SIAM J. Sci. Comput.
Vol. 21, No. 1, pp. 1–23

A SUBSPACE, INTERIOR, AND CONJUGATE GRADIENT METHOD FOR LARGE-SCALE BOUND-CONSTRAINED MINIMIZATION PROBLEMS*

MARY ANN BRANCH[†], THOMAS F. COLEMAN[‡], AND YUYING LI[‡]

Abstract. A subspace adaptation of the Coleman–Li trust region and interior method is proposed for solving large-scale bound-constrained minimization problems. This method can be implemented with either sparse Cholesky factorization or conjugate gradient computation. Under reasonable conditions the convergence properties of this subspace trust region method are as strong as those of its full-space version.

Computational performance on various large test problems is reported; advantages of our approach are demonstrated. Our experience indicates that our proposed method represents an efficient way to solve large bound-constrained minimization problems.

Key words. interior method, trust region method, negative curvature direction, inexact Newton step, conjugate gradients, bound-constrained problem, box constraints

AMS subject classifications. 65K05, 90C06, 90C30

PII. S1064827595289108

1. Introduction.

Recently Coleman and Li [1, 2, 3] proposed two interior and reflective Newton methods to solve the bound-constrained minimization problem, i.e.,

$$\min_{x\in\Re^n}\{f(x): \quad l \le x \le u\}, \tag{1.1}$$

where $l \in \{\Re \cup \{-\infty\}\}^n$, $u \in \{\Re \cup \{\infty\}\}^n$, $l < u$, and f is a smooth function, $f:\Re^n \to \Re^1$. Both algorithms are *interior* methods since the iterates $\{x_k\}$ are in the strict interior of the feasible region, i.e., $x_k \in int(\mathcal{F}) \overset{\text{def}}{=} \{x : l < x < u\}$. These two methods differ in that a line search to update iterates is used in [2, 3] while a trust region idea is used in [1]. However, in both cases convergence is accelerated with the use of a novel reflection technique.

The line search method version appears to be computationally viable for large-scale quadratic problems [3]. The trust region framework has more appeal, we feel, for nonlinear problems; however, the method given in [1], which we refer to as the trust region interior reflective (TIR) method, is not suitable for the large-scale case. Our main objective here is to investigate solving large-scale bound-constrained nonlinear minimization problems (1.1) using an adaptation of the TIR approach suitable for large problems.

The TIR method generalizes the trust region idea for unconstrained minimization to bound-constrained nonlinear minimization. A distinguishing feature of the

*Received by the editors July 14, 1995; accepted for publication (in revised form) April 4, 1998; published electronically August 3, 1999. This research was partially supported by the Applied Mathematical Sciences Research Program (KC-04-02) of the Office of Energy Research of the U.S. Department of Energy under grant DE-FG02-90ER25013.A000, by NSF, AFOSR, and ONR through grant DMS-8920550, and by the Cornell Theory Center which receives major funding from Cornell University, New York State, the National Center for Research Resources at the National Institutes of Health, the National Science Foundation, the Department of Defense Modernization Program, the United States Department of Agriculture, and members of the Corporate Partnership Program.

http://www.siam.org/journals/sisc/21-1/28910.html

[†] The MathWorks, Inc., 24 Prime Park Way, Natick, MA 01760 (branch@mathworks.com).

[‡] Computer Science Department and Center for Applied Mathematics, Cornell University, Ithaca, NY 14850 (coleman@cs.cornell.edu; yuying@cs.cornell.edu).

TIR method is that the main computation per iteration is solving a standard scaled trust region subproblem, that is, the same subproblem as in trust region methods for unconstrained minimization:

$$\min_{s \in \Re^n} \{\psi_k(s) : \ \|D_k s\|_2 \leq \Delta_k\}, \tag{1.2}$$

where $\psi_k(s)$ is a quadratic function, D_k is a positive diagonal matrix, and Δ_k is a positive scalar. The method of Moré and Sorensen [4] can be directly applied to (1.2) if sparse Cholesky factorizations can be computed efficiently. However, this method is unsuitable for large-scale problems if the Hessian matrix is not explicitly available or (sparse) Cholesky factorizations are too expensive. Recently, Sorensen [5] proposed a new method for solving the subproblem (1.2) using matrix-vector multiplications. Nonetheless, the effectiveness of the Sorensen approach for large-scale minimization, particularly in the context of our trust region algorithm, is yet to be investigated.

We take the view that solving the full-space trust region subproblem (1.2) is too costly for a large-scale problem. This view is shared by Steihaug [6], who proposes an approximate (conjugate gradient) approach. Steihaug's approach to (1.2) is viable, although the computational studies in [7, 8] indicate that important negative curvature information can be missed, causing a significant increase in the number of minimization iterations.

In this paper, we propose an alternative: an approximate *subspace* trust region approach (STIR). We verify that, under reasonable conditions, the convergence properties of STIR are as strong as those of its full-space version. We explore the use of sparse linear algebra techniques, i.e., sparse Cholesky factorization and preconditioned conjugate gradients, in the context of this approach.

In addition, we demonstrate the benefits of our affine scaling, reflection, and subspace techniques with computational results. First, for (1.1), our affine scaling technique outperforms the classical Dikin scaling [9], at least in the context of our algorithm. Second, we examine our method with and without reflection. We show that the reflection technique can substantially reduce the number of minimization iterations. Third, our computational experiments support the notion that the *subspace* trust region method is a promising way to solve large-scale bound-constrained nonlinear minimization problems. Finally, our subspace method is competitive with the active set method in LANCELOT [10] and we include results on test problems with negative curvature where our subspace method outperforms the LANCELOT method.

The paper is organized as follows. In section 2 we briefly summarize the existing TIR method. In section 3 we focus on how to solve the trust region subproblem (1.2): We summarize the Steihaug algorithm and our version of the subspace trust region method. We introduce a subspace method STIR and discuss its convergence properties in section 4. Issues concerning the computation of negative curvature vectors and inexact Newton steps are discussed in section 5; computational results are provided in section 6 indicating that performance typically is not impaired by using an inexact Newton step. Concluding remarks appear in section 7. The convergence analysis of the STIR method is included in the Appendix.

2. The TIR method. In this section we briefly review the full-space TIR method [1] sketched in Fig. 1. This method closely resembles a typical trust region method for unconstrained minimization, $\min_{x \in \Re^n} f(x)$. The key difference is the presence of the affine scaling (diagonal) matrices D_k and C_k. Next we briefly motivate

The TIR Method [1]

Let $0 < \mu < \eta < 1$, $0 < \Lambda_l < \Lambda_u$ and $\gamma_1 < 1 < \gamma_2$ be given. Let $x_0 \in \text{int}(\mathcal{F})$, $\Delta_0 < \Lambda_u$.

For $k = 0, 1, \ldots$

1. Compute f_k, g_k, D_k, H_k, and C_k; define the quadratic model
$$\psi_k(s) \overset{\text{def}}{=} g_k^T s + \tfrac{1}{2} s^T (H_k + C_k) s.$$
2. Compute a step s_k, with $x_k + s_k \in int(\mathcal{F})$, based on the subproblem
$$\min_s \{\psi_k(s) : \ \|D_k s\|_2 \le \Delta_k\}.$$
3. Compute
$$\rho_k = \frac{f(x_k + s_k) - f(x_k) + \frac{1}{2} s_k^T C_k s_k}{\psi_k(s_k)}.$$
4. If $\rho_k > \mu$ then set $x_{k+1} = x_k + s_k$. Otherwise set $x_{k+1} = x_k$.
5. Update Δ_k as specified below.

Updating Trust Region Size Δ_k

1. If $\rho_k \le \mu$ then set $\Delta_{k+1} \in (0, \gamma_1 \Delta_k]$.
2. If $\rho_k \in (\mu, \eta)$ then set $\Delta_{k+1} \in [\gamma_1 \Delta_k, \Delta_k]$.
3. If $\rho_k \ge \eta$ then

```
        if $\Delta_k > \Lambda_l$ then
            set $\Delta_{k+1} \in$ either $[\gamma_1 \Delta_k, \Delta_k]$ or $[\Delta_k, \gamma_2 \Delta_k]$,
        otherwise
            set $\Delta_{k+1} \in [\Delta_k, \min(\gamma_2 \Delta_k, \Lambda_u)]$.
```

FIG. 1. *The TIR method for minimization subject to bounds.*

these matrices and the TIR algorithm.

The trust region subproblem (1.2) and the affine scaling matrices D_k and C_k (the diagonal matrix C_k is defined subsequently in (2.8)) arise naturally from examining the Kuhn–Tucker conditions for (1.1),

$$D(x)^{-2} \nabla f(x) = 0, \tag{2.1}$$

where

$$D(x) \overset{\text{def}}{=} \text{diag}(|v(x)|^{-1/2}) \tag{2.2}$$

and the vector $v(x) \in \Re^n$ is defined below. For each $1 \le i \le n$,

(i) if $g_i < 0$ and $u_i < \infty$ then $v_i \overset{\text{def}}{=} x_i - u_i$;

(ii) if $g_i \ge 0$ and $l_i > -\infty$ then $v_i \overset{\text{def}}{=} x_i - l_i$;

(iii) if $g_i < 0$ and $u_i = \infty$ then $v_i \overset{\text{def}}{=} -1$;

(iv) if $g_i \ge 0$ and $l_i = -\infty$ then $v_i \overset{\text{def}}{=} 1$.

