SQP methods for large-scale nonlinear programming

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

Report 99/05 16 August 1999

Invited presentation at the 19th IFIP TC7 Conference
on System Modelling and Optimization,
July 12 - 16, 1999, Cambridge, England.

[1] Computational Science and Engineering Department, Rutherford Appleton Laboratory,
Chilton, Oxfordshire, OX11 0QX, England, EU.
Email : n.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

SQP methods for large-scale nonlinear programming

Nicholas I. M. Gould and Philippe L. Toint

August 16, 1999

Abstract

We compare and contrast a number of recent sequential quadratic programming (SQP) methods that have been proposed for the solution of large-scale nonlinear programming problems. Both line-search and trust-region approaches are considered, as are the implications of interior-point and quadratic programming methods.

1 Introduction

1.1 Perspectives

By the start of the 1980s, it was generally accepted that sequential quadratic programming (SQP) algorithms for solving nonlinear programming problems were the methods of choice. Such a view was based on strong convergence properties of such algorithms, and reinforced in the comparative testing experiments of Hock and Schittkowski (1981), in which SQP methods clearly outperformed their competitors. Although such claims of superiority were made for implementations specifically aimed at small-scale problems,—that is, those problems for which problem derivatives can be stored and manipulated as dense matrices—there was little reason to believe that similar methods would not be equally appropriate when the problem matrices were too large to be stored as dense matrices, but rather required sparse storage formats. Remarkably then, it is only in the latter part of the 1990s that SQP methods for sparse problems have started to appear in published software packages, while sparse variants of the methods that SQP was supposed to have superseded (for instance MINOS, see, Murtagh and Saunders, 1982, and LANCELOT, see, Conn, Gould and Toint, 1992) have been used routinely and successfully during the intervening years.

In our opinion, this curious divergence between what logically should have happened in the 1980s, and what actually came to pass may be attributed almost entirely to a single factor: quadratic programming (QP) methods (and their underlying sparse matrix technology) were not then capable of solving large problems. Witness the almost complete lack of software for solving large-scale (non-convex) quadratic programs even today, especially in view of the large number of available codes for the superficially similar linear programming problem.

The purpose of this paper is to survey modern SQP methods, and to suggest why at last that it is now reasonable to accept the widely-held view that SQP methods really are best. There have been a number of surveys of SQP methods over the past 20 years, and we refer the reader to Powell (1983, 1987), Boggs and Tolle (1995), Sargent (1995) and Conn, Gould and Toint (1997). Much of the material in this paper is covered in full detail in our forthcoming book on trust-region methods (Conn, Gould and Toint, 2000), which also contains a large number of additional references.

1.2 The problem

We consider the problem of minimizing a (linear or nonlinear) function f of n real variables x, for which the variables are required to satisfy a set of (linear or nonlinear) constraints $c_i(x) \geq 0$, $i = 1, \ldots, m$. For simplicity, we ignore the possibility that some of the constraints might be equations, since these are easily incorporated in what follows, nor shall we consider any special savings that can be made if some or all of the constraints have useful structure (e.g., might be linear). We remind the reader that if x_* is a local solution to the problem, and so long as a so-called constraint qualification holds to exclude pathological cases, it follows that the first-order criticality conditions

$$g(x_*) = A^T(x_*)y_*, \;\; c(x_*) \geq 0, \;\; y_* \geq 0 \;\text{ and }\; y_*^T c(x_*) = 0 \tag{1.1}$$

will hold. Here $c(x)$ is the vector whose components are the $c_i(x)$, $g(x) = \nabla_x f(x)$ is the gradient of f, $A(x) = \nabla_x c(x)$ is the Jacobian of c, and y_* are appropriate Lagrange multipliers. Notice that the first requirement in (1.1) is that the gradient $g(x,y) = \nabla_x \ell(x_*, y_*)$ of the Lagrangian function $\ell(x,y) = f(x) - y^T c(x)$ should vanish. For future reference, we also denote the Hessian of the Lagrangian function by $H(x,y) = \nabla_{xx} \ell(x,y)$, and will let c_- be the vector whose i-th component is $\min(c_i(x), 0)$, Throughout this paper, we shall make a blanket assumption that

A1. f and the c_i have twice Lipschitz-continuous derivatives (in the region of interest).

Throughout $\|\cdot\|$ will denote a generic norm. While there may be good practical reasons for choosing a specific norm, and while some of the given results have only been established in such a case, we suspect there are very few places where general results in arbitrary norms are not possible.

1.3 Generic SQP methods

For a most transparent derivation of the basic SQP method, we note that the final requirement in (1.1) (the complementarity condition) implies that Lagrange multipliers corresponding to inactive constraints (those for which $c_i(x_*) > 0$) must be zero. Thus (1.1) may equivalently be written as

$$g(x_*) - A_{\mathcal{A}_*}^T(x_*)(y_*)_{\mathcal{A}_*} = 0 \;\text{ and }\; c_{\mathcal{A}_*}(x_*) = 0, \tag{1.2}$$

where the subscript $\mathcal{A}_*$ indicates the components corresponding to the active set $\mathcal{A}_* = \{i \mid c_i(x_*) = 0\}$. Of course $\mathcal{A}_*$ depends on x_*, but suppose for the time being that we know $\mathcal{A}_*$. We then note that, if $\mathcal{A}_*$ has $m_{\mathcal{A}_*}$ elements, (1.2) is a set of $n+m_{\mathcal{A}_*}$ nonlinear equations in the $n+m_{\mathcal{A}_*}$ unknowns x and $y_{\mathcal{A}_*}$.

The best-known method for solving such systems (when it works) is Newton's method, and the basic SQP method is simply the Newton's iteration applied to (1.2). This leads to an iteration of the form

$$\begin{pmatrix} x_{k+1} \\ (y_{k+1})_{\mathcal{A}_*} \end{pmatrix} = \begin{pmatrix} x_k + s_k \\ (y_k)_{\mathcal{A}_*} + (v_k)_{\mathcal{A}_*} \end{pmatrix},$$

where

$$\begin{pmatrix} H_k & A_{\mathcal{A}_*}^T(x_k) \\ A_{\mathcal{A}_*}(x_k) & 0 \end{pmatrix} \begin{pmatrix} s_k \\ -(v_k)_{\mathcal{A}_*} \end{pmatrix} = -\begin{pmatrix} g(x_k) - A_{\mathcal{A}_*}^T(x_k)(y_k)_{\mathcal{A}_*} \\ c_{\mathcal{A}_*}(x_k) \end{pmatrix}, \tag{1.3}$$

to correct for the guess $(x_k,\ (y_k)_{\mathcal{A}_*})$. Here H_k is a "suitable" approximation of $H(x_k, y_k)$, where the nonzero components of y_k are those of $(y_k)_{\mathcal{A}_*}$. Since this is a Newton iteration, we can then expect a fast asymptotic convergence rate in many cases, so long as H_k is chosen appropriately. Interestingly, we do not require H_k to converge to $H(x_*, y_*)$ to obtain fast convergence, and considerable effort over the past 25 years has been devoted to obtaining minimal conditions, along with practical choices of H_k, which permit satisfactory convergence rates. We refer the interested reader to any of the previously-mentioned surveys, and the papers cited therein, for more details.

Most revealingly, we may rewrite (1.3) as

$$\begin{pmatrix} H_k & A^T_{\mathcal{A}_*}(x_k) \\ A_{\mathcal{A}_*}(x_k) & 0 \end{pmatrix} \begin{pmatrix} s_k \\ -(y_{k+1})_{\mathcal{A}_*} \end{pmatrix} = - \begin{pmatrix} g(x_k) \\ c_{\mathcal{A}_*}(x_k) \end{pmatrix},$$

which are the first-order criticality conditions for the (equality constrained) quadratic programming problem

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad s^T g(x_k) + \tfrac{1}{2} s^T H_k s \quad \text{subject to} \quad c_{\mathcal{A}_*}(x_k) + A_{\mathcal{A}_*}(x_k) s = 0,$$

with $(y_{k+1})_{\mathcal{A}_*}$ being its Lagrange multipliers. Notice that the constraints here are simply linearizations of the active constraints about the current estimate of the solution. But then, this suggests that to avoid having to estimate $\mathcal{A}_*$ in advance, it suffices to consider linearizations of all of the constraints, and to solve the (inequality constrained) quadratic programming problem

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad s^T g(x_k) + \tfrac{1}{2} s^T H_k s \quad \text{subject to} \quad c(x_k) + A(x_k) s \geq 0. \tag{1.4}$$

This provides the basic SQP method: given an estimate $(x_k,\ y_k)$ and a suitable H_k, solve (1.4) to find s_k, update $x_{k+1} = x_k + s_k$, and (if necessary) adjust y_{k+1} to guarantee that it converges to y_*. Remarkably, Robinson (1974) showed that, so long as x_0 is sufficiently close to x_*, H_0 is sufficiently close to $H(x_*, y_*)$, and $H_k = H(x_k, y_k)$ for $k \geq 1$, as well as

A2. the Jacobian of active constraints $A_{\mathcal{A}_*}(x_*)$ is of full rank,

A3. second-order necessary optimality conditions hold at $(x_*,\ y_*)$, and

A4. strict complementarity slackness occurs (i.e., $[y_*]_i > 0$ if $c_i(x_*) = 0$),

the SQP iteration converges Q-superlinearly, and the set of constraints which are active in (1.4) is precisely the set $\mathcal{A}_*$ for all sufficiently large k. If y_{k+1} are chosen to be the Lagrange multipliers for (1.4), the rate is actually Q-quadratic.

