ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

# NEWTON'S METHOD FOR LARGE BOUND-CONSTRAINED OPTIMIZATION PROBLEMS

**Chih-Jen Lin and Jorge J. Moré**

Mathematics and Computer Science Division

Preprint ANL/MCS-P724-0898

August 1998 (Revised April 1999)

This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Department of Energy, under Contract W-31-109-Eng-38, by the National Science Council of Taiwan grant NSC-88-2213-E-002-097 and by the the National Science Foundation grant CCR-9321550.

# Newton's Method for Large Bound-Constrained Optimization Problems

**Chih-Jen Lin*** **and Jorge J. Moré**†

**Abstract**

We analyze a trust region version of Newton's method for bound-constrained problems. Our approach relies on the geometry of the feasible set, not on the particular representation in terms of constraints. The convergence theory holds for linearly-constrained problems, and yields global and superlinear convergence without assuming neither strict complementarity nor linear independence of the active constraints. We also show that the convergence theory leads to an efficient implementation for large bound-constrained problems.

## 1 Introduction

We analyze a trust region version of Newton's method for the optimization problem

$$\min \{ f(x) : x \in \Omega \}, \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable mapping on the bound-constrained set

$$\Omega = \{ x \in \mathbb{R}^n : l \le x \le u \}. \tag{1.2}$$

Our analysis relies on the geometry of $\Omega$ and applies, without change, to the case where $\Omega$ is the linearly-constrained set

$$\Omega = \Big\{ x \in \mathbb{R}^n : l_i \le \langle c_i, x \rangle \le u_i, \ i \in \mathcal{I} \Big\}. \tag{1.3}$$

The convergence theory yields results that are independent of the representation of $\Omega$ in terms of constraints; in particular, we assume neither strict complementarity (nonzero multipliers) nor linear independence of the active constraints.

Our main interest is in algorithms for large optimization problems so the convergence theory that we develop emphasizes algorithms that use iterative techniques to solve the trust region subproblem, while retaining superlinear convergence of the trust region method. We show, in particular, how the convergence theory leads to an efficient implementation of Newton's method when the feasible set $\Omega$ is the bound-constrained set (1.2).

This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

*Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan (`cjlin@csie.ntu.edu.tw`). Work supported by in part by the National Science Council of Taiwan grant NSC-88-2213-E-002-097 and the National Science Foundation grant CCR-9321550.

†Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, Illinois 60439 (`more@mcs.anl.gov`).

Our development of a convergence theory for Newton's method yields three main results. We first establish *global convergence* to a stationary point; that is, if $\{x_k\}$ is the sequence generated by the trust region method, then every limit point of the sequence is a stationary point for problem (1.1). We then establish the *identification properties* of the algorithm by showing that if $\{x_k\}$ converges to some $x^*$, then there is an integer $k_0$ such that $x_k$ lands in the face *exposed* by $-\nabla f(x^*)$ for all $k \geq k_0$. Finally, we establish the *local convergence* properties of the algorithm. The main result shows that if a strong second-order sufficiency condition holds at a limit point $x^*$ of the trust region iterates, then the whole sequence $\{x_k\}$ converges to $x^*$ at a superlinear rate.

Global and superlinear convergence for linearly-constrained problems has been established, in almost all cases, under the assumption of strict complementarity. Moreover, the algorithms that have been analyzed usually require the exact solution of systems of linear equations. See, for example, [2, 22, 33, 18] for algorithms that use $\epsilon$-active constraints, [23, 20] for active set methods, [13, 25, 12, 21] for trust region methods, and [9, 16, 11, 10] for interior-point methods. In recent work Heinkenschloss, Ulbrich, and Ulbrich [24] analyzed an interior-point method without assuming strict complementarity, but proved only local convergence.

Lescrenier [25] and Facchinei and Lucidi [19] analyze algorithms for bound-constrained problems that are shown to be superlinearly convergent without assuming strict complementarity. Lescrenier analyses the trust region method of Conn, Gould, and Toint [13]. Facchinei and Lucidi analyze a line search algorithm based on a differentiable exact penalty function that, unlike the algorithms for bound-constrained problems that we have reviewed, generates iterates that need not be feasible.

We analyze a trust region method for the linearly-constrained optimization problem (1.3) based on the convergence theory of Moré [27] and Burke, Moré, and Toraldo [7]. The analysis relies on the geometric approach of Burke and Moré [6] for general linearly-constrained problems. We use projected searches [30] during the subspace minimization phase, and thus we are able to add many constraints during this phase. We show that global and superlinear convergence holds even if strict complementarity fails for the general linearly-constrained optimization problem (1.3).

The convergence theory for trust region methods presented in Section 2 depends on the definition of the Cauchy step $s_k^C$. The main result in this section shows that global convergence to a stationary point is guaranteed if the step $s_k$ in the trust region method achieves a fraction of the reduction achieved by the Cauchy step.

The standard development of identification properties for optimization algorithm shows that the active set settles down if the iterates converge to a stationary point $x^*$. This approach is not possible if strict complementarity does not hold at $x^*$. In Section 3 we show that the sequence generated by the trust region method is trapped by the face exposed by

$-\nabla f(x^*)$; Section 3 provides a precise definition of the face of a convex set exposed by a vector. If strict complementarity holds at $x^*$, this result implies that the active set settles down.

In Section 3 we also explore the concept of strict complementarity and its relationship to the concept of an exposed face. In this paper we use the term *nondegenerate stationary point* $x^*$ if strict complementarity holds at $x^*$ or, equivalently, if $x^*$ is in the relative interior of the face exposed by $-\nabla f(x^*)$.

Section 4 defines the projected searches that are used to explore the current face of the feasible set. Projected searches are an important ingredient of the optimization algorithm because they allow wider latitude in the choice of the next iterate. In particular, the active constraints are allowed to change arbitrarily, while requiring only the approximate solution of a linear system.

Section 5 contains the major convergence results for the trust region Newton's method. We show that if a strong second-order sufficiency condition holds at a limit point $x^*$ of the trust region iterates, then the whole sequence $\{x_k\}$ converges to $x^*$. Previous results assumed strict complementarity, and that the problem was bound-constrained. We also show that if the sequence $\{x_k\}$ converges to $x^*$, then the rate of convergence is at least superlinear.

Section 6 briefly outlines the implementation of TRON (version 1.0), a trust region Newton method for bound-constrained problems. Interesting features of this implementation include the use of projected searches and a preconditioned conjugate gradient method to determine the minor iterates, and the use of a limited-memory preconditioner. We use the incomplete Cholesky factorization icfs of Lin and Moré [26] as a preconditioner since this factorization does not require the choice of a drop tolerance, and the amount of storage can be specified in advance.

Section 7 presents the results of a comparison between TRON and the LANCELOT [14] and L-BFGS-B [36] codes. These results show that on the problems described in this section, TRON is generally more efficient, in terms of computing time, than LANCELOT and L-BFGS-B. Caution must be exercized in drawing conclusion from these results since, as noted in Section 7, there are many differences between TRON and LANCELOT.

## 2 Trust Region Methods

In this section we present a trust region method for the solution of optimization problems subject to linear constraints, but we emphasize the case where $\Omega$ is the bound-constrained set (1.2). The algorithm that we present was proposed by Moré [27] as a modification of the algorithm of Toint [35]. The development in this section follows Moré [27] and Burke, Moré, and Toraldo [7].

At each iteration of a trust region method there is an approximation $x_k \in \Omega$ to the solution, a bound $\Delta_k$, and a model $\psi_k : \mathbb{R}^n \to \mathbb{R}$ of the possible reduction $f(x_k + w) - f(x_k)$ for $\|w\| \le \Delta_k$. We assume that the model $\psi_k$ is the quadratic

$$\psi_k(w) = \langle \nabla f(x_k), w \rangle + \tfrac{1}{2}\langle w, B_k w \rangle$$

for some symmetric matrix $B_k$. The matrix $B_k$ is arbitrary for many of the results, but the rate of convergence results usually requires that $B_k$ be the Hessian matrix $\nabla^2 f(x_k)$. Of course, it is possible to choose $B_k = 0$, and then the model is linear.

The description of the algorithm in terms of the quadratic $\psi_k$ is appropriate when we are interested in the step $s_k$. However, we also use the quadratic

$$q_k(x) = \psi_k(x - x_k) = \langle \nabla f(x_k), x - x_k \rangle + \tfrac{1}{2}\langle x - x_k, B_k(x - x_k) \rangle$$

to describe the algorithm in terms of the iterates $x_k$.

The iterate $x_k$ and the bound $\Delta_k$ are updated according to rules that are standard in trust region methods for unconstrained minimization. Given a step $s_k$ such that $x_k + s_k \in \Omega$ and $\psi_k(s_k) < 0$, these rules depend on the ratio

$$\rho_k = \frac{f(x_k + s_k) - f(x_k)}{\psi_k(s_k)} \tag{2.1}$$

of the actual reduction in the function to the predicted reduction in the model. Since the step $s_k$ is chosen so that $\psi_k(s_k) < 0$, a step with $\rho_k > 0$ yields a reduction in the function. Given $\eta_0 > 0$, the iterate $x_k$ is updated by setting

$$x_{k+1} = \begin{cases} x_k + s_k & \text{if} \quad \rho_k > \eta_0 \\ x_k & \text{if} \quad \rho_k \le \eta_0. \end{cases} \tag{2.2}$$

Any step $s_k$ with $\rho_k > \eta_0$ is *successful*; otherwise the step in *unsuccessful*. Under suitable conditions, all steps (iterations) are eventually successful.

Updating rules for $\Delta_k$ depend on positive constants $\eta_1$ and $\eta_2$ such that $\eta_1 < \eta_2 < 1$, while the rate at which $\Delta_k$ is updated depends on positive constants $\sigma_1, \sigma_2$, and $\sigma_3$ such that $\sigma_1 < \sigma_2 < 1 < \sigma_3$. The trust region bound $\Delta_k$ is updated by setting

$$\begin{array}{lll} \Delta_{k+1} \in [\sigma_1 \min\{\|s_k\|, \Delta_k\}, \sigma_2 \Delta_k] & \text{if} & \rho_k \le \eta_1 \\ \Delta_{k+1} \in [\sigma_1 \Delta_k, \sigma_3 \Delta_k] & \text{if} & \rho_k \in (\eta_1, \eta_2) \\ \Delta_{k+1} \in [\Delta_k, \sigma_3 \Delta_k] & \text{if} & \rho_k \ge \eta_2. \end{array} \tag{2.3}$$

Similar rules are used in most modern trust region methods.

We choose a step $s_k$ that gives as much reduction in the model $\psi_k$ as the Cauchy step $s_k^C$ generated by the gradient projection method applied to the subproblem

$$\min \left\{ \psi_k(w) : x_k + w \in \Omega, \ \|w\| \le \Delta_k \right\}.$$

The Cauchy step $s_k^C$ is of the form $s_k(\alpha_k)$, where the function $s_k : \mathbb{R} \mapsto \mathbb{R}^n$ is defined by

$$s_k(\alpha) = P\left[x_k - \alpha \nabla f(x_k)\right] - x_k,$$

where $P : \mathbb{R}^n \mapsto \Omega$ is the projection into the feasible set $\Omega$. If $\Omega$ is the bound-constrained set (1.2), then the projection can be computed with at most $2n$ comparisons by

$$P(x) = \mathrm{mid}\,(l, x, u)\,,$$

where $\mathrm{mid}(\cdot)$ is the componentwise median (middle) of the three vectors in the argument. The trust region method that we describe can be implemented efficiently if there is an efficient algorithm for computing the projection $P$.

The scalar $\alpha_k$ that determines the Cauchy step $s_k^C$ is chosen so that $s_k(\alpha_k)$ produces a sufficient reduction. We require that

$$\psi_k(s_k(\alpha_k)) \le \mu_0 \left\langle \nabla f(x_k), s_k(\alpha_k) \right\rangle, \qquad \|s_k(\alpha_k)\| \le \mu_1 \Delta_k, \tag{2.4}$$

for positive constants $\mu_0$ and $\mu_1$ such that $\mu_0 < \frac{1}{2}$. We also require that there are positive constants $\gamma_1$, $\gamma_2$, and $\gamma_3$ such that

$$\alpha_k \in [\gamma_1, \gamma_3] \quad \text{or} \quad \alpha_k \in [\gamma_2 \widetilde{\alpha}_k, \gamma_3],$$

where $\widetilde{\alpha}_k > 0$ satisfies

$$\psi_k(s_k(\widetilde{\alpha}_k)) \ge (1 - \mu_0) \left\langle \nabla f(x_k), s_k(\widetilde{\alpha}_k) \right\rangle \quad \text{or} \quad \|s_k(\widetilde{\alpha}_k)\| \ge \mu_1 \Delta_k.$$

The requirements on the Cauchy step $s_k^C$ can be satisfied [27, 7] with a finite number of evaluations of $\psi_k$. For additional details, see Section 6.

