On the Convergence of a Trust Region SQP Algorithm for Nonlinearly Constrained Optimization Problems *

Paul T. Boggs † Jon W. Tolle ‡ Anthony J. Kearsley §

July 21, 1995

Abstract

In (Boggs, Tolle and Kearsley 1994*b*) the authors introduced an effective algorithm for general large scale nonlinear programming problems. In this paper we describe the theoretical foundation for this method. The algorithm is based on a trust region, sequential quadratic programming (SQP) technique and uses a special auxiliary function, called a merit function or line-search function, for assessing the steps that are generated. A global convergence theorem for a basic version of the algorithm is stated and its proof is outlined.

Keywords: Sequential Quadratic Programming, Merit Functions, Global Convergence, Trust Region

1. Introduction

We consider the inequality-constrained minimization problem,

$$\begin{array}{ll} \min\limits_{x} & f(x) \\ \text{subject to:} & g(x) \leq 0 \end{array} \tag{1}$$

where $x \in \mathcal{R}^n$, and $f : \mathcal{R}^n \rightarrow \mathcal{R}$, and $g : \mathcal{R}^n \rightarrow \mathcal{R}^m$ are smooth functions. One of the most successful methods for solving (1) is the sequential quadratic programming

*Contribution of the National Institute of Standards and Technology and not subject to copyright in the United States.

†Applied and Computational Mathematics Division, National Institute of Standards and Technology, Gaithersburg, MD 20899

‡Mathematics Department, University of North Carolina, Chapel Hill, NC 27599

§Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77251-1892

(SQP) method in which at each iteration a quadratic program is solved to obtain the step direction toward the next iterate. In particular, given a current approximation x^k to the solution x^*, one forms the quadratic program

$$\begin{array}{l}\min\limits_{\delta} \ \nabla f(x^k)^{\mathrm{t}}\delta + \frac{1}{2}\delta^{\mathrm{t}}B^k\delta \\ \text{subject to:} \ \ \nabla g(x^k)^{\mathrm{t}}\delta + g(x^k) \leq 0\end{array} \tag{2}$$

where B^k is usually taken to be a current approximation to the Hessian of the Lagrangian of (1). Let δ^k be the solution of (2). Then the next approximation, x^{k+1}, is calculated by

$$x^{k+1} = x^k + \alpha\delta^k \tag{3}$$

where α is a scalar *steplength*. This procedure is repeated until convergence. An enormous amount of research has been published on the theory of SQP methods and algorithms based on this method are among the most effective in solving general constrained nonlinear problems. For a survey of this topic the reader is referred to (Boggs and Tolle 1995).

In (Boggs et al. 1994*b*) the authors have developed an SQP method that is designed specifically for large scale problems of the form (1). An implementation of this algorithm has performed quite well on a wide variety of problems, including examples from optimal control, molecular chemistry, statistics, and engineering. The purpose of this paper is to describe the theoretical underpinnings for this method.

A key element in our algorithm is the use of a special merit function. By a merit function we mean a scalar-valued function that can be used as a test for ensuring that the potential step given by (3) will make x^{k+1} a better approximation to the solution than x^k. A merit function for testing steps is an essential ingredient of a globally convergent algorithm. For unconstrained optimization problems, the objective function serves this purpose; one simply chooses α such that $f(x^{k+1}) < f(x^k)$ (with appropriate restrictions on α to assure that a sufficient decrease is achieved). For constrained optimization, the possible decrease in the objective value must be weighed against the requirement that feasibility must also be achieved — at least in the limit. Therefore, if one is using a method, such as an SQP algorithm, that does not maintain feasibility at each step, the objective function by itself is not an acceptable merit function. Most merit functions for constrained optimization algorithms are specially designed functions that have an unconstrained minimum at the solutions of (1). (See (Boggs and Tolle 1995) for examples.) The merit function proposed here is different in that it does not have an unconstrained local minimum at x^* but, rather, a constrained minimum at (x^*, z^*) where the variables z are *nonnegatively constrained* slack variables. This merit function has excellent theoretical properties relative to the SQP method but needs to be augmented by a sequence of approximate merit functions in order to produce a globally convergent algorithm.

