On the Global Convergence of an SLP–Filter Algorithm

Roger Fletcher*, Sven Leyffer* and Philippe L. Toint†

*Department of Mathematics, University of Dundee, Dundee DD1 4HN, Scotland, UK.
†Department of Mathematics, University of Namur, 61 rue de Bruxelles, B-5000 Namur, Belgium.

Numerical Analysis Report NA/183, August 1998, (revised October 1999)

Abstract

A mechanism for proving global convergence in filter–type methods for nonlinear programming is described. Such methods are characterized by their use of the dominance concept of multiobjective optimization, instead of a penalty parameter whose adjustment can be problematic. The main point of interest is to demonstrate how convergence for NLP can be induced without forcing sufficient descent in a penalty-type merit function.

The proof technique is presented in a fairly basic context, but the ideas involved are likely to be more widely applicable. The technique allows a range of specific algorithm choices associated with updating the trust region radius and with feasibility restoration.

Keywords nonlinear programming, global convergence, filter, multiobjective optimization, SLP.

1 Introduction

In Fletcher and Leyffer [2] a new technique for globalizing methods for nonlinear programming (NLP) is presented. The idea is referred to as an NLP filter and is motivated by the aim of avoiding the need to choose penalty parameters, such as would occur with the use of l_1 penalty functions or augmented Lagrangian functions. Numerical experience with the technique in a Sequential Quadratic Programming (SQP) trust region algorithm is reported in [2] and is very promising. However, no global convergence proof is given in [2], although a number of heuristics are suggested to eliminate obvious situations in which the method might fail to converge.

This paper shows that the filter technique does provide a mechanism for forcing global convergence when used in an appropriate way. To present this result we consider a more basic situation in which a Sequential Linear Programming (SLP) algorithm is used in conjunction with a trust region and filter. A proof of global convergence is presented when the algorithm is used to solve NLP problems with just inequality constraints. It is hoped that this proof mechanism can be carried over to an SQP context, although there are a number of difficulties to be circumvented which are discussed further in Section 4. However, because our global result concerns first order conditions, it seems likely that little is lost by considering the SLP algorithm. The extension to include equality constraints should be straightforward, albeit at the expense of technical complications which would obscure the presentation.

The proposed algorithm contains an inner iteration for calculating a suitable trust region radius. In some ways this resembles the use of a backtracking line search. The motivation for this is that it provides certain conditions used in the convergence proof. To a large extent however, the approach allows conventional ideas to be used of halving or doubling (say) the previous trust region radius.

An interesting feature of the proof is that various of the heuristics used in [2] are shown to be unnecessary. These include the NW corner rule, the need to unblock the filter in some cases, and the consequent decision to reduce the strict upper bound on constraint infeasibility. In this paper we do use a feasibility restoration technique, but are not prescriptive as to how this is done. Another feature of some interest is that some points may be accepted by the algorithm, without a new entry in the filter being made. This contributes to the non-monotonic properties of the algorithm.

2 An SLP–filter Algorithm

In this paper we consider an NLP problem of the form

$$P \left\{ \begin{array}{ll} \underset{\mathbf{x}\in\mathbb{R}^n}{\text{minimize}} & f(\mathbf{x}) \\ \text{subject to} & c_i(\mathbf{x}) \le 0 \quad i = 1, 2, \ldots, m, \end{array} \right.$$

and refer to a local solution by $\mathbf{x}^*$. We assume for the purposes of our convergence proof that the constraint set includes some linear constraints (for example simple upper and lower bounds on $\mathbf{x}$) that define a non–empty bounded region X. The LP subproblem in our algorithm depends upon the value of the current iterate $\mathbf{x}$ and trust region radius ρ, ($\rho > 0$), and is defined by

$$LP(\mathbf{x}, \rho) \left\{ \begin{array}{ll} \underset{\mathbf{d}\in\mathbb{R}^n}{\text{minimize}} & \mathbf{g}^T\mathbf{d} \\ \text{subject to} & c_i + \mathbf{a}_i^T\mathbf{d} \le 0 \quad i = 1, 2, \ldots, m \\ & \quad \|\mathbf{d}\|_\infty \le \rho, \end{array} \right.$$

where we denote $\mathbf{g} = \nabla f(\mathbf{x})$, $c_i = c_i(\mathbf{x})$ and $\mathbf{a}_i = \nabla c_i(\mathbf{x})$. The l_∞ norm is used to define the trust region because it is readily implemented by adding simple bounds to the LP subproblem. Let $\mathbf{d}$ denote the solution (if it exists) of $LP(\mathbf{x}, \rho)$. Then we denote

$$\Delta l = -\mathbf{g}^T\mathbf{d}$$

as the *predicted reduction* in $f(\mathbf{x})$ and

$$\Delta f = f(\mathbf{x}) - f(\mathbf{x} + \mathbf{d})$$

as the *actual reduction* in $f(\mathbf{x})$. The measure of constraint infeasibility which we use in this paper is

$$h(\mathbf{c}) = ||\mathbf{c}^+||_\infty$$

where $c_i^+ = \max(0, c_i)$. This is different from the measure used in [2], although there is no compelling reason for this. The infinity norm is chosen merely to simplify the presentation of the convergence proof.

We now turn to the definition of an NLP filter as introduced in [2]. The two aims in an NLP problem are to minimize $f(\mathbf{x})$, and to satisfy the constraints, that is to minimize $h(\mathbf{c}(\mathbf{x}))$. In a filter we consider pairs of values (h, f) obtained by evaluating $h(\mathbf{c}(\mathbf{x}))$ and $f(\mathbf{x})$ for various values of $\mathbf{x}$. A pair (h_i, f_i) is said to *dominate* another pair (h_j, f_j) if and only if both $h_i \le h_j$ and $f_i \le f_j$, indicating that the former point is at least as good as the latter in respect of both measures. The NLP filter is defined to be a list of pairs (h_i, f_i) such that no pair dominates any other. This is illustrated in Figure 1. A point $\mathbf{x}$ is said to be "acceptable for inclusion in the filter" if its (h, f) pair is not dominated by any entry in the filter. This is the condition that

$$\text{either} \qquad h < h_i \qquad \text{or} \qquad f < f_i$$

for all $i \in \mathcal{F}$, where $\mathcal{F}$ denotes the current set of filter entries. We may also wish to "include a point $\mathbf{x}$ in the filter", by which we mean that its (h, f) pair is added to the list of pairs in the filter, and any pairs in the filter that are dominated by the new pair are removed. We use the filter as an alternative to a penalty function as a means of deciding whether or not to accept a new point in an NLP algorithm.

