Feasibility Control in Nonlinear Optimization*†

M. Marazzi ‡ Jorge Nocedal§

March 9, 2000

Report OTC 2000/04
Optimization Technology Center

Abstract

We analyze the properties that optimization algorithms must possess in order to prevent convergence to non-stationary points for the merit function. We show that demanding the exact satisfaction of constraint linearizations results in difficulties in a wide range of optimization algorithms. Feasibility control is a mechanism that prevents convergence to spurious solutions by ensuring that sufficient progress towards feasibility is made, even in the presence of certain rank deficiencies. The concept of feasibility control is studied in this paper in the context of Newton methods for nonlinear systems of equations and equality constrained optimization, as well as in interior methods for nonlinear programming.

*This work was supported by National Science Foundation grant CDA-9726385 and by Department of Energy grant DE-FG02-87ER25047-A004.

†To appear in the proceedings of the Foundations of Computational Mathematics Meeting held in Oxford, England, in July 1999

‡Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston Il 60208-3118, USA. Email: marazzi@ece.nwu.edu

§Department of Electrical and Computer Engineering, Northwestern University, Evanston Il 60208-3118, USA. Email: nocedal@ece.nwu.edu. Url: www.ece.nwu.edu/~nocedal

1 Introduction

We survey some recent developments in nonlinear optimization, paying particular attention to global convergence properties. A common thread in our review is the concept of "feasibility control", which is a name we give to mechanisms that regulate progress toward feasibility.

An example of lack of feasibility control occurs in line search Newton methods for solving systems of nonlinear equations. It has been known since the 1970s (see Powell [24]) that these methods can converge to undesirable points. The difficulties are caused by the requirement that each step satisfy a linearization of the equations, and cannot be overcome simply by performing a line search. The need for more robust algorithms has been one of the main driving forces behind the development of trust region methods. Feasibility control is provided in trust region methods by reformulating the step computation as an optimization problem with a restriction on the length of the step.

This weakness of Newton-type methods manifests itself in a variety of contexts, such as nonlinear systems of equations, equality constrained optimization, active set Sequential Quadratic Programming methods for nonlinear programming, and more surprisingly in interior methods for nonlinear optimization. In this paper we review various techniques for providing feasibility control in these contexts, paying special attention to interior methods.

2 Nonlinear Equations

We begin our study of feasibility control by considering its simplest context, which occurs when solving a system of nonlinear equations

$$c(x) = 0. \tag{2.1}$$

Throughout this section we assume that c is a mapping from $\mathbb{R}^n$ to $\mathbb{R}^n$, so that (2.1) represents a system of n equations in n unknowns.

A popular algorithm for solving (2.1) is the line search Newton method

$$A(x)d = -c(x) \tag{2.2a}$$

$$x^+ = x + \alpha d, \tag{2.2b}$$

where $A(x)$ is the Jacobian of c and α is a steplength parameter chosen to decrease a merit function, such as

$$\phi(x) \equiv \|c(x)\|_2^2. \tag{2.3}$$

Note that (2.2a) demands that linear approximations of the functions be exactly satisfied, and as a result, the step d can be exceedingly large, or even be undefined if the Jacobian is rank deficient.

Powell [24] showed that this iteration has a fundamental weakness in that it can converge to a point that is neither a solution of (2.1) nor a stationary point of ϕ. We now discuss this example, which plays an important role throughout this paper.

Example 1 (Powell [24])

Consider the nonlinear system

$$c(x) \equiv \begin{pmatrix} x_1 \\ 10x_1/(x_1+0.1)+2(x_2)^2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{2.4}$$

This problem has only one solution at $x^* = (0,0)$, which is also the only stationary point of the function (2.3) in the half plane $x_1 > -0.1$. The Jacobian of c is given by

$$A(x) = \begin{bmatrix} 1 & 0 \\ 1/(x_1+0.1)^2 & 4x_2 \end{bmatrix},$$

and is singular along the line

$$\mathcal{L} : (x_1, x_2) = (\theta, 0) \qquad \theta \in I\!R, \tag{2.5}$$

and in particular, at the solution point $x^* = (0,0)$. The graph of ϕ over the region of interest is plotted in Figure 1.

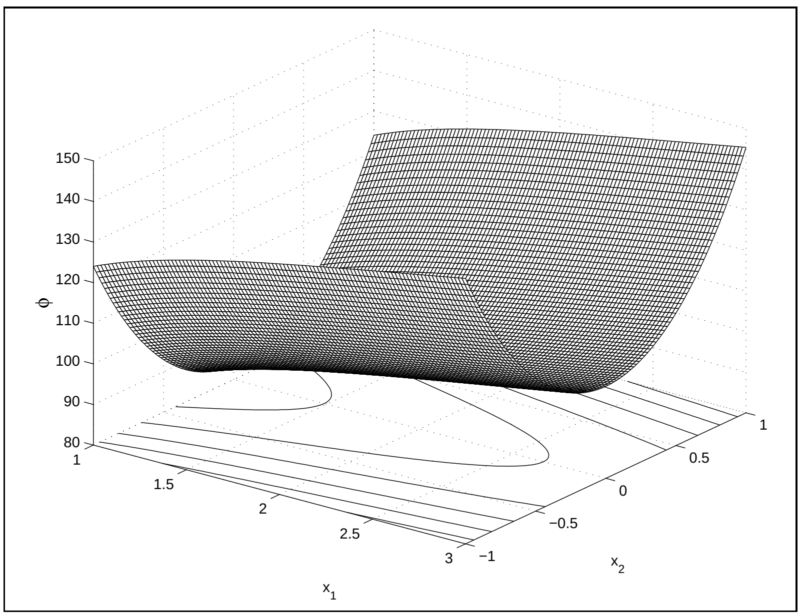

Figure 1: Merit function (2.3) of Example 1 on rectangle $1 \le x_1 \le 3$, $-1 \le x_2 \le 1$.

The analysis given in [24] studies the Newton iteration (2.2) using a steplength α that minimizes ϕ along the direction d. It shows that starting from $x^1 = (3,1)$, all the iterates remain within the rectangle $1 \le x_1 \le 3$, $-1 \le x_2 \le 1$, and that the Newton directions d^k quickly tend to become parallel to the x_2-axis, with lengths $\|d^k\|$ that diverge to infinity. As a consequence, the search directions become increasingly perpendicular to $\nabla\phi$ and the steplengths α_k tend to 0. It can be shown that the iterates converge to a point on $\mathcal{L}$ of the form $\tilde{x} = (\beta, 0)$, with $\beta > 1.4$. The value of β depends on the starting point, and a numerical computation reveals that starting from $x^1 = (3, 1)$, the iterates converge to $\tilde{x} \approx (1.8016, 0.000)$. It is clear from Figure 1 that this point is neither a solution to problem (2.4) nor a stationary point of ϕ, since the slope of ϕ at $\tilde{x}$ in the direction of the solution $(0, 0)$ is negative.

What is particularly striking in this example is that the Newton iteration (2.2), which generates directions d^k that are descent directions for ϕ (as is well known and readily verified), can converge to a point with non-zero slope of ϕ, even when an exact line search is performed. This suggests that the direction defined by (2.2a) has an intrinsic flaw, and must be altered in some circumstances. Before considering this, we will pay closer attention to the roles played by the line search and the merit function in this example.

To our surprise we found that, by using a relaxed line search—a backtracking scheme that only requires that ϕ be reduced at each step, trying $\alpha = 1$ first—the iteration converged to the solution $x^* = (0, 0)$. Using this inexact line search, the first step was a full Newton step with $\alpha = 1$ that led from $x^0 = (3, 1)$ to $x^1 = (0, -1.841)$. The first equation $c_1(x^1) = 0$ is satisfied at x^1, and due to (2.2), it will be satisfied at all subsequent iterates. Thus, for $k \geq 1$, the sequence of iterates (x_1^k, x_2^k) belongs to the line $(0, \theta)$, $\theta \in \mathbb{R}$, and converges to the solution $x^* = (0, 0)$ at a linear rate.