An outline of the paper is as follows. In Section 2 we define our proposed merit function and its approximations. We then set forth the conditions that are assumed to hold throughout and state the basic properties of the merit function and its approximations. In Section 3 we incorporate these ideas in a somewhat detailed algorithm. We then state a global convergence theorem. In Section 4 we describe enhancements to the algorithm that improve the performance in the large scale case. These include the automatic adjustment of the penalty parameter that occurs in the merit function and a trust region strategy for use in approximately solving the quadratic subproblems. This leads us to the discussion of an interior point quadratic program solver and an approach to dealing with inconsistent quadratic subproblems.

2. The merit functions and their properties

In order to obtain a useful merit function for the inequality-constrained problem we maintain, along with the iterates x^k, a sequence of nonnegatively constrained slack variables $\{z^k\}$. That is, if x^k is a current approximation to the solution of problem (1.1) we consider z^k to be the corresponding approximation to the optimal slack vector. Accordingly, if δ^k is the step computed at x^k using the quadratic program (1.2), then the corresponding step for z^k is taken to be

$$q^k = -(\nabla g(x^k)^{\mathrm{t}}\delta^k + g(x^k) + z^k), \tag{4}$$

i.e., $z^k + q^k$ is the slack vector for (1.2). (This choice for q^k is further motivated in (Boggs, Tolle and Kearsley 1995).) We then choose the new iterates as

$$(x^{k+1}, z^{k+1}) = (x^k, z^k) + \alpha\,(\delta^k, q^k)$$

for some steplength α.

Two important points need to be emphasized here. First, if z^k is nonnegative and $\alpha \in [0, 1]$ then $z^{k+1} \geq 0$. Thus the nonnegativity of the slack variables is easily maintained. Second, the slack variable updates are not obtained from an optimization procedure, they are computed algebraically. Consequently, we are *not* introducing any complexity into the computational process.

The purpose for the introduction of the slack variables is to make it possible to derive a useful merit function The following merit function is derived from the equality-constrained problem which results from the addition of slack variables; details are contained in (Boggs et al. 1995).

$$\psi_d(x, z) = f(x) + \bar{\lambda}(x, z)^{\mathrm{t}}\bar{c}(x, z) + \frac{1}{d}\bar{c}(x, z)^{\mathrm{t}}\bar{A}(x, z)^{-1}\bar{c}(x, z)$$

where

$$\begin{aligned}\bar{c}(x,z) &= g(x)+Ze\\ \bar{A}(x,z) &= \nabla g(x)^{\mathrm{t}}\nabla g(x)+Z\\ \bar{\lambda}(x,z) &= -\bar{A}(x,z)^{-1}\nabla g(x)^{\mathrm{t}}\nabla f(x),\end{aligned}$$

d is a small parameter, and

$$Z=\operatorname{diag}\{z_1,\ldots,z_m\}.$$

The function $\bar{\lambda}(x,z)$ can be interpreted as a least squares aproximation to the Lagrange multiplier vector for (1.1).

Since the z_i are to be interpreted as nonnegative slack variables for (1), the minimization of ψ_d has the form:

$$\begin{array}{l}\min\limits_{x,z}\ \psi_d(x,z)\\ \text{subject to:}\quad z\geq 0.\end{array}\tag{5}$$

A fundamental property of this merit function (under the assumptions to be discussed shortly) is that the local solutions to (5) correspond to local solutions of (1.1).