In fact this definition of a filter is not adequate for proving convergence as it allows points to accumulate in the neighbourhood of a filter entry that has $h_i > 0$. This is readily corrected by defining a small envelope around the current filter in which points are not accepted. Thus the condition for a point being acceptable to the filter is that its (h, f) pair satisfies

$$\text{either} \qquad h \le \beta h_i \qquad \text{or} \qquad f \le f_i - \gamma h_i \tag{2.1}$$

for all $i \in \mathcal{F}$, where β and γ are preset parameters such that $1 > \beta > \gamma > 0$, with β close to 1 and γ close to zero. The first of these inequalities is just sufficient reduction in h. The dependence on h_i in the second inequality is an artefact to enable convergence to be proved. Because γ is small the extra term has little practical impact. The desirability of this condition is discussed in more detail in Section 4. The role of the envelope in forcing convergence to a feasible point is demonstrated below in Lemma 3. It is also convenient to allow an upper bound

$$h(\mathbf{c}(\mathbf{x})) \le \beta u \tag{2.2}$$

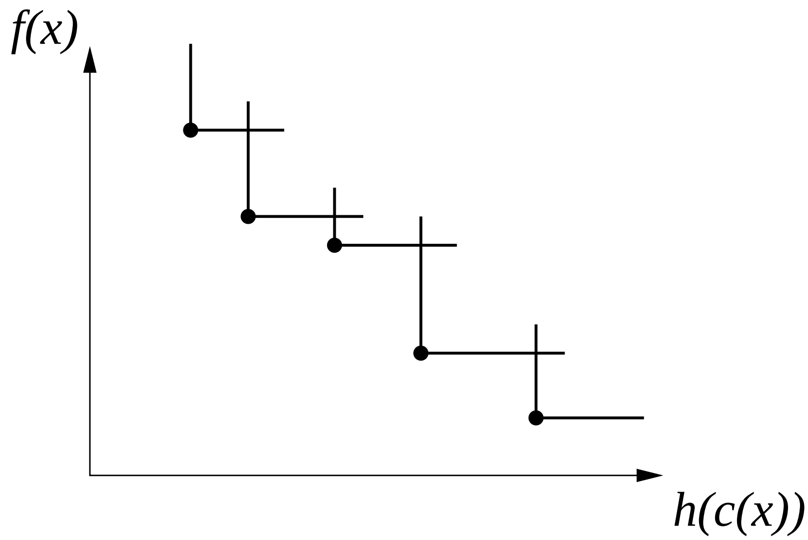

Figure 1: An NLP Filter

($u > 0$) on constraint infeasibility, and this is readily implemented by initializing the filter with the entry $(u, -\infty)$.

A common feature in a trust region algorithm for unconstrained minimization is the use of a sufficient reduction criterion

$$\Delta f \geq \sigma \Delta l, \tag{2.3}$$

where Δl is positive, and $\sigma \in [0, 1)$ is a preset parameter. However, in an NLP algorithm, Δl may be negative or even zero, in which case this test is no longer appropriate. A new feature of the algorithm in this paper is that it uses (2.3) (with $\sigma \geq \gamma$) only when Δl is sufficiently positive, which we shall measure by the inequality

$$\Delta l \geq \delta h^2, \tag{2.4}$$

where h refers to $h(\mathbf{c}(\mathbf{x}))$ evaluated at the current point, and $\delta > 0$ is a preset parameter that is close to zero. Again the dependence of the right hand side of (2.4) on h is an artefact to enable convergence to be proved.

We are now in a position to state our SLP–filter algorithm, which we do by means of the flow diagram of Figure 3. The sequence of points accepted by the algorithm is referred to by $\{\mathbf{x}^{(k)}\}$, and quantities derived from $\mathbf{x}^{(k)}$ are superscripted in a similar manner, for example $h^{(k)}$ refers to $h(\mathbf{c}(\mathbf{x}^{(k)}))$ and $f^{(k)}$ to $f(\mathbf{x}^{(k)})$. Also $\mathcal{F}^{(k)}$ refers to the set of filter entries at the start of iteration k. It can be seen in Figure 3 that there is an inner loop in which the trust region radius ρ is successively reduced until either certain tests are satisfied, or the current LP subproblem becomes incompatible (for clarity we avoid the use of the word 'infeasible' in this context). The inner loop is initialized with any value of $\rho \geq \rho^\circ$, where $\rho^\circ > 0$ is a preset parameter. We may assume without loss of generality that $\rho^\circ \leq |X|$ (the diameter of the region X defined by the set of linear constraints). The inner loop chooses a decreasing geometric sequence of values of ρ and generates corresponding values of $\mathbf{d}$, Δl and Δf (unsubscripted). The inner loop contains a test "is $\mathbf{x}^{(k)} + \mathbf{d}$ acceptable to the filter and $(h^{(k)}, f^{(k)})$". By this we mean that $\mathbf{x}^{(k)} + \mathbf{d}$ has to be acceptable to the filter formed of the current filter and $(h^{(k)}, f^{(k)})$, so that if $(h^{(k)}, f^{(k)})$ is subsequently entered into the filter, then $(h^{(k+1)}, f^{(k+1)})$ will still be acceptable to the new filter. When the inner iteration terminates, the current values of ρ, $\mathbf{d}$, Δl and Δf are denoted respectively by $\rho^{(k)}$, $\mathbf{d}^{(k)}$, $\Delta l^{(k)}$ and $\Delta f^{(k)}$. We observe that all points that are generated by the algorithm lie in the region X (a consequence of linearity).