We have described the requirements on the Cauchy step $s_k^C$ in terms of the quadratic $\psi_k$, but we could also use $q_k$. In particular,

$$q_k(x_k + s_k^C) \le q_k(x_k) + \mu_0 \left\langle \nabla q_k(x_k), s_k^C \right\rangle$$

is the sufficient reduction condition (2.4).

Given the Cauchy step $s_k^C$, we require that the step $s_k$ satisfy

$$\psi_k(s_k) \le \mu_0 \psi_k(s_k^C), \qquad \|s_k\| \le \mu_1 \Delta_k, \qquad x_k + s_k \in \Omega. \tag{2.5}$$

This requirement is quite natural and can always be satisfied by choosing $s_k = s_k^C$. However, this choice is likely to lead to slow convergence, because the method would then reduce to a version of steepest descent. In the next section we explore other choices that lead to superlinear and quadratic convergence.

Algorithm 2.1 summarizes the computations required to implement the trust region method. We assume that $f : \mathbb{R}^n \mapsto \mathbb{R}$ is continuously differentiable on $\Omega$ and that a $\Delta_0 > 0$ has been specified.

For $k = 0, \ldots,$

Compute the model $\psi_k$.

Compute the Cauchy step $s_k^C$.

Compute a step $s_k$ that satisfies (2.5).

Compute the ratio $\rho_k$ and update $x_k$ by (2.2).

Update $\Delta_k$ according to (2.3).

Algorithm 2.1: Major computational steps in a trust region method.

Burke, Moré, and Toraldo [7] analyzed the trust region method of Algorithm 2.1 in terms of the Cauchy point

$$x_k^C \equiv P\left[x_k + \alpha_k \nabla f(x_k)\right] = x_k + s_k^C.$$

Convergence results depend on a bound on the predicted decrease for the quadratic $\psi_k$. This bound is based on the inequality

$$-\left\langle \nabla f(x_k), s_k^C \right\rangle \geq \kappa_0 \left[\frac{\|x_k^C - x_k\|}{\alpha_k}\right] \min\left\{\Delta_k, \frac{1}{\|B_k\|}\left[\frac{\|x_k^C - x_k\|}{\alpha_k}\right], \right\} \tag{2.6}$$

where $\kappa_0$ is a positive constant. This bound was obtained by Moré [27]. Other bounds obtained for problems with bound constraints and, more generally, convex constraints [13, 35, 12] do not yield the same information because they are not expressed in terms of the Cauchy point.

The choice of $s_k^C$ is an important ingredient in the trust region method. Our choice of $s_k^C$ is simple and can be implemented efficiently provided there is an efficient algorithm for computing the projection $P$. For other choices, see [13, 35, 12].

Many of the convergence results in Burke, Moré, and Toraldo [7] are expressed in terms of the *projected gradient*

$$\nabla_\Omega f(x) \equiv P_{T(x)}\left[-\nabla f(x)\right] = \text{argmin}\{\|v + \nabla f(x)\| : v \in T(x)\},$$

where the *tangent cone* $T(x)$ is the closure of the cone of all feasible directions at $x \in \Omega$, and $\Omega$ is a general convex set. The term projected gradient is not entirely appropriate. Indeed, since

$$\min\left\{\langle \nabla f(x),\ v\rangle : v \in T(x), \|v\| \leq 1\right\} = -\|\nabla_\Omega f(x)\|, \tag{2.7}$$

it might be more appropriate to call $\nabla_\Omega f(x)$ the *projected steepest descent direction.* The optimality property (2.7) follows from the properties of the projection on convex cones; Calamai and Moré [8] provide a direct proof of (2.7).

The projected gradient should not be confused with the *reduced gradient.* When $\Omega$ is the bound-constrained set (1.2), the reduced gradient is the vector with components $\partial_i f(x)$

if $l_i < x_i < u_i$, while for the projected gradient

$$-[\nabla_\Omega f(x)]_i = \begin{cases} \partial_i f(x) & \text{if } x_i \in (l_i, u_i) \\ \min\{\partial_i f(x), 0\} & \text{if } x_i = l_i \\ \max\{\partial_i f(x), 0\} & \text{if } x_i = u_i \end{cases} \tag{2.8}$$

if $l_i < u_i$, with $[\nabla_\Omega f(x)]_i = 0$ in the exceptional case where $l_i = u_i$. The appearance of the minus sign in this expression for the projected gradient is only a minor nuisance because in our work we need only an expression for $\|\nabla_\Omega f(x)\|$.

The projected gradient $\nabla_\Omega f$ can be used to characterize stationary points because if $\Omega$ is a convex set, then $x \in \Omega$ is a stationary point of problem (1.1) if and only if $\nabla_\Omega f(x) = 0$. In general, $\nabla_\Omega f$ is discontinuous, but as proved by Calamai and Moré [8], if $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable on $\Omega$, then the mapping $x \mapsto \|\nabla_\Omega f(x)\|$ is lower semicontinuous on $\Omega$. This property implies that if $\{x_k\}$ is a sequence in $\Omega$ that converges to $x^*$, and if $\{\nabla_\Omega f(x_k)\}$ converges to zero, then $x^*$ is a stationary point of problem (1.1). In Section 3 we show that the continuity properties of the projected gradient are closely associated with the behavior of the optimization algorithm.

**Theorem 2.1** *Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be continuously differentiable on a closed, convex set $\Omega$, and let $\{x_k\}$ be the sequence generated by the trust region method. Assume that $\{B_k\}$ is uniformly bounded. If $x^*$ is a limit point of $\{x_k\}$ then there is a subsequence $\{x_{k_i}\}$ of successful steps that converges to $x^*$ with*

$$\lim_{i \to \infty} \|\nabla_\Omega f(x_{k_i}^C)\| = 0. \tag{2.9}$$

*Moreover, $\{x_{k_i}^C\}$ also converges to $x^*$, and thus $x^*$ is a stationary point for problem* (1.1).

This result is due to Burke, Moré, and Toraldo [7, Theorem 5.5]. Similar convergence results for bound-constrained and linearly-constrained optimization algorithms assert that every limit point of the algorithm is stationary, but do not yield any information on the projected gradient; in Sections 3 and 5 we show that (2.9) in Theorem 2.1 plays an important role in the convergence analysis. For a sampling of recent convergence results, see [12, 18, 9, 16, 20, 33].

## 3 Exposing Constraints

Identification properties are an important component of the convergence analysis of an algorithm for linearly-constrained problems. We show that if $x^*$ is a stationary point and $\Omega$ is the polyhedral set (1.3), then the iterates $\{x_k\}$ generated by the trust region method tend to lie in the face exposed by the direction $-\nabla f(x^*)$.

The notion of an exposed face arises in convex analysis, where the face of a convex set $\Omega$ exposed by the vector $d \in \mathbb{R}^n$ is

$$E[d] \equiv \operatorname{argmax}\left\{x \in \Omega : \langle d, x\rangle\right\}.$$

A short computation shows that when g$\Omega = [l, u]$ is the bound-constrained set (1.2) and $d = -\nabla f(x^*)$, then

$$E\left[-\nabla f(x^*)\right] = \Big\{x \in [l, u] : x_i = l_i \text{ if } \partial_i f(x^*) > 0 \text{ and } x_i = u_i \text{ if } \partial_i f(x^*) < 0\Big\}$$

is the face of (1.2) exposed by the direction $-\nabla f(x^*)$. A similar expression holds if $\Omega$ is the polyhedral set defined by (1.3). If $x^*$ is a stationary point of the optimization problem (1.1), then there are Lagrange multipliers such that

$$\nabla f(x^*) = \sum_{i \in \mathcal{A}(x^*)} \lambda_i^* c_i,$$

where $\lambda_i^*$ is unrestricted in sign if $l_i = u_i$, but

$$\lambda_i^* \geq 0 \ \text{ if } \ \langle c_i, x^*\rangle = l_i, \qquad \lambda_i^* \leq 0 \ \text{ if } \ \langle c_i, x^*\rangle = u_i,$$

and $\mathcal{A}(x)$ is the set of active constraints at $x \in \Omega$ defined by

$$\mathcal{A}(x) = \Big\{i \in \mathcal{I} : \langle c_i, x\rangle \in \{l_i, u_i\}\Big\}.$$

This definition of the active set does not distinguish between lower and upper bounds, and thus we need to interpret the inclusion $\mathcal{A}(x) \subset \mathcal{A}(y)$ to mean

$$\mathcal{A}_l(x) \subset \mathcal{A}_l(y), \qquad \mathcal{A}_u(x) \subset \mathcal{A}_u(y),$$

where

$$\mathcal{A}_l(x) = \Big\{i \in \mathcal{I} : \langle c_i, x\rangle = l_i\Big\}, \qquad \mathcal{A}_u(x) = \Big\{i \in \mathcal{I} : \langle c_i, x\rangle = u_i\Big\}.$$

With this interpretation, if $\langle c_i, x\rangle = l_i$ and $\mathcal{A}(x) \subset \mathcal{A}(y)$, then $\langle c_i, y\rangle = l_i$. For most results we only need to know that $\langle c_i, x\rangle \in \{l_i, u_i\}$, and then the first definition of the active set is suitable.

The face exposed by $-\nabla f(x^*)$ is determined by the nonzero multipliers. Indeed, a computation based on the definition of a face shows that

$$E\left[-\nabla f(x^*)\right] = \Big\{x \in \Omega : \langle c_i, x\rangle = l_i \text{ if } \lambda_i^* > 0 \text{ and } \langle c_i, x\rangle = u_i \text{ if } \lambda_i^* < 0\Big\}. \tag{3.1}$$

Note that this expression for $E\left[-\nabla f(x^*)\right]$ is valid for any choice of Lagrange multipliers.

Burke and Moré [6] provide additional information on exposed faces. In particular, they note that for $\Omega$ convex, $x^*$ is a stationary point for the optimization problem (1.1) if and only if $x^* \in E\left[-\nabla f(x^*)\right]$.

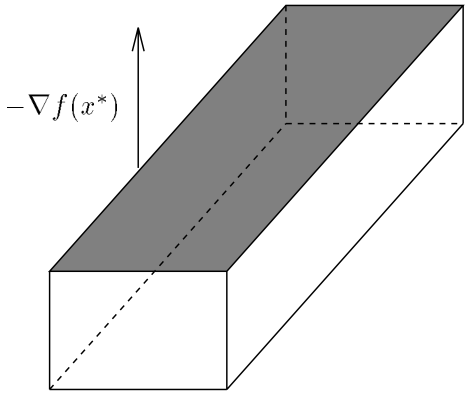


Figure 3.1: The exposed face $E[-\nabla f(x^*)]$ for a degenerate problem.

Dunn [17] defines $x^*$ to be a nondegenerate stationary point if $-\nabla f(x^*)$ lies in the relative interior of the normal cone

$$N(x^*) = \{u \in \mathbb{R}^n : \langle u, y - x^* \rangle \le 0, \ y \in \Omega\}.$$

Burke and Moré [6] relate nondegeneracy to the geometry of $E[-\nabla f(x^*)]$ by proving that $x^*$ is nondegenerate if and only if $x^*$ lies in the relative interior of the face $E[-\nabla f(x^*)]$. These two definitions rely only on the geometry of $\Omega$. If $\Omega$ is expressed in terms of constraints, then nondegeneracy can be shown [5] to be equivalent to the existence of a set of nonzero Lagrange multipliers. Thus, a stationary point $x^*$ is nondegenerate as defined by Dunn [17] if and only if strict complementarity holds at $x^*$. We can also show [6, Theorem 5.3] that

$$x \in E[-\nabla f(x^*)] \qquad \Longleftrightarrow \qquad \mathcal{A}(x^*) \subset \mathcal{A}(x) \tag{3.2}$$

whenever $x^*$ is nondegenerate. Thus, for nondegenerate problems, landing in the face $E[-\nabla f(x^*)]$ can be described in terms of active sets.

Figure 3.1 illustrates some of the properties of exposed faces. In this case $x^*$ is in the relative boundary of the face, so this problem is degenerate. Note that in this case (3.2) fails because $\mathcal{A}(x^*)$ may not be a subset of $\mathcal{A}(x)$ for $x \in E[-\nabla f(x^*)]$. Finally, note that $x - y$ is orthogonal to $\nabla f(x^*)$ whenever $x$ and $y$ are in $E[-\nabla f(x^*)]$. This last observation holds for any convex set $\Omega$ because the mapping $x \mapsto \langle \nabla f(x^*), x\rangle$ is constant on $E[-\nabla f(x^*)]$.

For nondegenerate problems we can show that eventually all iterates land in the relative interior of $E[-\nabla f(x^*)]$. For degenerate problems this is not possible, but we can show that eventually all iterates land in $E[-\nabla f(x^*)]$. We first prove a technical result that shows that if $\{x_k\}$ is any sequence that converges to a stationary point $x^*$, and $x_k$ lands in $E[-\nabla f(x^*)]$, then $x_k^C$ remains in $E[-\nabla f(x^*)]$. We need the following result (Theorem 4.2) of Burke and Moré [6].