Since $\psi_d(x,z)$ involves the gradients of the objective function and the constraints, carrying out line searches for this function is expensive. To ameliorate this difficulty, we define a sequence of (local) approximate merit functions by keeping the gradient terms fixed at the kth iteration. This produces functions that only require the evaluation of the function and the constraints, with no extra gradient evaluations, to check a prospective point. Our approximate merit function at the kth iterate is

$$\psi_d^k(x,z)=f(x)+\bar{c}(x,z)^{\mathrm{t}}\bar{\lambda}^k+\frac{1}{d}\bar{c}(x,z)^{\mathrm{t}}\bar{A}_k^{-1}\bar{c}(x,z)$$

where

$$\bar{A}_k=\nabla g(x^k)^{\mathrm{t}}\nabla g(x^k)+Z^k$$

and

$$\bar{\lambda}^k=-\bar{A}_k^{-1}\nabla g(x^k)^{\mathrm{t}}\nabla f(x^k).$$

While the $\psi_d^k(x,z)$ are cheap to evaluate, they cannot be used as a single merit function for measuring the progress towards a solution. However, as will be shown in Section 3, they can be employed as surrogates for $\psi_d(x,z)$ in a globally convergent algorithm.

To describe the basic theoretical properties of the merit functions and its local approximations, we first introduce appropriate notation and formulate some basic assumptions. We note that the feasible region for (1.1) can be thought of as the set of x such that $g(x)+z=0$ for some $z\geq 0$. Consequently, using the notation given above, we interpret

$$\mathcal{C}_0=\{(x,z):\bar{c}(x,z)=0,z\geq 0\}$$

as the *feasible set* and call

$$\mathcal{C}_\eta = \{(x,z) : ||\bar{c}(x,z)||^2 \leq \eta, z \geq 0\}$$

the η-*tube* surrounding the feasible set. We will denote by $\mathcal{G}$ a compact set of $\mathcal{R}^n \times \mathcal{R}^m_+$ and by $\mathcal{S}$ the set of points of $\mathcal{G}$ satisfying the first order conditions for (1.1), i.e.,

$$\mathcal{S} = \{(x,z) \in \mathcal{G} : \nabla f(x) + \lambda^{\mathrm{t}} \nabla g(x) = 0 \text{ and } \lambda^{\mathrm{t}} g(x) = 0 \text{ for some } \lambda \geq 0\}.$$

Also we will assume that the matrices used in (1.2) are chosen from $\mathcal{B}$, a compact set of positive definite $n \times n$ matrices.

The basic assumptions are the following:

A1: For each $(x,z) \in \mathcal{G}$ the matrix $\bar{A}(x,z)$ is positive definite.

A2: For each $(x,z) \in \mathcal{G}$ and $B^k \in \mathcal{B}$ the quadratic program (1.2) is feasible and its solution satisfies the strong second order sufficient conditions.

A3: $\mathcal{S}$ is not empty.

Note that the second assumption implies that the step (δ^k, q^k) will be a continuous function of (x^k,z^k) and B^k.

The merit functions can be shown to have the following basic properties.

Proposition 1: There exists a $\bar{d} > 0$ and a constant $\theta_1 > 0$ such that if $d \leq \bar{d}$ then for each $(x^k, z^k) \in \mathcal{G}$ and $B^k \in \mathcal{B}$

$$\nabla \bar{c}(x^k, z^k)^{\mathrm{t}} (\delta^k, q^k) = -||\bar{c}(x^k, z^k)||^2 \tag{6}$$

and

$$\nabla \psi_d^k (x^k, z^k)^{\mathrm{t}} (\delta^k, q^k) \leq -\theta_1 ||(\delta^k, q^k)||^2. \tag{7}$$

Proposition 2: For each d sufficiently small there exist an $\eta(d) > 0$ and a $\theta(d) > 0$ so that $(x^k, z^k) \in \mathcal{C}_{\eta(d)}$ and $B^k \in \mathcal{B}$ imply

$$\nabla \psi_d (x^k, z^k)^{\mathrm{t}} (\delta^k, q^k) \leq -\theta(d) ||(\delta^k, q^k)||^2. \tag{8}$$