Following our multiobjective thinking, we regard an iterate that satisfies (2.4) as being an *f–type iteration* (having the primary aim of improving f, and possibly allowing an increase in h). In this case we insist that the sufficient reduction condition (2.3) is satisfied. Thus the condition for iteration k to be an f–type iteration is that both

$$\Delta f^{(k)} \geq \sigma \Delta l^{(k)} \qquad \text{and} \qquad \Delta l^{(k)} \geq \delta (h^{(k)})^2 \tag{2.5}$$

are satisfied. If (2.4) is not satisfied, or if the current LP subproblem is incompatible, then the primary aim of the iteration is to reduce h (possibly allowing an increase in f) and we refer to the resulting iteration as an *h–type iteration.* As ρ is reduced in the inner loop, the value of Δl is reduced and the status of (2.4) may go from true to false, but not vice-versa. Thus the inner loop always samples the possibility for an f–type iteration before that of an h–type iteration. A key argument in the convergence proof is to show that if the limit point is not a Fritz–John point, and k is sufficiently large, then an f–type iteration will always be selected.

This algorithm differs in one important respect from that in [2] in that not all points $\mathbf{x}^{(k)}$ are included in the filter, even though they are acceptable to the filter. Also the current point $\mathbf{x}^{(k)}$ is not included in the current filter $\mathcal{F}^{(k)}$ but must be acceptable to it. The point $\mathbf{x}^{(k)}$ is included in the filter at the end of the iteration if and only if that iteration is an h–type iteration. A consequence of this is that $\mathbf{x}^{(k)} + \mathbf{d}$ need only be tested for acceptability to $(h^{(k)}, f^{(k)})$ if $\Delta l^{(k)} < \delta (h^{(k)})^2$. Another consequence is that all the current filter entries have $h_i > 0$, $i \in \mathcal{F}^{(k)}$. This is because if $h^{(k)} = 0$ then $LP(\mathbf{x}^{(k)}, \rho)$ must be compatible and also $\Delta l^{(k)} \geq 0$ occurs so that the test $\Delta l^{(k)} \geq \delta (h^{(k)})^2$ is always satisfied. Thus if $h^{(k)} = 0$, it follows that the resulting iteration is an f-type iteration and $\mathbf{x}^{(k)}$ is not entered into the filter. It is convenient to denote

$$\tau^{(k)} = \min_{i \in \mathcal{F}^{(k)}} h_i > 0. \tag{2.6}$$

It can be seen that our algorithm includes the provision for a feasibility restoration phase ("compatibility restoration" would be a more accurate description) if the current LP subproblem becomes incompatible. Any method for solving a nonlinear algebraic system of inequalities can be used to implement this calculation, such as for example a Newton-like scheme for minimizing $h(\mathbf{c}(\mathbf{x}))$. The restoration phase terminates if it finds a point that is both acceptable to the filter, and for which $LP(\mathbf{x}, \rho)$ is compatible for some $\rho \geq \rho^{\circ}$. (Essentially the latter condition only requires that $LP(\mathbf{x}^{(k)}, \infty)$ is compatible, since we can always take $\rho = \infty$ (or $\rho = |X|$)). There are various existing algorithms that might be used to implement this calculation: that of Madsen [5] (with suitable changes to include inequality constraints) has a convergence proof and is close to the spirit of this paper. Alternatively we can make use of the ideas expressed in [2]. Note that the restoration phase makes no demands on the resulting value of $f(\mathbf{x})$, which could be significantly worse than that at the previous point. If the restoration phase does terminate, then the point of termination becomes $\mathbf{x}^{(k+1)}$ and the resulting step from $\mathbf{x}^{(k)}$ to $\mathbf{x}^{(k+1)}$ is deemed to be an h–type iteration.

Of course is it not always possible to find a point which satisfies both the above conditions, and the restoration phase might converge to an infeasible point, for example a non–zero local minimum of $h(\mathbf{c}(\mathbf{x}))$. This is often an indication that the original problem P is incompatible. If, on the other hand, the restoration phase is converging to a feasible point, then it is usually able to terminate. This is so because $LP(\mathbf{x}, \infty)$ is usually compatible if $\mathbf{x}$ is sufficiently close to the feasible region, and because $\tau^{(k)} > 0$ allows such a point to be acceptable to the filter. However this outcome is not guaranteed, as it is possible for $LP(\mathbf{x}, \infty)$ to be incompatible for any infeasible point $\mathbf{x}$. Such an example is the pathological problem $\min(x_2 - 1)^2$ subject to $x_1 \leq 0$ and $x_1^2 \geq 0$, starting from $\mathbf{x} = (1, 0)^T$. A Newton-like iteration is likely to converge to the feasible point $\mathbf{x} = \mathbf{0}$, which is not a solution of the NLP, without finding a point at which the LP subproblem is compatible. However such a pathological problem (P) has the property that there exists an arbitrary small perturbation to P for which P is incompatible. Thus in this paper we content ourselves with the possibility that the restoration phase may fail to terminate, and regard this as an indication that P is incompatible (in a local sense) to within round-off error.

3 A Global Convergence Proof

In this section we present a proof of global convergence of the SLP–filter algorithm of Figure 3 when applied to problem P. We assume only that the set X defined by the linear constraints is non–empty and bounded, and that the problem functions $f(\mathbf{x})$ and $\mathbf{c}(\mathbf{x})$ are twice continuously differentiable on X.

When the algorithm is applied, one of four different possible outcomes can occur, which we itemize as follows.

($\mathbb{A}$) The restoration phase iterates infinitely and fails to find a point $\mathbf{x}$ which is acceptable to the filter and for which $LP(\mathbf{x}, \rho)$ is compatible for some $\rho \geq \rho^{\circ}$.

($\mathbb{B}$) A Kuhn–Tucker point is found ($\mathbf{d} = \mathbf{0}$ solves $LP(\mathbf{x}^{(k)}, \rho)$ for some k).

($\mathbb{C}$) All iterations are f–type iterations for k sufficiently large, and we consider the subsequence of consecutive f–type iterations.

($\mathbb{D}$) There exists an infinite subsequence of h–type iterations.