The important assumption here is A2, since this ensures that the Lagrange multipliers at x_*, as well as those for (1.4) for sufficiently large k, are unique. If $A_{\mathcal{A}_*}(x_*)$ is not of full rank, the limiting multipliers may not be unique, and the SQP method using the estimates obtained from (1.4) may not converge Q-quadratically. Of course A2 is a relatively strong first-order constraint qualification, and Wright (1999) shows that it is possible to replace this assumption by a weaker one due to Mangasarian and Fromovitz (1967) while still obtaining Q-quadratic convergence. To do so, the subproblem (1.4) must be modified slightly to ensure that its Lagrange multipliers are (locally) unique. In fact, Wright's subproblem is equivalent to that which would arise if an augmented Lagrangian function for (1.4) were minimized with respect to x and simultaneously maximized with respect to y while ensuring that $y \geq 0$. To ensure a Q-quadratic rate, the penalty parameter for the augmented Lagrangian must approach zero as $O\left(\max[\|x_k - x_*\|, \|y_k - y_*\|]\right)$. Bonnans and Launay (1992) and Hager (1998) show that it is also possible to remove A4 so long as A3 is strengthened.

Since the above iteration is essentially Newton's method, we must, of course, be cautious since such methods are not in general globally convergent. There have been traditionally two types of globalization schemes, linesearch and trust-region methods, and it is these that we now consider.

2 Linesearch methods

A traditional linesearch SQP method computes s_k to solve (1.4), and then obtains $x_{k+1} = x_k + \alpha_k s_k$ for some appropriately chosen stepsize α_k. The stepsize is selected so that x_{k+1} is closer in some way to a critical point than its predecessor, and linesearch methods achieve this by requiring that $\phi_k(x_{k+1})$ is significantly smaller than $\phi_k(x_k)$ for some so-called merit function ϕ_k. A highly desirable property of any merit function is that critical points of the merit function correspond to critical points for the underlying nonlinear programming problem. The most widely-used merit functions are non-smooth penalty functions,

$$\phi(x, \sigma) = f(x) + \sigma \|c(x)_-\|, \tag{2.1}$$

which depends on a positive penalty parameter σ, and the smooth exact penalty function, of which (variants of)

$$\phi(x, z, \sigma) = f(x) - y^T(x)(c(x) - z) + \sigma (c(x) - z)^T \left(A(x)A^T(x) + Z\right)^{-1} (c(x) - z), \tag{2.2}$$

where Z is a diagonal matrix with entries $z_i \geq 0$ and

$$\left(A(x)A^T(x) + Z\right) y(x) = A(x) \nabla_x f(x) \tag{2.3}$$

have proved to be popular. Relevant references include Pschenichny (1970), Fletcher (1973), Lucidi (1992), Di Pillo, Facchinei and Grippo (1992), and Boggs, Kearsley and Tolle (1999*a*). Note that none of these functions is actually ideal, since they may sometimes have critical points at values which do not correspond to those for the underlying nonlinear programming problem—these rogue values usually occur at points which are locally least infeasible. However, it can be shown that critical points for the two problems coincide so long as the former are feasible, and so long as the penalty parameter is smaller than a problem-dependent critical value—for smooth exact penalty functions, a further requirement like assumption A2 may also be required.

It is crucial that the SQP step s_k and the merit function $\phi(x)$ be compatible, in the sense that the directional derivative (slope in the smooth case) must be negative, for otherwise the linesearch may fail. In many cases, this condition is guaranteed when the penalty parameter is sufficiently large, and when $s_k^T H_k s_k \geq 0$. While the latter condition is likely to hold asymptotically, there is little reason why it should be true far from the solution, unless H_k is itself positive definite. For this reason, most active-set SQP methods work under the blanket assumption that H_k is positive definite, which is of course a far stronger assumption than A3.

When the function (2.1) is used, the penalty parameter may have to be as large as $\sigma > \|y_{k+1}\|_D$, where y_{k+1} are the Lagrange multipliers for (1.4) and $\|\cdot\|_D$ is the norm dual to $\|\cdot\|$. Such a condition is consistent with the problem-dependent critical value alluded to earlier, namely that $\sigma > \|y_*\|_D$.

An *a priori* bound on the size of the penalty parameter for (2.2) is harder to obtain, since it depends on the eigenvalues of H_k.

2.1 Second-order correction

The main disadvantage of functions like (2.1)—indeed, of any merit function which simply tries to balance f against constraint infeasibility—is that there is no guarantee that the SQP step together with a unit stepsize $\alpha_k = 1$ will lead to a reduction of the merit function, however close the iterates are to a critical point. Thus the full Newton (SQP) step may not be taken, and the iterates fail to converge at the anticipated Q-superlinear rate. Indeed, a famous example due to Maratos (1978) shows that this defect can actually occur. The Maratos effect happens because the linearization of the constraints fails to take adequate account of their nonlinear behaviour.

The idea of using a second-order correction to cope with the Maratos effect first appeared in a number of contemporary papers (see Mayne and Polak, 1982, Coleman and Conn, 1982, Fletcher, 1982*b*). The idea is to aim to replace the update $x_{k+1} = x_k + s_k$ by a corrected update

$$x_{k+1} = x_k + s_k + s_k^{\mathrm{C}}, \tag{2.4}$$

where s_k^{C} corrects for the "second-order" effects due to the constraint curvature. Let $\mathcal{A}_k$ be the set of active constraints at the solution to (1.4). Then a general second-order correction is the solution s_k^{C} to the system

$$\begin{pmatrix} H_k^{\mathrm{C}} & A_{\mathcal{A}_k}^T(x_k + p_k) \\ A_{\mathcal{A}_k}(x_k + p_k) & 0 \end{pmatrix} \begin{pmatrix} s_k^{\mathrm{C}} \\ -y_k^{\mathrm{CS}} \end{pmatrix} = - \begin{pmatrix} g_k^{\mathrm{C}} \\ c_{\mathcal{A}_k}(x_k + s_k) \end{pmatrix} \tag{2.5}$$

for some appropriate H_k^{C}, p_k and g_k^{C}. In order that the resulting step is suitable, we require that g^{C} and p are both small, indeed that

$$g_k^{\mathrm{C}} = O\left(\|x_k - x_*\| \max[\|x_k - x_*\|, \|y_k - y_*\|]\right) \text{ and } p_k = O(\|x_k - x_*\|). \tag{2.6}$$

Moreover, we also require that H_k^{C} is uniformly positive definite on the null-space of $A_{\mathcal{A}_k}(x_k + p_k)$ for all (x_k, y_k) close to (x_*, y_*). Provided that these conditions are satisfied, and so long as A2–A4 hold, it is possible to show that the corrected update (2.4) is guaranteed to reduce the merit function (2.1) close to (x_*, y_*). Two popular choices are

$$p_k = 0, \;\; g_k^{\mathrm{C}} = g(x_k), \;\text{ and }\; H_k^{\mathrm{C}} = H_k,$$

which gives the traditional second-order correction championed by Mayne and Polak (1982), Coleman and Conn (1982) and Fletcher (1982*b*), and

$$p_k = s_k, \;\; g_k^{\mathrm{C}} = g(x_k + s_k, y_{k+1}), \;\text{ and }\; H_k^{\mathrm{C}} = H(x_k + s_k, y_{k+1}),$$

which corresponds to a second SQP step, and provides the basis for the "watchdog technique" suggested by Chamberlain, Powell, Lemarechal and Pedersen (1982). Note that other authors (for example, Panier and Tits, 1991) have also shown that a small number (> 1) of SQP steps ensure that (2.1) decreases, but couch their proposal in the language of the non-monotone descent methods made famous for unconstrained minimization by Grippo, Lampariello and Lucidi (1986). In the linesearch context, a search should be made along the arc $x_k + \alpha s_k + \alpha^2 s_k^{\mathrm{C}}$, with the expectation that ultimately $\alpha_k = 1$ and (2.4) will occur.