Proposition 2 shows that at (x^k,z^k) sufficiently close to feasibility the step (δ^k,q^k) is a descent direction for the merit function while Proposition 1 gives the stronger results that the step is a descent direction everywhere for both the corresponding approximate merit function and the measure of infeasibility, $\bar{c}$. The rates of descent in the direction of the step can be used to guarantee that the *Wolfe conditions* (see (Nocedal 1992)) hold for each of the merit functions and the infeasibility function in the sets indicated in the propositions. For a given function ϕ and descent direction v at a point w the Wolfe conditions require the steplength α to satisfy

$$\phi(w+\alpha v) \leq \phi(w) + \sigma_1 \alpha \nabla\phi(w)^{\mathrm{t}} v$$

and

$$\nabla\phi(w+\alpha\, v)^{\mathrm{t}} v \geq \sigma_2 \nabla\phi(w)^{\mathrm{t}} v$$

for constants σ_1 and σ_2. The following result, which is a consequence of (7) and (8), relates the decreases in $\psi_d^k(x,z)$ and $\psi_d(x,z)$ and is important in establishing the global convergence of the algorithm. It shows that, under certain conditions, if α satisfies the first Wolfe condition for ψ_d^k (with constant σ) it also satisfies the Wolfe condition for ψ_d (with constant $\gamma\,\sigma$).

Proposition 3: Let $d > 0$ be sufficiently small and let $\gamma \in (0,1)$ and $\sigma \in (0,1/2)$ be specified. Let ϵ be a sufficiently small positive number and set

$$\mathcal{S}_\epsilon = \{(x,z) : ||(x,z)-(x^*,z^*)|| < \epsilon \text{ for some } (x^*,z^*) \in \mathcal{S}\}.$$

Then there exists an $\eta(\epsilon)$ such that if $(x^k,z^k) \in \mathcal{C}_{\eta(\epsilon)} - \mathcal{S}_\epsilon$ and

$$\psi_d^k(x^k+\alpha\,\delta^k, z^k+\alpha\, q^k) \leq \psi_d^k(x^k,z^k) + \sigma\,\alpha\,\nabla\psi_d^k(x^k,z^k)^{\mathrm{t}}(\delta^k,q^k)$$

then

$$\psi_d(x^k+\alpha\,\delta^k, z^k+\alpha\, q^k) \leq \psi_d(x^k,z^k) + \gamma\,\sigma\,\alpha\nabla\psi_d^k(x^k,z^k)^{\mathrm{t}}(\delta^k,q^k).$$

It is worth noting, although not emphasized here, that if the iterates are converging to a solution q-superlinearly, neither the merit function nor its approximations will interfere with this process, i.e., a steplength of $\alpha = 1$ is eventally acceptable. For details, see (Boggs et al. 1995).

3. A globally convergent algorithm

We first give a description of the algorithm and then proceed to state the basic global convergence theorem. The underlying idea of the algorithm is a standard one; the iterates are forced towards feasibility and then towards a solution. What distinguishes this algorithm is the use of the approximate merit functions that, far from feasibility, determine efficient steplengths that are likely to force the iterates toward optimality as well as feasibility and, near feasibility, provide relatively simple surrogates for the true merit function.

In the description of the algorithm, we drop the superscripts k on the iterates and the steps, denoting the current iterate by (x, z) and the step direction at this iterate by (δ, q). It is assumed that constants $d > 0$, $\eta > 0$, and $\sigma \in (0, 1/2)$ have been specified.