Note in case ($\mathbb{C}$) that there are only a finite number of h–type iterations. An example of case ($\mathbb{C}$) behaviour occurs when $h^{(k)} = 0$ for all k sufficiently large, such as when the iterates are converging to an unconstrained stationary point (or to a KT point at which only linear constraints are active).

Our global convergence theorem concerns Fritz–John necessary conditions and we first review the theory associated with these (see for example Bazaraa and Shetty [1] for a simple exposition). Necessary conditions for $\mathbf{x}^*$ to solve P are that $\mathbf{x}^*$ is a feasible point, and that the set of directions

$$\{\mathbf{s} \mid \mathbf{s}^T\mathbf{g}^* < 0 \tag{3.1}$$
$$\mathbf{s}^T\mathbf{a}_i^* < 0 \quad i \in \mathcal{A}^*\} \tag{3.2}$$

is empty, where $\mathcal{A}^*$ denotes the set of active constraints at $\mathbf{x}^*$. These are the conditions that $\mathbf{s}$ is respectively a *descent direction* and an *interior feasible direction*. When $\mathbf{x}^*$ is a feasible point at which the set in (3.1) and (3.2) is empty, we shall refer to $\mathbf{x}^*$ as an *FJ point*. That the set in (3.1) and (3.2) is empty is equivalent to the existence of multipliers, but we do not need that result in our presentation.

What we do need for our theorem is that if $\mathbf{x}^\infty$ is a feasible point, but not an FJ point, then there exists an $\varepsilon > 0$ and a vector $\mathbf{s}$ such that $\|\mathbf{s}\|_\infty = 1$ for which

$$\mathbf{s}^T\mathbf{g}(\mathbf{x}) \le -\varepsilon \tag{3.3}$$
$$\mathbf{s}^T\mathbf{a}_i(\mathbf{x}) \le -\varepsilon \quad i \in \mathcal{A}^\infty \tag{3.4}$$

for all $\mathbf{x}$ in some neighbourhood $\mathcal{N}^\infty$ of $\mathbf{x}^\infty$. This result is a staightforward consequence of (3.1), (3.2) and continuity of the vectors $\mathbf{g}(\mathbf{x})$ and $\mathbf{a}_i(\mathbf{x})$. We are now in a position to prove global convergence of the SLP-filter algorithm.

The global convergence proof is divided into three parts. First, two useful results following from Taylor's theorem are stated in Lemma 1. Next, Lemma 2 shows that the inner iteration always terminates. Thus if ($\mathbb{A}$) or ($\mathbb{B}$) do not occur, then the iteration sequence must be either of type ($\mathbb{C}$) or ($\mathbb{D}$). Lemma 3 shows the role of the filter (plus envelope) in forcing convergence to a feasible point. Finally, Theorem 1 establishes global convergence of the algorithm by showing that the iterates have an accumulation point that is an FJ point. The standard assumptions that are made throughout are summarized as follows.

Standard Assumptions *The set X is non-empty and bounded and the functions $f(\mathbf{x})$ and $\mathbf{c}(\mathbf{x})$ are twice continuously differentiable on an open set containing X.*

Lemma 1 *Let the standard assumptions hold and let $M > 0$ be an upper bound on all the terms $\frac{1}{2}\mathbf{s}^T\nabla^2 f(\mathbf{x})\mathbf{s}$ and $\frac{1}{2}\mathbf{s}^T\nabla^2 c_i(\mathbf{x})\mathbf{s}$ for all $\mathbf{x} \in X$ and all vectors $\mathbf{s}$ such that $\|\mathbf{s}\|_\infty = 1$. Let $\mathbf{d} \ne \mathbf{0}$ solve $LP(\mathbf{x}^{(k)}, \rho)$. It then follows that*

$$c_i(\mathbf{x}^{(k)} + \mathbf{d}) \le \rho^2 M \qquad i = 1, 2, \ldots, m, \tag{3.5}$$

and that

$$\Delta f \geq \Delta l - \rho^2 M. \tag{3.6}$$

Proof By Taylor's theorem and feasibility of $\mathbf{d}$ in $LP(\mathbf{x}^{(k)}, \rho)$,

$$c_i(\mathbf{x}^{(k)} + \mathbf{d}) = c_i^{(k)} + \mathbf{a}_i^{(k)T}\mathbf{d} + \tfrac{1}{2}\mathbf{d}^T\nabla^2 c_i(\mathbf{y}_i)\mathbf{d} \leq \rho^2 M \qquad i = 1, 2, \ldots, m,$$

where $\mathbf{y}_i$ denotes some point on the line segment from $\mathbf{x}^{(k)}$ to $\mathbf{x}^{(k)} + \mathbf{d}$. Likewise

$$f(\mathbf{x}^{(k)} + \mathbf{d}) = f^{(k)} + \mathbf{g}^{(k)T}\mathbf{d} + \tfrac{1}{2}\mathbf{d}^T\nabla^2 f(\mathbf{y})\mathbf{d}$$

and hence by definition of $\Delta l = -\mathbf{g}^{(k)T}\mathbf{d}$,

$$\Delta f = \Delta l - \tfrac{1}{2}\mathbf{d}^T\nabla^2 f(\mathbf{y})\mathbf{d} \geq \Delta l - \rho^2 M.$$

q.e.d.

Next we examine what can happen for fixed $\mathbf{x}^{(k)}$ as ρ is reduced in the inner iteration, and we show that the inner iteration must terminate.

Lemma 2: *Let the standard assumptions hold, then the inner iteration terminates finitely.*

Proof Clearly, if the inner iteration does not terminate finitely then the rule for decreasing ρ ensures that $\rho \to 0$. Two cases need to be considered, depending on whether $h^{(k)} > 0$ or $h^{(k)} = 0$.