The nonlinear system (2.1) is not differentiable everywhere; nondifferentiability occurs when $v_i = 0$. Hence we avoid such points by maintaining strict feasibility, i.e., restricting $x_k \in \text{int}(\mathcal{F})$. A Newton step for (2.1) is then defined and satisfies

$$\hat{M}_k D_k s_k^N = -\hat{g}_k, \tag{2.3}$$

where

$$\begin{aligned} \hat{g}_k &\overset{\text{def}}{=} D_k^{-1} g_k = \text{diag}(|v_k|^{1/2}) g_k, \\ \hat{M}_k &\overset{\text{def}}{=} D_k^{-1} H_k D_k^{-1} + \text{diag}(g_k) J_k^v. \end{aligned} \tag{2.4}$$

Here $J^v(x) \in \Re^{n \times n}$ plays the role of the Jacobian of $|v(x)|$. Each diagonal component of the diagonal matrix J^v equals zero or ± 1. If all the components of l and u are finite, $J^v = \text{diag}(\text{sgn}(g))$. At a point where $g_i = 0$, $|v_i|$ may not be differentiable. We define $J_{ii}^v = 0$ at such a point. Nondifferentiability of this type is not a cause for concern because, for such a component, it is not significant which value v_i takes. Furthermore, $|v_i|$ will still be discontinuous at this point, but the function $|v_i| \cdot g_i$ is continuous.

Equation (2.3) suggests the use of the affine scaling transformation $\hat{x} \overset{\text{def}}{=} D_k x$. This transformation reduces the constrained problem (1.1) into an unconstrained problem: A local minimizer of (1.1) corresponds to an unconstrained minimizer in the new coordinates $\hat{x}$ (for more details, see [1]). Therefore a reasonable way to improve x_k is to solve the trust region subproblem

$$\min_{\hat{s} \in \Re^n} \{\hat{\psi}_k(\hat{s}) : \ \|\hat{s}\|_2 \leq \Delta_k\}, \tag{2.5}$$

where

$$\hat{\psi}_k(\hat{s}) \overset{\text{def}}{=} \hat{g}_k^T \hat{s} + \tfrac{1}{2} \hat{s}^T \hat{M}_k \hat{s}.$$

Let $s = D_k^{-1} \hat{s}$. Subproblem (2.5) is equivalent to the following problem in the original variable space:

$$\min_{s \in \Re^n} \{\psi_k(s) : \ \|D_k s\|_2 \leq \Delta_k\}, \tag{2.6}$$

where

$$\psi_k(s) \overset{\text{def}}{=} s^T g_k + \tfrac{1}{2} s^T M_k s, \tag{2.7}$$

$$C_k \overset{\text{def}}{=} D_k \,\text{diag}(g_k) J_k^v D_k, \tag{2.8}$$

$$M_k \overset{\text{def}}{=} H_k + C_k. \tag{2.9}$$

In addition to the close resemblance to an unconstrained trust region method, the TIR algorithm has strong convergence properties with explicit conditions on steps for optimality. We now describe these conditions.

The TIR algorithm requires strict feasibility, i.e., $\{x_k\} \in \text{int}(\mathcal{F})$. Let τ_k^* denote the minimizer along d_k within the feasible trust region, i.e.,

$$\tau_k^* = \text{argmin}_\tau \{\psi_k(\tau d_k) : \|\tau D_k d_k\| \leq \Delta_k, x_k + \tau d_k \in \mathcal{F}\}$$

Given the current iterate x_k and a direction d_k, we define a stepsize $\alpha_k^*[d_k]$ to be the optimal step within $\text{int}(\mathcal{F})$ if $x_k + \tau d_k$ is strictly feasible; otherwise, $\alpha_k^*[d_k]$ is chosen just short of the step to the boundary. Let $\theta_k \in [\theta_l, 1]$ for some $0 < \theta_l < 1$ and $(\theta_k - 1) = O(\|d_k\|)$. Then

$$\alpha_k^*[d_k] \stackrel{\text{def}}{=} \theta_k \tau_k^* d_k \stackrel{\text{def}}{=} \begin{cases} \tau_k^* d_k & \text{if } x_k + \tau_k^* d_k \in \text{int}(\mathcal{F}), \\ \theta_k \tau_k^* d_k & \text{otherwise.} \end{cases} \tag{2.10}$$

The above definition implies that $\theta_k = 1$ if $x_k + \tau_k^* d_k \in \text{int}(\mathcal{F})$.

We use $\psi_k^*[d_k]$ to denote the minimum value of $\psi_k(s)$ along the direction d_k within the feasible trust region, i.e.,

$$\psi_k^*[d_k] \stackrel{\text{def}}{=} \psi_k(s_k^*) = \min\{\psi_k(s) : s = \tau d_k, \|D_k s\| \le \Delta_k, x_k + s \in \mathcal{F}\}.$$

Explicit conditions which yield first- and second-order optimality are analogous to those of trust region methods for unconstrained minimization [1]:

(AS.3) $\psi_k(s_k) < \beta \psi_k^*[-D_k^{-2} g_k]$, $\|D_k s_k\| \le \beta_0 \Delta_k$, $x_k + s_k \in \text{int}(\mathcal{F})$.

(AS.4) Assume that p_k is a solution to $\min_{s \in \Re^n} \{\psi_k(s) : \|D_k s\| \le \Delta_k\}$ and β^q and β_0^q are two positive constants. Then s_k satisfies $\psi_k(s_k) < \beta^q \psi_k^*[p_k]$, $\|D_k s_k\| \le \beta_0^q \Delta_k$, $x_k + s_k \in \text{int}(\mathcal{F})$.

Condition (AS.3) is necessary for first-order convergence; (AS.4), together with (AS.3), is necessary for second-order convergence. Both conditions (AS.3) and (AS.4) are extensions of convergence conditions for unconstrained trust region methods. In particular, when $l = -\infty$ and $u = \infty$, these assumptions are exactly what is required of trust region methods for unconstrained minimization problems.

Satisfaction of both conditions (AS.3) and (AS.4) is not difficult. For example, one can choose s_k so that $\psi_k(s_k)$ is the minimum of the values $\psi_k^*[p_k]$ and $\psi_k^*[-D_k^{-2} g_k]$. However, this does not lead to an efficient algorithm. In [3, 2], we utilized a reflection technique to permit further possible reduction of the objective function along a reflection path on the boundary. We found in [3, 2] that this reflection process significantly enhances performance for minimizing a general quadratic function subject to simple bounds.

In the context of TIR when solving nonlinear problems we use a (single) reflection step defined as follows. Given a step p_k, consider the first bound constraint crossed by p_k: assume it is the ith bound constraint (either the ith lower or the ith upper bound). Then the reflection step $p_k^R = p_k$ except in the ith component, where $p_{k_i}^R = -p_{k_i}$. Note that when the scaled steepest descent direction $-D_k^{-2} g_k$ points away from the ith constraint crossed by p_k, as in Fig. 2, the reflected step p_k^R permits further descent since $-D_{k_{ii}}^{-2} g_{k_i} * p_{k_i}^R > 0$.

For all the computational results in this paper, s_k is determined from the best of three points corresponding to $\psi_k^*[p_k]$, $\psi_k^*[-D_k^{-2} g_k]$, and $\psi_k^*[p_k^R]$, where p_k^R is p_k reflected on the first boundary it encounters as described above; see Fig. 2.

The updating of the trust region size Δ_k in Fig. 1 differs somewhat from the typical updating scheme for unconstrained problems. The updating is standard except in the third step, where we may either increase or decrease Δ_k when $\rho_k^f \ge \eta$. Such flexibility allows adjustment of the trust region size Δ_k for nonlinearity, as in the unconstrained case, and also for feasibility (this permits further reduction in Δ_k even when $\rho_k^f \ge \eta$) [1].

We can appreciate the convergence results for this approach by observing the role of the affine scaling matrix D_k in (2.1). For the components x_i which are approaching the "correct" bounds, the sequence of vectors $\{-D_k^{-2} g_k\}$ becomes increasingly

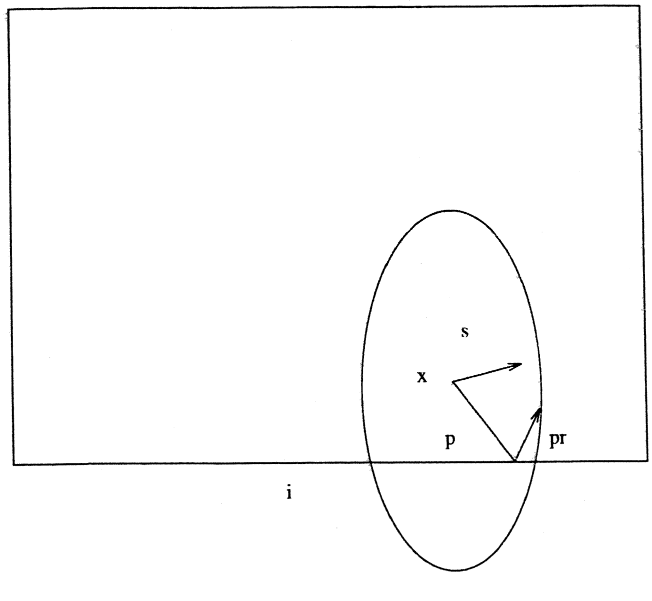

FIG. 2. *Reflection technique.*

tangential to these bounds. Hence, the bounds will not prevent a large stepsize along $\{-D_k^{-2}g_k\}$. For the components x_i which are approaching the "incorrect" bounds, $\{-D_k^{-2}g_k\}$ points away from these bounds in relatively large angles (the corresponding diagonal components of D_k are relatively large and g_k points away from these bounds). Hence, a reduction of at least $\psi_k^*[-D_k^{-2}g_k]$ implies the scaled gradient $\{D_k^{-2}g_k\}$ converges to zero (i.e., first-order optimality).

The scaling matrix used in our approach is related to, but is different from, the scaling typically used in affine scaling methods for linear programming. The affine scaling matrix $D_k^{\text{affine}} \overset{\text{def}}{=} \text{diag}(\min(x_k - l_k, u_k - x_k))$ [9], commonly used in linear programming, is formed from the distance of variables to their closest bounds. Our scaling matrix D_k^2 equals D_k^{affine} only when $\min(x_k - l_k, u_k - x_k) = |v_k|$. (Even in this case we employ the square root of the quantities used to define D_k^{affine}.)