2.2 Boggs, Kearsley and Tolle's approach

The Maratos effect does not occur for (2.2), and herein lies the popularity of methods based on this function. Traditionally such functions have been viewed somewhat unfavourably by most researchers since at a first glance they require a Jacobian value and the solution of the linear system (2.3) each time a function value is required—this may be very expensive if a number of different trial steps are required during the linesearch. This difficulty may be avoided by replacing (2.2) locally by a surrogate (approximation) merit function in which $y(x)$ and $A(x)$ are replaced by an appropriate y and A; it can be shown that such an approximation is valid, and that with care global convergence properties are retained. See Boggs et al. (1999*a*) for details.

Boggs, Kearsley and Tolle (1999*b*) provide details of the implementation and positive practical experience with such a method. Of particular note is that instead of trying to solve (1.4) directly, they pick three promising estimates of the required solution, and subsequently find the best approximation to this solution in the subspace spanned by these three vectors—at least one of the three directions is chosen to be a descent direction for the merit function. A trust-region (see Section 3) is used to limit the steps in the tests performed, and updated appropriately, but as yet this enhancement has no theoretical underpinning—the current theory for the linesearch version requires that H_k be positive definite.

2.3 SNOPT

SNOPT is a state-of-the-art linesearch-based SQP method for large-scale nonlinear programming due to Gill, Murray and Saunders (1997). At present, SNOPT uses a positive-definite approximation H_k to the Hessian of the Lagrangian, which exploits the fact that frequently many variables only appear linearly in the problem formulation, and retains information about previously-encountered curvature via a limited-memory secant update formula—we understand that a new version capable of using the exact Hessian of the Lagrangian is being tested. Special techniques are used to ensure that subsequent updates to H_k maintain positive definiteness. Feasibility with respect to linear constraints is attained from the outset. An augmented Lagrangian merit function is used to assess steps in both x and the Lagrange multiplier estimates y. The method is designed to be flexible, in that in theory it can use any quadratic programming algorithm, although by default it uses an null-space based active set method, which slightly limits the size and type of problems which can be handled. In practice, numerical tests have shown SNOPT to be most effective.

2.4 Feasible point approaches

A particularly appealing idea is to ensure that all iterates remain feasible, since then the objective function is itself a suitable merit function, and additionally the linearized constraints are sure to be consistent as $s = 0$ lies in the set $\{s \mid c(x_k) + A(x_k)s \geq 0\}$. In a sequence of papers, Herskovits (1986), Panier and Tits (1987, 1993) and Bonnans, Panier, Tits and Zhou (1992) show that this is in fact possible provided precautions are taken. It is easy to show that the SQP direction s_k from (1.4) at a feasible point x_k is a descent direction for $f(x)$ provided that H_k is positive definite. However, it may not be a feasible descent direction, i.e., it may not lie in the set

$$\left\{s \mid s^T g(x_k) < 0 \text{ and } s^T a_i(x_k) < 0 \text{ for all } i \in \mathcal{A}(x_k)\right\}, \tag{2.7}$$

since $s_k^T a_i(x_k)$ may be zero for one or more $i \in \mathcal{A}(x_k)$. Thus an arbitrarily small step along s_k may violate one or more of the (nonlinear) constraints active at x_k. To avoid this difficulty, any feasible descent direction s_k^{F}, and the "tilted" direction $s_k^{\text{T}} = (1 - \rho_k)s_k + \rho_k s_k^{\text{F}}$, for some $\rho_k \in (0, 1)$, are determined. The tilted direction is itself a feasible descent direction, and, so long as ρ_k converges to zero sufficiently fast, retains the fast asymptotic convergence properties of the original SQP direction. However, since these properties only arise if an asymptotic unit step is taken, a second-order correction rather like s_k^{C} from (2.5) may be necessary. The FSQP algorithm of Panier and Tits (1993) is based on these ideas, and can be shown to be globally convergent under suitable assumptions on H_k, and the requirement that (2.7) is non empty, which amounts to a constraint qualification, and Q-superlinearly convergent under the additional assumptions A3 and A4. Although this method has only been considered for small problems, it is not difficult to imagine how to generalize it by taking approximate solutions to the various subproblems. As with most linesearch methods, the requirement that H_k be positive definite is its major weakness.

3 Trust-region methods

The second important class of methods designed to ensure global convergence of locally convergent minimization algorithms are trust-region methods. Rather than controlling the step taken along the SQP direction (having computed the direction), trust-region methods aim to control the step at the same time as computing the search direction. Such methods hold a distinct advantage over linesearch methods, in that no requirement that H_k be positive definite is needed.

To simplify our discussion, consider first the unconstrained minimization of a nonlinear smooth function f. At the k-th iteration, a model $m_k(x_k + s)$ of $f(x_k + s)$ is used. This model is merely required to resemble f increasingly accurately as s approaches zero, and is believed to be a good approximation for all s within a trust region $\|s\|_k \leq \Delta_k$ for some appropriate, possibly iteration-dependent, norm $\|\cdot\|_k$ and radius $\Delta_k > 0$. If this is the case, an approximate minimizer of m_k should provide a good estimate of the minimizer of f within the same region. The first stage of a trust-region method is thus to compute a suitable approximate minimizer s_k of m_k within the trust region. If our hypothesis is correct, we would then expect $m_k(x_k) - m_k(x_k + s_k)$ to be a good approximation to $f(x_k) - f(x_k + s_k)$; if this confidence is repaid, we set $x_{k+1} = x_k + s_k$, and possibly increase the radius. On the other hand, when $m_k(x_k) - m_k(x_k + s_k)$ and $f(x_k) - f(x_k + s_k)$ are very different, our hypothesis is invalid, that is to say that Δ_k is too large. In this case, we set $x_{k+1} = x_k$, and ensure that Δ_{k+1} significantly less than Δ_k. This extremely simple framework is imbued with very powerful global convergence properties under extremely modest assumptions (see, for example, Conn et al., 2000). In practice, all that is required of the step is that it gives a reduction in the model of at least a fixed fraction of that which would be obtained by approximately minimizing the model within the trust-region along a gradient-related direction (such as $-g(x_k)$). This one-dimensional minimization problem is often trivial; the resulting point, the Cauchy point, plays a key role in the convergence theory for trust-region methods. A most important result is that is a Newton model—the first three terms of a Taylor expansion—is used, and if the model is minimized sufficiently accurately, the trust region constraint will asymptotically be inactive, and the resulting full Newton step will result in a Q-superlinear convergence rate.

Turning now to the constrained case, it is reasonable to expect to replace the objective function by a suitable merit function, and to build a model of this merit function. However, if we try to impose a trust-region constraint $\|s\|_k \leq \Delta_k$ on top of the linearized constraints $c(x_k) + A(x_k)s \geq 0$, we immediately see a difficulty. Simply, if $c(x_k)$ is nonzero, the intersection of linearized constraints with the trust region may be empty if Δ_k is too small. Thus, the strategy outlined in the previous paragraph, in which the radius is reduced until the model of the merit function proves to be adequate, is flawed in the constrained case.

In this section, we consider a number of ways of avoiding this potentially devastating discovery.

3.1 Sℓ_1QP-like approaches

This approach avoids the incompatibility issue altogether. Simply, rather than considering an SQP method directly, we instead aim to minimize the unconstrained, non-smooth penalty function (2.1). Since (2.1) is non-smooth, we cannot appeal directly to trust-region theory for smooth unconstrained minimization. However, the basic idea remains valid. We model $\phi(x_k + s, \sigma)$ as

$$m_k(x_k + s) = f(x_k) + s^T g(x_k) + \tfrac{1}{2} s^T H_k s + \sigma \| \left(c(x_k) + A(x_k)s \right)_- \|, \tag{3.1}$$

where H_k reflects the curvature in both f and c, and aim to approximately minimize this model within the trust region. All that is really required is to change the definition of the Cauchy point, since the gradient may not exist at x_k. Instead of the negative gradient, it suffices to consider the steepest descent direction

$$d(x_k) = -\arg \min_{g \in \partial\phi_k(x_k,\sigma)} \|g\|,$$

where $\partial\phi_k(x_k, \sigma)$ is the generalized gradient of $\phi_k(x, \sigma)$ at x_k. Because of the polyhedral convex structure of the non-differentiable term $\min(c, 0)$ in the definition of ϕ_k, it turns out that this generalized gradient may be found by solving a linear or convex quadratic program in the commonly occurring cases for which the ℓ_1, ℓ_2 or ℓ_∞ norm is used. While computing the Cauchy point is thus undoubtedly more expensive than in the smooth case, the alternative of exactly minimizing (3.1) within the trust region is usually even more expensive, since the latter problem may be non-convex (depending on H_k). The original idea here is due to Fletcher (1982*a*, 1987*a*, Section 14.4), who proposed minimizing (3.1) using the ℓ_1 norm within an ℓ_∞-norm trust region. The resulting so-called ℓ_1QP subproblem can be converted to an ordinary QP (with a given initial feasible point) by adding additional variables, but is probably best solved as is. A significant advantage of this method over almost all of its competitors is that an independence assumption like A2 is not required to assure global convergence to a critical point of the merit function.