If $h^{(k)} > 0$ and i is an index for which $c_i^{(k)} = h^{(k)}$ then for all $\mathbf{d}$ such that $\|\mathbf{d}\|_\infty \leq \rho$ it readily follows that

$$c_i^{(k)} + \mathbf{a}_i^{(k)T}\mathbf{d} \geq c_i^{(k)} - \rho\|\mathbf{a}_i^{(k)}\|_1 > 0$$

if $\rho < c_i^{(k)}/\|\mathbf{a}_i^{(k)}\|_1$. Thus for sufficiently small ρ, constraint i cannot be satisfied and $LP(\mathbf{x}^{(k)}, \rho)$ is incompatible. Note that if $\|\mathbf{a}_i^{(k)}\|_1 = 0$, then the constraint i cannot be satisfied for any ρ. Thus the inner iteration terminates in this case.

If $h^{(k)} = 0$, then by a similar argument, inactive constraints at $\mathbf{x}^{(k)}$ are inactive at the solution of $LP(\mathbf{x}^{(k)}, \rho)$ for sufficiently small ρ. In this case $LP(\mathbf{x}^{(k)}, \rho)$ is equivalent to the problem

$$\begin{array}{ll} \text{minimize} & \mathbf{g}^{(k)T}\mathbf{d} \\ \text{subject to} & \mathbf{a}_i^{(k)T}\mathbf{d} \leq 0 \qquad i \in \mathcal{A}^{(k)} \\ & \|\mathbf{d}\|_\infty \leq \rho, \end{array}$$

where $\mathcal{A}^{(k)}$ denotes the set of active constraints at $\mathbf{x}^{(k)}$. On dividing each expression by ρ we see that $\mathbf{d}$ solves this problem if and only if $\mathbf{s} = \mathbf{d}/\rho$ solves the problem

$$\begin{array}{ll} \text{minimize} & \mathbf{g}^{(k)T}\mathbf{s} \\ \text{subject to} & \mathbf{a}_i^{(k)T}\mathbf{s} \leq 0 \qquad i \in \mathcal{A}^{(k)} \\ & \|\mathbf{s}\|_\infty \leq 1. \end{array}$$

Thus, for sufficiently small ρ, $\Delta l = \eta\rho$ where $\eta = -\mathbf{g}^{(k)T}\mathbf{s}$ is independent of ρ. Because $\mathbf{x}^{(k)}$ is not stationary, it follows that $\eta > 0$. If $\rho \le (1-\sigma)\eta/M$, it then follows from (3.6) that $\Delta f \ge \sigma\Delta l$. Also if $\rho^2 \le \beta\tau^{(k)}/M$, it follows from (3.5) that $h(\mathbf{c}(\mathbf{x}^{(k)} + \mathbf{d})) \le \beta\tau^{(k)}$. Thus, for sufficiently small ρ, $\mathbf{x}^{(k)} + \mathbf{d}$ is acceptable to the filter, and the tests in (2.5) are both satisfied. Also, because $f(\mathbf{x}^{(k)} + \mathbf{d}) < f^{(k)}$, it follows that $\mathbf{x}^{(k)} + \mathbf{d}$ is acceptable to $(h^{(k)}, f^{(k)})$. Thus the inner iteration terminates in this case also, with an f–type step.
q.e.d.

The next result shows how the filter envelope contains a mechanism that forces the iterates towards a feasible point.

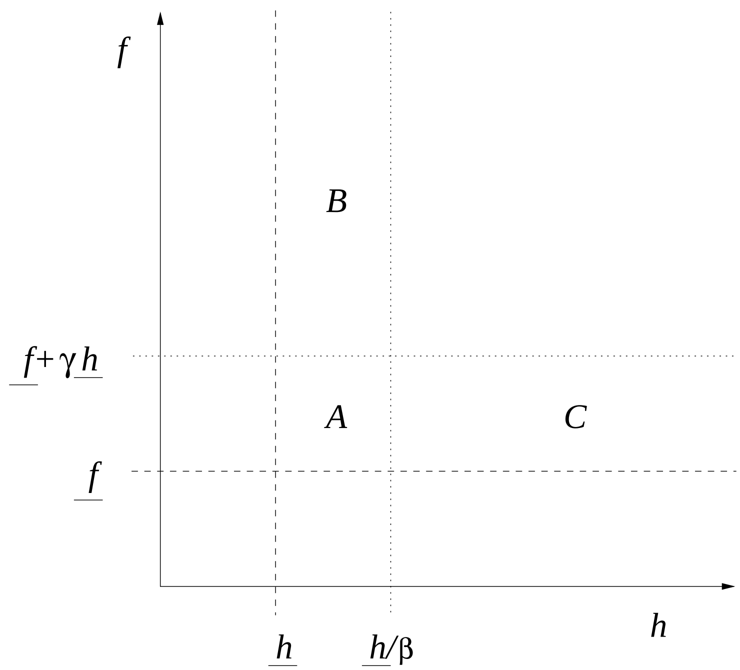

Figure 2: Construction for Lemma 3

Lemma 3: *Consider an infinite sequence of pairs (h, f) that are successfully entered into the filter using the envelope test (2.1). Then the sequence contains subsequences for which either $h \to 0$ or $f \to -\infty$ (or both).*

Proof If neither type of subsequence exists, then there exist constants $\underline{h} > 0$ and $\underline{f}$ such that both $h \ge \underline{h}$ and $f \ge \underline{f}$ for all entries in the main sequence. Assume that this is the case and consider the construction of Figure 2. If the sequence contains an entry in region A (with $h < \underline{h}/\beta$ and $f < \underline{f} + \gamma\underline{h}$) then the envelope test (2.1) excludes any further entries from being made in the filter. Thus the main sequence is finite, which is a contradiction. If the sequence contains an entry in region B (with $h < \underline{h}/\beta$), then it is no longer possible to satisfy the envelope test on h. Thus subsequent entries must reduce f by at least an amount $\gamma\underline{h}$. Since there is no subsequence for which $f \to -\infty$, we must again conclude that the main sequence is finite. Similarly, if the main sequence contains an entry in region C (with $f < \underline{f} + \gamma\underline{h}$), then it is no longer possible to satisfy the envelope test on f. Thus subsequent entries reduce h by a factor of at least β, again leading to the conclusion that the main sequence is finite.