Before we investigate a subspace adaptation of TIR, we demonstrate the effectiveness of our reflection idea and affine scaling technique. We consider random problem instances of molecule minimization [11, 12], which minimize a quartic subject to bounds on the variables. Tables 1 and 2 list the average number of iterations (more than 10 random test problem instances for each entry) required for the different techniques under comparison. The notation $>$ in front of a number indicates that the average number is at least this number because the iteration number exceeds 1000, the maximum allowed, for some instance. Details of the algorithm implementation are given in section 6.

Table 1 demonstrates the significant difference made by a single reflection. The only difference between the rows with and without reflection is the following. Without reflection, s_k is determined by the best of the two points based on $\psi_k^*[p_k]$ and $\psi_k^*[-D_k^{-2}g_k]$; with reflection, s_k is determined by the best of the three points based on $\psi_k^*[p_k]$, $\psi_k^*[-D_k^{-2}g_k]$, and $\psi_k^*[p_k^R]$ (with reflection). The advantage of using the reflection technique is clearly demonstrated with this problem.

In Table 2 we compare the computational advantage of the selection D_k over D_k^{affine}: the only difference is the scaling matrix. In this table, all the problems have

TABLE 1
The STIR algorithm with and without reflection: number of iterations.

	n				
	100	200	400	800	1000
With reflection	34.1	41.7	66.8	83.4	93.6
Without reflection	71.4	> 210.1	> 425.4	> 302.2	> 408.5

TABLE 2
Comparison of the STIR scaling D_k and Dikin scaling D_k^{affine} : number of iterations.

		n				
		100	200	400	800	1000
Unconstrained:	D_k	38.6	47.3	61.4	72.7	93.6
	D_k^{affine}	36.4	49	58.5	73.9	94.6
Constrained:	D_k	36.6	50.5	65.6	89.7	102.3
	D_k^{affine}	> 517.4	> 617.6	> 517.3	> 1000	> 1000

bound constraints, but we differentiate between problems that have an unconstrained solution (no bounds active at a solution) and those with a constrained solution. We observe that for the unconstrained solution problems there is no significant difference between the two scaling matrices. However, for the constrained solution problems we tested, the choice D_k is clearly superior. When D_k is used, the number of iterations for a constrained solution problem is roughly the same as that for the corresponding unconstrained solution problem. For D_k^{affine}, on the other hand, the number of iterations for a constrained problem is much larger than for the corresponding unconstrained solution problem. A single reflection strategy was used for these problems (with both scalings).

3. Approximation to the trust region solution in unconstrained minimization. We discuss two approaches that have been shown effective in approximating a full-space trust region solution in unconstrained minimization.

Shultz, Schnabel, and Byrd [13] suggest replacing the full trust region subproblem in the unconstrained setting by

$$\min_{s\in\Re^n}\{\psi_k(s) : \ \|s\|_2 \le \Delta_k, \ s \in \mathcal{S}_k\}, \tag{3.1}$$

where $\mathcal{S}_k$ is a low-dimensional subspace.

Another possible consideration for the approximation of (1.2) is the Steihaug idea [6], also proposed in the large-scale unconstrained minimization setting. In a nutshell, Steihaug proposes applying the method of preconditioned conjugate gradients (PCG) to the current Newton system until negative curvature is revealed, the Newton system residual is sufficiently reduced, or the current approximate solution reaches the boundary of the trust region.

We prefer the subspace approach given in [13]. In our implementation of the subspace method we apply the PCG method to the current Newton system. The PCG algorithm iterates until either a negative curvature direction is discovered or the Newton system residual is sufficiently decreased. Unlike the Steihaug method, our method ignores the boundary of the trust region during the PCG computation; thus this boundary will not keep us from finding either negative curvature or a step that sufficiently decreases the Newton system residual. Then we solve the subproblem (3.1) with the subspace $\mathcal{S}_k$ constructed from the gradient and the output of the PCG algorithm: $\mathcal{S}_k$ spans the gradient and a direction of negative curvature, or it spans the gradient and an inexact Newton step.

We believe that a subspace trust region approach leads to faster convergence because it better captures the negative curvature information than does the Steihaug approach [6]. This belief is based on computational studies, in the unconstrained minimization setting, given in [7, 8].

4. The STIR method. On the basis of the discussion in section 3 (and the computational studies in [7, 8]), we propose a large-scale subspace adaptation of the TIR method [1] for the bound-constrained problem (1.1).

In applying the unconstrained subspace approach to the box constrained setting, we replace the full trust region subproblem (1.2) by the following subspace subproblem:

$$\min_{s\in\Re^n}\{\psi_k(s):\ \|D_k s\|_2 \le \Delta_k,\ s\in\mathcal{S}_k\}, \tag{4.1}$$

where $\mathcal{S}_k$ is a small-dimensional subspace in $\Re^n$, e.g., a two-dimensional subspace. For example, a two-dimensional subspace for the trust region subproblem (2.5) can be defined by the span of $\{D_k^{-1}g_k, \hat{s}_k^N\}$ when $\hat{M}_k$ is positive definite and by the span of $\{D_k^{-1}g_k, D_k^{-1}\hat{w}_k\}$ when $\hat{M}_k$ is not positive definite and $\hat{w}_k$ is a vector of nonpositive curvature of $\hat{M}_k$. Will such subspace formulations succeed in achieving optimality? We examine this issue in more detail.

It is easy to prove that including the scaled gradient vector $D_k^{-2}g_k$ in $\mathcal{S}_k$ and satisfying (AS.3) will guarantee convergence to a point satisfying the first-order optimality conditions. Let us assume for now that $\{x_k\}$ converges to a first-order point x_*. To guarantee that x_* is also a second-order point, i.e., satisfies second-order necessary conditions, the following conditions must be met.

First, when the matrix $\hat{M}_k$ is indefinite or negative definite, a "sufficient negative curvature" condition must be carried over from the unconstrained setting [14]. To this end, we can require that sufficient negative curvature of the matrix $\hat{M}_k$ be captured if $\hat{M}_k$ is indefinite or negative definite; i.e., $\mathcal{S}_k$ must contain a vector $w_k = D_k^{-1}\hat{w}_k$ such that

$$\frac{\hat{w}_k^T\hat{M}_k\hat{w}_k}{\|\hat{w}_k\|} \le \max\{-\epsilon_{nc}, \tau\lambda_{\min}(\hat{M}_k)\}, \tag{4.2}$$

where ϵ_{nc} and τ are positive constants and $\lambda_{\min}(\cdot)$ is the smallest (most negative) eigenvalue.

Second, it is important that a solution to (4.1) lead to a sufficiently large step—the potential difficulty is running into a (bound) constraint immediately. This difficulty can be avoided if the stepsize sequence, along the trust region solution direction, is bounded away from zero. Subsequently, we define the *angle property*.

DEFINITION 1. *Let $\{x_k\}$ be a strictly feasible sequence and define diagonal matrix $D_k = D(x_k)$ via (2.2). A vector sequence $\{s_k\}$ has the* angle property *if* $\liminf_{k\to\infty}|D_k^2 s_k| < \infty$.

If fast local convergence is desired, then the subspace $\mathcal{S}_k$ should also contain a sufficiently accurate approximation to the Newton direction $D_k^{-1}\hat{s}_k^N$ when $\hat{M}_k$ is positive definite and $\hat{s}_k^N = -\hat{M}_k^{-1}\hat{g}_k$. An inexact Newton step $\hat{s}_k^N$ for problem (1.1) is defined as an approximate solution to

$$\hat{M}_k s = -\hat{g}_k$$

with accuracy η_k:

$$\hat{M}_k \hat{s}_k^{IN} = -\hat{g}_k + r_k \text{ such that } \|r_k\|/\|\hat{g}_k\| \leq \eta_k. \tag{4.3}$$

We can select two-dimensional subspaces satisfying all three properties and thus guarantee quadratic (superlinear) convergence to a second-order point. The subspace adaptation of the TIR algorithm (STIR) in Fig. 3 is an example of a subspace method capable of achieving the desired properties.

Our convergence results given below require the solution sequence of the subspace trust region subproblems (4.1) to have the angle property. Lemma 2 below indicates that this can be achieved if we set $\mathcal{S}_k = \text{span}\{w_k, z_k\}$, where $\{w_k\}$ and $\{z_k\}$ are the two angle property sequences of uniformly independent vectors in the sense that $\liminf\{\|z_k - w_k\|\} > 0$.

LEMMA 2. *Assume that $\{w_k\}$ and $\{z_k\}$ have the angle property with $\|D_k w_k\| = 1$ and $\|D_k z_k\| = 1$. Moreover, assume that $\lim inf_{k\to\infty}\{\|z_k - w_k\|\} > 0$. Then a sequence of solutions $\{p_k\}$ to subproblems (4.1) with $\mathcal{S}_k = \text{span}\{z_k, w_k\}$ has the angle property.*

Proof. The proof is straightforward and is omitted here. □

For the STIR method, a natural extension of the condition (AS.4) necessary for second-order optimality is as follows:

(AS.5) Assume that p_k is a solution to $\min_{s\in\Re^n}\{\psi_k(s) : \|D_k s\| \leq \Delta_k, s \in \mathcal{S}_k\}$ and β^q and β_0^q are two positive constants. Then s_k satisfies $\psi_k(s_k) < \beta^q \psi_k^*[p_k]$, where $\|D_k s_k\| \leq \beta_0^q \Delta_k$ and $x_k + s_k \in \text{int}(\mathcal{F})$.

Theorem 3 formalizes the convergence properties of STIR. The proof is provided in the appendix.

