Global Convergence of a Hybrid Trust-Region SQP-Filter Algorithm for General Nonlinear Programming

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

Report 01/07 August 30, 2001

[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 Hybrid Trust-Region SQP-Filter Algorithm for General Nonlinear Programming

Nick Gould and Philippe L. Toint

25 August 2001

Abstract

Global convergence to first-order critical points is proved for a variant of the trust-region SQP-filter algorithm analyzed in Fletcher, Gould, Leyffer and Toint (1999). This variant allows the use of two types of step strategies: the first decomposes the step into its normal and tangential components, while the second replaces this decomposition by a stronger condition on the associated model decrease.

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 (1997), 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 (2000), where the approximation of the objective function is quadratic, leading to sequential quadratic programming (SQP) methods, but at the price 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 was also considered in Fletcher et al. (1999), where the SQP step was decomposed in "normal" and "tangential" components. Although this latter procedure is computationally well-defined and considerably less complex than finding the global minimum of a general quadratic program, it may sometimes be costly, and a simpler strategy, where the step is computed "as a whole" can also be of practical interest whenever possible. The purpose of this paper, a companion of Fletcher et al. (1999), is to analyze a hybrid algorithm that uses the decomposition of the step into normal and tangential components as infrequently as possible.

2 A Hybrid Trust-Region SQP-Filter Algorithms

For the sake of completeness and clarity, we review briefly the main constituent parts of the SQP algorithm discussed in Fletcher et al. (1999). Sequential quadratic programming 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 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 filter

Unfortunately, due to the locally convergent nature of Newton's iteration, the step s_k may not always be very good. Thus, having computed (in a so far unspecified manner) a step s_k from our current iterate x_k, we need to decide whether the trial point $x_k + s_k$ is any better than x_k as an approximate solution to our original problem (1.1). This is achieved by using the notion of a filter, itself based on that of dominance.

If we define the feasibility measure

$$\theta(x) = \max\left[0, \max_{i\in\mathcal{E}} |c_i(x)|, \max_{i\in\mathcal{I}} -c_i(x)\right], \tag{2.2}$$

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 $i \neq j$. Fletcher et al. (1999) 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 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 (including that associated with x_k).

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 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 Fletcher et al. (1999) 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

$$\theta(x) < (1 - \gamma_\theta)\theta_j \text{ or } f(x) < f_j - \gamma_\theta \theta_j \text{ for all } (\theta_j, f_j) \in \mathcal{F}, \tag{2.3}$$

for some $\gamma_\theta \in (0, 1)$. We also say that x is "acceptable for the filter and x_k" if (2.3) 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, Fletcher et al. (1999) add (θ, f)-pairs to the filter. If an iterate x_k is acceptable for $\mathcal{F}$, this is done 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 $\theta_j \ge \theta_k$ and $f_j - \gamma_\theta \theta_j \ge f_k - \gamma_\theta \theta_k$. 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 introduction to the notion of a filter 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

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

is also be acceptable for the filter and x_k.

2.2 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

$$\begin{array}{ll} \text{minimize} & m_k(x_k + s) \\ \text{subject to} & c_{\mathcal{E}}(x_k) + A_{\mathcal{E}}(x_k)s = 0, \\ & c_{\mathcal{I}}(x_k) + A_{\mathcal{I}}(x_k)s \geq 0, \\ \text{and} & \|s\| \leq \Delta_k, \end{array} \tag{2.4}$$

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

$$m_k(x_k + s) = f_k + \langle g_k, s\rangle + \tfrac{1}{2}\langle s, H_k s\rangle, \tag{2.5}$$

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 of the existing 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 (1983) and Murray and Prieto (1995) to design conditions under which an approximate solution of the subproblem is acceptable. In contrast, the algorithm of Fletcher et al. (1999) is in spirit to the composite-step SQP methods pioneered by Vardi (1985), Byrd, Schnabel and Shultz (1987), and Omojokun (1989) and later developed by several authors, including Biegler, Nocedal and Schmid (1995), El-Alem (1995, 1999), Byrd, Gilbert and Nocedal (2000*a*), Byrd, Hribar and Nocedal (2000*b*), Bielschowsky and Gomes (1998), Liu and Yuan (1998) and Lalee, Nocedal and Plantenga (1998). It 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 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. The step s_k is then called *composite*. More formally, we write

$$s_k = n_k + t_k \tag{2.6}$$

and assume that

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

$$\|s_k\| \leq \Delta_k, \tag{2.8}$$

and

$$c_{\mathcal{E}}(x_k) + A_{\mathcal{E}}(x_k)s_k = 0, \quad c_{\mathcal{I}}(x_k) + A_{\mathcal{I}}(x_k)s_k \geq 0. \tag{2.9}$$