We can now repeat the argument with $\underline{h}/\beta$ replacing $\underline{h}$ and $\underline{f} + \gamma\underline{h}$ replacing $\underline{f}$. By doing this repeatedly, we can extend the regions A, B and C until they include the initial filter entry, and hence conclude that the sequence is finite. This contradicts the fact that the sequence is infinite. *q.e.d.*

A consequence of Lemma 2 is that, if the SLP-filter algorithm does not terminate with ($\mathbb{A}$) or ($\mathbb{B}$), then the iteration sequence is either of type ($\mathbb{C}$) or ($\mathbb{D}$). The following theorem completes the global convergence proof by showing that the the iterates have an accumulation point that is an FJ point. The argument has two strands: first it is shown that the filter plus envelope ultimately forces some iterates to become arbitrarily close to the boundary of the feasible region. Then, if the iterates are not close to an FJ point, it is shown that an f–type iteration giving a significant reduction in $f(\mathbf{x})$ can always be found. We use this in two ways to contradict the possibility of a subsequence converging to a non–FJ point. In case ($\mathbb{C}$) we make use of an argument similar to that in unconstrained optimization, showing that the objective function would decrease without bound, which is excluded by the assumption that X is bounded. In case ($\mathbb{D}$), we have a contradiction to the fact that all iterations are h–type and it has not proved possible to find an f–type iteration. Thus we deduce that there exists an accumulation point that is an FJ point.

Theorem 1 *If the standard assumptions hold, then for the SLP–filter algorithm of Figure 3, either ($\mathbb{A}$) or ($\mathbb{B}$) occurs, or the iterates have an accumulation point that satisfies Fritz-John conditions.*

Proof We need only consider the case in which neither ($\mathbb{A}$) nor ($\mathbb{B}$) occurs. Because the inner loop of each iteration is finite (Lemma 2), the (outer) iteration sequence is infinite. All iterates $\mathbf{x}^{(k)}$ lie in X, so existence of an accumulation point follows because X is bounded. We shall identify the existence of an infinite subsequence, indexed by $k \in \mathcal{S}$, such that any accumulation point $\mathbf{x}^\infty$ of $\{\mathbf{x}^{(k)}\}$ for $k \in \mathcal{S}$ is an FJ point. First we show that $\mathbf{x}^\infty$ is a feasible point.

In case ($\mathbb{C}$) $\mathcal{S}$ is just the tail of the main sequence, comprising all the iterations after the last h–type iteration. We note that $f^{(k)}$ is monotonically decreasing for $k \in \mathcal{S}$, and is bounded below because $\mathbf{x}^{(k)} \in X$, so we can deduce that $\sum_{k\in\mathcal{S}} \Delta f^{(k)}$ is convergent. Because all iterations are of f-type, we have from (2.5) that

$$\Delta f^{(k)} \geq \sigma\delta(h^{(k)})^2.$$

Since $\sum_{k\in\mathcal{S}} \Delta f^{(k)}$ is convergent, it follows that $h^{(k)} \to 0$ for $k \in \mathcal{S}$. Thus $h^\infty = 0$ for any accumulation point (h^∞ denotes $h(\mathbf{c}(\mathbf{x}^\infty))$).

In case ($\mathbb{D}$) we start by considering the subsequence of all h–type iterations. This is the sequence of all iterations for which the pair $(h^{(k)}, f^{(k)})$ is entered into the filter. Because X is bounded, there can be no subsequence for which $f^{(k)} \to -\infty$. Thus we can invoke Lemma 3 to show that the subsequence of h–type iterations contains a thinner subsequence, indexed by $\mathcal{S}$, such that $h^{(k)} \to 0$ for $k \in \mathcal{S}$.

Now we consider the possibility that $\mathbf{x}^\infty$ is not an FJ point and show that this leads to a contradiction. Let $\mathbf{s}$ be a vector defined by (3.3) and (3.4) that exists for all $\mathbf{x} \in \mathcal{N}^\infty$. For sufficiently large k and any $\rho \le |X|$, we consider the effect of a step $\rho\mathbf{s}$ in $LP(\mathbf{x}^{(k)}, \rho)$.

For active constraints at $\mathbf{x}^\infty$ we have from (3.4) that

$$c_i^{(k)} + \rho\mathbf{a}_i^{(k)T}\mathbf{s} \le h^{(k)} - \rho\varepsilon \qquad i \in \mathcal{A}^\infty. \tag{3.7}$$

For inactive constraints $i \notin \mathcal{A}^\infty$, if $k \ge K$ is sufficiently large, then there exist positive constants $\bar{c}$ and $\bar{a}$, independent of k, such that

$$c_i^{(k)} \le -\bar{c} \qquad \text{and} \qquad \mathbf{a}_i^{(k)T}\mathbf{s} \le \bar{a},$$

by continuity of c_i and boundedness of $\mathbf{a}_i$ on X. It follows that

$$c_i^{(k)} + \rho\mathbf{a}_i^{(k)T}\mathbf{s} \le -\bar{c} + \rho\bar{a} \qquad i \notin \mathcal{A}^\infty. \tag{3.8}$$

Denoting $\kappa = \bar{c}/\bar{a}$, it follows for $k \ge K$ that if

$$h^{(k)}/\varepsilon \le \rho \le \kappa, \tag{3.9}$$

then from (3.7) and (3.8),

$$c_i^{(k)} + \rho\mathbf{a}_i^{(k)T}\mathbf{s} \le 0 \qquad i = 1, 2, \ldots, m.$$

Thus if (3.9) holds, we are assured that $\rho\mathbf{s}$ is a feasible step, and hence that $LP(\mathbf{x}^{(k)}, \rho)$ is a compatible subproblem. It also follows by optimality of $\mathbf{d}$ that

$$\Delta l \ge -\rho\mathbf{g}^{(k)T}\mathbf{s} \ge \rho\varepsilon \tag{3.10}$$

from (3.3).