Asymptotically, if the trust-region constraint is inactive, the SQP and ℓ_1QP directions coincide, and thus we might expect a fast asymptotic convergence rate. However, as we noted in Section 2, the SQP direction may suffer from the Maratos effect, and the same is true of the Sℓ_1QP direction. Thus, the Sℓ_1QP direction may not be acceptable, and consequently the trust-region radius will be reduced to exclude this step. The cure is exactly as before, namely a second-order correction should be added to correct for constraint curvature. In view of (2.5), the appropriate correction is obtained by minimizing

$$s^{\mathrm{C}T} g_k^{\mathrm{C}} + \tfrac{1}{2} s^{\mathrm{C}T} H_k^{\mathrm{C}} s^{\mathrm{C}} + \sigma \| \left(c(x_k + s_k) + A(x_k + p_k) s^{\mathrm{C}} \right)_- \|$$

within the trust region $\|s_k + s^{\mathrm{C}}\|_k \leq \Delta_k$. A fairly intricate algorithm, based on such a correction, proposed by Fletcher (1982*b*) was shown by Yuan (1985) to ensure that the trust-region radius is asymptotically inactive, and thus that the iterates can converge Q-superlinearly under assumptions A2–A4. Perhaps more simply, all that is required is that the trust region radius is reset to at least a fixed positive value whenever a successful step is taken, for then the trust-region will not ultimately interfere with the next step s_k and, if needed, correction s_k^{C}. Thus either s_k or $s_k + s_k^{\mathrm{C}}$ will ultimately be accepted, and the radius is subsequently bounded away from zero. Convergence to a second-order critical point—one for which (weak) second-order necessary conditions hold—may also be guaranteed, so long as significant negative curvature is exploited in the model, and that this negative curvature is reflected in the true problem—this is the case, for example, if H_k converges to $H(x_*, y_*)$.

To date, it is unclear whether it is better to update σ as the iteration proceeds, or to wait until a critical point of $\phi(x, \sigma)$ has been found before doing so. The advantage of the former is that a sequence of problems will not be solved, while the disadvantage is that any automatic value which aims to predict the correct value may also over-estimate it, leading to a poorer conditioned penalty function.

3.2 Vardi-like approaches

The remaining approaches we shall consider are what may be termed composite-step methods. A composite step s_k is computed as the sum of two components n_k and t_k, each of which has different aims. The (quasi-) normal component n_k is simply intended to improve the linearized infeasibility in as much as is possible while still satisfying the trust region constraint. Thus the merit function is ignored in this part of the computation. By contrast, the tangential component t_k aims not to degrade the improved infeasibility obtained in the normal step, while now concentrating on reducing a model of the merit function.

For simplicity, we shall suppose in this and the next two sections, that our constraints are equations, $c(x) = 0$—we shall return to the inequality case in Section 3.5. Recognising that the set

$$\mathcal{F}_k = \{ n \mid c(x_k) + A(x_k) n = 0 \ \text{ and } \ \|n\|_k \leq \Delta_k \}$$

may be empty, Vardi (1985) and Byrd, Schnabel and Shultz (1987) instead relax the linearized constraints so that $\alpha_k c(x_k) + A(x_k) n = 0$ for some $0 < \alpha_k \leq 1$ for which

$$\mathcal{F}_k(\alpha_k) = \{ n \mid \alpha_k c(x_k) + A(x_k) n = 0 \ \text{ and } \ \|n\|_k \leq \Delta_k \}$$

is non-empty. Clearly $\mathcal{F}_k(0)$ is non-empty, and any value $\alpha_k \leq \alpha_{\max}$ is also suitable, where $\alpha_{\max}$ solves

$$\max_{\alpha \in (0,1]} \min_{\|n\|_k \leq \Delta_k} \|\alpha c(x_k) + A(x_k) n\| = 0.$$

As finding $\alpha_{\max}$ may be expensive,—it may require the computation of the projection $n^{\mathrm{C}}(x_k)$ of the origin onto the set $\{ n \mid c(x_k) + A(x_k) n = 0 \}$—in practice an approximation n_k^{C} to $n^{\mathrm{C}}(x_k)$ for which

$c(x_k) + A(x_k)n_k^{\mathrm{C}} = 0$ may be computed instead, and α_k subsequently found so that $n_k = \alpha_k n_k^{\mathrm{C}}$ lies within the trust region. In fact, this last requirement is strengthened so that n_k lies strictly within the trust region, allowing some "elbow room" for the subsequent tangential step. Notice, however, the implicit requirement that the linearized constraints be compatible—in fact, this is the weakest point of the whole approach.

Having found the normal step, the tangential step is chosen to reduce a model of the merit function. Specifically, if we consider a merit function of the form

$$\phi(x, \sigma) = f(x) + \sigma \|c(x)\|,$$

and model $\phi_k(x_k + s, \sigma)$ by $m_k(x_k + s) = q(x_k + s) + \sigma m_k^{\mathrm{N}}(x_k + s)$, where

$$q(x_k + s) = f(x_k) + s^T g(x_k) + \tfrac{1}{2} s^T H_k s \;\text{ and }\; m_k^{\mathrm{N}}(x_k + s) = \left\| \left(c(x_k) + A(x_k)s \right) \right\|,$$

we see that following the normal step, $m_k^{\mathrm{N}}(x_k + n_k)$ will have decreased, but $q(x_k + n_k)$ may have increased. To cope with this, we pick the tangential step so that

$$m_k^{\mathrm{N}}(x_k + n_k + t_k) = m_k^{\mathrm{N}}(x_k + n_k) \;\text{ and }\; q(x_k + n_k + t_k) < q(x_k + n_k)$$

by approximately solving the problem

$$\underset{t \in \mathbb{R}^n}{\text{minimize}} \;\; t^T (g(x_k) + H_k n_k) + \tfrac{1}{2} t^T H_k t \;\; \text{subject to} \;\; A(x_k)t = 0, \;\; \text{and} \;\; \|t\| \le \Delta_k - \|n_k\|.$$

A suitable Cauchy point for this problem is readily available. Both normal and tangential steps satisfying the above requirements may readily be computed using suitable conjugate-gradient methods, the accuracy required being measured by suitable measures of the violation of the criticality conditions for the underlying problem.

Note that there is no *a priori* guarantee that by choosing n_k and t_k separately that $m_k(x_k + n_k + t_k) < m_k(x_k)$. However, since $m_k^{\mathrm{N}}(x_k + n_k + t_k) < m_k^{\mathrm{N}}(x_k)$, one way of ensuring that the model of the merit function does decrease is to increase σ if necessary as the iteration proceeds. A simple rule is to increase the parameter to ensure that

$$m_k(x_k) - m_k(x_k + n_k + t_k) \ge \tau \sigma (m_k^{\mathrm{N}}(x_k) - m_k^{\mathrm{N}}(x_k + n_k + t_k)),$$

where the value $\tau > 0$ is arbitrary but preferably very small. These, then, are the essential ingredients in the algorithm, which otherwise follows the standard trust-region paradigm. Any limit point of such an algorithm can be shown to be first-order critical. Moreover, the penalty parameter cannot grow arbitrarily large.

Of course, as usual in methods based upon the non-smooth merit function ϕ, it does not follow automatically that a desirable rate of convergence occurs. The cure, as alway, is to include a second-order correction s_k^{C}, satisfying (2.5), when needed. A suitable rule is only to consider a second-order correction when the normal step n_k lies well within the trust region,—and thus by implication that this step is feasible for the linearized constraints—and when the original step s_k does not provide a sufficient reduction in the model. In fact, the actual form of second-order correction required depends on the form of s_k. If s_k is the standard SQP step (1.3), then any second-order correction for which (2.6) holds is permitted. On the other hand, if the standard SQP step lies outside the trust-region, a specific second-order correction for which $g_k^{\mathrm{C}} = 0$ is advised—since the first-order criticality conditions for the model are not satisfied, there is little sense in trying to correct for these, while it is still important to try to correct for constraint curvature. It can then be shown that with the usual assumptions A2–A4, the algorithm sketched above converges Q-superlinearly so long as the SQP step is attempted (asymptotically) whenever possible, so long as the second-order correction is discarded if it lies too far outside the trust region, and so long as the trust-region radius is not reduced when the SQP step is acceptable but has a "small" component n_k. The same conditions suffice to ensure convergence to at least one second-order critical point x_* under assumption A2 if $H_k = H(x_k, y_k)$ and y_k converges to the corresponding y_*.

3.3 Byrd–Omojokun-like approaches

A different composite-step approach is due to (Byrd and) Omojokun (1989) and Byrd, Gilbert and Nocedal (1996), and forms the basis of the ETR, NITRO and BECTR algorithms of Lalee, Nocedal and Plantenga (1998), Byrd, Hribar and Nocedal (1997) and Plantenga (1999), respectively. This approach has a major advantage over that in the previous section in that there is no requirement that the linearized constraints be compatible, but is otherwise quite similar.