Of course, this is a strong assumption, since in particular (2.7) or (2.8)/(2.9) 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

$$n_k = P_k[x_k] - x_k, \tag{2.10}$$

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 $\theta_k \stackrel{\mathrm{def}}{=} \theta(x_k)$ is sufficiently small, and that n_k is then reasonably scaled with respect to the values of the constraints. In other words, Fletcher et al. (1999) assume that

$$n_k \text{ exists and } \|n_k\| \leq \kappa_{\mathrm{usc}}\theta_k, \text{ whenever } \theta_k \leq \delta_n, \tag{2.11}$$

for some constants $\kappa_{\mathrm{usc}} > 0$ 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 problems with equality constraints only. It can be shown not to impose conditions on the constraints or the normal step itself that are unduly restrictive (see Fletcher et al. (1999) 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

$$x_k^{\mathrm{N}} = x_k + n_k, \tag{2.12}$$

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 (2.8) and (2.9), 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 condition by replacing our condition that $\|n_k\| \leq \Delta_k$ by the stronger requirement that

$$\|n_k\| \leq \kappa_\Delta \Delta_k \min[1, \kappa_\mu \Delta_k^{\mu_k}], \tag{2.13}$$

for some $\kappa_\Delta \in (0, 1]$, some $\kappa_\mu > 0$ and some $\mu_k \in [0, 1)$. If condition (2.13) does not hold, Fletcher et al. (1999) presume 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 (2.13) holds, $\mathrm{TRQP}(x_k, \Delta_k)$ is said to be *compatible for* μ. In this case at least a sufficient model decrease seems possible, in the form of a familiar Cauchy-point condition. In order to formalize this notion, we recall that the feasible set of $\mathrm{QP}(x_k)$ is convex, and we can therefore introduce the first-order criticality measure

$$\chi_k = \Big| \min_{\substack{A_{\mathcal{E}}(x_k)t=0 \\ c_{\mathcal{I}}(x_k)+A_{\mathcal{I}}(x_k)(n_k+t)\geq 0 \\ \|t\|\leq 1}} \langle g_k + H_k n_k, t\rangle \Big| \tag{2.14}$$

(see Conn, Gould, Sartenaer and Toint, 1993). Note that this function is zero if and only if x_k^{N} is a first-order critical point of the linearized "tangential" problem

$$\begin{array}{ll} \text{minimize} & \langle g_k + H_k n_k, t\rangle + \frac{1}{2}\langle H_k t, t\rangle \\ \text{subject to} & A_{\mathcal{E}}(x_k)t = 0 \\ & c_{\mathcal{I}}(x_k) + A_{\mathcal{I}}(x_k)(n_k + t) \geq 0, \end{array} \tag{2.15}$$

which is equivalent to $\mathrm{QP}(x_k)$ with $s = n_k + t$. The sufficient decrease condition then consists in assuming that there exists a constant $\kappa_{\mathrm{tmd}} > 0$ such that

$$m_k(x_k^{\mathrm{N}}) - m_k(x_k^{\mathrm{N}} + t_k) \geq \kappa_{\mathrm{tmd}} \chi_k \min\left[\frac{\chi_k}{\beta_k}, \Delta_k\right], \tag{2.16}$$

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^{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 (2.16) (see Gould, Hribar and Nocedal, 1998 for an example in the case where $c(x) = c_{\mathcal{E}}(x)$, or Toint, 1988 and Moré and Toraldo, 1991 for the case of bound constraints), but, of course, reduction of m_k beyond that imposed by (2.16) is often possible and desirable if fast convergence is sought. Also note that the minimization problem of the right-hand side of (2.14) reduces to a linear programming problem if we choose to use a polyhedral norm in its definition at iteration k.

If $\mathrm{TRQP}(x_k, \Delta_k)$ is not compatible for μ, 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 (2.13) fails, solving $\mathrm{TRQP}(x_k, \Delta_k)$ is most likely pointless, and 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. Fletcher et al. (1999) 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 $\Delta_{k+1} > 0$—another condition will actually be needed, which we will discuss shortly.

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

$$\min_{x \in \mathbb{R}^n} \theta(x) \tag{2.17}$$

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 An alternative step

Is it possible to find a cheaper alternative to computing a normal step, finding a generalized Cauchy point and explicitly checking (2.16)? Suppose, for now, that it is possible to compute a point $x_k + s_k$ directly to satisfy the constraints of TRQP(x_k, Δ_k) and for which

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\rm tmd} \min[\pi_k, \Delta_k] \tag{2.18}$$

and $\pi_k = \pi(x_k)$, where π is a continuous function of its argument that is a criticality measure for TRQP(x_k, Δ_k). Such a s_k could for instance be computed by applying any efficient method to this latter problem (we might think of interior point methods of the type described in Conn, Gould, Orban and Toint, 2000) for instance, and its performance could be assessed by computing

$$\pi(x) = \min_{y \mid y_{\mathcal{I}} \geq 0} \|g(x) - A(x)^T y\|.$$

Of course, nothing guarantees that such an s_k exists (depending on our choice of $\pi(x)$) or is cheaply computable for each x_k, which means that we may have to resort to the normal-tangential strategy of Fletcher et al. (1999) if such problems arise. However, if we can find s_k at a fraction of the cost of computing n_k and t_k, can we use it inside an SQP-filter algorithm and maintain the desirable convergence to first-order critical points?

Obviously, the answer to that question depends on the manner in which the use of s_k is integrated into a complete algorithm.

2.4 A hybrid 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, shown on the next page.

This algorithm differs from that of Fletcher et al. (1999) in that it contains the alternative step strategy, but also because it allows the normal step to satisfy (2.13) with $\mu = 0$ whenever (2.19) holds, that is whenever the current iterate is sufficiently feasible. (As we will see later, (2.13) with $\mu > 0$ can be viewed as an implicit technique to impose (2.19).)