**Theorem 3.1** *Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be continuously differentiable on the polyhedral set $\Omega$, and let $\{x_k\}$ be any sequence in $\Omega$ that converges to a stationary point $x^*$. Then*

$$\lim_{k \to +\infty} \|\nabla_\Omega f(x_k)\| = 0$$

*if and only if there is an index $k_0$ with $x_k \in E[-\nabla f(x^*)]$ for $k \geq k_0$.*

Theorem 3.1 is of interest because it provides a means to show that iterates land in the exposed face $E[-\nabla f(x^*)]$. Note that in this result $\{x_k\}$ can be any sequence in $\Omega$. We now show that if $x_k$ lands in $E[-\nabla f(x^*)]$, then $x_k^C$ remains in $E[-\nabla f(x^*)]$.

**Theorem 3.2** *Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be continuously differentiable on the polyhedral set $\Omega$, and let $\{x_k\}$ be any sequence that converges to a stationary point $x^*$. If $x_k \in E[-\nabla f(x^*)]$ for $k \geq k_0$, then*

$$P[x_k - \alpha_k \nabla f(x_k)] \in E[-\nabla f(x^*)]$$

*for $k$ sufficiently large.*

**Proof.** The proof relies on Theorem 3.1 of Burke and Moré [6], which shows that for any sequence $\{d_k\}$ in $\mathbb{R}^n$ that converges to $d^*$

$$E[d_k] \subset E[d^*] \tag{3.3}$$

for all $k$ sufficiently large. If $N(x)$ is the normal cone at $x \in \Omega$, the definition of the projection operator implies that

$$x_k - \alpha_k \nabla f(x_k) - P[x_k - \alpha_k \nabla f(x_k)] \in N\Big(P[x_k - \alpha_k \nabla f(x_k)]\Big).$$

The definition of the exposed face shows that $x \in E[d]$ if and only if $d \in N(x)$, and thus

$$P[x_k - \alpha_k \nabla f(x_k)] \in E\Big[-\alpha_k \nabla f(x_k) + x_k - P[x_k - \alpha_k \nabla f(x_k)]\Big] = E[d_k], \tag{3.4}$$

where we have defined the sequence $\{d_k\}$ by

$$d_k = -\nabla f(x_k) + \frac{x_k - P[x_k - \alpha_k \nabla f(x_k)]}{\alpha_k}.$$

We now claim that

$$\left\| \frac{P[x_k - \alpha_k \nabla f(x_k)] - x_k}{\alpha_k} \right\| \leq \|\nabla_\Omega f(x_k)\|. \tag{3.5}$$

If we accept this claim, we can complete the proof by noting that, since $\{x_k\}$ converges to $x^*$ and $x_k \in E[-\nabla f(x^*)]$, Theorem 3.1 and inequality (3.5) show that the sequence $\{d_k\}$ converges to $-\nabla f(x^*)$. Hence, (3.3) and (3.4) imply that $P[x_k - \alpha_k \nabla f(x_k)]$ belongs to $E[-\nabla f(x^*)]$ for all $k$ sufficiently large.

The proof of (3.5) requires two inequalities. First note that the optimality property (2.7) of the projected gradient $\nabla_\Omega f$ implies that

$$-\langle \nabla f(x), v \rangle \leq \|\nabla_\Omega f(x)\| \, \|v\|,$$

for any feasible direction $v$ at $x$. In particular,

$$-\langle \nabla f(x), s(\alpha) \rangle \leq \|\nabla_\Omega f(x)\| \, \|s(\alpha)\|,$$

where we have defined $s(\alpha) = P[x - \alpha \nabla f(x)] - x$. Next, note that the definition of the projection operator, $\langle P(x) - x, y - P(x) \rangle \geq 0$ for any $y \in \Omega$, implies that

$$-\langle \nabla f(x), s(\alpha) \rangle \geq \frac{\|s(\alpha)\|^2}{\alpha}.$$

The last two displayed inequalities imply (3.5) as desired. ■

We want to show that all iterates eventually stay in the exposed face $E[-\nabla f(x^*)]$. Theorems 2.1 and 3.1 show that if the sequence $\{x_k\}$ converges to $x^*$, then $x_k^C$ lands in $E[-\nabla f(x^*)]$ for some subsequence of successful iterates. We now restrict the step $s_k$ so that the next iterate does not leave $E[-\nabla f(x^*)]$. The following result makes use of the observation that

$$x \in E[-\nabla f(x^*)], \quad \mathcal{A}(x) \subset \mathcal{A}(y) \qquad \Longrightarrow \qquad y \in E[-\nabla f(x^*)].$$

This observation follows directly from the expression (3.1) for $E[-\nabla f(x^*)]$.

**Theorem 3.3** *Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be continuously differentiable on the polyhedral set $\Omega$, and let $\{x_k\}$ be the sequence generated by the trust region method. Assume that $\{B_k\}$ is uniformly bounded and that the step $s_k$ satisfies*

$$\mathcal{A}(x_k^C) \subset \mathcal{A}(x_k + s_k), \qquad k \geq 0. \tag{3.6}$$

*If $\{x_k\}$ converges to $x^*$, then there is an index $k_0$ such that*

$$x_k \in E[-\nabla f(x^*)], \quad x_k + s_k \in E[-\nabla f(x^*)], \qquad k \geq k_0.$$

**Proof.** Theorem 2.1 shows that there is a sequence $\mathcal{K}$ of successful iterates such that if $k \in \mathcal{K}$, then $\{x_k^C\}$ converges to $x^*$ and $\{\nabla_\Omega f(x_k^C)\}$ converges to zero. Hence, Theorem 3.1 shows that

$$x_k^C \in E[-\nabla f(x^*)], \qquad k \in \mathcal{K}.$$

Since every iterate in $\mathcal{K}$ is successful, assumption (3.6) implies that $x_{k+1} = x_k + s_k$ belongs to $E[-\nabla f(x^*)]$. In particular, there is an index $k_0$ such that $x_{k_0}$ belongs to $E[-\nabla f(x^*)]$. We now show that $x_k$ belongs to $E[-\nabla f(x^*)]$ for all $k \geq k_0$.

Assume that $x_k$ belongs to $E[-\nabla f(x^*)]$ for some $k \geq k_0$. Theorem 3.2 shows that $x_k^C \in E[-\nabla f(x^*)]$. Hence, assumption (3.6) on the step yields that $x_k + s_k \in E[-\nabla f(x^*)]$. If $x_{k+1} = x_k$, then $x_{k+1}$ clearly belongs to $E[-\nabla f(x^*)]$, while if $x_{k+1} = x_k + s_k$, then we also have $x_{k+1}$ in $E[-\nabla f(x^*)]$. Hence, in all cases $x_{k+1}$ belongs to $E[-\nabla f(x^*)]$.

We have shown that $x_k$ belongs to $E[-\nabla f(x^*)]$ for all $k \geq k_0$. Hence, Theorem 3.2 shows that $x_k^C \in E[-\nabla f(x^*)]$, and thus assumption (3.6) on the step yields that $x_k + s_k$ is in $E[-\nabla f(x^*)]$. ∎

## 4 Projected Searches

The convergence theory of the trust region Newton method depends on generating the step $s_k$ so that conditions (2.5) and (3.6) are satisfied. We determine $s_k$ by computing $m+1$ minor iterates $x_{k,1}, \ldots, x_{k,m+1}$, where $x_{k,1} = x_k^C$. We require that

$$x_{k,j} \in \Omega, \qquad \mathcal{A}(x_k^C) \subset \mathcal{A}(x_{k,j}), \qquad \|x_{k,j} - x_k\| \leq \mu_1 \Delta_k, \tag{4.1}$$

and that the decrease condition

$$q_k(x_{k,j+1}) \leq q_k(x_{k,j}), \qquad 1 \leq j \leq m, \tag{4.2}$$

be satisfied. If the step is defined by $s_k = x_{k,m+1} - x_k$, then (2.5) and (3.6) are satisfied. Also note that there is no loss in generality in fixing $m$ independent of the iteration; this only imposes an upper bound on the number of minor iterates because we can set $x_{k,j+1} = x_{k,j}$.

We can compute minor iterates that satisfy (4.1) and (4.2) by computing a descent direction for the subproblem

$$\min \left\{ q_k(x_{k,j} + w) : \langle c_i, w \rangle = 0, \; i \in \mathcal{A}(x_{k,j}) \right\}. \tag{4.3}$$

Given a descent direction direction $w_{k,j}$ with $\langle c_i, w_{k,j} \rangle = 0$ for $i \in \mathcal{A}(x_{k,j})$, we examine $q_k$ in the ray $x_{k,j} + \beta w_{k,j}$, with $\beta \geq 0$, and use a line search to choose $\beta_{k,j}$ so that $q_k$ is minimized. The minor iterate $x_{k,j+1} = x_{k,j} + \beta_{k,j} w_{k,j}$ may not be acceptable either because $x_{k,j+1}$ is not feasible or because $x_{k,j+1}$ does not satisfy the trust region constraint $\|x_{k,j+1} - x_k\| \leq \Delta_k$. Thus, if necessary, we modify $\beta_{k,j}$ so that both constraints are satisfied.

Instead of using a line search to determine $x_{k,j+1}$ we can use a projected search along the path defined by $P[x_{k,j} + \beta w_{k,j}]$. The advantage of this approach is that we would be able to add several constraints at once. For a line search we normally require a decrease of $q_k$ on the line segment $[x_{k,j}, x_{k,j+1}]$, but for a projected search we need only require a decrease at $x_{k,j+1}$ with respect to the base point $x_{k,j}$. We require that

$$q_k(x_{k,j+1}) \leq q_k(x_{k,j}) + \mu_0 \min \left\{ \langle \nabla q_k(x_{k,j}), x_{k,j+1} - x_{k,j} \rangle, 0 \right\}, \qquad 1 \leq j \leq m. \tag{4.4}$$

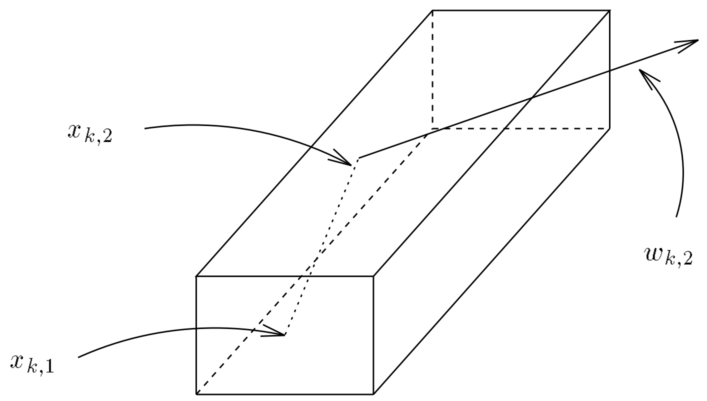


Figure 4.1: The minor iterates for a projected search.

In most cases we require only (4.2), but for rate of convergence results we need (4.4). For additional details on projected searches, see Moré and Toraldo [30, Section 4].

Figure 4.1 illustrates the projected search when $\Omega$ is the bound-constrained set (1.2). In this figure the iterate $x_{k,2}$ has been computed and the direction $w_{k,2}$ is determined that is orthogonal to the active constraint normals. If a line search is used, the search would be restricted to points in the ray $x_{k,2} + \beta w_{k,2}$ that lie in the feasible region. With a projected search, the search would continue along the piecewise linear path $P[x_{k,2} + \beta w_{k,2}]$. In either case, we require only that $x_{k,3}$ satisfy the decrease condition (4.4).

When $\Omega$ is the bound-constrained set (1.2), Lescrenier [25] determines the step $s_k$ by computing minor iterates, but he requires that the line segment $\alpha x_{k,j+1} + (1-\alpha)x_{k,j}$ be feasible for $\alpha \in [0,1]$ and that

$$q_k(x_{k,j+1}) \le q_k(\alpha x_{k,j+1} + (1-\alpha)x_{k,j}), \qquad \alpha \in [0,1]. \tag{4.5}$$

This requirement can be satisfied if a line search is used to choose the minor iterates, but it rules out the projected searches that we have proposed. Also note that assumption (4.5) on the minor iterates is stronger than (4.2). This observation can be verified by proving that if $\phi : \mathbb{R} \mapsto \mathbb{R}$ is a quadratic on $[0,1]$ with $\phi'(0) < 0$, and $\phi(1) \le \phi(\alpha)$ for $\alpha$ in $[0,1]$, then

$$\phi(1) \le \phi(0) + \tfrac{1}{2}\phi'(0) \le \phi(0) + \mu\phi'(0),$$

for any $\mu \in [0, \frac{1}{2}]$.

## 5 Convergence Results

We have been analyzing the trust region method under the assumption that $\{B_k\}$ is uniformly bounded. We now consider a trust region version of Newton's method so that $B_k$ is the Hessian matrix $\nabla^2 f(x_k)$. The assumption that $\{B_k\}$ is uniformly bounded is then satisfied if $\Omega$ is bounded or, more generally, if $\nabla^2 f$ is bounded on the level set

$$\mathcal{L}(x_0) \equiv \{x \in \Omega : f(x) \le f(x_0)\}.$$

We also assume that $\Omega$ is the polyhedral set (1.3).