We tried another inexact line search strategy and observed that the iteration was again successful. The only way in which we were able to make the Newton iteration fail from the starting point $(3, 1)$ was by performing an exact line search, which was unexpected. It is easy, however, to find *other* starting points for which the iteration (2.2) using a relaxed line search fails by converging to a point along the singular line (2.5) and with $x_1 > 0$. This illustrates how difficult it can be to predict when a failure will occur, and suggests that the interaction between the merit function and the search direction in these unfavorable circumstances is worthy of further investigation. We should stress that failures of the line search Newton iteration (2.2) are observed in practical applications, and are not purely of academic interest.

2.1 Feasibility Control for Nonlinear Equations

Since the deficiency of Newton's method just described occurs both with exact and inexact line searches, there is a strong indication that it is necessary to modify the search direction, at least sometimes, to obtain an algorithm with more satisfactory convergence properties. Let us therefore assume that instead of (2.2a) we demand that the step d satisfies

$$c(x) + A(x)d = r,$$

for some vector r.

One way to choose r is indirectly, by using a trust region approach. We can pose the step computation as an optimization problem in which we aim to satisfy (2.2a) as well as possible subject to the restriction that the step is no greater than a given trust region radius Δ. For example, the Levenberg-Marquardt method, which is the classical trust region method for nonlinear equations, makes use of the ℓ_2 norm and poses the step-generation subproblem as

$$\min_{d} \; ||c(x) + A(x)d||_2^2 \tag{2.6a}$$

$$\text{subject to} \quad ||d||_2 \leq \Delta. \tag{2.6b}$$

If the linear system (2.2a) admits a solution that is no larger in norm than Δ, it will be taken as the step of the optimization algorithm; otherwise the solution of (2.6) will in general be a vector pointing in a different direction from the pure Newton step (2.2a). We say that this iteration provides "feasibility control" since it automatically determines the degree to which the linear approximation (2.2a) is to be satisfied. The feasibility control mechanism provided by the formulation (2.6) relies on an important algorithmic feature introduced in the 1970s, and used in most[1] trust region methods: the radius Δ is adjusted adaptively during the course of the optimization according to the success that the linear model has in predicting the behavior of the nonlinear problem during the most recent iteration. The degree of feasibility control is therefore dependent on this trust region adjustment strategy.

It can be shown that trust region methods based on (2.6) cannot converge to a point where $\nabla\phi$ is nonzero, as is the case in Example 1. Indeed, one can prove (see [21]) that if $A(x)$ is Lipschitz continuous and bounded above in norm in the level set $\mathcal{T} = \{x : \phi(x) \leq \phi(x^0)\}$, then

$$\lim_{k\to\infty} \|\nabla\phi(x^k)\| = 0. \tag{2.7}$$

The iterates must therefore approach a stationary point of ϕ. This is not necessarily a solution of $c(x) = 0$, but since $\nabla\phi(x) = 2A(x)^T c(x)$, this result allows us to identify two possible outcomes. If the smallest singular value of the sequence $\{A(x^k)\}$ is bounded away from zero, then $c(x^k) \to 0$, and the iteration succeeds in finding a solution point. Otherwise, the method is attracted to a point (or a region) where the merit function ϕ cannot be improved to first order.

We can also view the trust region framework as a regularization technique since the solution of (2.6) is given by

$$\left(A(x)^T A(x) + \sigma I\right) d = -A(x)^T c(x), \tag{2.8}$$

for some non-negative parameter σ such that the coefficient matrix in this system is positive semi-definite. The original derivation of the Levenberg-Marquardt method was, in fact, motivated by the need to regularize the system $A(x)^T A(x)d = -A(x)^T c(x)$. We should point out, however, that regularization and feasibility control are fundamentally different. Attempting to improve the robustness of Newton's method by using a regularization approach has not been entirely successful due to the difficulty in choosing an appropriate value of the regularization parameter σ. A lower bound for σ is the value that ensures that (2.8) is numerically nonsingular, but we normally wish to use significantly larger values so that

[1] A notable exception is the filter method proposed in [15] where the trust region update strategy depends only on whether the new iterate is acceptable to the filter and on whether the trust region was active during the step generation.

the direction d is not unnecessarily long and potentially unproductive. Our view is that the selection of σ should not be based entirely on linear algebra considerations, but must be tied in with the convergence properties of the optimization process.

As we have mentioned, trust region approaches select σ indirectly through the choice of the trust region radius Δ, which is updated at each iteration according to information about the nonlinear problem gathered during the optimization calculation. In particular, trust region methods provide a strategy for driving the regularization parameter σ to zero whenever the iterates approach a solution point x^* with a full-rank Jacobian $A(x^*)$.

Variants of the Levenberg-Marquardt approach (2.6) have been proposed to take advantage of quasi-Newton approximations to the Jacobian $A(x)$, or to reduce the cost of computing the step. In the "dogleg method" [24] the subproblem takes the form

$$\min_{d} \ \|c(x) + A(x)d\|_2^2 \tag{2.9a}$$

$$\text{subject to} \quad x + d \ \in \ \mathcal{D} \tag{2.9b}$$

$$\|d\| \ \leq \ \Delta. \tag{2.9c}$$

The set $\mathcal{D}$ is a piecewise linear path that makes the solution of the problem easy to compute, and yet allows the iteration to be globally convergent in the sense that (2.7) holds. It is defined in terms of the "Cauchy step" p_{C} and the pure Newton step p_{N}, which are given by

$$p_{\mathrm{C}} = -\tau A(x)^T c(x), \qquad p_{\mathrm{N}} = -A(x)^{-1} c(x). \tag{2.10}$$

The Cauchy step p_{C} points in the direction of steepest descent for ϕ (see (2.3)) at the current point x, and τ is the steplength that minimizes the quadratic objective in (2.9a) along this steepest descent direction and subject to the trust region constraint (2.9c). The dogleg path $\mathcal{D}$ is the piecewise linear segment that starts at the current iterate x, passes through $x + p_{\mathrm{C}}$, and terminates at $x + p_{\mathrm{N}}$. As the set $\mathcal{D}$ includes the Cauchy step p_{C}, the resulting step d will attain a degree of feasibility of the linearized constraints that is no less than that achieved by p_{C}. This feature gives feasibility control to the method because it ensures—unlike the line search case—that sufficient progress is made at each iteration even when $A(x)$ is rank-deficient or nearly deficient. (In the rank-deficient case, as p_{N} is not defined, we can assume that the dogleg path terminates at $x + p_{\mathrm{C}}$.) The analysis presented in [24], which constitutes one of the founding blocks of trust region convergence theory, shows that any trust region method that decreases the objective (2.9a) at least as well as the Cauchy step p_{C} will enjoy the same global convergence properties as the Levenberg-Marquardt method. The restriction (2.9b) causes the step to be, in general, of lower quality than the Levenberg-Marquardt step, and this can slow progress toward the solution in some applications. In particular, we will see in section 4.2.2, that an interior method based on the dogleg approach can be inefficient on some simple problems.

The dogleg method was originally developed to take advantage of quasi-Newton approximations B_k to the Jacobian $A(x^k)$. Here computing the Newton-like step $B_k^{-1}c(x^k)$, updating B_k and computing the Cauchy point requires only $O(n^2)$ operations. As a result the dogleg quasi-Newton method allows a reduction in the iteration cost from $O(n^3)$ to $O(n^2)$, in the dense case, compared with the Levenberg-Marquardt method. The increasing availability of derivatives due to automatic differentiation techniques, and the fact that quasi-Newton updates for nonlinear equations are not as successful as those used in optimization, have, however, diminished the benefits of the dogleg method for nonlinear equations. Nevertheless, the dogleg method has seen a revival in the context of nonlinear programming because the generalizations of (2.9) to constrained optimization problems are more amenable to computation than the direct extensions of the Levenberg-Marquardt method (2.6). But as mentioned above, the dogleg method poses some tradeoffs that are currently the subject of investigation in the context of interior methods [6, 32].

3 Equality Constrained Optimization

Before we consider nonlinear programming problems in full generality it is convenient to focus on Newton-type methods for solving the equality constrained optimization problem

$$\min \; f(x) \tag{3.1a}$$

$$\text{subject to} \quad c(x) = 0, \tag{3.1b}$$

where $f : I\!R^n \to I\!R$ and $c : I\!R^n \to I\!R^t$ are smooth functions, and $t \leq n$.