Algorithm

1. Set FLAG = FALSE.
2. If $||\bar{c}(x, z)||^2 < \eta$, set FLAG = TRUE.
3. Compute (δ, q) and set $\alpha = 2$.
4. Set $\alpha = \alpha/2$ and $(x_\alpha, z_\alpha) = (x, z) + \alpha\,(\delta, q)$.
5. If
$$\psi_d^k(x_\alpha, z_\alpha) \geq \psi_d^k(x, z) + \alpha\,\sigma\,\nabla\psi_d^k(x, z)^{\mathsf{t}}(\delta, q),$$
return to Step 4.
6. If FLAG = FALSE, then
   a. if
   $$||\bar{c}(x_\alpha, z_\alpha)||^2 \geq (1 - 2\,\sigma\,\alpha)\,||\bar{c}(x, z)||^2,$$
   return to Step 4;
   b. else set $(x, z) = (x_\alpha, z_\alpha)$ and return to Step 2.
7. If FLAG = TRUE, then
   a. if
   $$||\bar{c}(x_\alpha, z_\alpha)||^2 \geq \eta,$$
   return to Step 4;
   b. elseif
   $$\psi_d(x_\alpha, z_\alpha) \geq \psi_d(x, z)$$
   set $\eta = \frac{1}{2}||\bar{c}(x, z)||^2$ and return to Step 1;
   c. elseif
   $$\psi_d(x_\alpha, z_\alpha) \geq \psi_d(x, z) + \tfrac{1}{2}\alpha\,\sigma_1\,\nabla\psi_d(x, z)^{\mathsf{t}}(\delta, q),$$
   set $\eta = \frac{1}{2}||\bar{c}(x, z)||^2$, set $(x, z) = (x_\alpha, z_\alpha)$, and return to Step 1;

d. else set $(x,z) = (x_\alpha, z_\alpha)$ and return to Step 3.

We first note that the steplength parameter α is always chosen from (0,1) and hence the variable z will remain nonnegative. The determination of the α is, of course, the crucial factor in establishing that the resulting sequence of iterates converges to a solution of the problem. In our algorithm, the criteria for choosing the parameter depend upon whether or not the current iterate is sufficiently close to feasibility; i.e., on whether $(x,z) \in \mathcal{C}_\eta$ for some sufficiently small η. Unfortunately, an appropriate value of η cannot be determined *a priori* and hence the value must be adjusted as the algorithm proceeds.

In the algorithm the logical variable FLAG is set to TRUE when the current iterate (x,z) is in $\mathcal{C}_\eta$ for the current value of η and is FALSE otherwise. Thus given a value of η, if (x,z) is not in $\mathcal{C}_\eta$ the algorithm forces the iterates toward feasibility. In particular, the inequality in Step 6 forces α to be chosen so that the first Wolfe condition for the function $||\bar{c}(x,z)||^2$ is satisfied for constant σ. It follows from (6) that eventually the iterates will enter the current η-tube. By Step 5 the α is also chosen to ensure that the Wolfe condition for the approximate merit function is also satisfied at *every* step. That this is possible is a consequence of (7). Once the iterate is inside the given η-tube, it is not allowed thereafter to leave it; α is reduced if necessary to guarantee this (Step 7a). Since the step is always in the direction of feasiblility in (x,z)-space this restriction is not severe. Now the computed step is tested to see if the true merit function $\psi_d(x,z)$ satisfies the Wolfe condition for the constant $\sigma/2$ (Steps 7b–7d). If η is small enough then Proposition 3 ensures that such a decrease will occur for any α that yields a decrease in the approximate merit function $\psi_d^k(x,z)$ for the constant σ. If the Wolfe condition for $\psi_d(x,z)$ is not satisfied for the value of α then we take this as a signal that the current value of η is too large and we decrease η to one-half the current value of $\bar{c}(x,z)$. Thus when the value of η is decreased, it is decreased by at least a factor of one-half, so that the sequence of η values either tends to zero or else the the Wolfe condition is satisfied for $\psi_d(x,z)$ for all (x^k,z^k) for k sufficiently large.