As in Fletcher and Leyffer (1997) and Fletcher and Leyffer (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

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

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

As in Fletcher et al. (1999), some comments on this algorithm are now in order. 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 is 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 (2.13), 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 is both compatible and acceptable for the filter.

For the restoration procedure in Step 3a to succeed, we have to evaluate whether 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 (2.13). Of course, once this is achieved, this normal step may be reused at iteration $k+1$, if the composite step strategy is used.

Algorithm 2.1: Hybrid 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 $\gamma_0 < \gamma_1 \leq 1 \leq \gamma_2$, $0 < \eta_1 \leq \eta_2 < 1$, $\gamma_\theta \in (0,1)$, $\kappa_\theta \in (0,1)$, $\kappa_\Delta \in (0,1]$, $\kappa_\mu > 0$, $\mu \in (0,1)$, $\psi > 1/(1+\mu)$, $\kappa_{\rm u} > 0$ and $\kappa_{\rm tmd} \in (0,1]$. Compute $f(x_0)$ and $c(x_0)$. Set $\mathcal{F} = \emptyset$ and $k = 0$.

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

Step 2: Alternative step. If

$$\theta_k > \kappa_{\rm u} \Delta_k \min[1, \Delta_k^\mu], \tag{2.19}$$

set $\mu_k = \mu$ and go to Step 3. Otherwise, attempt to compute a step s_k that satisfies the constraints of $\mathrm{TRQP}(x_k, \Delta_k)$ and (2.18). If this succeeds, go to Step 4. Otherwise, set $\mu_k = 0$.

Step 3: Composite step.

Step 3a: Normal component. Attempt to compute a normal step n_k. If TRQP (x_k, Δ_k) is compatible for μ_k, go to Step 3b. Otherwise, include x_k in the filter and compute a restoration step r_k for which $\mathrm{TRQP}(x_k + r_k, \Delta_{k+1})$ 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 3b: Tangential component. Compute a tangential 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 $n_{k+1} = n_k$ if Step 3 was executed, and go to Step 7.
- If

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_\theta \theta_k^\psi, \tag{2.20}$$

and

$$\rho_k \stackrel{\rm def}{=} \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)} < \eta_1, \tag{2.21}$$

again set $x_{k+1} = x_k$, choose $\Delta_{k+1} \in [\gamma_0 \Delta_k, \gamma_1 \Delta_k]$, set $n_{k+1} = n_k$ if Step 3 was executed, and go to Step 7.

Step 5: Test to include the current iterate in the filter. If (2.20) 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

$$\Delta_{k+1} \in [\Delta_k, \gamma_2 \Delta_k] \;\text{ if }\; \rho_k \geq \eta_2 \;\text{ and (2.20) holds.}$$

Step 7: Update the Hessian approximation. Determine H_{k+1}. Increment k by one and go to Step 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. It also makes it easier to satisfy (2.13). 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 extensive numerical experience will remain the ultimate measure of any suggestion at this level.

The role of condition (2.20) 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 in the filter. Fletcher and Leyffer (1997) 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 (2.20) 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 (1997) 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. 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 also require that (2.21) does not hold. In particular, if the iterate x_k is feasible, then (2.19) and (2.11) imply that $x_k = x_k^{\mathrm{N}}$ and we obtain that

$$\kappa_\theta \theta_k^\psi = 0 \le m_k(x_k^{\mathrm{N}}) - m_k(x_k + s_k) = m_k(x_k) - m_k(x_k + s_k). \tag{2.22}$$

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 (2.22) 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

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

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

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

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 (2.13), 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, 1998, for an example).

Notice also that (2.20) ensures that the denominator of ρ_k in (2.21) will be strictly positive whenever θ_k is. If $\theta_k = 0$, then $x_k = x_k^{\rm N}$, and the denominator of (2.21) will be strictly positive unless x_k is a first-order critical point because of (2.16).

The attentive reader will have observed that we have defined n_{k+1} in Step 4 in the cases where iteration k is unsuccessful (just before branching back to Step 2), while we may not use it if the alternative step of Step 2 is then used at iteration $k+1$. This is to keep the expression of the algorithm as general as possible: a more restrictive version would impose a branch back to Step 3b from Step 4 if iteration k is unsuccessful, but this would then prevent the use of an alternative step at iteration $k+1$. We have chosen not to impose that restriction, but we obviously require that n_{k+1} is used in Step 3a whenever it has been set at iteration k, instead of recomputing it from scratch.

Finally, note that Step 6 allows a relatively wide choice of the new trust-region radius Δ_{k+1}. While the stated conditions are 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 (2.20) fails or holds. If (2.20) 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 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, (2.20) 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.

3 Convergence to First-Order Critical Points

We now prove that our hybrid algorithm generates a globally convergent sequence of iterates, at least if the restoration iteration always succeeds. 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,

$$\mathcal{R} = \left\{ \begin{array}{ll} k \mid & \text{Step 3 is executed and} \\ & \text{either TRQP}(x_k, \Delta_k) \text{ has no feasible point} \\ & \text{or } \|n_k\| > \kappa_\Delta \Delta_k \min[1, \kappa_\mu \Delta_k^\mu] \end{array} \right\},$$

the set of *restoration* iterations,

$$\mathcal{A} = \{k \mid \text{the alternative step is used at iteration } k\},$$

and

$$\mathcal{C} = \{k \mid s_k = n_k + t_k\},$$

the set of iterations where a composite step is used (with $\mu_k \geq 0$). Note that (2.19) implies that

$$\theta_k \leq \kappa_{\mathrm{u}} \Delta_k \min[1, \Delta_k^\mu] \leq \kappa_{\mathrm{u}} \Delta_k^{1+\mu}, \tag{3.1}$$

for every $k \in \mathcal{A}$. Also note that $\{1, 2, \ldots\} = \mathcal{A} \cup \mathcal{C} \cup \mathcal{R}$ and that $\mathcal{R} \subseteq \mathcal{Z}$.