Classical sequential quadratic programming (SQP) methods define a search direction d from the current iterate x, as the solution of a model of the form

$$\min_d \; m(d) \tag{3.2a}$$

$$\text{subject to} \quad c(x) + A(x)d = 0, \tag{3.2b}$$

where $A(x)$ is, as before, the Jacobian of c. Here $m(\,\cdot\,)$ is a quadratic Taylor approximation to the Lagrangian of (3.1). A new iterate is computed as

$$x^+ = x + \alpha d, \tag{3.3}$$

where α is a steplength chosen to decrease a merit function.

There is a good reason for choosing the quadratic program (3.2) to generate a search direction, as this amounts (when $m(\,\cdot\,)$ is strictly convex on the null space of $A(x)$, and $A(x)$ has full rank) to applying Newton's method to the optimality conditions of (3.1); see for example [14]. The discussion of the previous section suggests, however, that a Newton iteration that uses the linearizations (3.2b) may encounter difficulties that cannot be resolved simply by an appropriate selection of the steplength parameter α. This is indeed the case, as one can show [26, 7] that the lack of feasibility control in (3.2) can lead an SQP method to converge to an infeasible point that is not stationary with respect to a measure of infeasibility, such as $\|c(x)\|_2^2$.

In order to obtain stronger convergence properties, we introduce feasibility control in (3.2). This can be done either by using trust region formulations, or by introducing relaxations of (3.2b) in line search SQP methods.

3.1 Feasibility Control and Trust Regions

We now review some of the most interesting trust region reformulations of (3.2), and discuss the degree to which they succeed in improving the global convergence properties of SQP-type iterations.

In Vardi's method [29] the step computation subproblem takes the form

$$\min_{d} \; m(d) \tag{3.4a}$$

$$\text{subject to} \quad A(x)d + \theta c(x) = 0 \tag{3.4b}$$

$$\|d\|_2 \le \Delta, \tag{3.4c}$$

where θ is a positive parameter that ensures that the constraints (3.4b) and (3.4c) are compatible. Interestingly, this method does not provide feasibility control, since θ only controls the *length* of the step and not its direction. To be more specific, since θ is given prior to the step computation, we can define $\hat{d} = \theta^{-1}d$, and rewrite (3.4) as

$$\min_{\hat{d}} \; m(\theta\hat{d}) \tag{3.5a}$$

$$\text{subject to} \quad A(x)\hat{d} + c = 0 \tag{3.5b}$$

$$\|\hat{d}\|_2 \le \Delta/\theta. \tag{3.5c}$$

Equations (3.5b) are a standard linearization of the constraints and do not ensure that progress towards feasibility is comparable to that of a steepest descent (Cauchy) step. We speculate that, when applied to the problem

$$\min \; 0 \quad \text{s.t.} \quad c(x) = 0,$$

where c is given by (2.4), Vardi's method may encounter the same difficulties as the line search Newton method on Example 1.

Due to the trust region constraint, the step cannot be unnecessarily large, but the system (3.5b) can be inconsistent, since it requires that $c(x)$ be in the range of $A(x)$, as in (3.2b). This approach, therefore, does not provide an adequate regularization in the case when $A(x)$ becomes rank-deficient.

Another potential drawback of (3.4) is that there does not appear to be a simple strategy for choosing the parameter θ so as to ensure good practical performance. It is easy to determine a range of values for which the constraints (3.4b)–(3.4c) are consistent, but the particular value chosen can strongly influence performance because it controls whether the step d tends more to satisfy constraint feasibility or to minimize the objective function. On the other hand, Vardi's approach is simpler than the formulations we discuss next.

In order to improve the theoretical deficiencies of Vardi's method, Byrd and Omojokun [4, 23] developed a scheme that emulates the Levenberg-Marquardt method (2.6) in its ability to solve systems of equations—or in the context of problem (3.1), to attain feasibility. They propose to first solve the auxiliary problem

$$\min_{p} \; q(p) \equiv \|c(x) + A(x)p\|_2^2 \tag{3.6a}$$

$$\text{subject to} \quad \|p\|_2 \le \Delta, \tag{3.6b}$$

which is the direct extension of (2.6). The step p computed in this manner will make sufficient progress toward feasibility and will prevent the iterates from converging to points that are not stationary for a variety of measures of infeasibility, such as $\phi(x) = \|c(x)\|_2^2$. It also provides regularization since (3.6) is well defined even when $A(x)$ is rank-deficient.

Of course, the step p will, in general, not contribute toward the minimization of f since (3.6) contains no information about the objective function. It will, however, be used to determine the level of linear feasibility that the step d of the optimization algorithm must attain. The step computation subproblem is thus formulated as

$$\min_d \; m(d) \tag{3.7a}$$

$$\text{subject to} \quad c(x) + A(x)d = c(x) + A(x)p \tag{3.7b}$$

$$\|d\|_2 \leq 1.2\,\Delta. \tag{3.7c}$$

The size of the trust region has been expanded to allow room for decreasing the objective function, because when $\|p\|_2 = \Delta$, the only solution of (3.7) could be p.

Methods based on (3.6)–(3.7) have been analyzed in [5, 10, 12]. Under reasonable assumptions[2] it has been shown that the iterates are not attracted by infeasible points that are not stationary for the measure of infeasibility $\|c(x)\|_2^2$—or for many other measures of infeasibility. In other words, convergence to undesirable points of the type discussed in Example 1, cannot take place.

The feasibility control mechanism is clearly illustrated by (3.7b), where one explicitly determines the degree to which the linear equations $c(x) + A(x)d$ are to be satisfied. As in the case of the Levenberg-Marquardt method (2.6), the degree of linear feasibility is based on the trust region methodology applied to the auxiliary problem (3.6), and not on linear algebra considerations. The subproblems (3.6)–(3.7) can be solved exactly [19] or approximately [20, 23]. In the latter case it is important that the approximate solution provides progress toward feasibility that is comparable to that attained by a Cauchy step p^c for (3.6). This step is defined as the minimizer of the quadratic q in (3.6a) along the steepest descent direction $-\nabla q(0)$, and subject to the trust region constraint (3.6b).

Feasibility control takes a different form in the method proposed by Celis, Dennis and Tapia [8]; see also Powell and Yuan [27]. In this method, the step-computation problem is given by

$$\min_d \; m(d) \tag{3.8a}$$

$$\text{subject to} \quad \|c(x) + A(x)d\|_2 \leq \pi \tag{3.8b}$$

$$\|d\|_2 \leq \Delta. \tag{3.8c}$$

There are several ways of defining the feasibility control parameter π. Perhaps the most appealing is to let

$$\pi = \|c(x) + A(x)p^c\|_2,$$

[2] These assumptions made in [5] are, in essence: the problem functions $f(\cdot)$ and $c(\cdot)$ are smooth, the sequence of function values $\{f(x_k)\}$ is bounded below, and the sequences $\{\nabla f(x^k)\}$, $\{c(x^k)\}$, $\{A(x^k)\}$ and the Lagrange Hessian approximations $\{B_k\}$ are all bounded.

where p^c is the Cauchy step for the problem (3.6).

By requiring that the step decreases linear feasibility at least as well as the Cauchy step, this method will not converge to non-stationary points of the type described in Example 1.

In summary, trust region methods based on (3.7) or (3.8) are endowed with feasibility control and regularization, whereas Vardi's approach does not provide either of these mechanisms. The formulation (3.8) proposed by Celis, Dennis and Tapia has not been used much in practice, due to the difficulties of solving a subproblem with two quadratic constraints, but the method of Byrd and Omojokun has been successfully used for large-scale equality constrained problems [20] and within interior methods for nonlinear programming, as we will discuss in the next section.

3.2 Feasibility Control and Line Search SQP

Most line search SQP methods compute a search direction d by solving the quadratic program (3.2). If at some iteration, the linear equations (3.2b) are found to be inconsistent, or if the Jacobian $A(x)$ is believed to be badly conditioned, a relaxation of the quadratic program is introduced.