The major, and essentially only, difference is in the computation of the normal step. Rather than shifting the linearized constraint, another possibility is to compute n_k to approximately

$$\underset{n \in \mathbb{R}^n}{\text{minimize}} \;\; \|c(x_k) + A(x_k)n\| \;\text{ subject to }\; \|n\|_k \leq \xi^{\mathrm{N}} \Delta_k \tag{3.2}$$

for some $0 < \xi^{\mathrm{N}} < 1$. Of course, this problem may have a large number of solutions—the minimum-norm solution will give a component which is normal to t_k. Since computing an exact solution to this problem may be expensive, a cheaper option is to compute an approximate solution giving a reduction in $\|c(x_k) + A(x_k)n\|$ no worse than a fraction of that achievable at a suitable Cauchy point for this problem—such a point is

$$n_k^{\mathrm{C}} = -\alpha_k^{\mathrm{C}} A^T(x_k) c(x_k), \tag{3.3}$$

where

$$\alpha_k^{\mathrm{C}} = \underset{0 \leq \alpha \leq \xi^{\mathrm{N}} \Delta_k / \|A^T(x_k) c(x_k)\|_k}{\arg\min} \;\; \|c(x_k) - \alpha A(x_k) A^T(x_k) c(x_k)\|.$$

As before, such a requirements is satisfied at the first iteration of a suitable conjugate-gradient method, and subsequent conjugate-gradient steps may be used to further reduce the violation. ¿From a theoretical point of view, the normal step needs to have a non-trivial component in the above-mentioned minimum-norm solution.

The resulting algorithm offers essentially the same guarantees as its predecessor. So long as $A(x_k)$ is of full rank, it follows that $\lim_{k \to \infty} A^T(x_k) c(x_k) = 0$, indicating that, at worst, limit points are locally least infeasible. If the limiting Jacobian is also of full rank, we deduce that $\lim_{k \to \infty} c(x_k) = 0$, the remaining first-order criticality conditions hold, and the penalty parameter remains finite.

Turning to the issue of fast convergence, essentially the same precautions as before may be used. Since we are then interested in using a full SQP step, we consider the case where either the normal step lies on the "shrunken" trust-region boundary, i.e., $\|n_k\| = \xi^{\mathrm{N}} \Delta_k$, or a step all the way to the boundary, but within the shrunken trust-region, is possible, i.e., $c(x_k) + A(x_k) n_k = 0$

and $\|n_k\| \leq \xi^{\mathrm{N}}\Delta_k$. In the latter case, if the standard SQP step (1.3) does not provide a sufficient reduction in the model, a second-order correction is attempted. As before, the exact form depends upon whether the SQP step satisfies the trust-region constraint, in which case a general correction is allowed, or if the SQP step lies outside, in which case a restricted correction in which $g_k^{\mathrm{C}} = 0$ is used. The resulting algorithm then converges at a Q-superlinear rate under exactly the same conditions as its predecessor, and at least one limit point x_* is second-order critical if additionally $H_k = H(x_k, y_k)$ and y_k converges to the corresponding y_*.

3.4 Celis–Dennis–Tapia-like approaches

A third way of dealing with the possibility that the linearized constraints and the trust region have no common feasible point is to replace the former by

$$\|c(x_k) + A(x_k)s\| \leq \theta_k, \tag{3.4}$$

where θ_k is chosen so that the intersection of (3.4) and the trust region has a solution. Clearly, since we wish to reduce the infeasibility, we should insist at the very least that

$$\min_{\|s\|_k \leq \Delta_k} \|c(x_k) + A(x_k)s\| \leq \theta_k \leq \|c(x_k)\|, \tag{3.5}$$

while another possibility is to require that

$$\min_{\|s\|_k \leq \xi_1\Delta_k} \|c(x_k) + A(x_k)s\| \leq \theta_k \leq \min_{\|s\|_k \leq \xi_2\Delta_k} \|c(x_k) + A(x_k)s\|, \tag{3.6}$$

where $0 < \xi_2 \leq \xi_1 < 1$. Since solving problems of the form

$$\min_{\|s\|_k \leq \xi\Delta_k} \|c(x_k) + A(x_k)s\|$$

for some $0 < \xi \leq 1$, which are needed to ensure that θ satisfies (3.5) or (3.6), may be expensive, a cheaper possibility is to find *any* step n which lies within the trust region but which also significantly reduces $\|c(x_k) + A(x_k)n\|$. The most popular choice is, of course, the Cauchy step (3.3), but any step which further decreases $\|c(x_k) + A(x_k)n\|$ is also possible.

Although the computation of a suitable value n to reduce the infeasibility is reminiscent of the composite step methods considered, n_k is actually only used to find

$$\theta_k = \|c(x_k) + A(x_k)n_k\| \leq \|c(x_k)\|, \tag{3.7}$$

where the inequality in (3.7) is strict unless $c(x_k) = 0$. The overall step is computed as an approximate solution to the problem

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad s^T g(x_k) + \tfrac{1}{2} s^T H_k s \tag{3.8a}$$

$$\text{subject to} \quad \|c(x_k) + A(x_k)s\| \leq \theta_k \quad \text{and} \quad \|s\| \leq \Delta_k. \tag{3.8b}$$

for some appropriate Lagrange multiplier estimates y_k and approximation, H_k, to the Hessian of the Lagrangian. Methods based on these suggestions have been proposed by Celis, Dennis and Tapia

(1985), Powell and Yuan (1990), and others. Notice that by considering the whole feasible region (3.8b) rather than successive normal and tangential components, there is a clear potential for greater reductions in the objective function (3.8a) than with the previous two approaches. Unfortunately, this advantage may also be regarded as its Achilles' heal.

The main disadvantage of these approaches is apparent if one considers (3.8). If polyhedral norms are used, this subproblem reduces to a (possibly non-convex) *inequality*-constrained quadratic program which may prove rather expensive to solve. On the other hand, if we choose the ℓ_2 norm, the subproblem involves *two* quadratic constraints. Thus it is unclear if or how the powerful techniques which have been developed for the simpler subproblem involving a single quadratic constraint (see, for example, Moré and Sorensen, 1983, and Gould, Lucidi, Roma and Toint, 1999) may or can be applied. In particular, it is far from evident how to compute the model minimizer, nor is it obvious how to derive a useful approximation—some results for convex models have been obtained by Yuan (1991), Heinkenschloss (1994), and others. Indeed, given that global and local convergence theories matching those of the other methods we have considered in this section can be developed, we can only surmise that the lack of any reported implementation based on the approach taken here may be attributed to this disadvantage. We mention in passing that all of the methods we are aware of that use this approach use smooth exact penalty functions like (2.2) to force global convergence, but see no reason why methods based on (2.1) might not be equally possible.

3.5 Inequality Constraints

We now return to the case where the constraints are inequalities, $c(x) \geq 0$. There are two basic approaches. The first is to extend the model problem to include inequalities. As we have already noted, it may be that the set

$$\mathcal{F}_k = \{s \mid c(x_k) + A(x_k)s \geq 0 \text{ and } \|s\|_k \leq \Delta_k\}$$

is empty when Δ_k is small. Thus, we may instead have to be content with a step which moves us towards a solution of the model problem. The methods we have considered in the three previous sections have achieved this by decomposing the step as $s_k = n_k + t_k$, where the normal step n_k is chosen to reduce the (linearized) infeasibility, and the tangential step t_k is then determined to reduce the model without worsening the infeasibility attained during the normal step. For the general problem, much the same approach is valid.

There are obvious variants of all of the three main approaches we have discussed. Consider first the normal step. To extend the Vardi-like methods, we sketched in Section 3.2, we need to compute a trial step n_k^{C} which satisfies the linear constraints $c(x_k) + A(x_k)s \geq 0$, and which is not significantly longer than the projection onto the linearized feasible region. We then take a step α_k in this direction as far, or almost as far, as we can within the trust region, and set $n_k = \alpha_k n_k^{\mathrm{C}}$. For the Byrd–Omojokun-like approaches of Section 3.3, the normal step should be calculated by finding n_k to approximately

$$\underset{n \in \mathbb{R}^n}{\text{minimize}} \ \| \left(c(x_k) + A(x_k)n\right)_- \| \ \text{ subject to } \ \|n\| \leq \xi^{\mathrm{N}} \Delta_k, \tag{3.9}$$

for some $0 < \xi^{\mathrm{N}} < 1$, essentially as we did in (3.2). Note that (3.9) can be reformulated as a convex quadratic program, and thus, in principle, there are effective methods for (approximately) solving it. Finally, to extend the Celis–Dennis–Tapia-like approaches of Section 3.4, we merely need the normal step to give us at least as much reduction in $\| \left(c(x_k) + A(x_k)n\right)_- \|$ as a step to a generalized Cauchy point for this problem.