THEOREM 3. *Let the level set $\mathcal{L} = \{x \in \Re^n : f(x) \leq f(x_0),\ x \in \mathcal{F}\}$ be compact and $f : \mathcal{F} \to \Re$ be twice continuously differentiable on $\mathcal{L}$. Let $\{x_k\}$ be the sequence generated by the STIR algorithm in Fig. 3. Then*

1. *If (AS.3) is satisfied, then the Kuhn–Tucker conditions are satisfied at every limit point.*
2. *Assume that both (AS.3) and (AS.5) are satisfied and $\hat{w}_k$ in Fig. 3 contains sufficient negative curvature information whenever $\hat{M}_k$ is indefinite (negative definite), i.e.,*

$$\frac{\hat{w}_k^T \hat{M}_k \hat{w}_k}{\|\hat{w}_k\|^2} \leq \max(-\epsilon_{nc}, \tau\lambda_{\min}(\hat{M}_k)),$$

with $\epsilon_{nc} > 0$ and $0 < \tau < 1$. Then

(a) *if every limit point of $\{x_k\}$ is nondegenerate, there is a limit point x_* at which both the first- and second-order necessary conditions are satisfied;*

(b) *if x_* is an isolated nondegenerate limit point, both the first- and the second-order necessary conditions are satisfied at x_*;*

(c) *if $\hat{M}_*$ is nonsingular for some limit point x_* of $\{x_k\}$ and $\hat{w}_k = \hat{s}_k^N$ whenever $\hat{M}_k$ is positive definite, $\hat{M}_*$ is positive definite, $\{x_k\}$ converges to x_*, all iterations are eventually successful, and $\{\Delta_k\}$ is bounded away from zero.*

The degeneracy definition is as in [1].

DEFINITION 4. *A point $x \in \mathcal{F}$ is* nondegenerate *if, for each index i,*

$$g(x)_i = 0 \implies l_i < x_i < u_i. \tag{4.4}$$

The STIR Method
Let $0 < \mu < \eta < 1, 0 < \Lambda_l < \Lambda_u$, and $\gamma_1 < 1 < \gamma_2$ be given. Let $x_0 \in \text{int}(\mathcal{F})$, $\Delta_0 < \Lambda_u$.

For $k = 0, 1, \ldots$

1. Compute f_k, g_k, D_k, H_k, and C_k; define the quadratic model
$$\psi_k(s) = g_k^T s + \tfrac{1}{2} s^T (H_k + C_k) s.$$
2. Compute a step s_k, with $x_k + s_k \in \text{int}(\mathcal{F})$, based on the subspace subproblem
$$\min_s \{\psi_k(s) : \ \|D_k s\|_2 \leq \Delta_k, \ s \in \mathcal{S}_k\},$$
where the subspace $\mathcal{S}_k$ is set up as below.
3. Compute
$$\rho_k = \frac{f(x_k + s_k) - f(x_k) + \frac{1}{2} s_k^T C_k s_k}{\psi_k(s_k)}.$$
4. If $\rho_k > \mu$ then set $x_{k+1} = x_k + s_k$. Otherwise set $x_{k+1} = x_k$.
5. Update Δ_k as specified in Fig. 1.

Determine Subspace $\mathcal{S}_k$
Assume that $w_k = D_k^{-1} \hat{w}_k$ where $\{w_k\}$ has the angle property. Let $0 < \tau < 1$ be a small positive constant.

```
IF $\hat{M}_k$ is positive definite
    $\mathcal{S}_k \overset{\text{def}}{=} \text{span}\{D_k^{-2} g_k, w_k\}$
ELSE $\hat{M}_k$ is not positive definite
    IF $(D_k^{-2}\text{sgn}(g_k))^T M_k (D_k^{-2}\text{sgn}(g_k)) < \tau \frac{\|D_k^{-2} g_k\|^2}{\|w_k\|^2} w_k^T M_k w_k$
        $\mathcal{S}_k \overset{\text{def}}{=} \text{span}\{D_k^{-2}\text{sgn}(g_k)\}$
    ELSE
        $\mathcal{S}_k \overset{\text{def}}{=} \text{span}\{D_k^{-2}\text{sgn}(g_k), w_k\}$
    END
END
```

FIG. 3. *The STIR method for minimization subject to bound constraints.*

We have established that in principle it is possible to replace the full-dimensional trust region subproblem with a two-dimensional variation. However, the equally strong convergence properties of STIR hinge on obtaining (guaranteed) sufficient negative curvature vectors with the angle property. We discuss this next.

5. Computing negative curvature vectors with the angle property. It is possible, in principle, to satisfy both the sufficient negative curvature requirement (4.2) and the angle property. Let u_k be a unit eigenvector of $\hat{M}_k$ corresponding to the most negative eigenvalue, i.e., $\hat{M}_k u_k = \lambda_{\min}(\hat{M}_k) u_k$. It is easily verified that for any convergent subsequence with $\lim_{k \to \infty} \lambda_{\min}(\hat{M}_k) < 0$, the sequence $\{D_k^{-1} u_k\}$ has the angle property.

However, it is not computationally feasible to compute the (exact) eigenvector

u_k. Therefore approximations and short cuts are in order. We next consider how to compute approximate eigenvectors with the angle property.

A good approximation to an eigenvector corresponding to an extreme eigenvalue can usually be obtained through a Lanczos process [15]. Using the Lanczos method for $\hat{M}_k$ with an initial vector $\hat{q}_k$, approximate eigenvectors at the jth step are computed in the Krylov space

$$\mathcal{K}(\hat{M}_k, \hat{q}_k, j) \overset{\text{def}}{=} \text{span}(\hat{q}_k, \hat{M}_k\hat{q}_k, \ldots, \hat{M}_k^{j-1}\hat{q}_k).$$

In the context of our algorithm, the vectors $D_k^{-1}\text{sgn}(g_k)$ or $D_k^{-1}g_k$ are natural choices for the initial vector $\hat{q}_k$ when applying the Lanczos method.

Our Key observation is as follows: If a sequence $\{D_k^{-1}\hat{q}_k\}$ has the angle property, then each sequence $\{D_k^{-1}\hat{M}_k^i\hat{q}_k\}$, $i = 1, \ldots j$ retains this property.

Now assume that $\hat{w}_k$ is the computed vector from the Lanczos method which contains the sufficient negative curvature information with respect to $\hat{M}_k$. It can be verified, based on the recurrence relation, that $\{D_k^{-1}\hat{q}_k^1, \ldots, D_k^{-1}\hat{q}_k^j\}$ all have the angle property if the Lanczos vectors $\{\hat{q}_k^1 \ldots, \hat{q}_k^j\}$ retain orthogonality. Since $\hat{w}_k$ is in the Krylov space $\mathcal{K}(\hat{M}_k, \hat{q}_k, j)$ it is clear that $\{w_k = D_k^{-1}\hat{w}_k\}$ has the angle property. In other words, in order to generate a negative curvature vector sequence with the angle property, orthogonality needs to be maintained in the Lanczos process. In practice maintaining orthogonality can be a delicate and expensive business [16].

A second (and cheaper) strategy is to employ a modified preconditioned conjugate gradient scheme, e.g., modified preconditioned conjugate gradient algorithm (MPCG) in Fig. 4. Unfortunately, this process is not guaranteed to generate sufficient negative curvature; nonetheless, as indicated in [7], the MPCG output will satisfy the angle property.

Finally we consider a modified Cholesky factorization, e.g., [17], to obtain a negative curvature vector.

LEMMA 5. *Assume that $\{\hat{M}_k\}$ is indefinite or negative definite and $\{d_k\}$ is obtained from the modified Cholesky method. Then the sequence $\{d_k = D_k^{-1}\hat{d}_k\}$ has the angle property under a nondegeneracy assumption.*

Proof. The negative curvature vector $\hat{d}_k = D_k d_k$ computed from the modified Cholesky method (see [17, p. 111]) satisfies

$$L_k^T\hat{d}_k = e_{j_k} \quad \text{and} \quad P_k^T\hat{M}_k P_k + E_k = L_k\,\text{diag}(\delta_k)L_k^T,$$

where L_k is a lower triangular matrix, P_k is a permutation matrix, and e_{j_k} is the j_kth elementary vector, i.e., $e_{j_k}(i) = 0$, if $i \neq j_k$ and $e_{j_k}(j_k) = 1$. Moreover, E_k is a bounded and nonnegative diagonal matrix. Without loss of generality, we assume that $P_k = I$.

We argue, by contradiction, that $\{d_k\}$ has the angle property. Assume that $\{d_k\}$ does not have this property. From $L_k^T\hat{d}_k = e_{j_k}$ and the fact that L_k is a lower triangular matrix with unit diagonals, it is clear[1] that $\hat{d}_k(j_k{+}1 : n) = 0$. Moreover, from $\hat{M}_k\hat{d}_k + E_k\hat{d}_k = \delta_{k_{j_k}}L_k e_{j_k}$, $\delta_k > \delta$ for some $\delta > 0$ and definition (2.4) of $\hat{M}_k$, the first $j_k - 1$ components of $\{D_k\hat{d}_k\}$ are bounded. This implies that $\{v_{j_k}\}$ converges to zero.