There is another parameter central to our algorithm, namely the penalty parameter d. Here, we make the assumption that this parameter is initially small enough so that the basic propositions of the preceding section are satisfied. In the implementation of our algorithm we do have a heuristic procedure for adjusting d, but for the theoretical convergence we do not include this modification (but see Section 4).

Theorem: Led $d > 0$ be sufficiently small and let an initial $\eta > 0$ and an initial point (x^0, z^0) be given. Assume that the iterates lie in a compact set in which assumptions **A1**–**A3** hold. If (x^*,z^*) is a limit point of the sequence of iterations then $(x^*, z^*) \in \mathcal{C}_0$

and there exists a $\lambda^* \geq 0$ such that

$$\nabla f(x^*) + \nabla g(x^*)^{\mathrm{t}} \lambda^* = 0 \tag{9}$$

and

$$g(x^*)^{\mathrm{t}} \lambda^* = 0. \tag{10}$$

Outline of Proof: Let $\{\eta_j\}$ be the set of *assigned* values of η in the algorithm. If there are only a finite number, say K, of values then at some point all of the iterates are in the tube $\mathcal{C}_{\eta_K}$ and satisfy the first Wolfe condition for ψ_d. By an argument similar to that employed in proving global convergence of descent methods for unconstrained optimization algorithms, it can then be shown that the steps $(\delta, q) \to 0$. In this case the assumptions guarantee that $\delta = 0$ is a solution of (2) for some $B \in \mathcal{B}$ and hence (9) and (10) are satisfied at any limit point. If, on the other hand, $\eta_j \to 0$ then it is possible that ψ_d does not satisfy the desired Wolfe condition at a sequence of iterations. Nevertheless, any limit point must be in $\mathcal{C}_0$ and it follows from the assumptions and Proposition 3 that some subsequence of the steps must converge to zero. Then a similar argument to that above yields the desired conditions.

It is observed that the standard global convergence proofs for unconstrained optimization require that the second Wolfe condition be satisfied. This condition is imposed to keep the steps from becoming too small. Here, however, we have followed the common practice of using a backtracking linesearch to avoid overly small steps.

4. Algorithmic enhancements for large scale problems

Almost all algorithms need to have certain enhancements to perform effectively in practice. In this section we describe the most important enhancements that we have used in adapting the algorithm to solve large scale problems. The numerical experiments for this implementation are reported in (Boggs et al. 1994*b*).

4.1. The dynamic adjustment of d

It follows from Proposition 1 that if (x^k, z^k) isn't feasible, then the penalty parameter d can be made small enough to ensure that the step is a direction of descent for ψ_d^k. In our implementation if, at some nonfeasible point, a good decrease in ψ_d^k is not achieved, we decrease d by an appropriate factor. This can be done without fundamentally altering the convergence theory. However, using too small of a value of d often causes the algorithm to slow dramatically, since it tends to force the iterates to stay close to the constraints. To avoid this, it is desireable to incorporate a means for allowing an increase in d when the steps are good descent steps for ψ_d^k. Since increasing d can theoretically cause the iterates to cycle, care has to be taken in implementing such a procedure. We have used a heuristic strategy for increasing d while avoiding cycling that has been effective in our numerical experiments. However, a theoretical proof of global convergence when this process is implemented is lacking.

4.2. A trust region approach for solving the quadratic subproblems

In applying the SQP method to large scale problems, it is necessary to obtain a solution (or else an approximate solution) to the quadratic subproblem quickly. Since the data for (2) are likely to be sparse this suggests the use of an iterative scheme for the solution. We have used such a procedure, an interior point algorithm, called the O3D method, that has special properties that make it particularly compatible with our SQP algorithm. The defining characteristic of this quadratic program solver is that it proceeds by solving a sequence of three-dimensional subspace approximations to the quadratic program. At a current iterate (of the quadratic program) three independent directions are generated and the quadratic program restricted to the subspace determined by these directions is solved. The next iterate is taken to be 99% of the distance to the boundary in the direction thus generated. A detailed description of this method is found in (Boggs, Domich and Rogers 1994*a*).

