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-scale test problems are reported; advantages of our approach are demonstrated. Our experience indicates our proposed method represents an efficient way to solve large-scale bound-constrained minimization problems.

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

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.,

$$\min_{x \in \Re^n} f(x), \;\; l \leq x \leq u, \tag{1.1}$$

where $l \in \{\Re \cup \{-\infty\}\}^n$, $u \in \{\Re \cup \{\infty\}\}^n$, $l < u$, and f is a smooth function, $f : \Re^n \rightarrow \Re^1$. 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]. Our main objective here is to investigate solving large-scale bound-constrained nonlinear minimization problems (1.1), using a large-scale adaptation of the Trust-region Interior Reflective (TIR) approach proposed in [1].

The TIR method [1], outlined in FIG. 1, elegantly generalizes the trust region idea for unconstrained minimization to bound-constrained nonlinear minimization. Here $g_k \overset{\text{def}}{=} \nabla f_k$, $H_k \overset{\text{def}}{=} \nabla^2 f_k$. The crucial role of the (diagonal) affine scaling matrices D_k and C_k will become clear in §2.

An attractive feature of the TIR method [1] is that the main computation per iteration is solving a

[*] Research 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, and by NSF, AFOSR, and ONR through grant DMS-8920550, and by the Advanced Computing Research Institute, a unit of the Cornell Theory Center which receives major funding from the National Science Foundation and IBM Corporation, with additional support from New York State and members of its Corporate Research Institute.

[†] Computer Science Department, Cornell University, Ithaca, NY 14850.

[‡] Computer Science Department and Center for Applied Mathematics, Cornell University, Ithaca NY 14850.

The TIR Method [1]

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

For $k = 0, 1, \cdots$

    1. Compute $f_k$, $g_k$, $D_k$, $H_k$, and $C_k$; define the quadratic model
           $\psi_k(s) \overset{\text{def}}{=} g_k^T s + \frac{1}{2} s^T (H_k + C_k) s.$
    2. Compute a step $s_k$, with $x_k + s_k \in int(\mathcal{F})$, based on the subproblem:
           $\min_s \{\psi_k(s) : \ \|D_k s\|_2 \leq \Delta_k\}.$
    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 $\Delta_k$ as specified below.

Updating Trust Region Size $\Delta_k$

    1. If $\rho_k \leq \mu$ 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 $\rho_k \geq \eta$ 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*

standard unconstrained trust region subproblem:

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

The method of Moré and Sorensen [4] can be directly applied to (1.2) if Cholesky factorizations of matrices with the structure of H_k can be computed efficiently. However, this method is unsuitable for large-scale problems if the Hessian H_k 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 this 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) seems viable although our computational experience (see Table 4) indicates 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 this STIR method 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 [7], at least in the context of our algorithm. Second, we examine our method with and without reflection. We show 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. Compared to the Steihaug [6] approach, the subspace approach is more likely to capture negative curvature information and consequently leads to better computational performance. Finally, our subspace method is competitive with, and often superior to, the active set method in LANCELOT [8].

The paper is organized as follows. In §2, we briefly summarize the existing TIR method. Then we provide a computational comparison of the subspace trust region method and the Steihaug algorithm in the context of unconstrained minimization in §3. We introduce a subspace method STIR, and discuss its convergence properties, in §4. Issues concerning the computation of negative curvature directions and inexact Newton steps are discussed in §5; computational results are provided indicating that performance is typically not impaired by using an inexact Newton step. Concluding remarks appear in §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 these matrices and the TIR algorithm.

The trust region subproblem (1.2) and the affine scaling matrices D_k and C_k arise naturally from examining the first-order Kuhn-Tucker conditions for (1): if a feasible point $l < x$ is a local minimizer, then $x_i g_i = 0$ for $1 \leq i \leq n$ and if $g_i < 0$ then x_i is not at any of its bounds. This characterization is expressed in the nonlinear system of equations

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

where

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

and the vector $v(x) \in \Re^n$ is defined below: for each $1 \leq i \leq n$,

(i). If $g_i < 0$ and $u_i < \infty$ then $v_i \stackrel{\text{def}}{=} x_i - u_i$;
(ii). If $g_i \geq 0$ and $l_i > -\infty$ then $v_i \stackrel{\text{def}}{=} x_i - l_i$;
(iii). If $g_i < 0$ and $u_i = \infty$ then $v_i \stackrel{\text{def}}{=} -1$;
(iv). If $g_i \geq 0$ and $l_i = -\infty$ then $v_i \stackrel{\text{def}}{=} 1$.