[1] We use MATLAB [18] notation to specify submatrices (subvectors). In particular, if $i \leq j$, $k \leq l$ are positive integers then $A(i : j, k : l)$ denotes the submatrix of matrix A defined by rows $i, i+1, \ldots, j$ and columns $k, k+1, \ldots, l$.

```
function $[p, d, posdef] = MPCG(H, g, P, \eta)$
% Note: $P$ is some preconditioning matrix for H.
% $P$ must be positive definite. $P = R^2$.
% $\epsilon$ is a small positive constant.
% $\eta$ is the relative residual stopping tolerance.
$n = length(g)$;
$kmax = n/2$; $posdef = 1$;
$k = 0$; $p_0 = 0$; $r_0 = -g$;
while $k < kmax$
        Step 1:  Solve $Pz_k = r_k$
        Step 2:  $k = k + 1$
        Step 3:  if $k = 1$
                     $d_1 = z_0$
                 else
                     $\beta_k = r_{k-1}^T z_{k-1} / r_{k-2}^T z_{k-2}$
                     $d_k = z_{k-1} + \beta_k d_{k-1}$
                 end
        Step 4:  $\gamma_k = d_k^T H d_k$
        Step 5:  if $\gamma_k \le \epsilon(d_k^T P d_k)$
                     $d = d_k$, $p = p_{k-1}$, $posdef = 0$, return
                 else
                     $\alpha_k = r_{k-1}^T z_{k-1} / \gamma_k$
                     $p_k = p_{k-1} + \alpha_k d_k$
                     $r_k = r_{k-1} - \alpha_k H d_k$
                 end
        Step 6:  if $\|R^{-1}\| \cdot \|r_k\| \le \eta \|R^{-1} g\|$
                     $p = p_k$, $d = 0$, return
                 end
end
$p = p_k$, $d = 0$, return
```

FIG. 4. *Modified preconditioned conjugate gradient algorithm.*

From the modified Cholesky factorization, the matrix $\hat{M}_k(1\!:\!j_k,\ 1\!:\!j_k)$ is indefinite (negative definite) but $\hat{M}_k(1\!:\!j_k\!-\!1,\ 1\!:\!j_k\!-\!1)$ is positive definite. However, this is impossible for sufficiently large k because, again using the definition (2.4) of $\hat{M}_k$, $\{\hat{M}_k(1\!:\!j_k,\ 1\!:\!j_k)\}$ converges to a matrix of the form

$$\begin{bmatrix} \hat{M}_k(1\!:\!j_k\!-\!1, 1\!:\!j_k\!-\!1) & 0 \\ 0 & \eta_k \end{bmatrix},$$

where η_k is positive (because of the nondegeneracy assumption). Therefore, we conclude that $\{d_k\}$ has the angle property. □

6. Computational experience. We demonstrate the computational performance of our STIR method given in Fig. 3. Below we report our experience with the modified Cholesky and the conjugate gradient (MPCG) implementations. We examine the sensitivity of the STIR method to a starting point. Finally, some limited

Updating Trust Region Size Δ_k

Let $\mu = 0.25$, $\eta = 0.75$, $\Lambda_l = 1$, $\Lambda_u = \max(\sqrt{\sum_i \min((u_i - l_i)^2, 1000)},\ 1)$, $\Delta_0 = \min(0.1\|g\|, \Lambda_u)$, $\gamma_0 = 0.0625$, $\gamma_1 = 0.5, \gamma_2 = 2$ be given.

```
1. If $\rho_k \leq 0$ then set $\Delta_{k+1} = \gamma_0 \Delta$.
2. If $\rho_k \in (0, \mu]$ then set $\Delta_{k+1} = \max(\gamma_0 \Delta_k, \gamma_1 \|D_k s_k\|)$.
3. If $\rho_k \in (\mu, \eta)$ then set $\Delta_{k+1} = \Delta_k$.
4. If $\rho_k \geq \eta$ then
       if $\Delta_k > \Lambda_l$
           $\Delta_{k+1} = \gamma_2 \Delta_k$
       otherwise
           $\Delta_{k+1} = \min(\max(\Delta_k, \gamma_2 \|D_k s_k\|), \Lambda_u)$.
```

FIG. 5. *Updating trust region size.*

comparisons with SBMIN of LANCELOT [10] are also made.

In the implementation of STIR, we compute s_k using a reflective technique as shown in Fig. 2. The exact trust region updating procedure is given in Fig. 5.

Our experiments were carried out on a Sun SPARC workstation using MATLAB [18].

The stopping criteria used are as follows. We stop if

either $f(x_k) - f(x_{k+1}) \leq \tau_1(1 + |f(x_k)|)$
or $\|x_{k+1} - x_k\|_2 \leq \tau_2$
or no negative curvature has been detected for $\hat{M}_k$ and $\|D_k^{-2} g_k\|_\infty \leq \tau_1$.

We define $\tau_1 = 10^{-10}$ and $\tau_2 = \sqrt{\tau_1}/10 = 10^{-6}$. We also impose an upper bound of 600 on the number of minimization iterations.

We first report the results of the STIR method using the modified Cholesky factorization. Table 3 lists the number of major iterations required for some standard test problems (for details of these problems, see [19]). (For all the results in this paper, the number of iterations is the same as the number of objective function evaluations.) The problem sizes vary from 100 to 10,000. The results in Table 3 indicate that, for these test problems at least, the number of iterations increases only slightly, if at all, with the problem size. This is true whether the solution is unconstrained (e.g., VAR U) or constrained (e.g., VAR C). This is depicted pictorially in Fig. 6. In this graph, the problem size is plotted versus iteration count. For each problem the corresponding points have been connected to show how the iteration count relates to the problem size.

Our second set of results are for the STIR algorithm, using a conjugate gradient implementation. We use algorithm MPCG in Fig. 4 to find the vectors needed to form the subspace $\mathcal{S}_k$. The stopping condition applied to the relative residual in MPCG is $\eta = 0.005$. The results are shown in Table 4 and Fig. 7. Again, for these problems the iteration counts are low and steady. The exception is for the problem VAR C with 10,000 variables, where the iteration count jumps to 86. This is one of several degenerate problems included in this test set. With a tighter bound η on the relative residual in MPCG, we could decrease the number of minimization iterations for this problem; for example, using $\eta = 0.0005$ the problem VAR C with 10,000 variables takes 57 minimization iterations as opposed to 86 iterations.

Next we include some results which indicate that our STIR method is fairly insensitive to the starting point (insensitive in the sense that the iteration counts do

TABLE 3
STIR method with exact Newton steps: number of iterations.

Problem	*n* 100	200	500	1000	10000
GENROSE U	25	25	25	25	25
GENROSE C	11	11	11	11	10
GENSING U	24	25	25	26	27
GENSING C	18	19	20	20	21
CHAINSING U	23	23	23	23	23
CHAINSING C	16	16	16	16	19
DEGENSING U	22	23	23	40	39
DEGENSING C	28	28	28	28	29
GENWOOD C	9	10	10	10	11
CHAINWOOD C	9	10	10	10	11
BROYDEN1A U	12	12	13	13	14
BROYDEN1A C	11	11	11	11	11
BROYDEN1B U	7	7	7	7	7
BROYDEN1B C	8	8	8	8	8
BROYDEN2A U	13	13	13	14	14
BROYDEN2A C	14	19	17	19	19
BROYDEN2B U	9	9	9	9	9
BROYDEN2B C	13	11	15	14	15
TOINTBROY U	8	8	8	8	8
TOINTBROY C	9	9	9	9	9
CRAGGLEVY U	16	14	15	16	15
CRAGGLEVY C	29	29	30	30	31
AUGMLAGN C	38	32	35	36	37
BROWN3 U	8	8	8	8	8
BROWN3 C	17	10	11	9	11
BVP U	9	10	9	8	8
BVP C	11	11	10	10	7
VAR U	9	9	10	12	15
VAR C	18	18	23	45	38

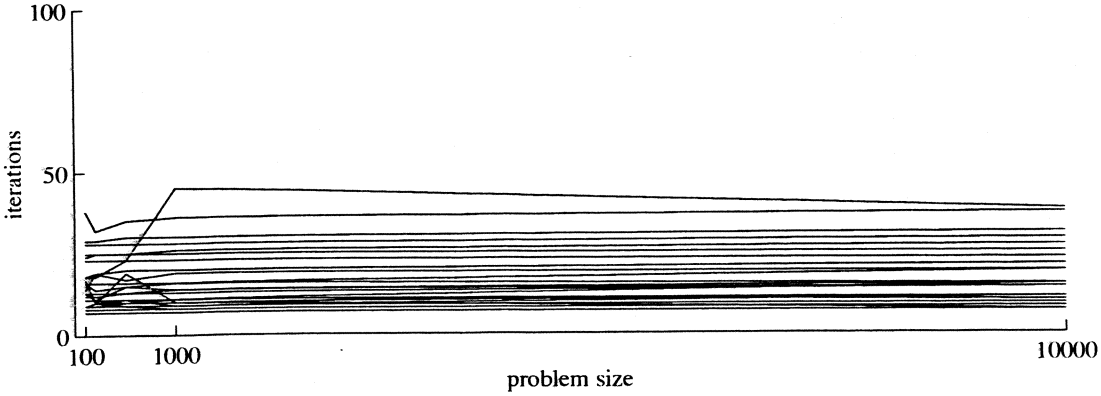

FIG. 6. *STIR performance with exact Newton steps.*

not appear correlated to the problem size for any start). The results in Table 5 were obtained using exact Newton steps on problems of dimension 1000. The results in Table 6 were obtained using the conjugate gradient implementation, also on problems with 1000 variables. The starting points are as follows: *original* is the suggested starting point according to [19]; *upper* starts all variables at upper bounds; *lower*

TABLE 4
STIR method with inexact Newton steps, $\|r\|/\|g\| \leq 0.005$: *number of iterations.*

	n				
Problem	100	200	500	1000	10000
GENROSE U	21	21	21	21	21
GENROSE C	10	10	10	10	17
GENSING U	23	23	24	24	25
GENSING C	16	16	16	16	16
CHAINSING U	21	21	21	21	21
CHAINSING C	14	17	19	19	20
DEGENSING U	32	32	33	33	35
DEGENSING C	33	56	35	33	31
GENWOOD C	8	8	8	8	8
CHAINWOOD C	8	8	8	8	8
BROYDEN1A U	11	11	11	11	12
BROYDEN1A C	9	8	8	8	8
BROYDEN1B U	6	6	6	6	6
BROYDEN1B C	7	7	7	7	7
BROYDEN2A U	15	15	19	17	20
BROYDEN2A C	10	10	10	10	10
BROYDEN2B U	8	8	8	8	9
BROYDEN2B C	9	9	9	9	9
TOINTBROY U	7	7	7	7	7
TOINTBROY C	8	8	8	8	8
CRAGGLEVY U	26	26	27	27	29
CRAGGLEVY C	26	26	26	26	27
AUGMLAGN C	26	33	29	34	27
BROWN3 U	7	7	7	7	7
BROWN3 C	7	7	7	7	8
BVP U	13	13	12	13	25
BVP C	15	15	14	14	15
VAR U	34	35	35	37	36
VAR C	19	21	32	36	86

