j-1$, iteration j cannot be the first iteration following k_1 for which (3.33) 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

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

□

We may now analyze the convergence of χ_k itself.

Lemma 3.14 Suppose that Algorithm 2.1 is applied to problem (1.1), that finite termination does not occur and that $|\mathcal{U}| < \infty$. Suppose also that AS1–AS3, (2.10) hold and (2.15) holds for $k \notin \mathcal{R}$. Then

$$\liminf_{k \to \infty} \chi_k = 0. \tag{3.37}$$

Proof. We start by observing that Lemma 3.12 implies that the second conclusion of (2.10) holds for k sufficiently large. Moreover, as in Lemma 3.12, we obtain (3.31) and therefore (3.32) for each $k \in \mathcal{S}$, $k \geq k_0$. Suppose now, for the purpose of obtaining a contradiction, that (3.10) holds and notice that

$$m_k(x_k) - m_k(x_k^+) = m_k(x_k) - m_k(x_k^{\rm N}) + m_k(x_k^{\rm N}) - m_k(x_k^+). \tag{3.38}$$

Moreover, note, as in Lemma 3.6, that

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

which in turn yields that

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

because of Lemma 3.12 and the second conclusion of (2.10). This limit, together with (3.31), (3.32) and (3.38), then gives that

$$\lim_{\substack{k\to\infty \\ k\in\mathcal{S}}} [m_k(x_k^{\rm N}) - m_k(x_k^+)] = 0. \tag{3.39}$$

But (2.15), (3.10), AS2 and Lemma 3.13 together imply that, for all $k \geq k_0$

$$m_k(x_k^{\rm N}) - m_k(x_k^+) \geq \kappa_{\rm tmd}\chi_k \min\left[\frac{\chi_k}{\beta_k}, \Delta_k\right] \geq \kappa_{\rm tmd}\epsilon \min\left[\frac{\epsilon}{\kappa_{\rm umh}}, \Delta_{\min}\right], \tag{3.40}$$

immediately giving a contradiction with (3.39). Hence (3.10) cannot hold and the desired result follows. □

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

Theorem 3.15 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 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 3.11, 3.12 and 3.14, we obtain that, for some subsequence $\{k_j\}$,

$$\lim_{j\to\infty} \theta_{k_j} = \lim_{j\to\infty} \chi_{k_j} = 0. \tag{3.41}$$

The conclusion then follows from Lemma 3.2. □

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 (2002), 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. It also differs from the algorithm of Fletcher et al. (2002*a*) in two main aspects. The first and most important is that the rule for deciding whether a trial point is acceptable for the filter is non-monotone, and allows, in some circumstances, acceptance of points that are dominated by other filter pairs. This gives hopes that an SQP filter algorithm can be developed without introducing second-order correction steps. The second is that the algorithm no longer relies on the definition of a "margin" around the filter, but directly uses the dominated area of the filter as an acceptance criterion.

References

R. H. Byrd, R. B. Schnabel, and G. A. Shultz. A trust region algorithm for nonlinearly constrained optimization. *SIAM Journal on Numerical Analysis*, **24**, 1152–1170, 1987.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-Region Methods*. Number 01 *in* 'MPS-SIAM Series on Optimization'. SIAM, Philadelphia, USA, 2000.

A. R. Conn, N. I. M. 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 Journal on Optimization*, **3**(1), 164–221, 1993.

J. E. Dennis and L. N. Vicente. On the convergence theory of trust-region based algorithms for equality-constrained optimization. *SIAM Journal on Optimization*, **7**(4), 927–950, 1997.

J. E. Dennis, M. El-Alem, and M. C. Maciel. A global convergence theory for general trust-region based algorithms for equality constrained optimization. *SIAM Journal on Optimization*, **7**(1), 177–207, 1997.

J. E. Dennis, M. El-Alem, and K. A. Williamson. A trust-region approach to nonlinear systems of equalities and inequalities. *SIAM Journal on Optimization*, **9**(2), 291–315, 1999.

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, Texas, USA, 1995.

R. Fletcher and S. Leyffer. User manual for filterSQP. Numerical Analysis Report NA/181, Department of Mathematics, University of Dundee, Dundee, Scotland, 1998.

R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, **91**(2), 239–269, 2002.

R. Fletcher, N. I. M. Gould, S. Leyffer, Ph. L. Toint, and A. Wächter. Global convergence of trust-region SQP-filter algorithms for nonlinear programming. *SIAM Journal on Optimization*, **13**(3), 635–659, 2002*a*.

R. Fletcher, S. Leyffer, and Ph. L. Toint. On the global convergence of an SLP-filter algorithm. Technical Report 98/13, Department of Mathematics, University of Namur, Namur, Belgium, 1998.

R. Fletcher, S. Leyffer, and Ph. L. Toint. On the global convergence of an filter-SQP algorithm. *SIAM Journal on Optimization*, **13**(1), 44–59, 2002*b*.