Turning to the tangential step, extensions to both the Vardi- and Byrd–Omojokun-like approaches require that the step approximately

$$\begin{array}{ll} \underset{t \in \mathbb{R}^n}{\text{minimize}} & t^T(g(x_k) + H_k n_k) + \tfrac{1}{2} t^T H_k t \\ \text{subject to} & A(x_k)t \geq -\max(c(x_k) + A(x_k)n_k, 0) \quad \text{and} \quad \|t\|_k \leq \Delta_k - \|n_k\|_k. \end{array}$$

Notice that the linearized infeasibility is made no worse, and attention turns instead to reducing the model value. In theory, all that is required is that the reduction in the model at t_k is a positive fraction of that attainable at a generalized Cauchy point, such as that proposed by Conn, Gould, Sartenaer and Toint (1993). For Celis–Dennis–Tapia-like approaches, the tangential step must be calculated to approximately

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \quad s^T g(x_k) + \tfrac{1}{2} s^T H_k s \quad \text{subject to} \quad \| \left(c(x_k) + A(x_k)s\right)_- \| \leq \theta_k \quad \text{and} \quad \|s\| \leq \Delta_k, \tag{3.10}$$

where $\theta_k = \|(c(x_k) + A(x_k)n_k)_-\|$. As before, this approach is less attractive in practice than its predecessors as effective methods for approximately minimizing (3.10) are not known. Other details extend in an obvious way. In particular, the same merit functions as before are appropriate, provided we replace all mention of $c(x)$ by $c(x)_-$.

The second way of moving from the equality-constrained to the general problem is to handle inequalities using barrier/interior-point methods (see, for example, Fiacco and McCormick, 1968, Vanderbei and Shanno, 1997, Forsgren and Gill, 1998, Gay, Overton and Wright, 1998, Sargent and Ding, 1998, Shanno and Vanderbei, 1999, and Conn, Gould, Orban and Toint, 1999). That is to say, we embed the inequality problem within a sequence of barrier problems of the form

$$\underset{x}{\text{minimize}} \quad f(x) + b(c(x), \mu_k),$$

where $b(c(x), \mu)$ is a barrier term like $-\mu \sum_{i=1}^m \log c_i(x)$, and $\{\mu_k\}$ is a sequence of barrier parameters which converge to zero from above. For this class of methods, we insist on starting from a strictly feasible point for the inequality constraints, that is that $c(x) > 0$, and require that all subsequent iterates remain strictly feasible for these constraints.

A typical trust-region method for such a problem models the barrier term using either a Newton (primal) or quasi-Newton (primal-dual) approximation. However, since such quadratic models have little influence in dissuading the iterates from violating one or more of inequality constraints, it is crucial to either adjust the shape of the trust region to keep the iterates feasible, or to add explicit extra constraints to the trust-region subproblems to do this (or both). The main difficulty when there are nonlinear inequality constraints present is that any additional constraints imposed on the trust-region subproblem may be nonlinear. For this reason, inequality constraints are often converted to equations by introducing slack variables. That is, we replace $c(x) \geq 0$ by the equivalent

$$c(x) - v = 0 \text{ and } v \geq 0,$$

and now solve a sequence of equality constrained minimization problems

$$\underset{x,\, v}{\text{minimize}} \quad f(x) + b(v, \mu_k) \text{ subject to } c(x) - v = 0.$$

The advantage of this is that we believe that the methods given throughout Section 3.2–3.4 are well-able to deal with nonlinear equality constraints, while any of the barrier/interior-point, affine scaling, or Coleman and Li (1996) algorithms are especially suited to linear, and particularly simple bound, constraints. Indeed, a careful combination of the Byrd-Omojokun and Coleman-Li approaches forms the basis of the algorithm proposed by Byrd et al. (1996) and implemented as NITRO by Byrd et al. (1997), while the method proposed by Yamashita, Yabe and Tanabe (1997) (see also Yamashita and Yabe, 1996*a*, 1996*b*, and Yabe and Yamashita, 1997) is essentially a Vardi-like primal-dual method. Both of these methods are reported to perform most effectively in practice.

There are still some disadvantages of adding slack variables. Firstly, we have most definitely increased the dimension of the problem. To counter this, it is important to realize that the dominant cost of most algorithms (at least when function values are inexpensive) tends to be that for the linear algebra. In practice, significant algebraic savings may be made be recognising that slack variables only occur linearly in the problem reformulation, and each slack variable is associated with a single constraint. The second disadvantage is that a suitable scaling of the slack variables is often difficult to find—in practice, it is more usual to pick the slacks so that $c(x) - Dv = 0$ and $v \geq 0$, where the diagonal matrix D is supposed to reflect "typical" values of $c(x)$, but the very fact that c is nonlinear indicates that a uniformly good D may be hard to determine. This has further repercussions for trust-region methods since it is usual to scale the trust-region norm to account for different scalings of the variables. We also note that in practice the trust-region scaling needs to reflect the interaction between the nonlinear constraints and the simple bounds (see Conn et al., 1999).

We conclude this short section on inequality constraints with the remark that blending good methods for coping with equality constraints with good ones for dealing with inequalities is an extremely active area of research. For this reason, we shall say no more here, but await further developments, and particularly comparisons of the numerous possibilities, with interest.

3.6 Filter methods

The last method we shall consider is the youngest, and certainly one of the most promising. The central idea is to dispense with the idea of using a merit function as a means of encouraging global convergence as far as is practically possible, and instead to use a mechanism which is less likely to reject candidate iterates. One such mechanism is a so-called filter.

Suppose $\theta(x)$ is some measure of the infeasibility of the constraints at x, for example $\theta(x) = \|c(x)_-\|$. A filter is a list of pairs $\{(f(x_i), \theta(x_i))\}$, with the property that no member of the filter is dominated by another, that is there are no two $(f(x_i), \theta(x_i))$ and $(f(x_j), \theta(x_j))$ $(i \neq j)$ for which

$$f(x_i) \leq f(x_j) \text{ and } \theta(x_i) \leq \theta(x_j).$$

The key point is that the filter may be used as a mechanism to accept or reject candidate iterates: a candidate will only be rejected if it gives "larger" values of both the function value and constraint violation than have been observed before. Contrast this to a merit function, which tries to combine these two (conflicting) requirements in a somewhat arbitrary way. An SQP-filter method aims to use the filter as a means of assessing iterates $x_k + s_k$, where s_k is a suitable approximation to the solution of the trust-region SQP subproblem

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \;\; m_k(x_k + s) \;\; \text{subject to} \;\; c(x_k) + A(x_k)s \geq 0 \;\; \text{and} \;\; \|s\|_k \leq \Delta_k, \tag{3.11}$$

where $m_k(x_k + s) = f(x_k) + s^T g(x_k) + \frac{1}{2} s^T H_k s$. The filter evolves as new iterates are accepted; the new iterate (or rather its (f, θ) pair) may be added to the filter, while the act of adding a new pair can result in the removal of previous members which are now dominated by the newcomer.

Of course, the reader will immediately object that such a simple-minded approach has obvious flaws. The first is, as always, that (3.11) may not have a solution because either the trust-region radius is too small, or because the linearized constraints are inconsistent. The cure is simply temporarily to abandon the objective function, and to enter a restoration phase, whose sole purpose is to reduce the infeasibility $\theta(x)$. The end of the restoration phase is reached at $x_k + r_k$, at which either the set $\{s \mid c(x_k + r_k) + A(x_k + r_k)s \geq 0 \;\text{and}\; \|s\|_{k+1} \leq \Delta_{k+1}\}$ is non empty for some $\Delta_{k+1} > 0$ and for which $\big(f(x_k + r_k), \theta(x_k + r_k)\big)$ is acceptable for the filter, or $x_k + r_k$ is a critical point for $\theta(x)$—in either case, such a point may be achieved by (approximately) minimizing $\theta(x)$. The second flaw is that it is easy to imagine a sequence of iterates each of which is barely acceptable to the (current) filter, but whose limit point is not critical—such a potential difficulty arises in most minimization methods, and the cure as always is to require that the iterates provide a "sufficient" improvement in the filter. A suitable rule is that an acceptable iterate must satisfy

$$f(x_k + s_k) < f(x_j) - \gamma\theta(x_j) \;\; \text{or} \;\; \theta(x_k + s_k) < (1 - \gamma)\theta(x_j),$$

where $\gamma \in (0, 1)$, for all x_j in the filter.