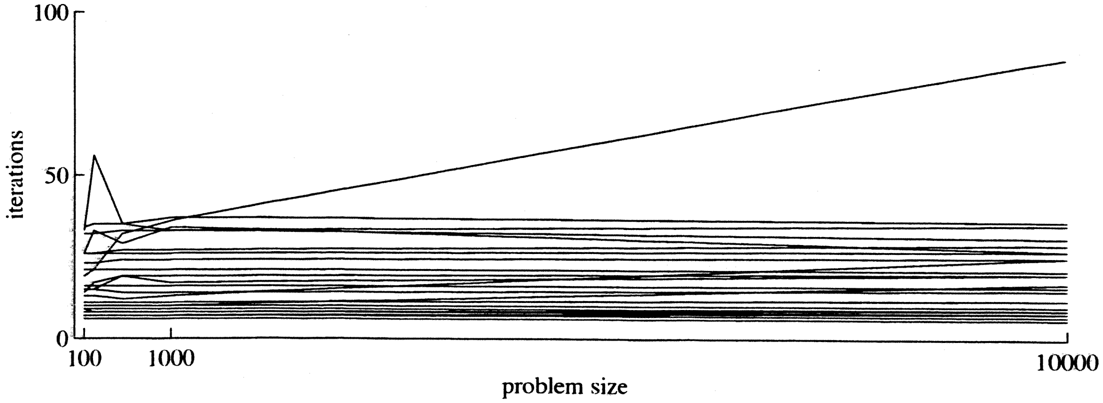

FIG. 7. *STIR method with inexact Newton steps.*

starts all variables at the lower bounds; *middle* starts at the midpoint between bounds; *zero* starts each variable at zero (the origin); *upper-lower* starts the odd variables at the upper and the even variables at the lower bounds; *lower-upper* is the reverse of this. For all of these, we perturb the starting point slightly if necessary to be strictly feasible. Note that for the problem BROWN3 C, the iteration count is not shown starting at *middle* and at *origin* as the gradient is undefined at both these starting

TABLE 5

STIR method with exact Newton steps for n = 1000: *number of iterations (* denotes gradient undefined at starting point).*

	Starting Point						
Problem	Original	Upper	Lower	Middle	Zero	Up-low	Low-up
GENROSE C	11	27	33	15	16	43	27
GENSING C	20	31	45	25	22	31	32
CHAINSING C	16	29	33	13	11	32	30
GENWOOD C	10	18	14	13	10	17	17
CHAINWOOD C	10	17	14	13	10	17	16
BROYDEN1A C	11	24	25	13	12	25	24
BROYDEN1B C	8	22	19	18	9	19	21
BROYDEN2A C	19	38	38	13	9	38	38
BROYDEN2B C	14	30	34	12	8	33	30
BROWN3 C	9	28	14	*	*	28	14
BVP C	10	17	8	9	10	11	17
VAR C	45	9	32	18	21	23	17

TABLE 6

STIR method with inexact Newton steps for n = 1000: *number of iterations (* denotes gradient undefined at starting point).*

	Starting Point						
Problem	Original	Upper	Lower	Middle	Zero	Up-low	Low-up
GENROSE C	10	23	37	17	20	37	23
GENSING C	16	27	57	26	22	30	29
CHAINSING C	19	29	33	11	10	33	28
GENWOOD C	8	14	10	11	8	13	13
CHAINWOOD C	8	14	10	11	8	13	13
BROYDEN1A C	8	24	21	13	8	21	24
BROYDEN1B C	7	21	16	13	8	16	21
BROYDEN2A C	10	35	35	13	8	36	35
BROYDEN2B C	9	28	32	12	8	31	28
BROWN3 C	7	29	53	*	*	29	53
BVP C	14	21	14	13	14	14	21
VAR C	36	7	34	29	25	28	8

points.

For all but four problems, the same solution point was found for all starting points. For the four problems BROYDEN1A C, BROYDEN1B C, BVP C, and VAR C, two different minima were found for each problem. For BROYDEN1A C, BROYDEN1B C, and VAR C, the solution points were the same except for when started at *upper* and *lower-upper*, which resulted in a second solution. For BVP C, a second local minimum was found when started at *lower* and *upper-lower*. The solutions found were consistent between the exact and inexact methods; for example, the same solution was found for BVP C when started at *lower* whether exact or inexact STIR was used.

The results in Tables 5 and 6 are also shown graphically in Figs. 8 and 9. From these graphs it is clear that both implementations of STIR are fairly robust when it comes to starting points. This is in contrast to active set methods where the starting point can have a more dramatic effect on the iteration count.

Last we contrast the performance of the STIR method using the conjugate gradient option with the SBMIN algorithm, an active set method, in the LANCELOT software package [10]. In particular, we choose problems where negative curvature is present or where it appears that the "active set" at the solution may be difficult

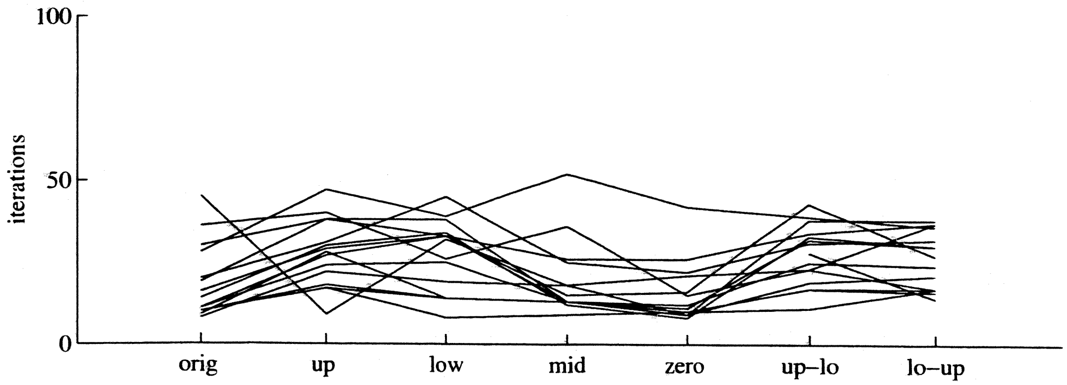

FIG. 8. *STIR method with exact Newton steps at varied starting points.*

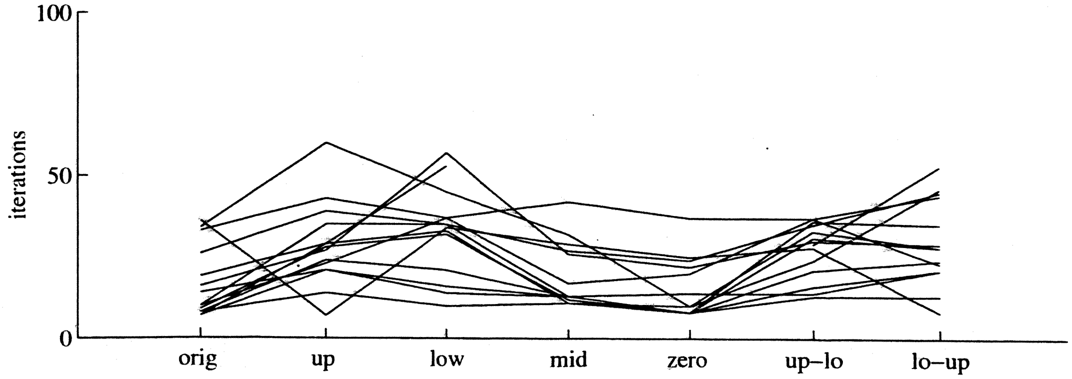

FIG. 9. *STIR method with inexact Newton steps at varied starting points.*

to find. We expect our STIR method to outperform an active set method in these situations; indeed, we have found this to be the case. For these problems, we used the following SPEC.SPC file

```
exact-second-derivatives-used
diagonal-preconditioned-cg-solver-used
gradient-tolerance 1.0D-6
```

which uses the default settings for LANCELOT, except we provide exact second derivatives, we use a diagonal preconditioner as opposed to a bandsolver preconditioner of semibandwidth 5 (because we also use a diagonal preconditioner with STIR), and we use a gradient tolerance of 1.0e-6 as opposed to 1.0e-5. (The use of a diagonal preconditioner seems to be more illustrative as far as the iterative nature of the methods. However, we note that LANCELOT takes many fewer iterations, as does the STIR method when used with the type of preconditioner that is LANCELOT's default.) We adjusted our STIR stopping conditions to be comparable if not more stringent: We stop if

either no negative curvature has been detected for $\hat{M}_k$ and $\hat{\psi}_k(\hat{s}) > -5.0e-12$,
or no negative curvature has been detected for $\hat{M}_k$ and $\|D_k^{-2}g_k\|_\infty < 1e-6$.

TABLE 7
STIR with inexact Newton steps vs. LANCELOT SBMIN on a convex quadratic: number of iterations.

	Inexact STIR		SBMIN	
$n = 800$	iteration	c.g. it	iteration	c.g. it
BIGGSB2	16	5451	243	40846

TABLE 8
STIR with inexact Newton steps versus LANCELOT SBMIN when negative curvature exists: number of iterations.

	Inexact STIR			SBMIN		
		n			n	
Problem	100	1000	10000	100	1000	10000
AUGMLAGN U	21	22	22	29	38	46
CHAINWOOD U	151	935	9298	6594	> 10000	> 10000
GENWOOD U	77	92	91	439	952	554
GENROSE U	21	21	21	76	76	76
CHAINWOOD NC	17	28	21	54	48	61
GENWOOD NC	13	25	24	47	1136	60

First consider a constrained convex quadratic problem.[2] The results, given in Table 7, show that our proposed STIR method is superior (by an order of magnitude) to SBMIN on this problem (e.g., it is the total number of conjugate gradient iterations). SBMIN takes many iterations on this problem when the starting point is near some of the bounds—the method misidentifies the correct active set at the solution and takes many iterations to recover. Our proposed STIR method, a strictly interior method, moves directly to the solution without faltering when started at the same point. For this problem, the STIR and SBMIN solution points agreed to within 7.8e-05, and within 7.2e-07 in the function values at these solutions.