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 $\kappa_{\mathrm{umh}} > 1$ such that

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

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

$$H_k = \nabla_{xx} f(x_k) + \sum_{i \in \mathcal{E} \cup \mathcal{I}} [y_k]_i \nabla_{xx} c_i(x_k), \tag{3.2}$$

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 (3.2) are replaced by bounded approximations.

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

$$|f(x_k + s_k) - m_k(x_k + s_k)| \leq \kappa_{\mathrm{ubh}} \Delta_k^2. \tag{3.3}$$

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

$$f^{\min} \leq f(x_k) \leq f^{\max} \text{ and } 0 \leq \theta_k \leq \theta^{\max} \tag{3.4}$$

for some constants $f^{\min}$, $f^{\max}$ 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

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

whose area, $\mathrm{surf}(\mathcal{M}_0)$, is clearly finite.

We also note the following simple consequence of (2.11) and AS3.

Lemma 3.1 Suppose that Algorithm 2.1 is applied to problem (1.1). Suppose also that (2.11) and AS3 hold, that $k \in \mathcal{C}$, and that

$$\theta_k \leq \delta_n.$$

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

$$\kappa_{\rm lsc}\theta_k \leq \|n_k\|. \tag{3.5}$$

Proof. Since $k \in \mathcal{C}$, we first obtain that n_k exists (as a consequence of (2.11)), and define

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

that is the subset of most-violated constraints. From the definitions of θ_k in (2.2) and of the normal step in (2.7) we obtain, using the Cauchy-Schwartz inequality, that

$$\theta_k \leq |\langle \nabla_x c_j(x_k), n_k\rangle| \leq \|\nabla_x c_j(x_k)\| \, \|n_k\| \tag{3.6}$$

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 in (3.6). □

Our assumptions and the definition of χ_k in (2.14) 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

$$\lim_{i\to\infty} \theta_{k_i} = 0, \quad \lim_{\substack{i\to\infty \\ k_i\in\mathcal{C}}} \chi_{k_i} = 0 \text{ and } \lim_{\substack{i\to\infty \\ k_i\in\mathcal{A}}} \pi_{k_i} = 0. \tag{3.7}$$

Then every limit point of the sequence $\{x_{k_i}\}$ 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_*. If $\{k_\ell\}$ contains infinitely many indices of $\mathcal{A}$, the definition of π_k implies that x_* is a first-order critical point for problem (1.1). If this is not the case, the fact that $k_\ell \notin \mathcal{R}$ implies that n_{k_ℓ} satisfies (2.11) 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 (2.12) that $\{x^{\rm N}_{k_\ell}\}$ also converges to x_*. Since the minimization problem occuring in the definition of χ_{k_ℓ} (in (2.14)) is convex, we then obtain from classical perturbation theory (see, for instance, Fiacco, 1983, pp. 14-17), AS1 and the first part of (3.7) that

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

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

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 also 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

$$\theta_{k_i} \geq \epsilon \tag{3.8}$$

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 $\mathcal{M}_0$. But the area of each of these squares is $\gamma_\theta^2\epsilon^2$. Thus the set $\mathcal{M}_0$ is completely covered by at most a finite number of such squares. This puts a finite upper bound on the number of iterations in $\{k_j\}$, and the conclusion follows. $\square$

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

Lemma 3.4 Suppose that Algorithm 2.1 is applied to problem (1.1). Suppose also that AS1 and AS3 hold and that n_k satisfies (3.5) for $k \in \mathcal{C}$. Then there exists a constant $\kappa_{\rm ubt} > 0$ such that

$$\theta_k \leq \kappa_{\rm ubt} \Delta_k^{1+\mu} \tag{3.9}$$

and

$$\theta(x_k + s_k) \leq \kappa_{\rm ubt} \Delta_k^2. \tag{3.10}$$

for all $l \notin \mathcal{R}$.

Proof. Assume first that $k \in \mathcal{C}$ with $\mu_k = \mu$. Since $k \notin \mathcal{R}$, we have from (3.5) and (2.13) that

$$\kappa_{\rm lsc} \theta_k \leq \|n_k\| \leq \kappa_\Delta \kappa_\mu \Delta_k^{1+\mu}, \tag{3.11}$$

which gives (3.9). On the other hand, (3.1) implies that an inequality of the form (3.9) holds for $k \in \mathcal{A}$ or $k \in \mathcal{C}$ with $\mu_k = 0$. Now, for any k, the i-th constraint function at $x_k + s_k$ can be expressed as

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

for $i \in \mathcal{E} \cup \mathcal{I}$, where we have used AS1, 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 (2.9), the Cauchy-Schwartz inequality, and (2.8) we have that