Fletcher and Leyffer (1997) demonstrate that an SQP-filter method based on the above, but including a number of other heuristics, is most effective in practice. The stated goal of requiring minimal interference from the filter is vindicated, and evidence is provided to show that other SQP methods (specifically the Sℓ_1QP method discussed in Section 3.1) frequently require more interference from their merit functions. In order to prove convergence of an SQP-filter method, specific rules for when to include an iterate in the filter, what sort of approximate step may be tolerated, and how to adjust the trust-region radius are required. The first-such convergence result (for an SLP-filter) was provided by Fletcher, Leyffer and Toint (1998), and this has now been extended for the SQP case by Fletcher, Gould, Leyffer and Toint (1999). The step is computed as the composite $s_k = n_k + t_k$, essentially as we have considered in the previous four sections—an infeasible subproblem is detected during the normal-step calculation, and the restoration phase started straight away. (A variation in which the step s_k is computed as a whole is also possible, although this may have to retreat to the composite step under unfavourable circumstances.) Once s_k has been computed, it is rejected if either it is unacceptable for the filter or if $m_k(x_k + s_k)$ offers a "sufficient" improvement over $m_k(x_k)$ but this predicted improvement does not translate into an actual improvement in $f(x)$. The trust-region radius is reduced whenever the step is rejected. The iterate is added to the filter if either it leads to a restoration phase, or if it has been accepted despite $m_k(x_k + s_k)$ not giving a "sufficient" improvement over $m_k(x_k)$. Second-order convergence issues are still open, but under investigation.

4 QP methods

Without a doubt, in our opinion, the primary reason SQP methods are back in the ascendant is that large-scale quadratic programming (QP) methods have matured considerably over the past few years. There have been a number of reasons for this. At the start of the 1980s, the vast majority of QP methods (see the surveys by Fletcher, 1987*a*, Chapter 10, and 1987*b*, and the bibliography in Cottle, Pang and Stone, 1992) were of the active set variety, most were specifically designed for convex (H positive semi-definite) or even strictly convex (H positive definite) problems, and few (if any) were capable of solving even medium size problems (for exceptions, see Gould, 1991, and Fletcher, 1993). The latter defect was due to two factors. Firstly, the dominant linear algebraic requirements usually treated all relevant matrices and associated factorizations as dense—while it was easy to anticipate using sparse factorizations, this ruled out some of the most successful (orthogonal transformation) methods developed for the dense case. Secondly, as problem size increased, the number of iterations rose quite rapidly—in the worst case, an exponential number of changes in the active set was possible, and while the expected and observed behaviour did not get close to such dire predictions, this was clearly a cause for concern.

By the turn of the decade, the theoretical (polynomially bounded) promise of Karmarkar's (1984) interior-point linear programming (LP) approach, and its successors, had been shown to be realized in practice, and theoretical extensions to convex QP were immediate—we note that, as in the LP case, only a small percentage of the methods proposed and analysed have ever been implemented (for exceptions, see Carpenter, Lustig, Mulvey and Shanno, 1993, and Vanderbei, 1994), most of these differ from their theoretical counterparts in order to obtain good practical performance, and all of them appear to perform considerably better than their worse-case polynomial bound. It is now accepted that, at the very least, interior-point and active-set methods are useful alternatives, but frequently the former are the methods of choice when the number of variables is very large. Of course, modern SQP methods often require the (approximate) solution of non-convex QPs, for which the above-mentioned interior-point methods cannot offer the same theoretical guarantees, since non-convex QP is known to be an NP-hard problem. Nonetheless, it is possible to construct interior-point-like methods, which are both globally convergent, and whose asymptotic convergence behaviour is as if they are locally convex (see, for example, Vanderbei and Shanno, 1997, Conn, Gould and Toint, 1996, and Conn et al., 1999). Early computational experience indicates considerable promise for large (say $n = O(10^5)$) problems.

One of the means by which methods for unconstrained minimization made the transition from small to large problems was the recognition that it is not necessary to solve the relevant model problem very accurately, at least when far from the solution. As we have indicated in Section 3, the same is true for SQP methods. However, at present, this either requires that the step is computed as a composite, in which two Cauchy points are determined (see Sections 3.2, 3.3, and 3.6), or as a single step in which an auxiliary computational may be necessary (see Section 3.1). As yet, the only method we are aware of that allows a direct truncation of the QP subproblem is the active set method due to Murray and Prieto (1995). The subproblems in both active-set and interior-point methods may be solved by iterative (conjugate gradient-like) methods, although it is crucial, especially for the latter, to use suitable preconditioners.

Finally, as to which of the two QP alternatives we suggest is appropriate for SQP methods, our answer is both! To justify this, we believe that interior-point methods probably hold the advantage for early SQP iterations when the active set has far from settled down. By contrast, when the active set is essentially know, a few active-set iterations is often cheaper than applying an interior-point method, since the latter is difficult to "warm start", i.e., start from a known near optimal (but possibly un-centred) value. Thus we contend that any new SQP method for large-scale nonlinear programming should have access to both interior-point and active-set non-convex QP algorithms.

5 Conclusions

In this paper, we have surveyed many of the most recent SQP algorithms for nonlinear programming. The majority of these are well-suited to large-scale problems, and recent numerical results (see, for example, Fletcher and Leyffer, 1997, and Yamashita et al., 1997) indicate that such methods are often considerably better than the state-of-the-art for other nonlinear programming algorithms (such as MINOS and, it hurts us to say, LANCELOT). We should add that the tests performed are all on what might now be considered to be medium-sized problems (say $n = O(10^3)$–$O(10^4)$), but we see no reason that this trend will not continue for large ones (say $n = O(10^5)$–$O(10^6)$) in the near future, provided that options for solving core linear systems by iterative (say, preconditioned conjugate-gradient) methods are incorporated. It still remains to be seen if the major way in which modern SQP algorithms will benefit from interior-point technology is in the improvements these give to quadratic programming algorithms, or in the ways these suggest we should handle inequality constraints. It also remains to be seen if new ideas, such as the SQP-filter method described in Section 3.6, fulfil their early promise.

References

P. T. Boggs and J. W. Tolle. Sequential quadratic programming. *Acta Numerica*, **4**, 1–51, 1995.

P. T. Boggs, A. J. Kearsley, and J. W. Tolle. A global convergence analysis of an algorithm for large scale nonlinear programming problems. *SIAM Journal on Optimization*, **9**(to appear), 1999*a*.

P. T. Boggs, A. J. Kearsley, and J. W. Tolle. A practical algorithm for general large scale nonlinear optimization problems. *SIAM Journal on Optimization*, **9**(3), 755–778, 1999*b*.

J. F. Bonnans and G. Launay. Implicit trust region algorithm for constrained optimization. Technical report, Institute Nationale Recherche Informatique et Automation, Le Chesnay, France, 1992.

J. F. Bonnans, E. R. Panier, A. L. Tits, and J. L. Zhou. Avoiding the Maratos effect by means of a nonmonotone line search II. inequality constrained problems – feasible iterates. *SIAM Journal on Numerical Analysis*, **29**(4), 1187–1208, 1992.

R. H. Byrd, J. Ch. Gilbert, and J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. Technical Report 2896, INRIA, Rocquencourt, France, 1996.

R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large scale nonlinear programming. Technical Report 97/05, Optimization Technology Center, Argonnne National Laboratory, Argonne, Illinois, USA, 1997.

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.

T. J. Carpenter, I. J. Lustig, J. M. Mulvey, and D. F. Shanno. Higher-order predictor-corrector interior point methods with application to quadratic objectives. *SIAM Journal on Optimization*, **3**(4), 696–725, 1993.

M. R. Celis, J. E. Dennis, and R. A. Tapia. A trust region strategy for nonlinear equality constrained optimization. *In* P. T. Boggs, R. H. Byrd and R. B. Schnabel, eds, 'Numerical Optimization 1984', pp. 71–82, SIAM, Philadelphia, USA, 1985.

R. M. Chamberlain, M. J. D. Powell, C. Lemarechal, and H. C. Pedersen. The watchdog technique for forcing convergence in algorithms for constrained optimization. *Mathematical Programming Studies*, **16**, 1–17, 1982.

T. F. Coleman and A. R. Conn. Non-linear programming via an exact penalty-function: global analysis. *Mathematical Programming*, **24**(2), 137–161, 1982.

T. F. Coleman and Y. Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, **6**(2), 418–445, 1996.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. LANCELOT*: a Fortran package for Large-scale Nonlinear Optimization (Release A)*. Springer Series in Computational Mathematics. Springer Verlag, Heidelberg, Berlin, New York, 1992.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. A primal-dual algorithm for minimizing a non-convex function subject to bound and linear equality constraints. Technical Report RAL-TR-96-096, Rutherford Appleton Laboratory, Chilton, Oxfordshire, England, 1996. To appear in 'Nonlinear Optimization and Applications 2', (G. Di Pillo and F. Giannnessi, eds), 1999.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Methods for nonlinear constraints in optimization calculations. *In* I. Duff and A. Watson, eds, 'The State of the Art in Numerical Analysis', pp. 363–390, Oxford University Press, Oxford, England, 1997.