Table 8 summarizes the performances of STIR and SBMIN on a set of constrained problems exhibiting negative curvature. (Again the problems are from [19], except the last two have been constrained differently to display negative curvature: the bounds are set to $-0.1 \leq x_i \leq 0.9 : \ i = 1, 3, 5, \ldots, n$ instead of $1.1 \leq x_i \leq 2.1 : \ i = 1, 3, 5, \ldots, n$; the start point is $(0, -1, 0, -1, 0, 0, \ldots, 0)$ not $(-3, -1, -3, -1, -2, 0, -2, 0, \ldots, -2, 0)$.) For all these problems except CHAINWOOD U, the STIR and SBMIN solution points agreed to within 1.5e-06 and within 1.5e-06 in the function values at these solutions. CHAINWOOD U is an exception. When the problem size is 100, SBMIN and STIR converge to two different solutions where the function values are 4.6713e+01 and 4.5743e+00, respectively. For problem size 1000 and 10000, SBMIN did not converge so solution comparison is not possible. STIR is significantly better on these problems—probably because negative curvature is better exploited in our subspace trust region approach than in the Steihaug trust region method, which SBMIN employs. This is consistent with the remarks in section 3 and [7, 8].

[2] BIGGSB2 is a modification of problem BIGGSB1 [20]. Active set methods, such as LANCELOT, are likely to struggle with the identification of the correct active set on this modified version (from the given starting point). BIGGSB2: $f(x) = (x_1 - 1)^2 + (1 - x_n)^2 + \sum_{i=1}^{n-1}((x_{i+1} - x_i)^2 + 0.00001x_i)$, where $0.0 \leq 0.9$ for $i = 1$ and x_n is free. The starting point is $x(i) = 0.01$ for $i = 1$ to n. The problem size is $n = 800$.

7. Conclusion. Based on the TIR method in [1], we have proposed a *subspace* TIR method (STIR) suitable for large-scale minimization with bound constraints on the variables. In particular, we consider a two-dimensional STIR in which a subspace is formed from the scaled gradient and (inexact or exact) Newton steps or a negative curvature vector.

We have designed and reported on a variety of computational experiments. The results strongly support the different components of our approach: the "subspace idea," the novel affine scaling matrix, the modified Cholesky factorization and conjugate gradient variations, and the "reflection technique." Moreover, preliminary experimental comparisons with code SBMIN, from LANCELOT [10], indicate that our proposed STIR method performs as well as, and can outperform, an active set approach for some large-scale problems.

Appendix A. The convergence results (Theorem 2) for the STIR algorithm can be obtained in a manner similar to Theorem 3.10 for the full-space trust region and interior reflective method (TIR) [1]. Indeed, first-order optimality is a direct consequence of the condition (AS.3). The second-order optimality rests on the fact that the solution subsequence of the subspace trust region subproblem would have the angle property if the corresponding $\{\hat{M}_*\}$ were indefinite (negative definite) at a limit point (see Lemma 6, below). Moreover, if $\hat{M}_k$ is positive definite at a limit point then we prove that the stepsize along the subspace trust region solution is sufficiently large in the following sense:

$$\alpha_k \geq \frac{\chi_\alpha}{\nabla_k} \min(\|D_k^{-1} g_k\|, \|D_k s_k^N\|) \qquad \text{for some } \chi_\alpha > 0.$$

Here α_k is the stepsize, along p_k, to the boundary of the feasible region (see Lemma 6). On the basis of this inequality, it follows that the trust region size is bounded away from zero and Newton steps are eventually successful.

Assume that p_k is a solution to a subspace trust region subproblem (4.1) with $S_k = \text{span}\{w_k, z_k\}$. Assume that the columns of Y_k form an orthonormal basis for span $\{D_k z_k, D_k w_k\}$. Then $p_k = D_k^{-1} Y_k y_k$ where y_k solves

$$(A.1) \qquad (Y_k^T \hat{M}_k Y_k + \lambda_k I) y_k = -Y_k^T D_k^{-1} g_k, \qquad Y_k^T \hat{M}_k Y_k + \lambda_k I = R_k^T R_k,$$

and

$$(A.2) \qquad (D_k p_k)^T \hat{M}_k D_k p_k + \lambda_k \|y_k\|^2 = -(D_k p_k)^T \hat{g}_k.$$

Next we prove that the subspace trust region solution sequence from the STIR algorithm in Fig. 3 has the angle property if the corresponding sequence $\{\hat{M}_k\}$ satisfies that $\lim_{k \to \infty} \lambda_{\min}(\hat{M}_k) < 0$.

LEMMA 6. *If lim sup $\lambda_{min}(\hat{M}_k) < 0$ for a subsequence, then the corresponding solution subsequence $\{p_k\}$ of the subspace trust region subproblem* (4.1) *has the angle property.*

Proof. Consider two subsequences of $\{\hat{M}_k\}$: one sequence satisfies $|S_k| = 1$ and the other sequence has $|S_k| = 2$.

For the subsequence with $|S_k| = 1$, the corresponding trust region solution sequence clearly has the angle property.

For the subsequence with $|S_k| = 2$, it is clear that

$$\liminf \left\{ \left\| \frac{D_k^{-2} \text{sgn}(g_k)}{\|D_k^{-2} \text{sgn}(g_k)\|} - \frac{w_k}{\|w_k\|} \right\| \right\} > 0.$$

Since $\{z_k\} = \{D_k^{-2}\mathrm{sgn}(g_k)\}$ and $\{w_k\}$ have the angle property, we have that $\{p_k\}$ has the angle property following Lemma 2. □

We state the following result, which is similar to Lemma 8 in [1], and omit the proof.

LEMMA 7. *Assume that* (AS.4) *is satisfied. Then*

$$-\psi_k(s_k) \geq \frac{\beta^q}{2}[\min\{1, \alpha_k^2\}\lambda_k\Delta_k^2 + \min\{1, \alpha_k\}\|R_k y_k\|^2],$$

where α_k is the stepsize along $p_k = D_k^{-1}Y_k y_k$ to the boundary and y_k is defined by (A.1).

Let s_k^N denote the Newton step (2.3) of (2.1). Then

$$\mathrm{diag}(g_k)J_k^u D_k s_k^N = -D_k^{-1}gk - D_k^{-1}H_k D_k^{-1}D_k s_k^N. \tag{A.3}$$

The next result is required to establish that Newton steps s_k^N will eventually lead to successful steps.

LEMMA 8. *Assume that p_k is a solution to the subspace subproblem* (4.1) *with $\mathcal{S}_k = \mathrm{span}\{D_k^{-2}gk, s_k^N\}$. If $\{x_k\}$ converges to a nondegenerate point x_* where the second-order sufficiency conditions are satisfied, then*

$$\alpha_k \leq \frac{X_a}{\Delta_k} min(\|D_k^{-1}g_k\|, \|D_k s_k^N\|) \tag{A.4}$$

for k sufficiently large, where α_k is the stepsize to the boundary along p_k.

Proof. By definition

$$\alpha_k = \min_i \left(\max \left(\frac{l_{ki} - x_{ki}}{p_{ki}}, \frac{u_{ki} - x_{ki}}{p_{ki}} \right) \right).$$

For any k, if $D_k^{-1}gk = \rho_k D_k s_k^N$ for some $\rho_k \in \Re^1$, then $p_k = -\frac{\Delta_k}{\|D_k^{-1}gk\|}D_k^{-2}gk$. Hence if $\frac{1}{\chi_\alpha} \geq \|g_k\|_\infty$ we have

$$\alpha_k \geq \frac{\chi_\alpha}{\Delta_k}\min(\|D_k^{-1}gk\|, \|D_k s_k^N\|).$$

Assume that $D_k^{-1}gk \neq \rho_k D_k s_k^N$. We first show that if we can establish

$$p_k = \gamma_k(-D_k^{-2}gk) + \beta_k s_k^N, \tag{A.5}$$

where $\beta_k \geq 0$ and $\gamma_k \geq 0$, then (A.4) holds. From (A.5) and $(D_k s_k^N)^T(-D_k^{-1}gk) \geq 0$, we have

$$(\beta_k D_k s_k^N)^T(\gamma_k(-D_k^{-1}gk)) \geq 0.$$

Using $\hat{p}_k = D_k p_k = \gamma_k(-D_k^{-1}gk) + \beta_k D_k s_k^N$, again

$$\|\hat{p}_k\|^2 = \gamma_k^2\|D_k^{-1}gk\|^2 + 2(\gamma_k D_k s_k^N)^T(\beta_k(-D_k^{-1}gk)) + \beta_k^2\|D_k s_k^N\|^2,$$

but $\|\hat{p}_k\| \leq \Delta_k$. Hence

$$0 \leq \beta_k \leq \frac{\Delta_k}{\|D_k^{-1}gk\|} \quad \text{and} \quad 0 \leq \gamma_k \leq \frac{\Delta_k}{\|D_k s_k^N\|}.$$

Hence, from (A.3), the boundedness of $g_k, D_k^{-1}gk, D_k s_k^N$, and the fact that x_* is a nondegenerate first-order point, it is easy to verify that

$$\alpha_k \geq \frac{X_\alpha}{\Delta_k}\min(\|D_k^{-1}g_k\|, \|D_k s_k^N\|) \text{ for some } \chi_\alpha > 0.$$