For example, in the recently developed SNOPT package [18], the iteration enters "elastic mode" if the subproblem (3.2) is infeasible, unbounded, or its Lagrange multipliers become large (which can happen if $A(x)$ is nearly rank deficient). In this case, the nonlinear program is reformulated as

$$\min_{x,v,w} \; f(x) + \gamma\, e^T(v+w) \tag{3.9a}$$

$$\text{subject to} \quad c(x) - v + w = 0 \tag{3.9b}$$

$$v \geq 0,\; w \geq 0, \tag{3.9c}$$

where γ is a nonnegative penalty parameter. By applying the SQP approach (3.2) to this relaxed problem (3.9), the search direction (d_x, d_v, d_w) is required to satisfy the new linearized constraints

$$\begin{aligned} c(x) + A(x)\, d_x - (v + d_v) + (w + d_w) &= 0 \\ v + d_v \geq 0,\; w + d_w &\geq 0, \end{aligned}$$

which are always consistent. Once the auxiliary variables v and w are driven to zero, SNOPT leaves elastic mode and returns to the standard SQP subproblem (3.2).

In this approach feasibility control is therefore introduced only when needed. The relaxed problem is an instance of the Sℓ_1QP method advocated by Fletcher [14], which is known to be robust in the presence of Jacobian rank deficiencies. Various other related strategies for relaxing the linear constraints (3.2b) have been proposed; see for example [25] and the references therein.

In a different approach advocated, among others by Biggs [2], the linearized constraints are relaxed at every iteration. This approach can be motivated [22] using the classical penalty function

$$P(x;\nu) = f(x) + \tfrac{1}{2}\nu \textstyle\sum_{i=1}^{t} c_i(x)^2,$$

where $\nu > 0$ is the penalty parameter. The minimizer x satisfies

$$\nabla P(x;\nu) = \nabla f(x) + \nu \sum_{i=1}^{t} c_i(x) \nabla c_i(x) = 0. \tag{3.10}$$

Defining the Lagrange multiplier estimates

$$y_i = -c_i(x)\nu, \quad i = 1, \ldots, t,$$

we can rewrite the optimality condition (3.10) as

$$\nabla f(x) - A^T(x)y = 0 \tag{3.11a}$$

$$c(x) + y/\nu = 0. \tag{3.11b}$$

Considering this as a system in the variables x and y, and applying Newton's method to it, we obtain

$$\begin{bmatrix} W(x,y) & -A(x)^T \\ A(x) & \nu^{-1}I \end{bmatrix} \begin{bmatrix} d_x \\ d_y \end{bmatrix} = - \begin{bmatrix} \nabla f(x) - A(x)^T y \\ c(x) + y/\nu \end{bmatrix}, \tag{3.12}$$

where

$$W(x,y) = \nabla^2 f(x) - \sum_{i=1}^{t} y_i \nabla^2 c_i(x).$$

At every iteration we increase ν in such a way that the sequence $\{\nu_k\}$ diverges. In the limit the term $\nu^{-1}I$ will therefore vanish and the second equation in (3.12) will tend to the standard linearized equation $A(x)d_x + c(x) = 0$.

This type of feasibility control has not been implemented in most SQP codes, but has recently received attention [16] in the context of interior methods for nonlinear programming.

4 Interior Methods

Let us now focus on the general nonlinear programming problem with equality and inequality constraints, which can be written as

$$\min_x \; f(x) \tag{4.1a}$$

$$\text{subject to} \quad c_{\mathrm{E}}(x) = 0 \tag{4.1b}$$

$$c_{\mathrm{I}}(x) \geq 0 \tag{4.1c}$$

where $c_{\mathrm{E}} : I\!R^n \mapsto I\!R^t$ and $c_{\mathrm{I}} : I\!R^n \mapsto I\!R^l$, with $t \leq n$. Interior (or barrier) methods attempt to find a solution to (4.1) by approximately solving a sequence of barrier problems of the form

$$\min_{x,s} \; f(x) - \mu \sum_{i=1}^{l} \log(s_i) \tag{4.2a}$$

$$\text{subject to} \quad c_{\mathrm{E}}(x) = 0 \tag{4.2b}$$

$$c_{\mathrm{I}}(x) - s = 0, \tag{4.2c}$$

for decreasing values of μ (see e.g. [13]). Here $s = (s_1, \ldots, s_l)$ is a vector of positive slack variables, and l is the number of inequality constraints. The formulation (4.2) allows one to develop "infeasible" algorithms that can start from an initial guess x^0 that does not satisfy the constraints (4.1b)–(4.1c). Line search and trust region techniques have been developed to solve (4.2), and both must efficiently deal with the implicit constraint that the slacks s must remain positive. The practical performance and theoretical properties of interior methods for nonlinear programming are not yet well understood, and in this section we discuss some recent analytical contributions.

4.1 Line Search Interior Methods

Line search interior methods generate search directions by applying Newton's method to the KKT conditions of the barrier problem (4.2), which can be written in the form

$$\begin{aligned} \nabla f(x) - A_{\rm E}(x)^T y - A_{\rm I}(x)^T z &= 0 &\quad (4.3a)\\ Sz - \mu e &= 0 &\quad (4.3b)\\ c_{\rm E}(x) &= 0 &\quad (4.3c)\\ c_{\rm I}(x) - s &= 0. &\quad (4.3d) \end{aligned}$$

Here $A_{\rm E}(x)$ and $A_{\rm I}(x)$ are the Jacobian matrices of the functions $c_{\rm E}$ and $c_{\rm I}$, respectively, and y and z are their Lagrange multipliers; we also define $S = \operatorname{diag}(s)$ and $Z = \operatorname{diag}(z)$. The Newton equations for this system,

$$\begin{bmatrix} W & 0 & {A_{\rm E}}^T & {A_{\rm I}}^T \\ 0 & Z & 0 & -S \\ A_{\rm E} & 0 & 0 & 0 \\ A_{\rm I} & -I & 0 & 0 \end{bmatrix} \begin{bmatrix} d_x \\ d_s \\ -d_y \\ -d_z \end{bmatrix} = - \begin{bmatrix} \nabla f - {A_{\rm E}}^T y - {A_{\rm I}}^T z \\ Sz - \mu e \\ c_{\rm E} \\ c_{\rm I} - s \end{bmatrix} \tag{4.4}$$

define the *primal dual* direction $d = (d_x, d_s, d_y, d_z)$. A line search along d determines a steplength $\alpha \in (0, 1]$ such that the new iterate

$$(x^+, s^+, y^+, z^+) = (x, s, y, z) + \alpha\, (d_x, d_s, d_y, d_z) \tag{4.5}$$

decreases a merit function and satisfies $s^+ > 0$, $z^+ > 0$. In (4.4), the $n \times n$ matrix W denotes the Hessian, with respect to x, of the Lagrangian function

$$\mathcal{L}(x, y, z) = f(x) - y^T c_{\rm E}(x) - z^T c_{\rm I}(x). \tag{4.6}$$

Many interior methods follow this basic scheme [1, 13, 17, 28, 31]; they differ mainly in the choice of the merit function, in the mechanism for decreasing the barrier parameter μ, and in the way of handling nonconvexities. A careful implementation of a line search interior method is provided in the LOQO software package [28].

This line search approach is appealing due to its simplicity and its close connection to interior methods for linear programming, which are well developed. Numerical results reported, for example, in [11, 28] indicate that these methods represent a very promising approach for solving large scale nonlinear programming problems.

Nevertheless, Wächter and Biegler [30] have recently shown that *all* interior methods based on the scheme (4.4)–(4.5) suffer from convergence difficulties reminiscent of those affecting the Newton iteration in Example 1. We will see that, due to the lack of feasibility control in (4.4), these iterations may not be able to generate a feasible point.

Example 2 (Wächter and Biegler [30]) Consider the problem

$$\begin{aligned} &\min\ f(x) && (4.7a)\\ \text{subject to}\quad &(x_1)^2 - x_2 - 1 = 0 && (4.7b)\\ &x_1 - x_3 - 2 = 0 && (4.7c)\\ &x_2 \geq 0,\ x_3 \geq 0, && (4.7d)\end{aligned}$$

where the objective $f(x)$ is any smooth function. (This is a special case of the example presented in [30].) Let us apply an interior method of the form (4.4)–(4.5), starting from the initial point $x^0 = (-2, 1, 1)$. From the third equation in (4.4) we have that the initial search direction d will satisfy the linear system

$$c_{\mathrm{E}}(x^0) + A_{\mathrm{E}}(x^0) d_x = 0, \tag{4.8}$$

whose solution set can be written in the parametric form

$$d_x = \begin{pmatrix} 0 \\ 2 \\ -5 \end{pmatrix} + \theta \begin{pmatrix} 1 \\ -4 \\ 1 \end{pmatrix} \qquad \theta \in I\!R.$$

Figure 2 illustrates the feasible region (the dotted segment of the parabola) and the set of possible steps d_x, all projected onto the x_1–x_2 plane.