A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust-region methods.* To appear. SIAM, Philadelphia, 2000.

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. Technical Report (in preparation), Department of Mathematics, FUNDP, Namur, Belgium, 1999.

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. W. Cottle, J.-S. Pang, and R. E. Stone. *The Linear Complementarity Problem*. Academic Press, London, 1992.

G. Di Pillo, F. Facchinei, and L. Grippo. An RQP algorithm using a differentiable exact penalty function for inequality constrained problems. *Mathematical Programming*, **55**(1), 49–68, 1992.

A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. J. Wiley and Sons, Chichester, England, 1968. Reprinted as *Classics in Applied Mathematics 4*, SIAM, Philadelphia, USA, 1990.

R. Fletcher. An exact penalty function for nonlinear programming with inequalities. *Mathematical Programming*, **5**(2), 129–150, 1973.

R. Fletcher. A model algorithm for composite nondifferentiable optimization problems. *Mathematical Programming Studies*, **17**, 67–76, 1982*a*.

R. Fletcher. Second-order corrections for non-differentiable optimization. *In* G. A. Watson, ed., 'Proceedings Dundee 1981', pp. 85–114, Springer Verlag, Heidelberg, Berlin, New York, 1982*b*. Lecture Notes in Mathematics.

R. Fletcher. *Practical Methods of Optimization*. J. Wiley and Sons, Chichester, England, second edn, 1987*a*.

R. Fletcher. Recent developments in linear and quadratic programming. *In* Iserles and Powell (1987), pp. 213–243.

R. Fletcher. Resolving degeneracy in quadratic programming. *Annals of Operations Research*, **46-47**(1-4), 307–334, 1993.

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, N. I. M. Gould, S. Leyffer, and Ph. L. Toint. Global convergence of trust-region SQP algorithms for nonlinear programming. Technical Report 99/03, Department of Mathematics, FUNDP, 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, FUNDP, Namur, Belgium, 1998.

A. Forsgren and P. E. Gill. Primal-dual interior methods for nonconvex nonlinear programming. *SIAM Journal on Optimization*, **8**(4), 1132–1152, 1998.

D. M. Gay, M. L. Overton, and M. H. Wright. A primal-dual interior method for nonconvex nonlinear programming. *In* Y. Yuan, ed., 'Advances in Nonlinear Programming', pp. 31–56, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: an SQP algorithm for large-scale constrained optimization. Technical Report SOL97-3, Department of EESOR, Stanford University, Stanford, California 94305, USA, 1997.

N. I. M. Gould. An algorithm for large-scale quadratic programming. *IMA Journal of Numerical Analysis*, **11**(3), 299–324, 1991.

N. I. M. Gould, S. Lucidi, M. Roma, and Ph. L. Toint. Solving the trust-region subproblem using the Lanczos method. *SIAM Journal on Optimization*, **9**(2), 504–525, 1999.

L. Grippo, F. Lampariello, and S. Lucidi. A nonmonotone line search technique for Newton's method. *SIAM Journal on Numerical Analysis*, **23**(4), 707–716, 1986.

W. W. Hager. Stabilized sequential quadratic programming. Technical report, Department of Mathematics, University of Florida, Gainsville, Florida, USA, 1998. To appear in *Computational Optimization and Applications*.

M. Heinkenschloss. On the solution of a two ball trust region subproblem. *Mathematical Programming*, **64**(3), 249–276, 1994.

J. Herskovits. A 2-stage feasible directions algorithm for nonlinear constrained optimization. *Mathematical Programming*, **36**(1), 19–38, 1986.

W. Hock and K. Schittkowski. *Test Examples for Nonlinear Programming Codes.* Number 187 *In* 'Lecture Notes in Economics and Mathematical Systems'. Springer Verlag, Heidelberg, Berlin, New York, 1981.

A. Iserles and M. J. D. Powell, editors. *The State of the Art in Numerical Analysis*, Oxford University Press, Oxford, England, 1987.

N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, **4**, 373–395, 1984.

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.

S. Lucidi. New results on a continuously differentiable exact penalty function. *SIAM Journal on Optimization*, **2**(4), 558–574, 1992.

O. L. Mangasarian and S. Fromovitz. The Fritz John necessary optimality conditions in the presence of equality and inequality constraints. *Journal of Mathematical Analysis and Applications*, **17**, 37–47, 1967.

N. Maratos. *Exact penalty function algorithms for finite-dimensional and control optimization problems.* PhD thesis, University of London, London, England, 1978.

D. Q. Mayne and E. Polak. A superlinearly convergent algorithm for constrained optimization problems. *Mathematical Programming Studies*, **16**, 45–61, 1982.

J. J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, **4**(3), 553–572, 1983.

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.

B. A. Murtagh and M. A. Saunders. A projected Lagrangian algorithm and its implementation for sparse non-linear constraints. *Mathematical Programming Studies*, **16**, 84–117, 1982.

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

E. Panier and A. L. Tits. Avoiding the Maratos effect by means of a nonmonotone linesearch I. general constrained problems. *SIAM Journal on Numerical Analysis*, **28**, 1183–1195, 1991.

E. R. Panier and A. L. Tits. A superlinearly convergence feasible method for the solution of inequality constrained optimization problems. *SIAM Journal on Control and Optimization*, **25**(4), 934–950, 1987.

E. R. Panier and A. L. Tits. On combining feasibility, descent and superlinear convergence in inequality constrained optimization. *Mathematical Programming*, **59**(2), 261–276, 1993.

T. D. Plantenga. A trust-region method for nonlinear programming based on primal interior point techniques. *SIAM Journal on Scientific Computing*, **20**(1), 282–305, 1999.

M. J. D. Powell. Variable metric methods for constrained optimization. *In* A. Bachem, M. Grötschel and B. Korte, eds, 'Mathematical Programming: The State of the Art', pp. 288–311, Springer Verlag, Heidelberg, Berlin, New York, 1983.

M. J. D. Powell. Methods for nonlinear constraints in optimization calculations. *In* Iserles and Powell (1987), pp. 325–358.

M. J. D. Powell and Y. Yuan. A trust region algorithm for equality constrained optimization. *Mathematical Programming*, **49**(2), 189–213, 1990.

B. N. Pschenichny. Algorithms for general problems of mathematical programming. *Kibernetica*, **6**, 120–125, 1970.

S. M. Robinson. Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear programming algorithms. *Mathematical Programming*, **7**(1), 1–16, 1974.

R. W. H. Sargent. The development of the SQP algorithm for nonlinear programming. Technical Report C95-35, Center for Process Sstems Engineering, Imperial College, London, England, 1995.

R. W. H. Sargent and M. Ding. A new SQP algorithm for large-scale nonlinear programming. Technical Report C98-5, Center for Process Sstems Engineering, Imperial College, London, England, 1998.

D. F. Shanno and R. J. Vanderbei. Interior-point methods for nonconvex nonlinear programming: orderings and higher-order methods. Technical Report SOR 99-05, Program in Statistics and Operations Research, Princeton University, New Jersey, USA, 1999.

R. J. Vanderbei. LOQO: an interior point code for quadratic programming. Technical Report SOR 94-15, Program in Statistics and Operations,Research, Princeton University, New Jersey, USA, 1994.

R. J. Vanderbei and D. F. Shanno. An interior point algorithm for nonconvex nonlinear programming. Technical Report SOR 97-21, Program in Statistics and Operations Research, Princeton University, New Jersey, USA, 1997.

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

S. J. Wright. Superlinear convergence of a stabilized SQP method to a degenerate solution. *Computational Optimization and Applications*, **11**(3), 253–275, 1999.

H. Yabe and H. Yamashita. Q-superlinear convergence of primal-dual interior point quasi-Newton methods for constrained optimization. *Journal of the Operations Research Society of Japan*, **40**(3), 415–436, 1997.

H. Yamashita and H. Yabe. Nonmonotone SQP methods with global and superlinear convergence properties. Technical report, Mathematical Systems, Inc., Sinjuku-ku, Tokyo, Japan, 1996*a*.

H. Yamashita and H. Yabe. Superlinear and quadratic convergence of some primal-dual interior point methods for constrained optimization. *Mathematical Programming, Series A*, **75**(3), 377–397, 1996*b*.

H. Yamashita, H. Yabe, and T. Tanabe. A globally and superlineraly convergent primal-dual point trust region method for large scale constrained optimization. Technical report, Mathematical Systems, Inc., Sinjuku-ku, Tokyo, Japan, 1997.

Y. Yuan. On the superlinear convergence of a trust region algorithm for nonsmooth optimization. *Mathematical Programming*, **31**(3), 269–285, 1985.

Y. Yuan. A dual algorithm for minimizing a quadratic function with two quadratic constraints. *Journal of Computational mathematics*, **9**(4), 348–359, 1991.