Finally, we need to establish (A.5) under the linearly independent assumption $D_k^{-1}gk \neq \rho_k D_k s_k^N$. Assume that the columns of Y_k form an orthonormal basis for span $\{D_k^{-1}gk, D_k s_k^N\}$. Then $Y_k Y_k^T D_k^{-1}gk = D_k^{-1}gk$, $Y_k Y_k^T D_k s_k^N = D_k s_k^N$, and $Y_k^T Y_k = I_2$ where I_2 is the 2-by-2 identity matrix. Moreover,

$$p_k = -D_k^{-1}Y_k[Y_k^T(\hat{M}_k + \lambda_* I)Y_k]^{-1}Y_k^T D_k^{-1}gk,$$

where $\lambda_* \geq 0$ and if $\lambda_* > 0, \|D_k p_k\| = \Delta_k > 0$. Let $\hat{p}(\lambda) = D_k p_k(\lambda)$ and

$$\hat{p}(\lambda) \overset{\text{def}}{=} -Y_k[Y_k^T(\hat{M}_k + \lambda I)Y_k]^{-1}Y_k^T D_k^{-1}gk \qquad \text{for } \lambda \geq 0. \tag{A.6}$$

Then there exists $\beta(\lambda)$ and $\gamma(\lambda)$ such that

$$\hat{p}(\lambda) = \beta(\lambda)(-D_k^{-1}gk) + \gamma(\lambda)D_k s_k^N.$$

First, it is clear that $\beta(0) = 0, \gamma(0) = 1$. From (A.6),

$$\lim_{\lambda \to +\infty} \frac{\hat{p}(\lambda)}{\|\hat{p}(\lambda)\|} = \frac{D_k^{-1}gk}{\|D_k^{-1}gk\|},$$

and by the linear independence assumption $D_k^{-1}gk \neq \rho_k D_k s_k^N$ we have

$$\lim_{\lambda \to +\infty} \frac{\beta(\lambda)}{\|\hat{p}(\lambda)\|} = \frac{1}{\|D_k^{-1}gk\|}.$$

Hence, for λ sufficiently large, $\beta(\lambda) > 0$.

We now prove that $\gamma(\lambda^*) \geq 0$ by contradiction. Assume that $\gamma(\lambda^*) < 0$ (this means that $\|D_k s_k^N\| > \Delta_k$). From continuity of $\gamma(\lambda), \gamma(0) = 1$ and $\gamma(\lambda^*) < 0$, there exists $0 < \bar{\lambda} < \lambda^*$ so that $\gamma(\bar{\lambda}) = 0$. This implies that

$$\beta(\bar{\lambda})D_k^{-1}gk = -Y_k(Y_k^T(\hat{M}_k + \bar{\lambda}I)Y_k)^{-1}Y_k^T D_k^{-1}gk.$$

From $\hat{M}_k \hat{s}_k^N = -D_k^{-1}gk, Y_k Y_k^T \hat{s}_k^N = \hat{s}_k^N$, and the fact that the columns of Y_k are linearly independent, there exists χ such that

$$[Y_k^T \hat{M}_k Y_k]Y_k^T D_k^{-1}gk = \chi[Y_k^T \hat{M}_k Y_k]Y_k^T \hat{s}_k^N.$$

Again using $Y_k Y_k^T D_k^{-1}gk = D_k^{-1}gk$ and $Y_k Y_k^T \hat{s}_k^N = \hat{s}_k^N$, we have

$$D_k^{-1}gk = \chi D_k s_k^N,$$

which contradicts the assumption $D_k^{-1}g_k \neq \rho_k D_k s_k^N$.

Similarly, we can prove that $\beta(\lambda^*) \geq 0$ based on $\beta(\lambda) > 0$ for sufficiently large λ. Therefore (A.5) holds. This completes the proof. □

Now we establish the convergence properties of the STIR algorithm.

THEOREM 9. *Let the level set $\mathcal{L} = \{x \in \Re^n : f(x) \leq f(x_0),\ x \in \mathcal{F}\}$ be compact and $f : \mathcal{F} \to \Re$ be twice continuously differentiable on $\mathcal{L}$. Let $\{x_k\}$ be the sequence generated by the STIR algorithm in* Fig. 3. *Then the following hold:*

1. *If* (AS.3) *is satisfied, the Kuhn–Tucker condition is satisfied at every limit point.*
2. *Assume that both* (AS.3) *and* (AS.5) *are satisfied and* $\hat{w}_k$ *in Fig.* 3 *contains sufficient negative curvature information whenever* $\hat{M}_k$ *is indefinite* (*negative definite*), *i.e.*,

$$\frac{\hat{w}_k^T \hat{M}_k \hat{w}_k}{\|\hat{w}_k\|^2} \leq \max(-\epsilon_{nc}, \tau \lambda_{\min}(\hat{M}_k)),$$

with $\epsilon_{nc} > 0$ *and* $0 < \tau < 1$. *Then the following hold:*

(a) *If every limit point is nondegenerate, there is a limit point* x_* *at which both the first- and second-order necessary conditions are satisfied.*

(b) *If* x_* *is an isolated nondegenerate limit point, both the first- and second-order necessary conditions are satisfied at* x_*.

(c) *If* $\hat{M}_*$ *is nonsingular for some limit point* x_* *of* $\{x_k\}$ *and* $\hat{w}_k = \hat{s}_k^N$ *whenever* $\hat{M}_k$ *is positive definite,* $\hat{M}_*$ *is positive definite,* $\{x_k\}$ *converges to* x_*, *all iterations are eventually successful, and* $\{\Delta_k\}$ *is bounded away from zero.*

Proof. Using Lemma 6, for any subsequence with $\lim_{k\to\infty} \lambda_{\min}(\hat{M}_k) < 0$, the corresponding $\{p_k\}$ has the angle property. Therefore there exists $\epsilon_0 > 0$ such that $\alpha_k > \epsilon_0$ for k sufficiently large. Hence using Lemma 7, for some $\epsilon_1 > 0$,

$$-\psi_k(s_k) \geq \epsilon_1 \frac{\beta^q}{2} [\lambda_k \nabla_k^2 + \|R_k D_k p_k\|^2].$$

Condition (AS.3) then implies that

$$f(x_k) - f(x_{k+1}) \geq \mu \epsilon_1 \frac{\beta^q}{2} [\lambda_k \Delta_k^2 + \|R_k D_k p_k\ |^2]. \tag{A.7}$$

Now assume that $\hat{M}_*$ is positive definite and $\{x_k\}$ converges to x_*. From Lemma 7 we have

$$-\psi_k(s_k) \geq \frac{\beta^q}{2} [\min\{1, \alpha_k^2\} \lambda_k \Delta_k^2 + \min\{1, \alpha_k\} \|R_k y_k\|^2],$$

where α_k is the stepsize along p_k. Let $\epsilon > 0$ be a lower bound for the eigenvalues of $\hat{M}_k$.

From (A.1), (A.4) in Lemma 8, and $\|D_k s_k^N\| \leq \frac{1}{\epsilon} \|D_k^{-1} g_k\|$, there exists $\chi > 0$ such that

$$|\psi_k^*[p_k]| \geq \chi \min\{\Delta_k^2, \|D_k s_k^N\|^2\}. \tag{A.8}$$

Using (A.7) and (A.8), the proof is essentially the same as that of Theorem 3.10 in [1]: replacing (3.21) in [1] by (A.7) and (3.22) in [1] by (A.8). □

Acknowledgments. We thank the referees and the associate editor for their suggestions, which led to improvements on the original version of this paper.

REFERENCES

[1] 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.

[2] T. F. Coleman and Y. Li, *On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to bounds*, Math. Programming, 67 (1994), pp. 189–224.

[3] T. F. Coleman and Y. Li, *A reflective Newton method for minimizing a quadratic function subject to bounds on some of the variables*, SIAM J. Optim., 6 (1996), pp. 1040–1058.

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

[5] D. C. Sorensen, *Minimization of a Large scale Quadratic Function Subject to an Ellipsoidal Constraint*, Technical report TR94-27, Department of Computational and Applied Mathematics, Rice University, Houston, TX, 1994.

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

[7] M. A. Branch, *Inexact Reflective Newton Methods for Large-Scale Optimization Subject to Bound Constraints*, Ph.D. thesis, Department of Computer Science, Cornell University, Ithaca, NY, 1996.

[8] M. A. Branch, T. F. Coleman, and Y. Li, *A Computational Study of Two Approximations to the Trust-Region Subproblem*, Technical report, Cornell University, Ithaca, NY, 1998, in progress.

[9] I. I. Dikin, *Iterative solution of problems of linear and quadratic programming*, Dokl. Akad. Nauk SSSR, 174 (1967), pp. 747–748.

[10] A. R. Conn, N. I. M. Gould, and P. L. Toint. *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*, Springer-Verlag, Berlin, New York, 1992.

[11] B. A. Hendrickson, *The Molecule Problem: Determining Conformation from Pairwise Distances*, Ph.D. thesis, Cornell University, Ithaca, NY, 1991.

[12] T. F. Coleman, *Large-scale numerical optimization: Introduction and overview*, in Encyclopedia of Computer Science and Technology, A. Kent and J. G. Williams, eds., Marcel Dekker, New York, 1993, pp. 167–195.

[13] G. A. Shultz, R. B. Schnabel, and R. H. Byrd, *A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties*, SIAM J. Numer. Anal., 22 (1985), pp. 47–67.

[14] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, *Approximate solution of the trust region problem by minimization over two-dimensional subspaces*, Math. Programming, 40 (1988), pp. 247–263.

[15] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, MD, 1989.

[16] J. K. Cullum and R. A. Willoughby, *Lanczos Algorithms for Large symmetric eigenvalue computations. Vol. 1: Theory*, Birkhaüser, Boston-Basel-Stuttgart, 1985.

[17] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, New York, 1981.

[18] *MATLAB* 4.2c *for Unix*, The MathWorks, Natick, MA.

[19] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Testing a class of methods for solving minimization problems with simple bounds on the variables*, Math. Comput., 50 (1988), pp. 399–430.

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