The local convergence analysis for the trust region version of Newton's method requires that we assume that some subsequence of the iterates $\{x_k\}$ generated by the trust region method converges to a stationary point $x^*$ that satisfies a regularity condition. We assume that the Hessian matrix $\nabla^2 f(x^*)$ is positive definite on the subspace

$$S(x^*) = \text{aff}\{E[-\nabla f(x^*)] - x^*\}, \tag{5.1}$$

where aff$\{S\}$ denotes the affine hull of the set $S$. Thus, we require that the Hessian matrix be positive definite on the smallest subspace that contains $E[-\nabla f(x^*)] - x^*$. In the convergence analysis we use this regularity condition in the equivalent form

$$\left\langle v, \nabla^2 f(x^*) v \right\rangle \ge \kappa \|v\|^2, \qquad v \in S(x^*), \qquad \kappa > 0. \tag{5.2}$$

The *strong second-order sufficiency condition* (5.2) is equivalent to the standard second-order sufficiency condition if $x^*$ is nondegenerate, but is stronger than the standard second-order sufficiency condition for degenerate problems.

The strong second-order condition (5.2) is satisfied if $\nabla^2 f(x^*)$ is positive definite on the subspace

$$\{v \in \mathbb{R}^n : \langle c_j, v \rangle = 0, \; j \in \mathcal{B}(x^*)\}, \tag{5.3}$$

where $\mathcal{B}(x^*)$ is the set of *strictly binding* constraints

$$\mathcal{B}(x^*) = \Big\{ i \in \mathcal{I} : \lambda_i^* > 0 \text{ if } \langle c_i, x^* \rangle = l_i \text{ and } \lambda_i^* < 0 \text{ if } \langle c_i, x^* \rangle = u_i \Big\}.$$

Gay [23], Lescrenier [25], and Robinson [32] use this condition in their work. A disadvantage of working with (5.3) is that $\mathcal{B}(x^*)$ depends on the representation of $\Omega$ and the choice of multipliers. On the other hand, (5.2) depends only on the geometry of $\Omega$.

Burke and Moré [6] provide additional information on the regularity condition (5.2). In particular, they present an example where (5.2) holds but the Hessian matrix is not positive definite on (5.3).

The strong second-order sufficiency condition simplifies considerably when $\Omega$ is the bound-constrained set (1.2). In this case (5.2) requires that $\nabla^2 f(x^*)$ be positive definite on the subspace

$$S(x^*) = \{w \in \mathbb{R}^n : w_i = 0, \; i \in \mathcal{B}(x^*)\},$$

of vectors orthogonal to the strictly binding constraints

$$\mathcal{B}(x^*) = \{i \in \mathcal{A}(x^*) : \partial_i f(x^*) \ne 0\}.$$

**Theorem 5.1** *Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be twice continuously differentiable on the polyhedral set $\Omega$, and let $\{x_k\}$ be the sequence generated by the trust region Newton method. Assume that $\nabla^2 f$ is bounded on the level set $\mathcal{L}(x_0)$ and that the step $s_k$ satisfies* (3.6). *If $\{x_k\}$ has a limit point $x^*$ that satisfies the strong second-order sufficiency condition* (5.2), *then $\{x_k\}$ converges to $x^*$.*

**Proof.** We first claim that (5.2) implies that $x^*$ is an isolated limit point of $\{x_k\}$. This claim follows by noting that (5.2) implies that $x^*$ is an isolated stationary point, and that every limit point of $\{x_k\}$ is stationary.

The proof is by contradiction. If we assume that $\{x_k\}$ does not converge to $x^*$, then Lemma 4.10 of Moré and Sorensen [29] shows that when $x^*$ is an isolated limit point of $\{x_k\}$, there is a subsequence $\mathcal{K}$ such that $\{x_k\}$ converges to $x^*$ for $k \in \mathcal{K}$, and an $\epsilon > 0$ with

$$\|x_{k+1} - x_k\| \geq \epsilon, \qquad k \in \mathcal{K}.$$

In particular, $\|s_k\| \geq \epsilon$ for $k \in \mathcal{K}$. We now prove that if the sequence $\{w_k\}$ is defined by

$$w_k = \frac{s_k}{\|s_k\|}, \qquad k \in \mathcal{K},$$

then any limit point $w^*$ is a feasible direction at $x^*$. Note that $\|s_k\| \geq \epsilon$ implies that $x_k + \tau w_k$ belongs to $\Omega$ for $\tau$ in $[0, \epsilon]$, and hence $x^* + \tau w^*$ also belongs to $\Omega$. This shows that $w^*$ is a feasible direction at $x^*$.

We now show that $\langle \nabla f(x^*), w^* \rangle = 0$. Note that requirements (2.4), (2.5), and (2.6) on $s_k$ show that if the iteration is successful, then

$$f(x_k) - f(x_{k+1}) \geq \eta_0 \mu_0 \kappa_0 \left[ \frac{\|x_k^C - x_k\|}{\alpha_k} \right] \min \left\{ \Delta_k, \frac{1}{\|\nabla^2 f(x_k)\|} \left[ \frac{\|x_k^C - x_k\|}{\alpha_k} \right] \right\}. \tag{5.4}$$

Our assumptions guarantee that the Hessian matrices $\nabla^2 f(x_k)$ are bounded, and since $\|s_k\| \leq \mu_1 \Delta_k$, and $\|s_k\| \geq \epsilon$ for $k \in \mathcal{K}$, the trust region bounds $\Delta_k$ are bounded away from zero. Hence, inequality (5.4) implies that

$$\lim_{k \in \mathcal{K}, k \to \infty} \frac{\|x_k^C - x_k\|}{\alpha_k} = 0.$$

Moreover, since $\{\alpha_k\}$ is bounded above, $\{\|x_k^C - x_k\|\}$ also converges to zero for $k \in \mathcal{K}$. Hence, Lemma 5.1 in Burke, Moré, and Toraldo [7] implies that

$$\lim_{k \in \mathcal{K}, k \to \infty} \left\| \nabla_\Omega f(x_k^C) \right\| = 0.$$

Theorem 3.1 now shows that $x_k^C$ is in $E[-\nabla f(x^*)]$ for $k \in \mathcal{K}$, and thus assumption (3.6) on the step $s_k$ implies that $x_k + s_k$ belongs to $E[-\nabla f(x^*)]$ for $k \in \mathcal{K}$. In particular,

$$\langle \nabla f(x^*), (x_k + s_k - x^*) \rangle = 0, \qquad k \in \mathcal{K}.$$

A computation using $\|s_k\| \geq \epsilon$ now shows that $\langle \nabla f(x^*), w^* \rangle = 0$.

We have shown that $w^*$ is a feasible direction at $x^*$ with $\langle \nabla f(x^*), w^* \rangle = 0$. Thus, $w^*$ belongs to $S(x^*)$, and $\langle w^*, \nabla^2 f(x^*) w^* \rangle > 0$. On the other hand, $\psi_k(s_k) \leq 0$ implies that

$$\tfrac{1}{2}\|s_k\| \left\langle w_k, \nabla^2 f(x_k) w_k \right\rangle \leq - \langle \nabla f(x_k), w_k \rangle .$$

Since $\{x_k\}$ converges to $x^*$, $\{w_k\}$ converges to $w^*$, and $\|s_k\| \geq \epsilon$ for $k \in \mathcal{K}$, this inequality implies that

$$0 < \tfrac{1}{2}\epsilon \left\langle w^*, \nabla^2 f(x^*) w^* \right\rangle \leq - \langle \nabla f(x^*), w^* \rangle = 0.$$

This contradiction proves the result. ■

Theorems 5.1 improves on previous convergence results for linearly-constrained optimization algorithms because it does not assume strict complementarity. For recent convergence results, see [19, 12, 18, 9, 16, 20, 33].

Rate of convergence results depend on showing that eventually the trust region bound is not active. These results require additional assumptions on the step $s_k$. We assume that the minor iterates satisfy (4.1) and the decrease condition (4.4). We now estimate the decrease of the quadratic $q_k$ if the minor iterates satisfy (4.4). The following result appears in Moré [28], but for completeness we provide the proof.

**Lemma 5.2** *Assume that $\phi : \mathbb{R} \mapsto \mathbb{R}$ is twice differentiable on $[0, 1]$ and that $\phi''(\alpha) \geq \varepsilon$ on $[0, 1]$ for some $\varepsilon > 0$. If*

$$\phi(1) \leq \phi(0) + \mu \phi'(0) \tag{5.5}$$

*for some $\mu \in (0, 1)$, then*

$$\phi(0) - \phi(1) \geq \frac{\mu}{2(1-\mu)} \varepsilon.$$

**Proof.** The mean value theorem shows that

$$\phi(1) = \phi(0) + \phi'(0) + \tfrac{1}{2}\phi''(\theta)$$

for some $\theta \in (0, 1)$, and thus (5.5) implies that $\frac{1}{2}\phi''(\theta) \leq (1-\mu)(-\phi'(0))$. Hence,

$$\phi(0) - \phi(1) \geq \mu(-\phi'(0)) \geq \frac{\mu}{2(1-\mu)} \phi''(\theta) \geq \frac{\mu}{2(1-\mu)} \varepsilon,$$

as desired. ■

If we assume that the sequence $\{x_k\}$ converges to $x^*$, then Theorem 3.3 guarantees that all iterates belong to $E[-\nabla f(x^*)]$, and hence (4.1) shows that all the minor iterates also belong to $E[-\nabla f(x^*)]$. Now define

$$\phi(\alpha) = q_k \left( \alpha x_{k,j+1} + (1-\alpha) x_{k,j} \right),$$

and note that the decrease condition (4.4) guarantees that

$$q_k(x_{k,j+1}) \le q_k(x_{k,j}) + \mu_0 \langle \nabla q_k(x_{k,j}), x_{k,j+1} - x_{k,j} \rangle,$$

and thus (5.5) holds. Hence, if we assume that the strong second-order condition (5.2) holds, then Lemma 5.2 implies that there is a $\kappa_0 > 0$ such that

$$q_k(x_{k,j}) - q_k(x_{k,j+1}) \ge \kappa_0 \|x_{k,j+1} - x_{k,j}\|^2. \tag{5.6}$$

We need this estimate for our next result.

**Theorem 5.3** *Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be twice continuously differentiable on the polyhedral set $\Omega$, and let $\{x_k\}$ be the sequence generated by the trust region Newton method. Assume that $\{x_k\}$ converges to a solution $x^*$ of* (1.1) *that satisfies the regularity condition* (5.2)*. If the minor iterates satisfy* (4.1) *and* (4.4)*, then there is an index $k_0$ such that all steps $s_k$ with $k \ge k_0$ are successful and the trust region bound $\Delta_k$ is bounded away from zero.*

**Proof.** In the proof we bound $|\rho_k - 1|$, where $\rho_k$ is defined by (2.1), and show that the bounds converge to zero; the rules for updating $\Delta_k$ then show that all steps $s_k$ are ultimately successful, and that $\Delta_k$ is bounded away from zero. We begin by noting that

$$\rho_k - 1 = \frac{f(x_k + s_k) - f(x_k) - \psi_k(s_k)}{\psi_k(s_k)}. \tag{5.7}$$

The denominator of (5.7) is estimated by noting that (5.6) implies that the decrease generated by $s_k$ satisfies

$$-\psi_k(s_k) = q_k(x_k) - q_k(x_k + s_k) \ge \kappa_0 \sum_{j=0}^{m} \|x_{k,j+1} - x_{k,j}\|^2 \ge \kappa_0 \max_{0 \le j \le m} \left\{ \|x_{k,j+1} - x_{k,j}\|^2 \right\}.$$

On the other hand,

$$\|s_k\| \le \sum_{j=0}^{m} \|x_{k,j+1} - x_{k,j}\| \le (m+1) \max_{0 \le j \le m} \left\{ \|x_{k,j+1} - x_{k,j}\| \right\}.$$

Hence, $-\psi_k(s_k) \ge \kappa_1 \|s_k\|^2$ for $\kappa_1 = \kappa_0/(m+1)^2$. We estimate the numerator of (5.7) by noting that the mean value theorem implies that

$$|f(x_k + s_k) - f(x_k) - \psi_k(s_k)| \le \sigma_k \|s_k\|^2, \qquad \sigma_k = \sup_{0 \le \theta \le 1} \left\{ \|\nabla^2 f(x_k + \theta s_k) - \nabla^2 f(x_k)\| \right\}.$$

These estimates show that $|\rho_k - 1| \le \sigma_k / \kappa_0$, so that the our result will be established if we show that $\{\sigma_k\}$ converges to zero.