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

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

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

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

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

This gives the desired bound for any

$$\kappa_{\rm ubt} \geq \max[\kappa_1, \kappa_{\rm u}, \kappa_\Delta \kappa_\mu / \kappa_{\rm lsc}].$$

$\square$

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 and (2.16) hold, that $k \in \mathcal{C}$, that, for some $\epsilon > 0$,

$$\chi_k \geq \epsilon \tag{3.12}$$

Suppose furthermore that

$$\Delta_k \leq \min\left[1, \frac{\epsilon}{\kappa_{\rm umh}}, \left(2\frac{\kappa_{\rm ubg}}{\kappa_{\rm umh}\kappa_\Delta\kappa_\mu}\right)^{\frac{1}{1+\mu}}, \left(\frac{\kappa_{\rm tmd}\epsilon}{4\kappa_{\rm ubg}\kappa_\Delta\kappa_\mu}\right)^{\frac{1}{\mu}}\right] \stackrel{\rm def}{=} \delta_m, \tag{3.13}$$

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

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

This last inequality also holds if $k \in \mathcal{A}$, if (3.13) holds and

$$\pi_k \geq \epsilon. \tag{3.14}$$

Proof. Assume first that $k \in \mathcal{C}$. We note that, by (2.16), AS2, (3.12) and (3.13),

$$m_k(x_k^{\rm N}) - m_k(x_k + s_k) \geq \kappa_{\rm tmd}\chi_k \min\left[\frac{\chi_k}{\kappa_{\rm umh}}, \Delta_k\right] \geq \kappa_{\rm tmd}\epsilon\Delta_k. \tag{3.15}$$

Now

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

and therefore, using the Cauchy-Schwartz inequality, AS2, (2.13) and (3.13) that

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

We thus conclude from this last inequality and (3.15) that the desired conclusion holds for $k \in \mathcal{C}$. If we now assume that $k \in \mathcal{A}$ (that is iteration k uses an alternative step), then (2.18), (3.13) and the inequality $\kappa_{\rm umh} \geq 1$ directly yields that

$$m_k(x_k) - m_k(x_k + s_k) \geq \kappa_{\rm tmd}\min[\epsilon, \Delta_k] \geq \tfrac{1}{2}\kappa_{\rm tmd}\Delta_k$$

as desired. □

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, (2.16) and (3.12) hold, that $k \notin \mathcal{R}$, and that

$$\Delta_k \leq \min\left[\delta_m, \frac{(1-\eta_2)\kappa_{\rm tmd}\epsilon}{2\kappa_{\rm ubh}}\right] \stackrel{\rm def}{=} \delta_\rho. \tag{3.16}$$

Then

$$\rho_k \geq \eta_2.$$

Proof. Using (2.21), (3.3), Lemma 3.5 and (3.16), we find that

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

from which the conclusion immediately follows. □

Note that this proof could easily be extended if the definition of ρ_k in (2.21) were altered to be of the form

$$\rho_k \stackrel{\rm def}{=} \frac{f(x_k) - f(x_k+s_k) + \Theta_k}{m_k(x_k) - m_k(x_k+s_k)} \tag{3.17}$$

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.

Now, we also show that the test (2.20) 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, (2.16) and (3.12) hold, that $k \notin \mathcal{R}$, that n_k satisfies (3.5) if $k \in \mathcal{C}$, and that

$$\Delta_k \leq \min\left[\delta_m, \left(\frac{\kappa_{\rm tmd}\epsilon}{2\kappa_\theta\kappa_{\rm ubt}^\psi}\right)^{\frac{1}{\psi(1+\mu)-1}}\right] \stackrel{\rm def}{=} \delta_f. \tag{3.18}$$

Then

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

Proof. This directly results from the inequalities

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

where we successively used Lemma 3.4, (3.18) 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, (2.16), (3.12) and (3.16) hold, that $k \notin \mathcal{R}$, that n_k satisfies (3.5) if $k \in \mathcal{C}$, and that

$$\theta_k \leq \kappa_{\rm ubt}^{-\frac{1}{\mu}} \left(\frac{\eta_2 \kappa_{\rm tmd} \epsilon}{2\gamma_\theta} \right)^{\frac{1+\mu}{\mu}} \stackrel{\rm def}{=} \delta_\theta. \tag{3.19}$$

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 (3.12), (3.16), $k \notin \mathcal{R}$ and n_k satisfies (3.5) for $k \in \mathcal{C}$—and (3.19), we obtain that

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

and the desired inequality follows. □

We now establish that if the trust-region radius and the constraint violation are both small at a non-critical 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 that AS1–AS3, (2.11), and (3.12) hold, that (2.16) holds for $k \notin \mathcal{R}$, and that

$$\Delta_k \leq \min \left[\gamma_0 \delta_\rho, \left(\frac{1}{\kappa_\mu} \right)^{\frac{1}{\mu}}, \left(\frac{\gamma_0^2 (1 - \gamma_\theta) \kappa_\Delta \kappa_\mu}{\kappa_{\rm usc} \kappa_{\rm ubt}} \right)^{\frac{1}{1-\mu}} \right]. \tag{3.20}$$