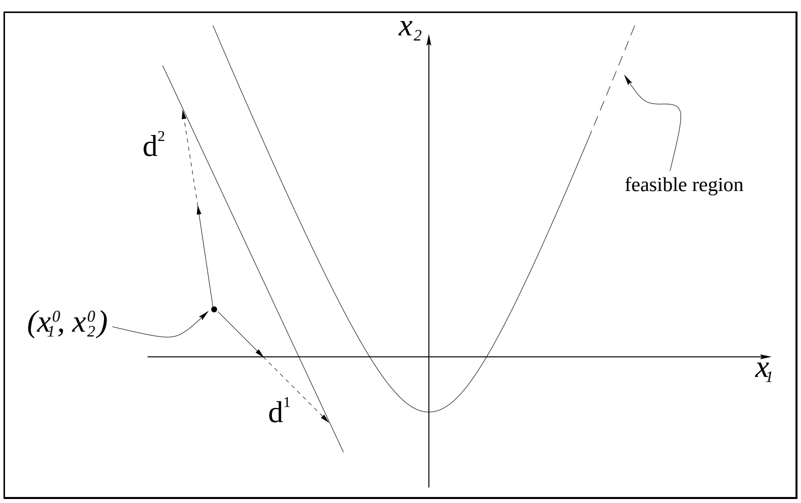

Figure 2: Example 2, projected onto the x_1–x_2 plane.

The steplength $\alpha = 1$ will not be acceptable because, for any value of θ, the point

$$x^0 + d_x = \begin{pmatrix} -2 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 2 \\ -5 \end{pmatrix} + \theta \begin{pmatrix} 1 \\ -4 \\ 1 \end{pmatrix}$$

violates the positivity bounds (4.7d). The steplength will therefore satisfy $0 < \alpha < 1$. The parabola in Figure 2 represents the constraint (4.7b), and the straight line to the left of it is its linearization at x^0. We have drawn two displacements, d^1 and d^2 that satisfy (4.8). Note how the step d^1 is restricted by the bound $x_2 > 0$; the shortened step is drawn as a solid line. The second step, d^2 is restricted by the bound $x_3 > 0$ (which cannot be drawn in this picture), and the shortened step is again drawn as a solid line.

Now, regardless of the value of $\alpha \in (0, 1)$, we see that the new iterate will be confined to the set

$$\{(x_1, x_2) \ : \ x_1 \leq -\sqrt{x_2 + 1}, \ x_2 \geq 0, \ x_3 \geq 0\}. \tag{4.9}$$

All points in this region are infeasible, as the constraint (4.7c) and the bound on x_3 in (4.7d) imply that any feasible point x must satisfy $x_1 \geq 2$.

The argument can now be repeated from the new iterate: one can show that the full step toward the linearized constraints violates at least one of the bounds and that a steplength $\alpha < 1$ must be employed. Using this, Wächter and Biegler showed that the sequence of iterates $\{x^k\}$ never leaves the region (4.9), and cannot generate a feasible iterate (even in the limit). This behavior is obtained for any starting point $x^0 = (x_1^0, x_2^0, x_3^0)$ that belongs to (4.9).

This convergence failure affects *any* method that generates directions that satisfy the linearized constraints (4.8) and that enforces the bounds (4.7d) by means of a backtracking line search. The merit function can only restrict the steplength further, and therefore is incapable of resolving the difficulties. The strategy for reducing μ is also irrelevant, since the proof only makes use of the third equation in the Newton iteration (4.4).

Example 2 is not pathological, in the sense that the Jacobian $A_{\rm E}(x)$ of the equality constraints given by (4.7b)–(4.7c) has full rank for all $x \in I\!R^3$. Without specifying the objective function and the merit function, all that can be said is that the algorithm never reaches feasibility, but by choosing f appropriately one can make the sequence of iterates converge to an infeasible point.

More specifically, Wächter and Biegler performed numerical tests defining $f(x) = x_1$ in Example 2, using various starting points, and report convergence to points of the form

$$x = (-\beta, 0, 0), \qquad \text{with } \beta > 0. \tag{4.10}$$

In other words, the iteration can converge to a point on the boundary of the set $\{(x_1, x_2) : x_2 \geq 0, \ x_3 \geq 0\}$—a situation that barrier methods are supposed to prevent! It is interesting to note that the points (4.10) are not stationary points for any measure of infeasibility of the form $\|c_{\rm E}(x)\|_p$ $1 \leq p \leq \infty$ (the bound constraints (4.7d) need not be taken into account in the infeasibility measure, as the limit points (4.10) satisfy both of them) and that the steplengths α_k converge to zero.

We should note that it is not essential that the parabola (4.7b) crosses the horizontal axis for the failure to occur; the analysis given in [30] includes the case in which the entire parabola lies above the x_1-axis.

An intriguing question is whether the failure of the iteration (4.4)–(4.5) is a manifestation of the limitations of Newton's method discussed in Example 1, or whether it is caused by the specific treatment of inequalities in these interior methods. To try to answer this question, we first ask whether the iteration matrix in (4.4) becomes singular in the limit, as is the case in Example 1. By defining $f(x) = x_1$, once more, we can consider x_2 and x_3 as slack variables, and the KKT matrix (4.4) for problem (4.7) is given by

$$\begin{bmatrix} W & 0 & 0 & 2x_1 & 1 \\ 0 & z_1 & 0 & x_2 & 0 \\ 0 & 0 & z_2 & 0 & x_3 \\ 2x_1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

This matrix is indeed singular on the manifold $\{(x_1, 0, 0) : x_1 \in I\!R\}$. In Example 1, however, the steplengths α_k are forced to converge to zero by the merit function, whereas in Example 2 this is a consequence of the bounds $x_2 \geq 0$, $x_3 \geq 0$. Thus, at least superficially, there appears to be a fundamental difference in these two phenomena, but this topic is worthy of further investigation.

Wächter and Biegler make another interesting observation about their example. At each step the linearized constraints (4.8) and the bounds (4.7d) are inconsistent. In active set SQP methods, this set of conditions comprises the feasible region of the quadratic program, and hence this inconsistency is readily detected. By introducing a relaxation of the quadratic program, as discussed in section 3.2, these methods are able to generate a step. In the context of interior methods, however, all that is known in these unfavorable situations is that the step must be shortened, which is a very common occurrence, and therefore cannot be used as a reliable warning sign. When convergence to non-stationary limit points such as (4.10) is taking place, the steplength will eventually become so small that difficulties will be apparent, but by this time a large amount of computation will have been wasted. This appears to be a theoretical deficiency of the algorithmic class (4.4)–(4.5).

We do not know whether failures of this type are rare in practice, and whether this deficiency of the scheme (4.4)–(4.5) manifests itself more often as inefficient behavior, rather than as outright convergence failures. An important practical question, which is the subject of current research, is whether a simple modification of the scheme (4.4)–(4.5) can resolve the difficulties. A line search interior method in which the steps do not satisfy the linearization of the constraints was proposed by Forsgren and Gill [16]. We do not know, however, if the convergence properties of this method are superior to those of the basic line search scheme (4.4)–(4.5).

In the next section we will see that a trust region interior method that employs feasibility control will not fail in Example 2, and thus possesses more desirable global convergence properties.

4.2 Trust Region Interior Methods

Let us now discuss interior methods that, in contrast to the schemes just described, generate steps using trust region techniques. For simplicity we will assume that the nonlinear program has only inequality constraints; the extension to the general problem (4.1) is straightforward.