Since $\{x_k\}$ converges to $x^*$, the sequence $\{\sigma_k\}$ converges to zero if $\{s_k\}$ converges to zero. Theorem 3.3 shows that $x_k$ and $x_k + s_k$ belong to $E[-\nabla f(x^*)]$, and thus the definition

(5.1) implies that $s_k \in S(x^*)$. In particular, $s_k = P_{S(x^*)} s_k$, where $P_{S(x^*)}$ is the orthogonal projection onto $S(x^*)$. Since $\psi_k(s_k) \le 0$,

$$\tfrac{1}{2} \left\langle s_k, \nabla^2 f(x_k) s_k \right\rangle \le - \langle \nabla f(x_k), s_k \rangle ,$$

and thus $s_k = P_{S(x^*)} s_k$ and the regularity condition (5.2) imply that there is a $\nu_0 > 0$ with

$$\|s_k\| \le \nu_0 \| P_{S(x^*)} \nabla f(x_k) \|.$$

The gradient $\nabla f(x^*)$ is orthogonal to $S(x^*)$ because $\langle \nabla f(x^*), x \rangle = \langle \nabla f(x^*), x^* \rangle$ whenever $x$ is in $E[-\nabla f(x^*)]$, and since $\{x_k\}$ converges to $x^*$, this implies that $\{P_{S(x^*)} \nabla f(x_k)\}$ converges to zero. Thus, the previous estimate shows that $\{s_k\}$ converges to zero, as desired. ■

Lescrenier [25] proved an analogous result, but he assumed that the feasible set was bound-constrained, that the quadratic was decreasing on the line segment $[x_{k,j}, x_{k,j+1}]$, and that the minor iterates satisfied (4.5). In particular, his result did not cover projected searches. Our assumptions in Theorem 5.3 are considerably weaker.

When the iterate $x_k$ is far away from the solution, the step $s_k$ is usually determined because the trust region bound $\|x_{k,j} - x_k\| \le \mu_1 \Delta_k$ is encountered during the computation of $x_{k,j+1}$. However, as we converge, Theorem 5.3 shows that the trust region does not interfere with the computation of the step, so that we are free to reduce $q_k$ further by searching the feasible set.

We propose to compute the step $s_k$ by computing minor iterates $x_{k,j}$ that satisfy (4.1) and the decrease condition (4.4). For each minor iterate $x_{k,j}$ let the columns of $Z_{k,j}$ form an orthonormal basis for the subspace

$$V_{k,j} = \{ w \in \mathbb{R}^n : \langle c_i, w \rangle = 0, \; i \in \mathcal{A}(x_{k,j}) \} .$$

Given $x_{k,j}$, we find an approximate minimizer of $q_k$ on $x_{k,j} + V_{k,j}$. We require that if $x_{k,m+1}$ is the final iterate generated according to (4.1) and (4.4), then the step $s_k = x_{k,m+1} - x_k$ satisfies

$$\left\| Z_{k,m}^T [\nabla f(x_k) + \nabla^2 f(x_k) s_k] \right\| \le \xi_k \left\| Z_{k,m}^T \nabla f(x_k) \right\| , \qquad x_k + s_k \in \Omega. \tag{5.8}$$

We motivate these requirements by noting that if $\Psi_{k,m}(v) = q_k \left( x_{k,m} + Z_{k,m} v \right)$, then

$$\nabla \Psi_{k,m}(v) = Z_{k,m}^T [\nabla f(x_k) + \nabla^2 f(x_k)(x_{k,m} - x_k + Z_{k,m} v)],$$

where we have set $x_{k,0} = x_k$. Thus, the first condition in (5.8) is equivalent to finding $v_{k,m}$ such that

$$\| \nabla \Psi_{k,m}(v_{k,m}) \| \le \xi_k \left\| Z_{k,m}^T \nabla f(x_k) \right\| ,$$

and setting $s_k = x_{k,m} - x_k + Z_{k,m}v_{k,m}$. In particular, $x_{k,m+1} = x_{k,m} + Z_{k,m}v_{k,m}$ is a minimizer of $q_k$ on $x_{k,m} + V_{k,m}$ if we choose $\xi_k = 0$.

At first sight it is not clear that we can always find a step that satisfies (5.8) since satisfying the first condition in (5.8) may violate the second condition. The simplest method of generating minor iterates $x_{k,j}$ that guarantees (5.8) is to set $x_{k,j+1}$ to the minimizer of $q_k$ on $x_{k,j} + V_{k,j}$. With this choice $s_k = x_{k,j+1} - x_k$ satisfies the first condition in (5.8). If $x_k + s_k$ lies in $\Omega$ for this choice of $x_{k,j+1}$, then we are done. Otherwise, we can set $x_{k,j+1}$ to any point in $\Omega$ that satisfies (4.4) and such that $\mathcal{A}(x_{k,j+1})$ has at least one more active variable. This choice guarantees that, after computing at most $n$ minor iterates, we reach a minor iterate with all variables active, and then (5.8) is trivially satisfied.

The procedure that we have outlined generates iterates $x_{k,j}$ that satisfy (4.1) and (4.4) with $\mathcal{A}(x_{k,j}) \subset \mathcal{A}(x_{k,j+1})$. The step $s_k = x_{k,m+1} - x_k$ satisfies (5.8), where $Z_{k,m}$ is defined by $x_{k,m}$. Geometrically this procedure searches for an approximate minimizer in the face defined by the active set $\mathcal{A}(x_{k,j})$, terminating if the approximate minimizer is on the relative interior of this face; otherwise, the search continues on the lower dimensional face defined by $\mathcal{A}(x_{k,j+1})$.

We have already noted that the step $s_k$ is usually determined because the trust region bound $\|x_{k,j} - x_k\| \le \mu_1 \Delta_k$ is encountered during the computation of $x_{k,j+1}$. Thus, we need only assume that the step $s_k$ satisfies (5.8) if $\|s_k\| \le \mu_* \Delta_k$ for some $\mu_* < \mu_1$.

Rate of convergence results when strict complementarity holds depend on the result that $\mathcal{A}(x_k) = \mathcal{A}(x^*)$ for all $k$ sufficiently large. This result fails without strict complementarity. In this case the proof relies on showing that

$$V(x) \equiv \Big\{ w \in \mathbb{R}^n : \langle c_i, w \rangle = 0,\ i \in \mathcal{A}(x) \Big\} \subset S(x^*), \qquad x \in E[-\nabla f(x^*)]. \tag{5.9}$$

The subspace $V(x)$ is the largest subspace contained in the tangent cone $T(x)$.

For the rate of convergence results we assume that the sequence $\{x_k\}$ generated by the trust region Newton method converges to $x^*$. Theorems 3.2 and 3.3 show that $x_k$ and $x_k^C$ eventually land in $E[-\nabla f(x^*)]$ for all $k \ge k_0$. Since (4.1) guarantees that $\mathcal{A}(x_k^C)$ is a subset of $\mathcal{A}(x_{k,j})$ for any minor iterate $x_{k,j}$, we also have $x_{k,j} \in E[-\nabla f(x^*)]$. In particular, $x_{k,m} \in E[-\nabla f(x^*)]$. We shall need this result in the proof.

**Theorem 5.4** *Let $f : \mathbb{R}^n \mapsto \mathbb{R}$ be twice continuously differentiable on the polyhedral set $\Omega$, and let $\{x_k\}$ be the sequence generated by the trust region Newton method. Assume that $\{x_k\}$ converges to a solution $x^*$ of* (1.1) *that satisfies the strong second-order sufficiency condition* (5.2)*. If the step $s_k$ is calculated by the algorithm outlined above, and* (5.8) *holds whenever $\|s_k\| \le \mu_* \Delta_k$ for some $\mu_* < \mu_1$, then $\{x_k\}$ converges $Q$-linearly to $x^*$ when $\xi^*$ is sufficiently small, where*

$$\xi^* = \limsup_{k \to +\infty} \xi_k.$$

*The rate of convergence is Q-superlinear when $\xi^* = 0$.*

**Proof.** We first prove that (5.9) holds. The proof begins by noting that expression (3.1) for $E[-\nabla f(x^*)]$ shows that if $\lambda_i^*$ are Lagrange multipliers, then

$$\{i : \lambda_i^* \neq 0\} \subset \mathcal{A}(x), \qquad x \in E[-\nabla f(x^*)].$$

Hence, if $w \in V(x)$ then $\langle \nabla f(x^*), w \rangle = 0$. Since any $w \in V(x)$ is a feasible direction, we also have that $x + \alpha w$ for all $\alpha$ sufficiently small. Hence, $\langle \nabla f(x^*), w \rangle = 0$ implies that $x + \alpha w$ belongs to $E[-\nabla f(x^*)]$. Moreover, since $x \in E[-\nabla f(x^*)]$ and $S(x^*)$ is a subspace,

$$\alpha w = \Big( [x + \alpha w - x^*] - [x - x^*] \Big) \in S(x^*).$$

Hence, $w \in S(x^*)$ as desired, and thus (5.9) holds.

We proved (5.9) for any $x \in E[-\nabla f(x^*)]$ because this result sheds light on the geometry behind the rate of convergence results but for this proof we only need to show that

$$V_{k,m} \subset S(x^*). \tag{5.10}$$

Since have already noted that $x_{k,m} \in E[-\nabla f(x^*)]$, (5.9) implies that (5.10) holds.

We analyze the convergence rate in terms of the projection $P_k = Z_{k,m} Z_{k,m}^T$ onto the subspace $V_{k,m}$. Note, in particular, that since $V_{k,m}$ is a subspace and $V_{k,m} \subset S(x^*)$, an orthogonal basis for $V_{k,m}$ can be extended to a basis for $S(x^*)$, and thus

$$\|P_k w\| \leq \|P_{S(x^*)} w\|, \qquad w \in \mathbb{R}^n. \tag{5.11}$$

The main estimate needed for the rate of convergence result is obtained by noting that

$$\|P_k \nabla f(x_{k+1})\| \leq \left\| P_k[\nabla f(x_{k+1}) - \nabla f(x_k) - \nabla^2 f(x_k) s_k] \right\| + \left\| P_k[\nabla f(x_k) + \nabla^2 f(x_k) s_k] \right\|,$$

assumption (5.8) on the step, and standard bounds yield that

$$\|P_k \nabla f(x_{k+1})\| \leq \varepsilon_k \|s_k\| + \xi_k \left\| P_k \nabla f(x_k) \right\|, \tag{5.12}$$

for some sequence $\{\varepsilon_k\}$ converging to zero. Also note that the argument at the end of Theorem 5.3 shows that there is a constant $\nu_0$ with

$$\|s_k\| \leq \nu_0 \|P_{S(x^*)} \nabla f(x_k)\|. \tag{5.13}$$

If we make use of this estimate and (5.11) in (5.12) we obtain that

$$\limsup_{k \to +\infty} \frac{\|P_k \nabla f(x_{k+1})\|}{\|P_{S(x^*)} \nabla f(x_k)\|} \leq \limsup_{k \to +\infty} \xi_k. \tag{5.14}$$

We complete the proof by estimating $\|P_k \nabla f(x_{k+1})\|$ and $\|P_{S(x^*)} \nabla f(x_k)\|$. We first show that

$$\|P_k \nabla f(x_{k+1})\| \geq (\nu_1 - \varepsilon_k)\|x_{k+1} - x^*\| \tag{5.15}$$

for some sequence $\{\varepsilon_k\}$ converging to zero.

The proof of (5.15) requires some preliminary results. We first show that $x_{k+1} - x^*$ is in $V_{k,m}$ for all $k$ sufficiently large. This follows from the definition of $V_{k,m}$ because $\mathcal{A}(x_{k,m}) \subset \mathcal{A}(x_{k+1})$ and $\mathcal{A}(x_{k,m}) \subset \mathcal{A}(x^*)$. We also need to show that $P_k \nabla f(x^*) = 0$. This result follows because, as noted at the end of Theorem 5.3, $\nabla f(x^*)$ is orthogonal to $S(x^*)$, and since $V_{k,m}$ is a subspace of $S(x^*)$, we must also have $\nabla f(x^*)$ orthogonal to $V_{k,m}$. In particular, $P_k \nabla f(x^*) = 0$. The last result that we need for the proof of (5.15) is that

$$\|P_k \nabla^2 f(x^*) P_k v\| \geq \kappa \|v\|, \qquad v \in V_{k,m}. \tag{5.16}$$

To prove this result, note that if $v \in V_{k,m}$ then $P_k v = v$, and in view of (5.10), $P_k v$ is in $E[-\nabla f(x^*)]$. Hence, the regularity assumption (5.2) shows that (5.16) holds.

We now have all the ingredients to prove (5.15). Since $P_k \nabla f(x^*) = 0$,

$$P_k \nabla f(x_{k+1}) = P_k \nabla^2 f(x^*)(x_{k+1} - x^*) + P_k[\nabla f(x_{k+1}) - \nabla f(x^*) - \nabla^2 f(x^*)(x_{k+1} - x^*)],$$

and thus estimates of the last term to show that

$$\left\|P_k \nabla^2 f(x^*)(x_{k+1} - x^*)\right\| \leq \|P_k \nabla f(x_{k+1})\| + \varepsilon_k \|x_{k+1} - x^*\|,$$

where $\{\varepsilon_k\}$ converges to zero. Since $x_{k+1} - x^*$ is in $V_{k,m}$ for all $k$ sufficiently large, (5.16) shows that

$$\|P_k \nabla^2 f(x^*) P_k (x_{k+1} - x^*)\| \geq \kappa \|x_{k+1} - x^*\|.$$

The last two inequalities show that (5.15) holds with $\nu_1 = \kappa$.