Suppose furthermore that

$$\theta_k \leq \min[\delta_\theta, \delta_n]. \tag{3.21}$$

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

Proof. If an alternative step is used at iteration k, then $k \notin \mathcal{R}$. Assume therefore that $k \notin \mathcal{A}$. Because $\theta_k \leq \delta_n$, we know from (2.11) and Lemma 3.1 that (2.11) and (3.5) hold. Moreover, since $\theta_k \leq \delta_\theta$, we have that (3.19) also holds. Assume, for the purpose of deriving a contradiction, that $k \in \mathcal{R}$, which implies that

$$\|n_k\| > \kappa_\Delta \kappa_\mu \Delta_k^{1+\mu}, \tag{3.22}$$

where we have used (2.13) and the fact that $\kappa_\mu \Delta_k^{\mu_k} \leq \kappa_\mu \Delta_k^\mu \leq 1$ because of (3.20). 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 (3.20), the fact that $\theta_k = \theta_{k-1}$, (2.11), and (3.19), we obtain that

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

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

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

But Lemma 3.4 and the mechanism of the algorithm then imply that

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

Combining this last bound with (3.22) and (2.11), we deduce that

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

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 (3.20), 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 (3.22), (2.11) and (3.10) that

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

which is again impossible because of (3.20) and because $(1-\gamma_\theta) < 1$. Hence our initial assumption (3.22) 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 and (2.11) hold and that (2.16) holds for $k \notin \mathcal{R}$. Suppose furthermore that $|\mathcal{Z}| = \infty$. Then there exists an infinite subsequence $\{k_j\} \subseteq \mathcal{Z}$ such that

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

and

$$\lim_{\substack{j\to\infty\\ k_j\in\mathcal{C}}} \chi_{k_j} = 0 \;\text{ and }\; \lim_{\substack{j\to\infty\\ k_j\in\mathcal{A}}} \pi_{k_j} = 0. \tag{3.24}$$

Proof. Let $\{k_i\}$ be any infinite subsequence of $\mathcal{Z}$. We observe that (3.23) follows from Lemma 3.3. Suppose now that, for some $\epsilon_2 > 0$.

$$\chi_{k_i} \geq \epsilon_2 \tag{3.25}$$

for all i such that $k_i \in \mathcal{C}$ and

$$\pi_{k_i} \geq \epsilon_2 \tag{3.26}$$

for all i such that $k_i \in \mathcal{A}$. Suppose furthermore that there exists $\epsilon_3 > 0$ such that, for all $i \geq i_0$,

$$\Delta_{k_i} \geq \epsilon_3. \tag{3.27}$$

If $k_i \notin \mathcal{A}$, (3.23) and (2.11) ensure that n_{k_i} exists for $i \geq i_0$, say, and also that

$$\lim_{i\to\infty} \|n_{k_i}\| = 0. \tag{3.28}$$

Thus (3.27) ensures that (2.13) holds for sufficiently large i and $k_i \notin \mathcal{R}$. We may then decompose the model decrease in its normal and tangential components, that is

$$m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i}+s_{k_i}) = m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i}^{\rm N}) + m_{k_i}(x_{k_i}^{\rm N}) - m_{k_i}(x_{k_i}+s_{k_i}). \tag{3.29}$$

Consider the normal component first. As we noted in the proof of Lemma 3.5,

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

which in turn, with (3.28), yields that

$$\lim_{i\to\infty} [m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i}^{\rm N})] = 0. \tag{3.30}$$

If we now consider the normal component, (3.25), (3.27) (2.16) and AS2 yield that

$$m_{k_i}(x_{k_i}^{\rm N}) - m_{k_i}(x_{k_i}+s_{k_i}) \geq \kappa_{\rm tmd}\epsilon_2 \min\left[\frac{\epsilon_2}{\kappa_{\rm umh}}, \epsilon_3\right] \stackrel{\rm def}{=} \delta_1 > 0. \tag{3.31}$$

Substituting (3.30) and (3.31) into (3.29), we find that, for $k_i \in \mathcal{C}$,

$$m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i} + s_{k_i}) \geq \delta_1 > 0.$$

If, on the other hand, $k_i \in \mathcal{A}$, then (3.26), (3.27) and (2.18) give that

$$m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i} + s_{k_i}) \geq \kappa_{\rm tmd} \min[\epsilon_2, \epsilon_3] \stackrel{\rm def}{=} \delta_2 > 0.$$

Thus

$$\liminf_{i \to \infty} [m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i} + s_{k_i})] \geq \min[\delta_1, \delta_2] \stackrel{\rm def}{=} \delta > 0. \tag{3.32}$$

We now observe that, because x_{k_i} is added to the filter at iteration k_i, the mechanism of the algorithm imposes that either iteration $k_i \in \mathcal{R}$ or (2.20) must fail. Since we already verified that $k_i \notin \mathcal{R}$ for $i \geq i_0$ sufficiently large, we obtain that (2.20) must fail for such i, that is

$$m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i} + s_{k_i}) < \kappa_\theta \theta_{k_i}^\psi. \tag{3.33}$$

Combining this bound with (3.32), we find that θ_{k_i} is bounded away from zero for i sufficiently large, which is impossible in view of (3.23). We therefore deduce that (3.27) 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 (3.18), (3.19) and (3.20) hold, which is possible by definition of the subsequence and because of (3.23). 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, (3.33) must hold for ℓ sufficiently large. However, we may also apply Lemma 3.7, which contradicts (3.33), and therefore (3.25) and (3.26) cannot hold together, yielding the desired result. □

Thus, if an infinite subsequence of iterates is added to the filter, Lemma 3.2 ensures that it converges to 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) and that finite termination does not occur. Suppose also that AS1–AS3 and (2.11) hold and that (2.16) holds for $k \notin \mathcal{R}$. Then we have that

$$\lim_{k \to \infty} \theta_k = 0. \tag{3.34}$$

Furthermore, n_k exists and satisfies (3.5) for all $k \geq k_0$ sufficiently large.