If $\rho^2 \le \beta\tau^{(k)}/M$, it follows from (3.5) that $h(\mathbf{c}(\mathbf{x}^{(k)} + \mathbf{d})) \le \beta\tau^{(k)}$. Also we deduce from (3.6) and (3.10) that

$$\frac{\Delta f}{\Delta l} \ge 1 - \frac{\rho^2 M}{\Delta l} \ge 1 - \frac{\rho M}{\varepsilon},$$

so if $\rho \le (1-\sigma)\varepsilon/M$ it follows that $\Delta f \ge \sigma\Delta l$. Putting these results with (3.9), we see for sufficiently large k that if ρ satisfies

$$\frac{h^{(k)}}{\varepsilon} \le \rho \le \min\left\{\frac{(1-\sigma)\varepsilon}{M}, \sqrt{\frac{\beta\tau^{(k)}}{M}}, \kappa, |X|\right\} \tag{3.11}$$

then $LP(\mathbf{x}^{(k)}, \rho)$ is compatible, $h(\mathbf{c}(\mathbf{x}^{(k)} + \mathbf{d})) \leq \beta\tau^{(k)}$ and $\Delta f \geq \sigma\Delta l$. Also from (3.10) and (3.11), $\Delta l \geq h^{(k)}$ and hence $\Delta l \geq \delta(h^{(k)})^2$. Moreover, it follows from $\Delta f \geq \sigma\Delta l$, $\Delta l \geq h^{(k)}$ and $\sigma \geq \gamma$ that $f(\mathbf{x}^{(k)} + \mathbf{d}) \leq f^{(k)} - \gamma h^{(k)}$. Thus for sufficiently large $k \in \mathcal{S}$ there is a range of values (3.11) that guarantee that (i) $\mathbf{x}^{(k)} + \mathbf{d}$ is acceptable to the filter and to $(h^{(k)}, f^{(k)})$, and (ii) the conditions for an f–type step are satisfied.

If the subsequence $\mathcal{S}$ arises from case ($\mathbb{C}$), then $\tau^{(k)}$ is fixed and the right hand side of (3.11) is just a number, $\bar{\rho}$ say ($\bar{\rho} > 0$), whilst the left hand side converges to zero. Thus, as ρ is reduced in the inner loop, either it must eventually fall within this interval or a value to the right of the interval is accepted. Hence for sufficiently large k we can guarantee that a value $\rho^{(k)} > \frac{1}{2}\min(\bar{\rho}, \rho^\circ)$ will be chosen. We then deduce from (3.10) that $\Delta f^{(k)} > \frac{1}{2}\sigma\varepsilon\min(\bar{\rho}, \rho^\circ)$ which contradicts the fact that $\sum_{k\in\mathcal{S}} \Delta f^{(k)}$ is convergent.

Finally we look at case ($\mathbb{D}$). Because $h^{(k)} \to 0$ for $k \in \mathcal{S}$ it follows that $\tau^{(k)} \to 0$ and there is an infinite subsequence of $\mathcal{S}$ for which $\tau^{(k+1)} = h^{(k)} < \tau^{(k)}$. On this subsequence, for sufficiently large k, the range (3.11) becomes

$$\frac{h^{(k)}}{\varepsilon} \leq \rho \leq \sqrt{\frac{\beta\tau^{(k)}}{M}}. \tag{3.12}$$

In the limit, because $h^{(k)} < \tau^{(k)}$, the upper bound in (3.12) is more than twice the lower bound. Hence, as in the previous paragraph, reducing ρ in the inner loop will eventually locate a value in the interval (3.12), or to the right of this interval, which provides the conditions for an f–type step to occur. It is not possible for a larger value of ρ to produce an h–type step since Δl increases monotonically as ρ increases. Thus if $k \in \mathcal{S}$ is sufficiently large, an f–type step will be taken. This contradicts the fact that case ($\mathbb{D}$) is formed by a subsequence of h–type steps.

Thus a contradiction has been obtained in both case ($\mathbb{C}$) and case ($\mathbb{D}$) and the theorem is proved. *q.e.d.*

4 Discussion

Of course, the algorithm of Figure 3 is only a guide to what might be successfully implemented in practice, and is incomplete in various ways. For example, it is necessary to make a specfic choice of algorithm to implement the restoration phase. Also the rule for adjusting ρ in the inner iteration could be more intricate, based partly on interpolation. Another possibility is to allow the pair $(h^{(k)}, f^{(k)})$ to be entered into the filter if $h^{(k)} \geq \tau^{(k)}$ as this does not affect the convergence proof. Yet another possibility, when $\Delta l < \delta(h^{(k)})^2$, is to test the condition $h \leq h^{(k)}$ rather than testing for acceptability to $(h^{(k)}, f^{(k)})$. This is more in keeping with the aim of an h–type step being to provide an improvement in h.

The choice of an initial value of ρ for the inner iteration requires that the condition $\rho \geq \rho^\circ$ is satisfied, but is otherwise unspecific. We envisage that in practice ρ° is close to zero (say 10^{-4}) so that the effect of this restriction is small. Thus to a large extent the algorithm of Figure 3 allows the more usual trust region procedure in which one may double or halve (say) the value of ρ from the previous iteration, only setting $\rho = \rho^\circ$ if it would otherwise be less than ρ°. The potential danger of just taking ρ from the previous iteration is that the existence of a successful f-type step may not be recognised in the limit of the ($\mathbb{D}$) subsequence. The algorithm of Figure 3 ensures that the upper bound in (3.12) exceeds the lower bound by a factor of 2 or more if FJ conditions are not satisfied in the limit. By starting with $\rho \geq \rho^\circ$, we thus ensure that an f-type step will be taken. Adjusting the trust region in this sort of way has featured in other recent work, see for example [3], [4] and references contained therein.

Most important however is the need to include second order information, such as by using SQP steps rather than SLP steps. However, from the point of view of a convergence proof, this would introduce difficulties due to the possible existence of local minima in the QP subproblem, and the possibility of multiplier estimates becoming unbounded, which at present we cannot resolve, other than by assuming that these situations do not occur. Also it would be important to pay attention to the asymptotic behaviour of the algorithm to ensure that the second order convergence property of the SQP iteration is not compromised. It is not yet clear how best to do this. We hope to address these issues in our future work.