We estimate $\|P_{S(x^*)} \nabla f(x_k)\|$ by proving that

$$\|P_{S(x^*)} \nabla f(x_k)\| \leq (\nu_2 + \varepsilon_k)\|x_k - x^*\| \tag{5.17}$$

for some sequence $\{\varepsilon_k\}$ converging to zero. Since $P_{S(x^*)} \nabla f(x^*) = 0$,

$$P_{S(x^*)} \nabla f(x_k) = P_{S(x^*)} \nabla^2 f(x^*)(x_k - x^*) + P_{S(x^*)}[\nabla f(x_k) - \nabla f(x^*) - \nabla^2 f(x^*)(x_k - x^*)],$$

and thus standard estimates of the last term show that

$$\|P_{S(x^*)} \nabla f(x_k)\| \leq \left\|P_{S(x^*)} \nabla^2 f(x^*)(x_k - x^*)\right\| + \varepsilon_k \|x_k - x^*\|,$$

where $\{\varepsilon_k\}$ converges to zero. Since $P_{S(x^*)}(x_k - x^*) = x_k - x^*$, we obtain that

$$\|P_{S(x^*)} \nabla f(x_k)\| \leq \nu_2 \|x_k - x^*\| + \varepsilon_k \|x_k - x^*\|, \qquad \nu_2 = \|P_{S(x^*)} \nabla^2 f(x^*) P_{S(x^*)}\|,$$

where $\{\varepsilon_k\}$ converges to zero. This proves (5.17).

Linear and superlinear convergence rates are obtained by noting that (5.14), together with estimates (5.15) and (5.17), show that

$$\limsup_{k\to+\infty} \frac{\|x_{k+1}-x^*\|}{\|x_k-x^*\|} \le \left(\frac{\nu_2}{\nu_1}\right) \limsup_{k\to+\infty} \xi_k = \left(\frac{\nu_2}{\nu_1}\right)\xi^*.$$

Linear convergence takes place if $\nu_2\xi^* < \nu_1$, and superlinear convergence holds if $\xi^* = 0$. ■

A modification of the proof of Theorem 5.4 shows linear convergence for any $\xi^* < 1$ if the vectors $x_k - x^*$ lie in a fixed subspace $V$ of $S(x^*)$ for all $k$ sufficiently large. This result holds when $x^*$ is non-degenerate (strict complementarity holds at $x^*$) since in this case $x_k - x^*$ belongs to $V(x_k) = S(x^*)$ for all $k$ sufficiently large.

There are several interesting variations on Theorem 5.4. Note, in particular, that the minor iterate $x_{k,m}$ enters into the proof via the subspace $V_{k,m}$ and that the proof holds if $P_k$ is a projection into any subspace of $S(x^*)$ that contains $x_{k+1} - x^*$. Thus we could have set $P_k$ to the projection into $V(x_{k+1})$ and eliminated $x_{k,m}$ from the analysis. We did not make this simplification because with our choice of $P_k$ the minor iterate $x_{k,m+1}$ is an approximate minimizer of $q_k$ on $x_{k,m} + V_{k,m}$.

Lescrenier [25] and Facchinei and Lucidi [19] proved rate of convergence results without assuming strict complementarity, but the analysis was restricted to bound-constrained problems. Other convergence results for bound-constrained and linearly-constrained optimization algorithms require strict complementarity. For recent convergence results, see [12, 18, 9, 16, 20, 33].

We can also show that quadratic convergence holds in Theorem 5.4 if we assume that $\nabla^2 f$ satisfies a Lipschitz condition at $x^*$, and if

$$\xi_k \le \kappa_0 \|P_k \nabla f(x_k)\|, \qquad k \ge 0,$$

for a positive constant $\kappa_0$. With these assumptions we can follow the proof of Theorem 5.4. The main difference is that the inequality (5.12) can be replaced by

$$\|P_k \nabla f(x_{k+1})\| \le \kappa \|s_k\|^2 + \xi_k \|P_k \nabla f(x_k)\|,$$

where $\kappa$ is the Lipschitz constant, and thus (5.11) and (5.13) yield that

$$\limsup_{k\to+\infty} \frac{\|P_k \nabla f(x_{k+1})\|}{\|P_{S(x^*)} \nabla f(x_k)\|^2} \le \kappa\nu_0^2 + \kappa_0.$$

The result now follows from estimates (5.15) and (5.17).

## 6 Implementation Issues

We now provide a brief outline of the implementation issues for a trust region Newton method for bound-constrained problems. We concentrate on discussing our choices for the trust region bound $\Delta_k$, the Cauchy step, and the subspace step.

For the initial $\Delta_0$ we used $\|\nabla f(x_0)\|$. This choice is appropriate in many cases, but more sophisticated choices are possible. We update the trust region bound $\Delta_k$ as outlined in Section 2. We choose $\eta_0 = 10^{-3}$ in the algorithm (2.2) to update the current iterate, $\eta_1 = 0.25$, $\eta_2 = 0.75$ as the constants that determine when to increase or decrease the trust region $\Delta_k$, and $\sigma_1 = 0.25$, $\sigma_2 = 0.5$, and $\sigma_3 = 4.0$ as the constants that govern the update of $\Delta_k$ in (2.3).

Given a step $s_k$, we attempt to choose $\Delta_{k+1}$ as $\alpha_k^* \|s_k\|$, where $\alpha_k^*$ is the minimum of a quadratic that interpolates the function $\alpha \mapsto f(x_k + \alpha s_k)$. In other words, we consider the quadratic $\phi$ such that

$$\phi(0) = f(x_k), \qquad \phi'(0) = \langle \nabla f(x_k), s_k \rangle, \qquad \phi(1) = f(x_{k+1}),$$

and determine $\alpha_k^*$ as the minimum of this quadratic. If $\phi$ does not have a minimum, we set $\alpha_k^* = +\infty$. We choose $\Delta_{k+1}$ as $\alpha_k^* \|s_k\|$ if it falls in the desired interval; otherwise we set $\Delta_{k+1}$ to the closest endpoint.

The Cauchy step $s_k^C$ is chosen by an iterative scheme that is guaranteed to terminate in a finite number of steps. Recall that the Cauchy step $s_k^C$ is of the form $s_k(\alpha_k)$, where the function $s_k : \mathbb{R} \mapsto \mathbb{R}^n$ is defined by

$$s_k(\alpha) = P[x_k - \alpha \nabla f(x_k)] - x_k,$$

and $\alpha_k$ satisfies the conditions specified in Section 2. The simplest scheme is to set $\alpha_k^{(0)}$ to a constant and then generate a sequence $\{\alpha_k^{(l)}\}$ of trial values by decreasing the trial values by a constant factor until the sufficient decrease condition (2.4) is satisfied. We use a more sophisticated scheme. Given $\alpha_k^{(0)}$, we generate a sequence $\{\alpha_k^{(l)}\}$ of trial values. The sequence can be either increasing or decreasing, but in all cases we require that

$$\alpha_k^{(l+1)} \in [\beta_1 \alpha_k^{(l)}, \beta_2 \alpha_k^{(l)}],$$

where $\beta_1 \le \beta_2 < 1$ for a decreasing sequence, and $1 < \beta_1 \le \beta_2$ for an increasing sequence. The decision to generate an increasing sequence or a decreasing sequence depends of the initial $\alpha_k^{(0)}$. If the initial $\alpha_k^{(0)}$ fails to satisfy the sufficient decrease condition (2.4), we decrease the trial values until (2.4) fails, and set $\alpha_k$ to the last trial value that satisfies (2.4). If the initial $\alpha_k^{(0)}$ satisfies (2.4), we increase the trial values until (2.4) fails, and set $\alpha_k$ to the last trial value that satisfies (2.4).

We use $\alpha_k^{(0)} = 1$ on the first iteration, but on all other iterations we use $\alpha_{k-1}$. We use $\mu_0 = 10^{-2}$ and $\mu_1 = 1.0$ in the sufficient decrease condition (2.4).

The minor iterates generated in the trust region method are required to satisfy conditions (4.1) and (4.4). We generate the step between the minor iterates along the lines specified in Section 4, but specialized to the case of bound constraints. Specifically, we compute the step from the trust region subproblem

$$\min \left\{ q(x+w) : w_i = 0, \;\; i \in \mathcal{A}(x), \;\; \|Dw\| \le \Delta \right\},$$

where $D$ is a scaling matrix. If $i_1, \ldots, i_m$ are the indices of the free variables, and the matrix $Z$ is defined as the matrix in $\mathbb{R}^{n \times m}$ whose $k$th column is the $i_k$th column of the identity matrix in $\mathbb{R}^{n \times n}$, then this subproblem is equivalent to

$$\min \{ q_F(v) : \|DZv\| \le \Delta \},$$

where $q_F$ is the quadratic in the free variables defined by

$$q_F(v) \equiv q(x+Zv) - q(x) = \tfrac{1}{2} v^T A v + r^T v.$$

The matrix $A$ and the vector $r$ are, respectively, the reduced Hessian matrix of $q$ and reduced gradient of $q_F$ at $x$ with respect to the free variables.

Given a descent direction $w$ for this subproblem, a projected line search guarantees that we can determine $\beta > 0$ such that the next iterate $x_+ = P[x + \beta w]$ satisfies conditions (4.1) and (4.4). The conditions in (4.1) are satisfied for any $\beta > 0$ provided $D$ has a condition number that is bounded independent of the iterate. We use $\mu_0 = 10^{-2}$ in the sufficient decrease condition (4.4).

We generate the descent direction $w$ with a preconditioned conjugate gradient method as suggested by Steihaug [34]. The conjugate gradient iterates are generated until the trust region is violated, a negative curvature direction is generated, or the convergence condition (5.8) is satisfied. As noted in Section 5, this condition can be satisfied by choosing the minor iterates so that $\mathcal{A}(x_{k,j}) \subset \mathcal{A}(x_{k,j+1})$. For additional details, see the discussion in Lin and Moré [26].

In our algorithms we choose $D$ from an incomplete Cholesky factorization. From a theoretical viewpoint, the choice of $D$ is not important, but the numerical results are strongly dependent on the choice of $D$. We use the incomplete Cholesky factorization icfs of Lin and Moré [26]. The icfs incomplete Cholesky factorization does not require the choice of a drop tolerance. Moreover, the amount of storage used by the factorization can be specified in advance as $p \cdot n$, where $p$ is set by the user, and $n$ is the number of variables. In our numerical results we use $p = 5$.

## 7 Computational Experiments

We now compare the performance of an implementation TRON (version 1.0) of the trust region method outlined in Section 6 with the LANCELOT [14] and L-BFGS-B [36] codes. All computational experiments were done with the -O optimization compiler option on a Sun UltraSPARC2 workstation with 1024 MB RAM.

LANCELOT implements Newton's method with a trust region strategy but differs from TRON in significant issues. In particular, LANCELOT does not use projected searches, and the default is a banded preconditioner. The L-BFGS-B code is a limited-memory variable metric method. An advantage of L-BFGS-B is that only the gradient is required, while Newton codes require an approximation to the Hessian matrix. On the other hand, for sparse problems the Hessian matrix can usually be obtained efficiently with differences of gradients if the sparsity pattern of the Hessian matrix is provided.

Our first set of computational results uses a set of bound-constrained problems from the CUTE collection [3]. We used the select tool to choose problems representative of problems that arise in applications and where the number of variables $n$ could be changed. Since we are interested in large problems, we refined this selection by only considering problems where the number of variables was at least $5,000$. These requirements lead to a list of nine problems, with some of the problems having more than one version.

Table 7.1 presents the results of the first set of computational experiments. LANCELOT and L-BFGS-B were used with the default options. For LANCELOT, exact second derivatives and a preconditioned conjugate gradient method with a banded preconditioner were used; all other default options are shown in Table 5 of [15]. In Table 7.1 we used the LANCELOT termination test

$$\|P[x - \nabla f(x)] - x\|_\infty \le 10^{-5}, \tag{7.1}$$

where $P$ is the projection into the feasible set (1.2).

The first column in Table 7.1 is the name of the test problem, and the second column is the number of variables $n$. For TRON and LANCELOT we record the number of Hessian evaluations nh, function evaluations nf, and conjugate gradient iterations ncg. For L-BFGS-B we record only the number of function and gradient evaluations nfg because L-BFGS-B always evaluates the function and gradient at the same time. The execution time (in seconds) is reported in the time column. In these results, all three codes obtained the same optimal function value at the final iterate.