Proof. Consider any successful iterate with $k \geq k_0$. Then we have that

$$f(x_k) - f(x_{k+1}) \geq \eta_1[m_k(x_k) - m_k(x_k + s_k)] \geq \eta_1 \kappa_\theta \theta_k^\psi \geq 0. \tag{3.35}$$

Thus the objective function does not increase for all successful iterations with $k \geq k_0$. But AS1 and AS3 imply (3.4) and therefore we must have, from the first part of this statement, that

$$\lim_{\substack{k \in \mathcal{S} \\ k \to \infty}} f(x_k) - f(x_{k+1}) = 0. \tag{3.36}$$

(3.34) then immediately follows from (3.35) 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 (2.11) 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 3.12 Suppose that Algorithm 2.1 is applied to problem (1.1) and that finite termination does not occur. Suppose also that AS1—AS3 hold and that (2.16) holds for $k \notin \mathcal{R}$. Suppose furthermore that (3.12) 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 (3.21) holds and thus that (2.11) also holds 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

$$\Delta_j \leq \gamma_0 \min\left[\delta_\rho, \sqrt{\frac{(1-\gamma_\theta)\theta^{\rm F}}{\kappa_{\rm ubt}}}, \Delta_{k_1}\right] \stackrel{\rm def}{=} \gamma_0 \delta_s, \tag{3.37}$$

where

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

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 (3.37), 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 (3.37) imply that

$$\Delta_{j-1} \leq \frac{1}{\gamma_0} \Delta_j \leq \delta_s \tag{3.38}$$

and Lemma 3.6, which is applicable because (3.37) and (3.38) together imply (3.16) with k replaced by $j - 1$, then ensures that

$$\rho_{j-1} \geq \eta_2. \tag{3.39}$$

Furthermore, since $n - j - 1$ satisfies (2.11), Lemma 3.1 implies that we can apply Lemma 3.4. This together with (3.37) and (3.38), gives that

$$\theta(x_{j-1} + s_{j-1}) \le \kappa_{\rm ubt} \Delta_{j-1}^2 \le (1 - \gamma_\theta)\theta^{\rm F}. \tag{3.40}$$

We may also apply Lemma 3.8 because (3.37) and (3.38) ensure that (3.16) holds and because (3.19) also holds for $j - 1 \ge k_1$. Hence we deduce that

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

This last relation and (3.40) ensure that $x_{j-1} + s_{j-1}$ is acceptable for the filter and x_{j-1}. Combining this conclusion with (3.39) and the mechanism of the algorithm, we obtain that $\Delta_j \ge \Delta_{j-1}$. As a consequence, and since (2.20) also holds at iteration $j - 1$, iteration j cannot be the first iteration following k_1 for which (3.37) holds. This contradiction shows that $\Delta_k \ge \gamma_0 \delta_s$ 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 3.13 Suppose that Algorithm 2.1 is applied to problem (1.1) and that finite termination does not occur. Suppose also that AS1–AS3, (2.11) hold, and that (2.16) holds for $k \notin \mathcal{R}$. Then there exists a subsequence $\{k_j\}$ such that

$$\liminf_{\substack{j \to \infty \\ k_j \in \mathcal{C}}} \chi_{k_j} = 0 \quad \text{and} \quad \liminf_{\substack{j \to \infty \\ k_j \in \mathcal{A}}} \pi_{k_j} = 0. \tag{3.41}$$

Proof. We start by observing that Lemma 3.11 implies that the second conclusion of (2.11) holds for k sufficiently large. Moreover, as in Lemma 3.11, we obtain (3.35) and therefore (3.36) for each $k \in \mathcal{S}$, $k \ge k_0$. Suppose now, for the purpose of obtaining a contradiction, that (3.12) and (3.14) hold. Assume first that $k \in \mathcal{C}$. In this case, and notice that

$$m_k(x_k) - m_k(x_k + s_k) = m_k(x_k) - m_k(x_k^{\rm N}) + m_k(x_k^{\rm N}) - m_k(x_k + s_k). \tag{3.42}$$

Moreover, note, as in Lemma 3.5, that

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

which in turn yields that

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

because of Lemma 3.11 and the second conclusion of (2.11). This limit, together with (3.35), (3.36) and (3.42), then gives that

$$\lim_{\substack{k\to\infty\\ k\in\mathcal{S}}} [m_k(x_k^{\rm N}) - m_k(x_k+s_k)] = 0. \tag{3.43}$$

But (2.16), (3.12), AS2 and Lemma 3.12 together imply that, for all $k \geq k_0$

$$m_k(x_k^{\rm N}) - m_k(x_k+s_k) \geq \kappa_{\rm tmd}\chi_k \min\left[\frac{\chi_k}{\beta_k}, \Delta_k\right] \geq \kappa_{\rm tmd}\epsilon \min\left[\frac{\epsilon}{\kappa_{\rm umh}}, \Delta_{\rm min}\right], \tag{3.44}$$

immediately giving a contradiction with (3.43).