The referees for the paper both make the point that the links between f and $\mathbf{c}$ that are implicit in both (2.4) and the second part of (2.1) are undesirable. Aesthetically we agree that it would be preferable not to have such links, although we submit that their effect is minimal. We stress that γ and δ are intended to be close to zero (perhaps 10^{-4}) so that these tests are close to $\Delta l > 0$ and $f < f_i$ respectively, in which case there is no linkage. We have successfully implemented this type of algorithm in practice (in an SQP context), with results of similar quality to those in [2], and changing to $\Delta l > 0$ and $f < f_i$ causes negligible difference to the outcome. It may well be desirable to take the relative scaling of f and h into account, but this is readily done. Also it may be appropriate to modify (2.4) in some way, for example replacing h by $\min(1, h)$ or something similar. (In passing, we note that the test

$$\Delta l \geq \delta h$$

(in place of (2.4)) would also enable convergence to be proved. The squared term in (2.4) was introduced with an eye to SQP where it is possible that the predicted reduction is of order h^2.)

In any event it is by no means clear how to avoid the linkage between f and $\mathbf{c}$. The predicted step $\mathbf{d}$ is not be a descent direction for f when $\Delta l \leq 0$ so we cannot use any analogue of the Goldstein or Wolfe–Powell tests from unconstrained optimization. We feel that our proposals are noteworthy in that they enable a convergence proof to be made in such a way that the linkages between f and h are small and the impact on practical performance is negligible.

5 References

[1] Bazaraa M.S. and Shetty C.M. (1979), *NONLINEAR PROGRAMMING Theory and Algorithms*, John Wiley and Sons, New York.

[2] Fletcher R. and Leyffer S. (1997), Nonlinear Programming Without a Penalty Function, Dundee University, Dept. of Mathematics, Report NA/171.

[3] Jiang H., Fukushima M., Qi L. and Sun D. (1998), A Trust Region Method for Solving Generalized Complementarity Problems, *SIAM J. Optim.*, **8**, pp. 140–158.

[4] Kanzow Ch. and Zupke M. (1997), Inexact Trust-Region Methods for Nonlinear Complementarity Problems, University of Hamburg, Institute of Applied Mathematics, Report Number 127.

[5] Madsen K., An Algorithm for Minimax Solution of Overdetermined Systems of Nonlinear Equations, *J. Inst. Maths Applns.*, **16**, pp. 321–328.

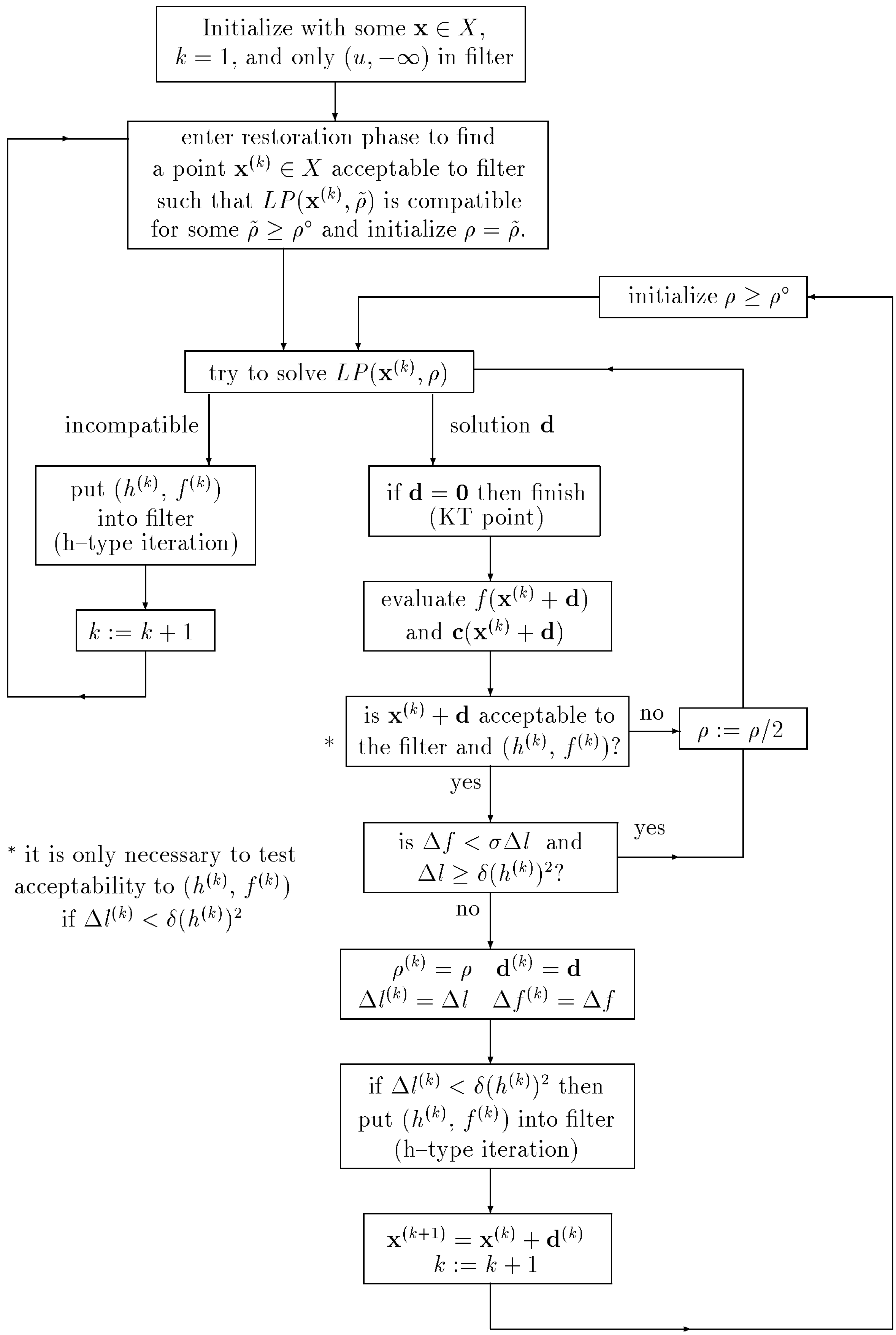

Figure 3: An SLP Filter Algorithm