For a given value of the barrier parameter μ, we can apply a trust region method to the barrier problem (4.2), which we now write as

$$\min \; f(x) - \mu \sum_{i=1}^{l} \log(s_i) \tag{4.11a}$$

$$\text{subject to} \qquad c(x) - s = 0 \tag{4.11b}$$

(we have omitted the subscript in $c_{\text{I}}(x)$ for simplicity). The interior methods described in [6, 9, 32] differ in the trust region method used to solve (4.11). We will first consider the approach described in [5, 6], which has been implemented in the NITRO software package.

4.2.1 The Algorithm in NITRO

The presence of the implicit bounds $s > 0$ in (4.11) suggests that the trust region should be scaled to discourage the generation of steps that immediately leave the feasible region. We can define, for example,

$$\|(d_x, S^{-1}d_s)\|_2 \leq \Delta, \tag{4.12}$$

where we have decomposed the step in terms of its x and s components, i.e., $d = (d_x, d_s)$. This trust region becomes more elongated as the iterates approach the boundary of the set $s \geq 0$, and permits the generation of steps that point away from this boundary. In addition, we impose the so-called "fraction to the boundary rule"

$$s + d_s \geq (1 - \tau)s, \qquad \text{with e.g., } \tau = 0.995, \tag{4.13}$$

to ensure that the slack variables remain strictly positive at every iteration.

The step-generation subproblem is given by

$$\begin{aligned} &\min_d \; m(d) && (4.14a)\\ \text{subject to} \qquad & c(x) - s + A(x)d_x - Sd_s = r && (4.14b)\\ & ||d||_2 \leq \Delta && (4.14c)\\ & d_s \geq -\tau e. && (4.14d) \end{aligned}$$

Here $m(\cdot)$ is a quadratic model of the Lagrangian of the barrier problem (4.11), r is a vector that provides feasibility control, $A(x)$ is the Jacobian of $c(x)$, and e is the vector of all ones. To obtain (4.14) we have introduced the change of variables

$$d_s \leftarrow S^{-1}d_s, \tag{4.15}$$

so that the trust region constraint (4.12) has the familiar spherical form (4.14c).

The vector r is determined by solving a Levenberg-Marquardt type of subproblem, which in this context takes the form

$$\min_p \ \|c(x) - s + A(x)p_x - Sp_s\|_2^2 \tag{4.16a}$$

$$\text{subject to} \qquad \|p\| \ \leq \ \xi\Delta \tag{4.16b}$$

$$p_s \ \geq \ -\tau\xi e. \tag{4.16c}$$

We call the solution $p = (p_x, p_s)$ of this problem the "normal step" because, as we discuss below, it is often chosen to be normal to the constraints. The trust region radius has been reduced by a factor of ξ (e.g., $\xi = 0.8$) to provide room in the computation of the total step d given by (4.14). Whereas in the equality constrained case (see (3.6)), feasibility control is only provided by the trust region constraint, in this formulation control is also exercised through the bound (4.16c).

After the normal step p has been computed, we define

$$r = c(x) - s + A(x)p_x - Sp_s, \tag{4.17}$$

so that the level of linear feasibility provided by the step d is that obtained by the normal step.

To complete the description of the interior algorithm proposed in [5, 6], we note that the step d is accepted or rejected depending on whether or not it provides sufficient reduction of the merit function

$$\psi(x, s) = f(x) - \mu \sum_{i=1}^{l} \log(s_i) + \nu ||c(x) - s||_2,$$

where ν is a positive penalty parameter. The trust region radius Δ is updated according to rules that can be viewed as the direct extension of the strategies used in trust region methods for unconstrained and equality constrained optimization.

To obtain a primal dual interior method, as opposed to a primal method, it is necessary to define the model $m(\cdot)$ in (4.14) appropriately. Consider the quadratic function

$$m(d) = (\nabla f, -\mu e)^T (d_x, d_s) + \tfrac{1}{2}(d_x, d_s)^T \begin{bmatrix} W & \\ & S\Sigma S \end{bmatrix} (d_x, d_s), \tag{4.18}$$

where W denotes the Hessian of the Lagrangian function (4.6) of the nonlinear programming program, and Σ is a diagonal matrix. If we choose $\Sigma = \mu S^{-2}$, then $m(\cdot)$ is a quadratic approximation of the Hessian of the barrier function (4.11), and this approach gives rise to a primal method. By defining $\Sigma = S^{-1}Z$, on the other hand, the step generated by (4.14) can be considered as a primal dual method in the sense that, when the problem is locally convex and the trust region is inactive, its solution coincides with the primal dual step defined by (4.4). The choice $\Sigma = S^{-1}Z$ has been observed to provide better performance than the primal method.

Behavior on Example 2. We have seen that the problem described in Example 2 will cause failure of any line search interior method of the form (4.4)–(4.5). Let us now consider the performance of the trust region method just outlined. Since providing analytic expressions for the iterates of this algorithm is laborious when the inequality constraints in the subproblems (4.14) and (4.16) are active, we will simply report the result of running the NITRO package on Example 2.

The iterates generated by the trust region method are plotted in Figure 3. During the first three iterations, which start at x^0 and finish at x^3, the trust region is active in both the normal (4.16) and in the step-generation subproblem (4.14). In particular, in these iterations, the normal step is a linear combination of the Cauchy and the Newton steps on the function (4.16a). Hence, the first three displacements did *not* satisfy the linearization of the constraints $c(x) - s + A(x)^T d_x - S d_s = 0$. We note that x^1 has already left the area in which line search methods get trapped.

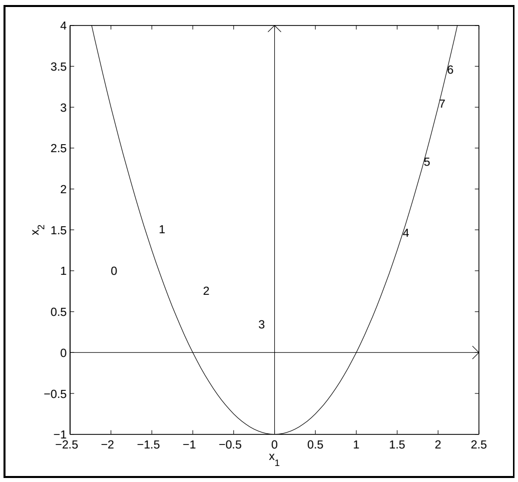

Figure 3: Iterates generated by a trust region interior method with feasibility control on Example 2, projected onto the x_1-x_2 plane. Each iterate x^k is indicated by the integer k.

For all the remaining steps, $x^4, x^5, \ldots$ the trust region was inactive in both subproblems, so that the linearized constraints were satisfied, and the steps were similar to those produced by a line search algorithm of the form (4.4)–(4.5). The trust region algorithm therefore performed very well on this problem, correctly deciding when to include a steepest descent like component on the step.

That this trust region method cannot fail on Example 2 is not surprising in the light of the convergence theory developed in [5]. It shows that the iterates cannot converge to a non-stationary point of the measure of feasibility $\|c(x)\|_2^2$. This result is possible because sufficient progress towards feasibility is required at every iteration, as we now discuss.

The normal step p given by (4.16) plays a similar role to that of the Levenberg-Marquardt step (2.6) for systems of nonlinear equations. In practice we may prefer to solve (4.16) inaccurately, using for example a dogleg method. In this case two conditions are required of the approximate solution p of the normal step. The first is that it must provide a decrease of linear feasibility that is proportional to that attained by the Cauchy step p_{C} for (4.16). This is of the form

$$p_{\mathrm{C}} = \tau\,[A \ \ -S]^T\,(c - s),$$

for some scalar τ.

Interestingly, this "Cauchy decrease condition" is not sufficient since it is possible for a very long step to provide the desired degree of linear feasibility but produce a very large increase in the model $m(\cdot)$ defined in (4.18). To control the length of the step we can impose the condition that p lie on the range space of the constraint normals of (4.14b), i.e.

$$p \in \mathcal{R}(\,[A \ \ -S]^T\,). \tag{4.19}$$

An alternative to (4.19) that has been studied by El-Alem [12] in the context of equality constrained optimization, is to require that

$$\|p\| = O(\|p_{\mathrm{MN}}\|)$$

where p_{MN} is the minimum norm solution of (4.16a).

These are requirements that the step must satisfy to improve feasibility. To make progress toward optimality, the step should also provide a decrease in the model $m(\cdot)$ that is proportional to that attained by the Cauchy step for (4.14). We will not describe this in more detail since it is not central to the main topic of this article.