A general observation on the results in Table 7.1 is that the number of function evaluations for TRON and LANCELOT is at most one more than the number of Hessian evaluations. Thus, for these problems all the iterations of the Newton codes are successful. We conclude that these problems do not fully test TRON or LANCELOT.

In analyzing our computational results we do not discuss problems where L-BFGS-B requires less than 50 function and gradient evaluations. In general, we feel that if a limited-memory variable metric algorithm converges is less than 50 function and gradient evaluations on a problem with $10,000$ variables, then the starting point is exceptionally good.

An important observation on the results in Table 7.1 is that on these problems TRON

Table 7.1: Performance on the CUTE problems: Default options

| | | TRON | | | | LANCELOT | | | | L-BFGS-B | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Problem | $n$ | nh | nf | ncg | time | nh | nf | ncg | time | nfg | time |
| BDEXP | 5000 | 11 | 11 | 10 | 1.43 | 10 | 11 | 12 | 1.19 | 15 | 0.60 |
| CVXBQP1 | 10000 | 2 | 2 | 0 | 0.24 | 1 | 2 | 1 | 0.81 | 2 | 0.08 |
| JNLBRNG1 | 15625 | 26 | 26 | 33 | 15.22 | 24 | 25 | 2029 | 165.42 | 999 | 198.75 |
| JNLBRNG2 | 15625 | 16 | 16 | 27 | 9.21 | 14 | 15 | 898 | 74.16 | 577 | 105.18 |
| JNLBRNGA | 15625 | 23 | 23 | 29 | 12.46 | 21 | 22 | 1584 | 117.64 | 332 | 54.56 |
| JNLBRNGB | 15625 | 10 | 10 | 15 | 5.29 | 8 | 9 | 419 | 30.71 | 999 | 160.32 |
| MCCORMCK | 10000 | 6 | 7 | 6 | 1.46 | 4 | 5 | 4 | 1.10 | 15 | 1.76 |
| NCVXBQP1 | 10000 | 2 | 2 | 0 | 0.24 | 4 | 5 | 0 | 3.01 | 2 | 0.08 |
| NCVXBQP2 | 10000 | 10 | 10 | 10 | 1.44 | 6 | 7 | 84 | 3.35 | 178 | 6.85 |
| NCVXBQP3 | 10000 | 10 | 10 | 10 | 1.39 | 6 | 7 | 163 | 2.96 | 388 | 14.87 |
| NOBNDTOR | 14884 | 38 | 38 | 71 | 22.03 | 36 | 37 | 1386 | 123.66 | 213 | 36.38 |
| NONSCOMP | 10000 | 9 | 9 | 8 | 1.44 | 8 | 9 | 8 | 1.45 | 51 | 4.24 |
| OBSTCLAE | 15625 | 27 | 27 | 51 | 14.48 | 5 | 6 | 7452 | 821.46 | 660 | 116.18 |
| OBSTCLAL | 15625 | 25 | 25 | 39 | 12.64 | 24 | 25 | 604 | 43.64 | 156 | 24.51 |
| OBSTCLBL | 15625 | 20 | 20 | 42 | 12.81 | 18 | 19 | 2088 | 199.04 | 272 | 49.28 |
| OBSTCLBM | 15625 | 8 | 8 | 15 | 5.41 | 5 | 6 | 1378 | 152.87 | 146 | 25.90 |
| OBSTCLBU | 15625 | 21 | 21 | 33 | 11.85 | 19 | 20 | 621 | 56.68 | 194 | 33.94 |
| TORSION1 | 14884 | 39 | 39 | 64 | 19.85 | 37 | 38 | 1148 | 86.08 | 224 | 35.36 |
| TORSION2 | 14884 | 19 | 19 | 43 | 11.10 | 14 | 15 | 2063 | 173.28 | 521 | 91.56 |
| TORSION3 | 14884 | 20 | 20 | 26 | 9.06 | 19 | 20 | 332 | 21.13 | 76 | 10.66 |
| TORSION4 | 14884 | 18 | 18 | 27 | 8.98 | 14 | 15 | 653 | 34.99 | 417 | 65.78 |
| TORSION5 | 14884 | 11 | 11 | 12 | 4.67 | 9 | 10 | 93 | 5.74 | 40 | 5.06 |
| TORSION6 | 14884 | 15 | 15 | 18 | 7.07 | 8 | 9 | 151 | 8.54 | 362 | 53.99 |
| TORSIONA | 14884 | 39 | 39 | 64 | 21.45 | 37 | 38 | 1147 | 98.23 | 205 | 37.38 |
| TORSIONB | 14884 | 24 | 24 | 50 | 14.54 | 15 | 16 | 1982 | 186.69 | 371 | 70.13 |
| TORSIONC | 14884 | 20 | 20 | 26 | 9.80 | 19 | 20 | 332 | 24.65 | 89 | 13.97 |
| TORSIOND | 14884 | 18 | 18 | 26 | 9.70 | 14 | 15 | 634 | 39.70 | 409 | 69.59 |
| TORSIONE | 14884 | 11 | 11 | 12 | 5.06 | 9 | 10 | 93 | 6.55 | 38 | 5.44 |
| TORSIONF | 14884 | 15 | 15 | 19 | 7.71 | 7 | 8 | 154 | 9.36 | 341 | 56.83 |

requires less time than L-BFGS-B. These results support the conclusion that TRON is preferable to L-BFGS-B if the Hessian matrix can be obtained explicitly. We also expect TRON to outperform L-BFGS-B for sparse problems if the sparsity pattern of the Hessian matrix is provided because with this information the Hessian matrix can be obtained efficiently from differences of gradients.

The results in Table 7.1 also show that on these problems TRON requires less time than LANCELOT and significantly fewer conjugate gradient iterations than LANCELOT. Reducing the number of conjugate gradient iterations is important because this number is likely to increase as the number of variables increases. We note that since for these problems the cost of the conjugate gradient iterations is significant, fewer conjugate gradient iterations translates into smaller computing times.

Another observation that can be made on the results of Table 7.1 is that LANCELOT usually requires fewer major iterations than TRON. Differences in the number of major iterations are due, in part, to the choice of Cauchy point and the use of projected searches. These algorithmic choices in TRON tend to add many constraints, and on some of these problems, they lead to a larger number of major iterations. We also note that a detailed examination of the output shows that even when both codes require the same number of iterations, the algorithms visit different faces of the feasible set.

As a minor point, note that TRON almost always requires the same number of function and Hessian evaluations. This is an algorithmic decision since we always evaluate the gradient and Hessian at successful iterates. On the other hand, if an iterate satisfies the termination criteria (7.1), LANCELOT returns without evaluating the Hessian matrix at the final iterate.

The number of conjugate gradient iterations in LANCELOT can usually be reduced by using other preconditioners instead of the default banded preconditioner. Other preconditioners, however, usually require more memory and more floating point operations per conjugate gradient iteration.

In Table 7.2 we present the results of using LANCELOT with Munksgaard's ma31 preconditioner [31], which is an incomplete Cholesky factorization with a drop tolerance. A disadvantage of using the ma31 preconditioner with LANCELOT is that the memory requirements are unpredictable. The user is asked to allocate a given amount of memory, and if this amount is not sufficient, then an error message is issued. On the other hand, the incomplete Cholesky factorization icfs used in TRON does not require the choice of a drop tolerance, and the amount of storage can be specified in advance. For the results presented in this section icfs uses $5n$ additional (double precision) words. For a comparison of ma31 with icfs, see Lin and Moré [26].

Comparison of the LANCELOT results in Table 7.1 with those in Table 7.2 show that in all cases the number of function evaluations and Hessian evaluations for both preconditioners are identical, and that the main difference is the number of conjugate gradient iterations. Also note that, with the exception of problems OBSTCLBL and OBSTCLBM, the number of conjugate gradient iterations and the time required to solve the problems with LANCELOT decreased when the ma31 preconditioner was used. Overall, these results show that for these problems the ma31 preconditioner is preferable in LANCELOT.

The results in Table 7.2 show that TRON requires fewer conjugate gradient iterations, and on most problems, less time than LANCELOT with the ma31 preconditioner. Also note that there were five problems (OBSTCLAE, OBSTCLBL, OBSTCLBM, TORSION2, and TORSIONB) where LANCELOT required more than 1000 conjugate gradient iterations, and that on these problems the reductions in time over the default preconditioner were not substantial. For these problems the differences in conjugate gradient iterations are not due to the use of different preconditioners but to the methods used by TRON and LANCELOT

Table 7.2: Performance on the CUTE problems: LANCELOT with ma31

| | | TRON | | | | LANCELOT (ma31) | | | |
|---|---|---|---|---|---|---|---|---|---|
| Problem | $n$ | nh | nf | ncg | time | nh | nf | ncg | time |
| BDEXP | 5000 | 11 | 11 | 10 | 1.43 | 10 | 11 | 10 | 1.32 |
| CVXBQP1 | 10000 | 2 | 2 | 0 | 0.24 | 1 | 2 | 1 | 0.80 |
| JNLBRNG1 | 15625 | 26 | 26 | 33 | 15.22 | 24 | 25 | 179 | 28.69 |
| JNLBRNG2 | 15625 | 16 | 16 | 27 | 9.21 | 14 | 15 | 70 | 13.09 |
| JNLBRNGA | 15625 | 23 | 23 | 29 | 12.46 | 21 | 22 | 166 | 24.29 |
| JNLBRNGB | 15625 | 10 | 10 | 15 | 5.29 | 8 | 9 | 46 | 7.56 |
| MCCORMCK | 10000 | 6 | 7 | 6 | 1.46 | 4 | 5 | 4 | 1.41 |
| NCVXBQP1 | 10000 | 2 | 2 | 0 | 0.24 | 4 | 5 | 0 | 3.03 |
| NCVXBQP2 | 10000 | 10 | 10 | 10 | 1.44 | 7 | 8 | 93 | 3.34 |
| NCVXBQP3 | 10000 | 10 | 10 | 10 | 1.39 | 6 | 7 | 124 | 2.61 |
| NOBNDTOR | 14884 | 38 | 38 | 71 | 22.03 | 36 | 37 | 176 | 36.61 |
| NONSCOMP | 10000 | 9 | 9 | 8 | 1.44 | 8 | 9 | 8 | 1.66 |
| OBSTCLAE | 15625 | 27 | 27 | 51 | 14.48 | 2 | 3 | 7154 | 809.04 |
| OBSTCLAL | 15625 | 25 | 25 | 39 | 12.64 | 24 | 25 | 79 | 15.62 |
| OBSTCLBL | 15625 | 20 | 20 | 42 | 12.81 | 22 | 21 | 2346 | 307.67 |
| OBSTCLBM | 15625 | 8 | 8 | 15 | 5.41 | 5 | 6 | 1554 | 213.38 |
| OBSTCLBU | 15625 | 21 | 21 | 33 | 11.85 | 19 | 20 | 165 | 22.72 |
| TORSION1 | 14884 | 39 | 39 | 64 | 19.85 | 37 | 38 | 159 | 27.97 |
| TORSION2 | 14884 | 19 | 19 | 43 | 11.10 | 14 | 15 | 1592 | 143.66 |
| TORSION3 | 14884 | 20 | 20 | 26 | 9.06 | 19 | 20 | 52 | 9.02 |
| TORSION4 | 14884 | 18 | 18 | 27 | 8.98 | 14 | 15 | 438 | 25.91 |
| TORSION5 | 14884 | 11 | 11 | 12 | 4.67 | 9 | 10 | 14 | 2.99 |
| TORSION6 | 14884 | 15 | 15 | 18 | 7.07 | 8 | 9 | 116 | 7.46 |
| TORSIONA | 14884 | 39 | 39 | 64 | 21.45 | 37 | 38 | 175 | 31.80 |
| TORSIONB | 14884 | 24 | 24 | 50 | 14.54 | 15 | 16 | 1606 | 153.55 |
| TORSIONC | 14884 | 20 | 20 | 26 | 9.80 | 19 | 20 | 52 | 9.76 |
| TORSIOND | 14884 | 18 | 18 | 26 | 9.70 | 14 | 15 | 445 | 29.13 |
| TORSIONE | 14884 | 11 | 11 | 12 | 5.06 | 9 | 10 | 13 | 3.27 |
| TORSIONF | 14884 | 15 | 15 | 19 | 7.71 | 7 | 8 | 107 | 7.46 |

to compute the minor iterates. LANCELOT uses a line search, and thus only one constraint is added at each minor iteration. As a result many minor iterates can be generated, and determining a minor iterate almost certainly requires at least one conjugate gradient iteration. For these five problems LANCELOT generated, respectively, 7155, 1710, 1184, 1533, and 1541 minor iterates. TRON, on the other hand, uses a projected search and is thus able to add many constraints at each minor iteration. For these problems TRON generated 27, 26, 10, 19, and 24 minor iterates.

These results support the conclusion that TRON tends to require significantly fewer minor iterations than LANCELOT. Moreover, the use of projected searches is the major reason for TRON requiring a small number of minor iterates.