The O3D solver is employed in the following way for the SQP algorithm. At iteration k in the main algorithm, the quadratic program (2) is formed and the iteration procedure is begun. The algorithm is stopped when a prescribed tolerance for an optimality condition is satisfied *or* when an implicit *trust region constraint* of the form

$$||\delta|| \leq \tau_k$$

is violated. If the final iterate, δ_j^k, is optimal (or very nearly optimal) for (2) then, of course, the theory described in the preceding section applies. The significance of this trust region approach lies in the case where the jth iterate is not optimal. The remarkable fact is that if q^k is set equal to $-(\nabla g(x^k)^{\mathrm{t}}\delta_j^k + g(x^k) + z^k)$ then the step (δ_j^k, q^k) satisfies the strong descent conditions given in Propositions 1 and 2 despite the fact that it is not optimal. This implies that the algorithm described in the preceding section can be applied using approximate solutions to the quadratic subproblems rather than the true optimal solutions. Moreover, together with the fact that the trust region constraint will become irrelevant near the solution, it means that a global convergence analysis of this trust region algorithm modeled after that of Section 3 is feasible.

The trust region parameter, τ_k, is adjusted in a manner similar in spirit to that used in most trust region methods; that is, the decision to increase or decrease τ_k is based on a comparison of the predicted and actual reductions of the merit function. In our implementation we use either ψ_d or ψ_d^k depending on the current status of the point (x^k,z^k). For details of this procedure, see (Boggs et al. 1994*b*).

4.3. Inconsistent quadratic subproblems

One of the major theoretical weaknesses of SQP methods is the assumption that the quadratic subproblems, (2), are feasible. It is a troublesome practical problem as well since it is a reasonably common occurrence to have the linearized constraints be inconsistent. Therefore, any implementation must take into account this possibility.

The use of the O3D quadratic program solver gives a natural way to handle this difficulty in our algorithm. The O3D solver uses an artificial variable together with an adaptive "big M" method for generating a starting point, so if the quadratic problem is infeasible then the algorithm will terminate with a step in the x variables, δ_j^k, that (together with a value of the artificial variable) is either optimal in the modified problem or exceeds the trust region constraint. In either case it can be shown that, under an additional assumption on the "optimal" artificial variable, the step (δ_j^k, q^k) is still a descent step for $\bar{c}, \psi_d^k$, and ψ_d in the sets prescribed in Propositions 1 and 2. Therefore using this step allows us to move away from a point where the quadratic subproblem is inconsistent in a direction that is compatible with the goals of the algorithm. A theoretical difficulty that arises from this approach is that the resulting point could be such that z^{k+1} is negative (since $\nabla g(x^k)^{\mathrm{t}}\delta_j^k + g(x^k)$ will have positive components). Our procedure has been to accept such steps and simply reset any negative components of z to be positive. Although this procedure has been quite successful in practice, it has not been possible to prove global convergence in this case.

References

P. T. Boggs and J. W. Tolle (1995), 'Sequential quadratic programming', *Acta Numerica* **1995**, 1–52.

P. T. Boggs, P. D. Domich and J. E. Rogers (1994*a*), An interior-point method for general large scale quadratic programming problems, Internal Report 5406 (accepted in *Annals of Operations Research*), National Institute of Standards and Technology.

P. T. Boggs, J. W. Tolle and A. J. Kearsley (1994*b*), A practical algorithm for general large scale nonlinear optimization problems, Internal report, National Institute of Standards and Technology.

P. T. Boggs, J. W. Tolle and A. J. Kearsley (1995), A basic global convergence analysis of a large scale algorithm for general nonlinear programming problems, Internal report, National Institute of Standards and Technology.

J. Nocedal (1992), 'Theory of algorithms for unconstrained optimization', *Acta Numerica* **1991**, 199–242.