E. O. Omojokun. *Trust region algorithms for optimization with nonlinear equality and inequality constraints*. PhD thesis, University of Colorado, Boulder, Colorado, USA, 1989.

A. Sartenaer. Automatic determination of an initial trust region in nonlinear programming. *SIAM Journal on Scientific Computing*, **18**(6), 1788–1803, 1997.

Ph. L. Toint. Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space. *IMA Journal of Numerical Analysis*, **8**(2), 231–252, 1988.

Ph. L. Toint. A non-monotone trust-region algorithm for nonlinear optimization subject to convex constraints. *Mathematical Programming*, **77**(1), 69–94, 1997.

A. Vardi. A trust region algorithm for equality constrained minimization: convergence properties and implementation. *SIAM Journal on Numerical Analysis*, **22**(3), 575–591, 1985.

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, USA, 2001.

Y. Yuan. Trust region algorithms for nonlinear programming. *in* Z. C. Shi, ed., 'Contemporary Mathematics', Vol. 163, pp. 205–225, Providence, Rhode-Island, USA, 1994. American Mathematical Society.

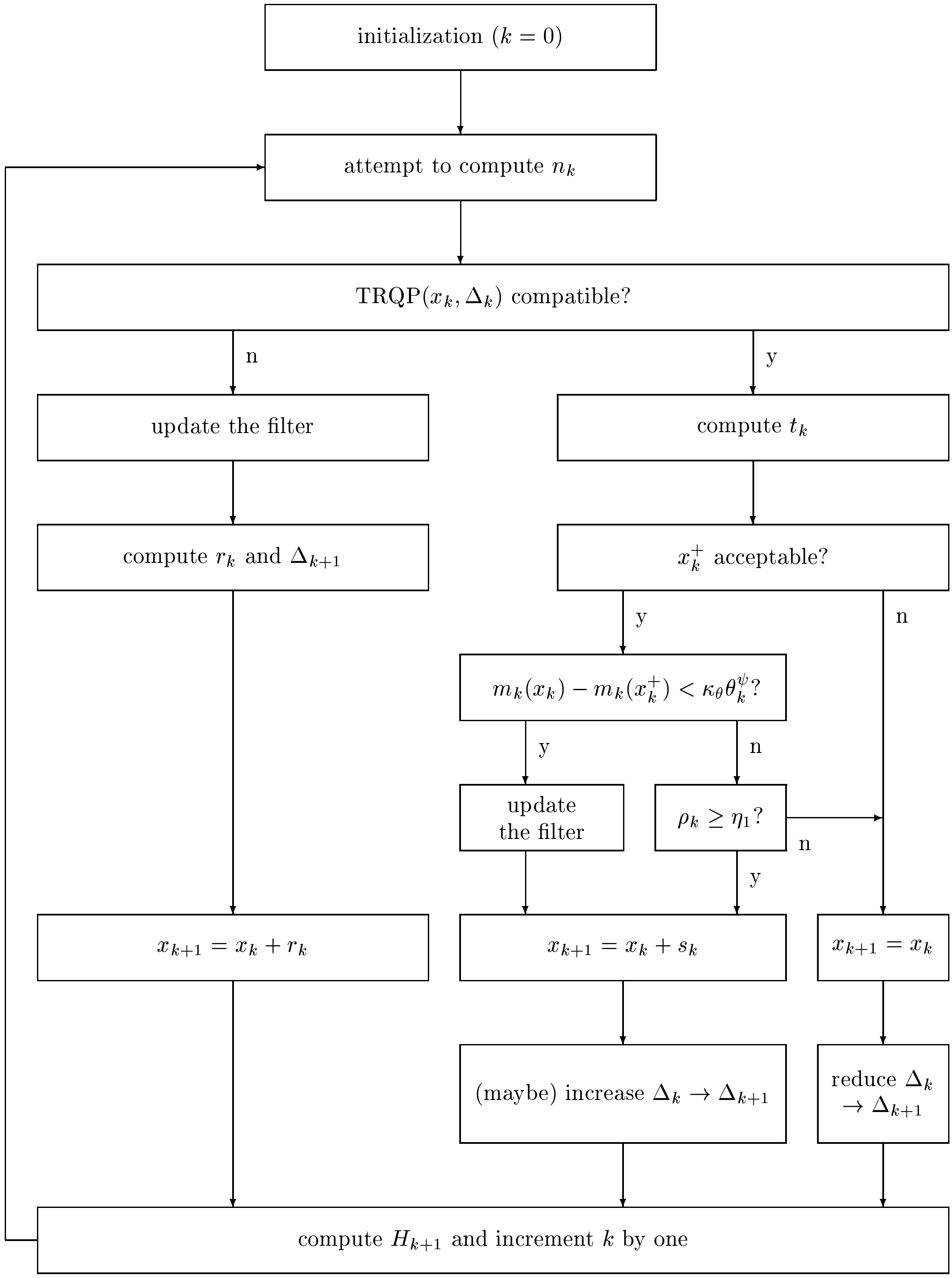

Figure 6.1: Flowchart of the algorithm.