On the other hand, if $k \in \mathcal{A}$, then (3.14) and (2.18) immediately imply that

$$m_k(x_k) - m_k(x_k+s_k) \geq \kappa_{\rm tmd} \min[\epsilon, \Delta_{\rm min}] > 0,$$

which, together with (2.21) and the fact that $k \in \mathcal{S}$, contradicts the boundedness of f. Hence (3.12) and (3.14) cannot hold together and the desired result follows. □

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–AS3 and (2.11) hold, and that (2.16) 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 (2.17), 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 $\{k_j\}$,

$$\lim_{j\to\infty} \theta_{k_j} = \lim_{\substack{j\to\infty\\ k_j\in\mathcal{C}}} \chi_{k_j} = \lim_{\substack{j\to\infty\\ k_j\in\mathcal{A}}} \pi_{k_j} = 0. \tag{3.45}$$

The conclusion then follows from Lemma 3.2. □

Can we dispense with AS3 to obtain this result? Firstly, this assumption ensures that the objective and constraint functions remain bounded above and below (see (3.4)). This is crucial for the rest of the analysis because the convergence of the iterates to feasibility depends on the fact that the area of the filter is finite. Thus, if AS3 does not hold, we have to verify that (3.4) 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

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

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 (2.20). If $\mathcal{C}(\theta^{\max})$ is bounded, AS3 is most certainly satisfied. If this is not the case, we could assume that

$$f^{\min} \leq f(x) \leq f^{\max} \text{ and } 0 \leq \theta(x) \leq \theta^{\max} \text{ for } x \in \mathcal{C}(\theta^{\max}) \tag{3.46}$$

for some values of $f^{\min}$ and $f^{\max}$ 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 (3.46) holds.

The reader should note that AS4 no longer ensures the existence of a limit point, but only that (3.45) holds for some subsequence $\{k_j\}$. Furthermore, the comments following the statement of (2.11) no longer apply if limit points at infinity are allowed.

4 Conclusion and Perspectives

We have a introduced a hybrid trust-region SQP-filter algorithm for general nonlinear programming, that mixes composite steps with potentially cheaper alternative steps, and we have shown this algorithm to be globally convergent to first-order critical points. This hybrid algorithm has the potential of being numerically more efficient than its version that only uses composite steps, as analyzed in Fletcher et al. (1999). However, the authors are well aware that this potential must be confirmed by numerical experiments.

References

L. T. Biegler, J. Nocedal, and C. Schmid. A reduced Hessian method for large-scale constrained optimization. *SIAM Journal on Optimization*, **5**(2), 314–347, 1995.

R. H. Bielschowsky and F. A. M. Gomes. Dynamical control of infeasibility in nonlinearly constrained optimization. Presentation at the Optimization 98 Conference, Coimbra, 1998.

R. H. Byrd, J. Ch. Gilbert, and J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming, Series A*, **89**(1), 149–186, 2000*a*.

R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large scale nonlinear programming. *SIAM Journal on Optimization*, **9**(4), 877–900, 2000*b*.

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, D. Orban, and Ph. L. Toint. A primal-dual trust-region algorithm for minimizing a non-convex function subject to bound and linear equality constraints. *Mathematical Programming*, **87**(2), 215–249, 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.

R. S. Dembo and U. Tulowitzki. On the minimization of quadratic functions subject to box constraints. School of Organization and Management Working paper Series B no. 71, Yale University, Yale, USA, 1983.

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-Alem. Global convergence without the assumption of linear independence for a trust-region algorithm for constrained optimization. *Journal of Optimization Theory and Applications*, **87**(3), 563–577, 1995.

M. El-Alem. A global convergence theory for a general class of trust-region-based algorithms for constrained optimization without assuming regularity. *SIAM Journal on Optimization*, **9**(4), 965–990, 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.

A. V. Fiacco. *Introduction to sensitivity and stability analysis in nonlinear programming.* Academic Press, London, 1983.

R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. Numerical Analysis Report NA/171, Department of Mathematics, University of Dundee, Dundee, Scotland, 1997.

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, N. I. M. Gould, S. Leyffer, and Ph. L. Toint. Global convergence of trust-region SQP-filter algorithms for nonlinear programming. Technical Report 99/03, Department of Mathematics, University of Namur, Namur, Belgium, 1999.

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 SQP-filter algorithm. Technical Report 00/??, Department of Mathematics, University of Namur, Namur, Belgium, 2000.

N. I. M. Gould, M. E. Hribar, and J. Nocedal. On the solution of equality constrained quadratic problems arising in optimization. Technical Report RAL-TR-98-069, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 1998.

M. Lalee, J. Nocedal, and T. D. Plantenga. On the implementation of an algorithm for large-scale equality constrained optimization. *SIAM Journal on Optimization*, **8**(3), 682–706, 1998.

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.

J. J. Moré and G. Toraldo. On the solution of large quadratic programming problems with bound constraints. *SIAM Journal on Optimization*, **1**(1), 93–113, 1991.

W. Murray and F. J. Prieto. A sequential quadratic programming algorithm using an incomplete solution of the subproblem. *SIAM Journal on Optimization*, **5**(3), 590–640, 1995.

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.

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

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.