4.2.2 Tradeoffs in this Trust Region Approach

Even though the trust region interior method implemented in NITRO has more desirable global convergence properties than line search interior methods, we now show through an example that it can be inefficient on some problems when certain cost-saving techniques are used in the solution of the subproblems (4.14) and (4.16).

Example 3. Consider the one variable problem

$$\min\ x \qquad \text{s.t.} \quad x \geq b,$$

where b is a constant, say $b = 500$. The barrier problem is given by

$$\min_{x,s}\ x - \mu \log(s) \quad \text{subject to} \quad x - b - s = 0, \tag{4.20}$$

and its optimal solution is

$$(x, s) = (b + \mu, \mu). \tag{4.21}$$

Let us focus on the first barrier problem, with say $\mu = 0.1$. We choose the initial point $x^0 = 1$, and to simplify the analysis, we set the initial slack to its optimal value $s^0 = \mu$; see Figure 4.

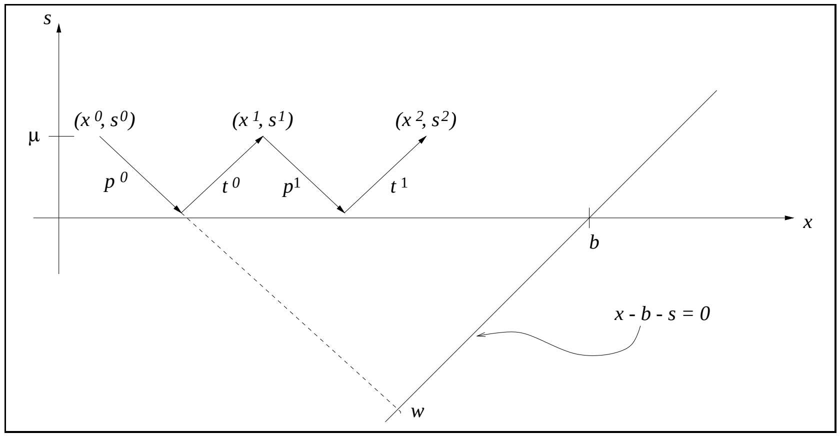

Figure 4: First iterates of a trust region interior method on the barrier problem (4.20).

Suppose that the radius Δ is large enough at x^0 that the trust region is inactive when solving both (4.14) and (4.16). The step generated by the trust region algorithm just described is

$$\left(\frac{\mu - s^0}{z^0} - (x^0 - b - s^0),\ \frac{\mu - s^0}{s^0 z^0} \right), \tag{4.22}$$

which can be regarded as a good step: it leads to the point $(b - \mu, 0)$, which is close to the minimizer (4.21).

Since it is expensive, however, to solve the subproblems (4.14) and (4.16) exactly, in the NITRO software package these subproblems are solved inaccurately. In particular, the normal step is required to be in the range of the constraint normals, i.e., p satisfies (4.19). We will now show that for such a normal step, the algorithm can be very inefficient, even if (4.14) is solved exactly.

In this example, there is a unique normal step p^0 from x^0 that lies in the range of the constraint normals; it is illustrated in Figure 4. If p^0 were not restricted by the fraction to the boundary rule (4.16c), it would lead to the point w (see Figure 4) where the objective in (4.16a) becomes zero. But after being cut back, the ability of the normal step p^0 to reduce infeasibility is greatly impaired. Since the total step d must retain the level of infeasibility provided by the normal step, we can view it as the sum $d = p + t$, for some displacement t perpendicular to p. It is easy to show that, in this example, d_x can be very small compared to b, and that the slack variable will be unchanged after the first step, i.e. $s^1 = s^0$. This behavior can be sustained for a large number of iterations, so that the sequence of iterates x^k moves slowly toward feasibility.

We can show that the number of iterations $n_{\rm iter}$ needed to attain feasibility satisfies

$$n_{\rm iter} \geq \frac{4}{9\mu}(b - x^0), \tag{4.23}$$

when the primal version of the algorithm is used. This number can be very large; consider e.g. $\mu = 0.1$ and $|b - x^0| = 500$. The bound (4.23) is established under the conditions that

(i) $\Delta_0 \geq (9/4)\mu$, (ii) $s^0 = \mu$, and that the merit function is of the form

$$\phi(x,s) = f(x) - \mu \sum_{i=1}^{l} \log(s_i) + \nu \|c(x) - s\|_p$$

for some $p \in [1, \infty]$. It is possible to show that, prior to the generation of a feasible iterate, all the steps are accepted by the merit function, the normal step is always restricted by (4.16c), and the trust region is inactive in both subproblems. The normal and tangential steps then take a simple closed form that enables one to obtain a recurrence relation for the iterates (x^k, s^k), from which (4.23) can be readily established.

We should stress that the interior method will not fail on Example 3, but that it will be inefficient. This is of concern, as we have observed this behavior in several of the problems in the CUTE [3] collection (e.g. in problem `GOFFIN`). In these problems many iterations are needed to attain feasibility, but convergence to the solution is fast after feasibility is achieved.

In contrast, a line search method of the form (4.4)–(4.5) solves Example 3 in a few iterations. We performed the calculations using an option in the NITRO package that computes steps of the form (4.4)–(4.5). We set $\mu = 0.1$, and report information about the first few iterations in Table 1. The first 3 steps where restricted significantly by the fraction to the boundary rule (4.13), (see the "steplength" column) providing small improvement in infeasibility, $\|c(x) - s\|$. But as the value of x became small, the algorithm produced a search direction that was sufficiently horizontal to allow a unit steplength and thus satisfy the linear constraints.

iter	infeasibility	steplength
0	5.000E+02	
1	4.996E+02	8.95E-04
2	4.982E+02	2.75E-03
3	4.853E+02	2.58E-02
4	3.876E+02	2.01E-01
5	6.706E-14	1.00E+00

Table 1: Line search interior method on the barrier problem (4.20).

We conclude that it is desirable to develop new techniques for approximately solving the subproblems (4.14) and (4.16) that perform well in the situation illustrated in Example 3.

4.3 The Trust Region Method in NUOPT

Yamashita, Yabe and Tanabe [32] have proposed a trust region interior method that promotes global convergence in a different manner from the method implemented in NITRO. They report excellent numerical results on a wide range of problems.

Their algorithm is an infeasible method based on the barrier problem (4.11). Each step d is computed using an extension of the dogleg method described in section 2.1. Specifically, $d = (d_x, d_s)$ is of the form

$$d = \beta d_{\mathrm{SD}} + (1-\beta) d_{\mathrm{PD}}, \tag{4.24}$$

where β is a parameter in $[0,1]$. Here d_{PD} denotes the x-s components of the primal dual step (4.4), and the so-called steepest descent step d_{SD} is also defined by (4.4) but replacing W by a positive definite diagonal matrix D, e.g. $D = I$.

The value of β must be chosen so that the step d satisfies the following three conditions

$$\|d\| \leq \Delta, \qquad s + d_s \geq (1-\tau)s, \tag{4.25a}$$

$$M(0) - M(\alpha d) \geq \tfrac{1}{2}\{M(0) - M(\alpha_{\mathrm{SD}} d_{\mathrm{SD}})\}, \tag{4.25b}$$

where $M(\cdot)$ is the following model of the barrier problem (4.2),

$$M(d) = (\nabla f, -\mu S^{-1} e)^T (d_x, d_s) + \tfrac{1}{2}(d_x, d_s)^T \begin{bmatrix} W & \\ & \Sigma \end{bmatrix} (d_x, d_s) + \nu \sum_{i=1}^{l} \left(|c_i(x) + \nabla c_i(x)^T d| - |c_i(x)| \right),$$

and $\Sigma = S^{-1} Z$. This model differs from (4.18) in the inclusion of terms measuring changes in the constraints. It is derived from the ℓ_1 merit function

$$f(x) - \mu \sum_{i=1}^{l} \log(s_i) + \nu \sum_{i=1}^{t} \|c(x) - s\|_1, \tag{4.26}$$

where ν is a positive parameter. The steplengths α and α_{SD} in (4.25b) are the values in $[0,1]$ that minimize $M(\cdot)$ along the directions d and d_{SD} respectively, and subject to the trust region constraint and the fraction to the boundary rule (4.25a).