General conclusions cannot be drawn from these results because, as already noted, this problem set does not fully test these algorithms. Our numerical results are also affected by non-algorithmic differences between TRON and LANCELOT. We have already noted that these codes differ in the amount of memory required, but TRON and LANCELOT differ in other ways. For example, LANCELOT uses the partial separability structure, while TRON only uses the sparsity structure.

Table 7.3: Parameters for the MINPACK-2 test problems.

| Problem | $n$ | $n_x$ | $n_y$ | $\lambda$ | $l$ | $u$ |
|---|---|---|---|---|---|---|
| EPT1 | 10000 | 200 | 50 | 1.0d0 | default | default |
| EPT2 | 10000 | 200 | 50 | 5.0d0 | default | default |
| EPT3 | 10000 | 200 | 50 | 10.0d0 | default | default |
| PJB1 | 10000 | 100 | 100 | 0.1d0 | default | 1.0d2 |
| PJB2 | 10000 | 100 | 100 | 0.5d0 | default | 1.0d2 |
| PJB3 | 10000 | 100 | 100 | 0.9d0 | default | 1.0d2 |
| MSA1 | 10000 | 200 | 50 | 0.0d0 | -0.4d0 | 0.4d0 |
| MSA2 | 10000 | 200 | 50 | 0.0d0 | -0.2d0 | 0.2d0 |
| MSA3 | 10000 | 200 | 50 | 0.0d0 | -0.1d0 | 0.1d0 |
| SSC1 | 10000 | 100 | 100 | 5.0d0 | 1.0d-1 | 1.0d0 |
| SSC2 | 10000 | 100 | 100 | 5.0d0 | 1.0d-2 | 1.0d0 |
| SSC3 | 10000 | 100 | 100 | 5.0d0 | 1.0d-3 | 1.0d0 |
| SSC4 | 10000 | 100 | 100 | 5.0d0 | 1.0d-4 | 1.0d0 |
| DGL2 | 10000 | 50 | 50 | 2.0d0 | -1.0d20 | 1.0d20 |

We also compared TRON with L-BFGS-B on a test set drawn from the MINPACK-2 collection of large-scale problems [1]. The MINPACK-2 problems defined by Table 7.3 are finite-dimensional approximation of an infinite-dimensional variational problem defined over a grid with $n_x$ and $n_y$ grid points in each coordinate direction. The column labeled $\lambda$ in Table 7.3 defines the value of a parameter associated with the problem, while the last two columns define the lower and upper bounds on the variables. For these results we used the termination test

$$\|\nabla_\Omega f(x)\|_2 \le 10^{-5} \|\nabla f(x_0)\|_2, \tag{7.2}$$

where $\nabla_\Omega f$ is the projected gradient (2.8). This termination test is generally preferable to (7.1) because (7.2) is invariant to changes in the scale of $f$.

The number of grid points $n_x$ and $n_y$ and the parameter $\lambda$ can be easily modified in the MINPACK-2 problems, thereby providing a convenient means to generate difficult problems. In general, the problems become more difficult as the ratio $n_y/n_x$ deviates from unity. We have restricted the testing to problems where this ratio lies in the interval $[0.25, 1]$, which leads to relatively easy problems. In some cases, the choice of $\lambda$, and of lower and upper bounds also affects the performance of optimization algorithms.

In the first two problems in Table 7.4 we examine the behavior of TRON and L-BFGS-B as $\lambda$ changes. For problem EPT (elastic-plastic torsion) the parameter $\lambda$ is the force constant, and for this problem the number of active constraints increases as $\lambda$ increases. The results in Table 7.4 show that EPT becomes easier to solve as $\lambda$ increases. This finding is reasonable because the EPT problem tends to be increasingly linear as $\lambda$ increases. The results for problem PJB (pressure in a journal bearing) show that this problem becomes increasingly harder to solve as $\lambda$ approaches unity. For this problem $\lambda$ is the eccentricity of the journal bearing, so this result is reasonable.

Table 7.4: Performance on the MINPACK-2 problems with $n = 10,000$.

| | TRON | | | | L-BFGS-B | |
|---|---|---|---|---|---|---|
| Problem | nh | nf | ncg | time | nfg | time |
| EPT1 | 30 | 30 | 96 | 9.38 | 466 | 35.17 |
| EPT2 | 31 | 31 | 61 | 7.69 | 445 | 27.81 |
| EPT3 | 21 | 21 | 31 | 4.06 | 229 | 10.66 |
| PJB1 | 22 | 22 | 42 | 5.92 | 717 | 49.25 |
| PJB2 | 13 | 13 | 29 | 3.38 | 542 | 31.29 |
| PJB3 | 7 | 7 | 17 | 1.76 | 2765 | 150.91 |
| MSA1 | 27 | 48 | 94 | 19.06 | 776 | 65.35 |
| MSA2 | 16 | 22 | 65 | 10.47 | 613 | 50.50 |
| MSA3 | 19 | 19 | 48 | 9.89 | 487 | 39.79 |
| SSC1 | 5 | 5 | 23 | 3.28 | 347 | 36.32 |
| SSC2 | 6 | 6 | 25 | 4.11 | 345 | 36.83 |
| SSC3 | 6 | 6 | 26 | 3.96 | 377 | 40.26 |
| SSC4 | 6 | 6 | 26 | 3.99 | 293 | 30.91 |
| GL2 | 8 | 8 | 364 | 34.73 | 3521 | 372.89 |

In problems MSA and SSC we examine the behavior of TRON and L-BFGS-B as the lower and upper bounds $l$ and $u$ change. The results of this testing were somewhat disappointing because for these problems there does not seem to be a strong correlation between the choice of bounds and the number of iterations. The most dramatic change in performance occurs for L-BFGS-B and the MSA problem. Note, on the other hand, that the performance of TRON is relatively insensitive to the choice of bounds.

Problem GL2 is unconstrained but is included in these results because it is a hard problem for algorithms that do not use second-order information. The reason seems to be that the GL2 problem has a saddle point that attracts L-BFGS-B.

The most striking feature of the results in Table 7.4 is that TRON requires far fewer function and gradient evaluations than L-BFGS-B and that this translates into smaller computing times. This advantage is likely to increase as the number of variables increases because the number of iterations in a Newton method tends to grow slowly, while the number of iterations in limited-memory variable metric methods tends to grow rapidly as the number of variables increases. See, for example, the results of Bouaricha, Moré, and Wu [4].

## Acknowledgments

The implementation of the Newton code benefited from the work of Ali Bouaricha and Zhijun Wu on the unconstrained version of the code. We also thank Gail Pieper for her careful reading of the manuscript.

We also thank the referees and the associate editor, Nick Gould, for their comments. One of the referees provided an extremely detailed and helpful report that led to improvements in the amount of detail and explanations in the paper.

## References

[1] B. M. Averick, R. G. Carter, J. J. Moré, and G.-L. Xue, *The MINPACK-2 test problem collection*, Preprint MCS-P153-0694, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois, 1992.

[2] D. P. Bertsekas, *Projected Newton methods for optimization problems with simple constraints*, SIAM J. Control Optim., 20 (1982), pp. 221–246.

[3] I. Bongartz, A. R. Conn, N. I. M. Gould, and P. L. Toint, CUTE*: Constrained and unconstrained testing environment*, ACM Trans. Math. Software, 21 (1995), pp. 123–160.

[4] A. Bouaricha, J. J. Moré, and Z. Wu, *Preconditioning Newton's method*, Preprint ANL/MCS-P715-0598, Argonne National Laboratory, Argonne, Illinois, 1998.

[5] J. V. Burke and J. J. Moré, *On the identification of active constraints*, SIAM J. Numer. Anal., 25 (1988), pp. 1197–1211.

[6] ——, *Exposing constraints*, SIAM J. Optim., 4 (1994), pp. 573–595.

[7] J. V. Burke, J. J. Moré, and G. Toraldo, *Convergence properties of trust region methods for linear and convex constraints*, Math. Programming, 47 (1990), pp. 305–336.

[8] P. H. Calamai and J. J. Moré, *Projected gradient methods for linearly constrained problems*, Math. Programming, 39 (1987), pp. 93–116.

[9] T. F. Coleman and Y. Li, *An interior trust region approach for nonlinear minimization subject to bounds*, SIAM J. Optim., 6 (1996), pp. 418–445.

[10] ——, *A trust region and affine scaling interior point method for nonconvex minimization with linear inequality constraints*, Technical Report TR97-1642, Cornell University, Ithaca, New York, 1997.

[11] ——, *Combining trust region and affine scaling for linearly constrained nonconvex minimization*, in Advances in Nonlinear Programming, Y. Yuan, ed., Kluwer Academic Publishers, 1998, pp. 219–250.

[12] A. R. Conn, N. I. M. Gould, A. Sartenaer, and P. L. Toint, *Global convergence of a clas of trust region algorithms for optimization using inexact projections on convex constraints*, SIAM J. Optim., 3 (1993), pp. 164–221.

[13] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Global convergence of a class of trust region algorithms for optimization problems with simple bounds*, SIAM J. Numer. Anal., 25 (1988), pp. 433–460. Correction in SIAM J. Numer. Anal., 26 (1989), 764-767.

[14] ——, LANCELOT, no. 17 in Springer Series in Computational Mathematics, Springer-Verlag, 1992.

[15] ——, *Numerical experiments with the* LANCELOT *package (Release A) for large-scale nonlinear optimization*, Math. Programming, 73 (1996), pp. 73–110.

[16] J. E. Dennis and L. N. Vicente, *Trust–region interior–point algorithms for minimization problems with simple bounds*, in Applied Mathematics and Parallel Computing, Festschrift for Klaus Ritter, H. Fisher, B. Riedmüller, and S. Schäffler, eds., Physica–Verlag, Springer–Verlag, 1996, pp. 97–107.

[17] J. C. Dunn, *On the convergence of projected gradient processes to singular critical points*, J. Optim. Theory Appl., 55 (1987), pp. 203–216.

[18] F. Facchinei, J. Júdice, and J. Soares, *An active set Newton's method for large-scale nonlinear programs with box constraints*, SIAM J. Optim., 8 (1998), pp. 158–186.

[19] F. Facchinei and S. Lucidi, *A class of methods for optimization problems with simple bounds*, Technical Report R.336, IASI-CNR, Rome, Italy, 1992.

[20] A. Forsgren and W. Murray, *Newton methods for large-scale linear inequality-constrained minimization*, SIAM J. Optim., 7 (1997), pp. 162–176.

[21] A. Friedlander, J. M. Martínez, and S. A. Santos, *A new trust region algorithm for bound constrained minimization*, Appl. Math. Optim., 30 (1994), pp. 235–266.

[22] E. M. Gafni and D. P. Bertsekas, *Two-metric projection methods for constrained optimization*, SIAM J. Control Optim., 22 (1984), pp. 936–964.

[23] D. M. Gay, *A trust region approach to linearly constrained optimization*, in Numerical Analysis, D. F. Griffiths, ed., Lecture Notes in Mathematics 1066, Springer-Verlag, 1984, pp. 72–105.

[24] M. Heinkenschloss, M. Ulbrich, and S. Ulbrich, *Superlinear and quadratic convergence of affine-scaling interior-point Newton methods for problems with simple bounds without strict complementarity assumption*, Technical Report TR97-30, Rice

University, Houston, Texas, 1997. Revised September 1998. To appear in Mathematical Programming.

[25] M. Lescrenier, *Convergence of trust region algorithms for optimization with bounds when strict complementarity does not hold*, SIAM J. Numer. Anal., 28 (1991), pp. 476–495.

[26] C.-J. Lin and J. J. Moré, *Incomplete Cholesky factorizations with limited memory*, Preprint MCS-P682-0897, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois, 1997. To appear in the SIAM Journal on Scientific Computing.

[27] J. J. Moré, *Trust regions and projected gradients*, in Systems Modelling and Optimization, M. Iri and K. Yajima, eds., Lecture Notes in Control and Information Sciences 113, Springer-Verlag, 1988, pp. 1–13.

[28] ——, *Global methods for nonlinear complementarity problems*, Math. Oper. Res., 21 (1996), pp. 589–614.

[29] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[30] J. J. Moré and G. Toraldo, *On the solution of large quadratic programming problems with bound constraints*, SIAM J. Optim., 1 (1991), pp. 93–113.

[31] N. Munksgaard, *Solving sparse symmetric sets of linear equations by preconditioned conjugate gradients*, ACM Trans. Math. Software, 6 (1980), pp. 206–219.

[32] S. M. Robinson, *Strongly regular generalized equations*, Math. Oper. Res., 5 (1980), pp. 43–62.

[33] A. Schwartz and E. Polak, *Family of projected descent methods for optimization problems with simple bounds*, J. Optim. Theory Appl., 92 (1997), pp. 1–31.

[34] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[35] P. L. Toint, *Global convergence of a class of trust region methods for nonconvex minimization in Hilbert space*, IMA J. Numer. Anal., 8 (1988), pp. 231–252.

[36] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, *L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization*, ACM Trans. Math. Software, 23 (1997), pp. 550–560.