The definition (4.25b) implies that the step d must provide a decrease in the model that is proportional to that given by the steepest-descent step d_{SD}. Once the step is computed, one follows standard trust region techniques to determine if it is acceptable for the merit function (4.26) and to update the trust region radius.

In the implementation described in [32], the parameter β is computed by a backtracking search, using the initial value $\beta = 0$, which corresponds to the primal dual step d_{PD}, and decreasing it by 0.1 until (4.25) is satisfied.

The similarities of this approach and the one used in the NITRO package are apparent. Both use trust region techniques to generate the step, and employ models of the barrier problem to determine the minimum quality of a step. The algorithms, however, differ in fundamental ways. Since both d_{PD} and d_{SD} are computed by a system of the form (4.4) they satisfy the linearization of the constraints. The total step d therefore does not provide feasibility control, and as a result, this method will fail on the problem described in Example 2.

Let us examine the steepest descent step d_{SD} more closely. Its name derives from an analogy with unconstrained optimization, where a Newton step applied to a quadratic function whose Hessian is the identity, coincides with a steepest descent step. The step d_{SD} is, however, not a steepest descent step for $\|c(x) - s\|_2$, and does not guarantee sufficient progress toward feasibility. Nor is it a steepest descent step for the merit function (4.26). This is in contrast with the model (4.16) which demands that the normal step provides a steepest-descent like reduction in the linearized constraints. Even though the method does not resolve the convergence difficulties of standard line search Newton iterations, the dogleg-like scheme between d_{SD} and d_{PD} constitutes a novel approach to handling nonconvexities, and may provide stability to the iteration in the case when the Hessian W is ill-conditioned.

5 Final Remarks

This article has focused on possible failures of optimization algorithms near non-stationary points of the measure of infeasibility of the problem. At these points the Jacobian is rank-deficient, and they can be avoided by introducing appropriate "feasibility control" in the optimization iteration. We have reviewed a variety of techniques for preventing convergence to non-stationary points, and discussed to what extent they are successful in doing so.

Whereas feasibility control mechanisms have been extensively studied in the context of nonlinear systems of equations and equality constrained optimization, they are only beginning to be investigated within interior methods for nonlinear programming. Designing algorithms with favorable global convergence properties, fast local rate of convergence, and low computational cost, remains an interesting topic for further research.

Acknowledgements. We thank Richard Byrd, Andreas Wächter and Richard Waltz, who provided many useful comments during the preparation of this article, and Mike Powell for an illuminating discussion concerning Example 1.

References

[1] I. Akrotirianakis and B. Rustem. A primal-dual interior point algorithm with an exact and differentiable merit function for general nonlinear programming problems. Technical Report 98-09, Department of Computing, Imperial College, London, UK, February 1999.

[2] M. C. Bartholomew-Biggs. Recursive quadratic programming methods for nonlinear constraints. In M. J. D. Powell, editor, *Nonlinear Optimization 1981*, pages 213–222. Academic Press, London, New York, 1982.

[3] I. Bongartz, A. R. Conn, N. I. M. Gould, and Ph. L. Toint. CUTE: Constrained and unconstrained testing environment. *ACM Transactions on Mathematical Software*, 21(1):123–160, 1995.

[4] R. H. Byrd. Robust trust region methods for constrained optimization. Third SIAM Conference on Optimization, Houston, Texas, May 1987.

[5] R. H. Byrd, J. Ch. Gilbert, and J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. Technical Report 96/02, Optimization Technology Center, Northwestern University, Evanston, IL, USA, 1996. Revised August 1998. Also available as Rapport de Reccherche 2896, INRIA.

[6] R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.

[7] R. H. Byrd, M. Marazzi, and J. Nocedal. Failures due to rank deficiency in nonlinear optimization. In preparation, 2000.

[8] M. R. Celis, J. E. Dennis, and R. A. Tapia. A trust region strategy for nonlinear equality constrained optimization. In R. B. Schnabel P. T. Boggs, R. H. Byrd, editor, *Numerical Optimization 1984*, pages 71–82, Philadelphia, 1985. SIAM.

[9] 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 general inequality and lineal equality constraints. Technical Report RAL-TR-1999-054, Computational Science and Engineering Department, Atlas Centre, Rutherford Appleton Laboratory, Oxon, OX11 0Qx, July 1999.

[10] A. R. Conn, N. I. M. Gould, and Ph. Toint. *Trust region methods*. SIAM publications, 2000.

[11] 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. In G. Di Pillo and F. Gianessi, editors, *Nonlinear Optimization and Applications 2*, Dordrecht, The Netherlands, 1999. Kluwer Academic Publishers. To appear.

[12] M. El-Alem. A global convergence theory for Dennis, El-Alem and Maciel's class of trust region algorithms for constrained optimization without assuming regularity. *SIAM Journal on Optimization*, 9(4), 1999.

[13] A. S. El-Bakry, R. A. Tapia, T. Tsuchiya, and Y. Zhang. On the formulation and theory of the Newton interior-point method for nonlinear programming. *Journal of Optimization Theory and Application*, 89(3):507–541, June 1996.

[14] R. Fletcher. *Practical methods of optimization.* John Wiley and Sons, New York, USA, second edition, 1987.

[15] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. Technical Report NA/171, Univ. of Dundee Numerical Analysis Report, Dundee, Scotland, U.K., September 1997.

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

[17] D. M. Gay, M. L. Overton, and M. H. Wright. A primal-dual interior method for nonconvex nonlinear programming. In Y. Yuan, editor, *Advances in Nonlinear Programming*, pages 31–56. Kluwer Academic Publishers, Dordretch, 1998.

[18] P. E. Gill, W. Murray, and M. A. Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. Technical Report 97-2, Dept. of Mathematics, University of California, San Diego, USA, 1997.

[19] F. A. M. Gomes, M. C. Maciel, and J. M. Martinez. Nonlinear programming algorithms using trust regions and augmented lagrangians with nonmonotone penalty parameters. *Mathematical Programming*, 84(1):161–200, 1999.

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

[21] J. Moré. Recent developments in algorithms and software for trust region methods. In A. Bachem, M. Grötschel, and B. Korte, editors, *Mathematical Programming: The State of the Art*, pages 258–287. Springer-Verlag, Berlin, 1983.

[22] W. Murray. An algorithm for constrained optimization. In R. Fletcher, editor, *Optimization*. Academic Press, London, 1969.

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

[24] M. J. D. Powell. A hybrid method for nonlinear equations. In P. Rabinowitz, editor, *Numerical Methods for Nonlinear Algebraic Equations*, pages 87–114. Gordon and Breach Science Publishers, London, New York, Paris, 1970.

[25] M. J. D. Powell. Variable metric methods for constrained optimization. In Bachem, A., Grötschel, M., and Korte, B., editors, *Mathematical Programming : The State of the Art, Bonn 1982*. Springer-Verlag, 1983.

[26] M. J. D. Powell. Methods for nonlinear constraints in optimization calculations. In A. Iserles and M. J. D. Powell, editors, *The state of the art in numerical analysis*, chapter 12, pages 325–357. Clarendon Press, Oxford, 1987.

[27] M. J. D. Powell and Y. Yuan. A trust region algorithm for equality constrained optimization. *Mathematical Programming*, 1990.

[28] R. J. Vanderbei and D. F. Shanno. An interior-point algorithm for nonconvex nonlinear programming. Technical Report SOR-97-21, Statistics and Operations Research, Princeton University, 1997.

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

[30] A. Wächter and L. T. Biegler. Failure of global convergence for a class of interior point methods for nonlinear programming. Technical Report CAPD B-99-07, Department of Chemical Engineering, Carnegie Mellon University, Pittsburg, PA 15213, USA, December 1999.

[31] H. Yamashita. A globally convergent primal-dual interior-point method for constrained optimization. Technical report, Mathematical System Institute, Inc., Tokyo, Japan, MAy 1994. Revised March 1994.

[32] H. Yamashita, H. Yabe, and T. Tanabe. A globally and superlinearly convergent primal-dual interior point trust region method for large scale constrained optimization. Technical report, Mathematical Systems Inc., Tokyo, Japan, July 1997.