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 $x_k \in int(\mathcal{F})$. 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

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

Here $J^v(x) \in \Re^{n \times n}$ corresponds to the Jacobian of $|v(x)|$. Each diagonal component of the diagonal matrix J^v equals to zero or ± 1. If all the components of l and u are finite, $J^v = \text{diag}(\text{sgn}(g))$. If $v_i = 0$, we define $J_{ii}^v = 1$.

Equation (2.3) suggests the use of the affine scaling transformation: $\hat{x} \stackrel{\text{def}}{=} D_k x$. 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

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

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

$$\begin{aligned} \psi_k(s) &\overset{\text{def}}{=} s^T g_k + \frac{1}{2} s^T M_k s, \\ C_k &\overset{\text{def}}{=} D_k \text{diag}(g_k) J_k^v D_k, \\ M_k &\overset{\text{def}}{=} H_k + C_k. \end{aligned}$$

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., $x_k + s_k \in int(\mathcal{F})$. We use $\alpha_k^*[d_k]$ to denote the step obtained from d_k with a possible step-back for strict feasibility. Let τ_k^* denote the minimizer along d_k within the *feasible trust region*, i.e., $\tau_k^* = \text{argmin}_\tau \{\psi_k(\tau d_k) : \|\tau D_k d_k\| \leq \Delta_k, x_k + \tau d_k \in \mathcal{F}\}$. Let $\theta_k \in [\theta_l, 1]$ for some $0 < \theta_l < 1$ and $\theta_k - 1 = O(\|d_k\|)$. Then

$$\alpha_k^*[d_k] \overset{\text{def}}{=} \theta_k \tau_k^* d_k \overset{\text{def}}{=} \begin{cases} \tau_k^* d_k & \text{if } x_k + \tau_k^* d_k \in int(\mathcal{F}), \\ \theta_k \tau_k^* d_k & \text{otherwise.} \end{cases} \tag{2.7}$$

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

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

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

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

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 computation process. In [3] and [2], we have utilized a reflection technique to permit further possible reduction of the objective function along a reflection path on the boundary. We have found in [3] and [2] that this reflection process significantly enhances performance for minimizing a general quadratic function subject to simple bounds.

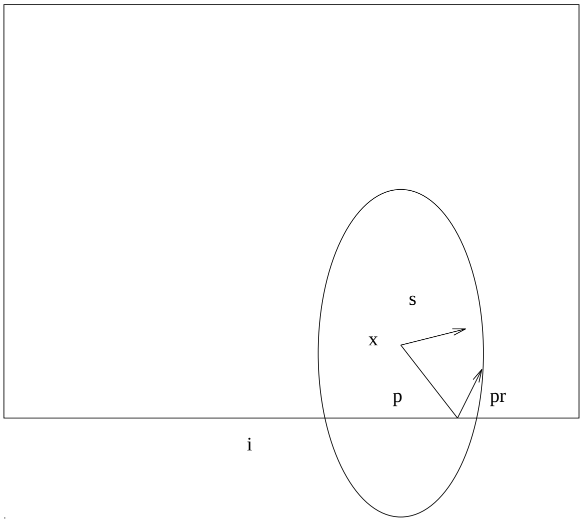

FIG. 2. *Reflection Technique*

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 denotes the piecewise direction path with p_k reflected on the first boundary it encounters, see FIG. 2.

We can appreciate the convergence results for this approach by observing the role of the affine scaling matrix D_k. For the components x_i which are approaching the "correct" bounds, the sequence of directions $\{-D_k^{-2}g_k\}$ becomes increasingly tangential to these bounds. Hence, the bounds will not prevent a large step size along $\{-D_k^{-2}g_k\}$ from being taken. 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 different from, the scaling typically used in affine scaling methods for linear programming. The affine scaling matrix $D_k^{\text{affine}} \overset{\text{def}}{=} \text{diag}(\min(x_k - l_k, u_k - x_k))$ [7], commonly used in affine scaling methods for linear programming, is formed from the distance of variables to their closest bounds. Our scaling matrix D_k^2 equals to D_k^{affine} only when $\min(x_k - l_k, u_k - x_k) = |v_k|$. (Note that 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 [9, 10], which minimize a quartic subject to bounds on the variables. Table 1 and 2 list the average number of iterations (over ten 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. The details of the algorithm implementation are given in §6.

Table 1 demonstrates the significant difference made by a single reflection. The only difference

	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 1
The STIR algorithm with and without reflection: 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

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

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 superiority 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. 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 unconstrained problems, there is no significant difference between the two scaling matrices. However, for the constrained problems we tested, the choice D_k is clearly superior. We observe that when D_k is used, the number of iterations for a constrained problem is roughly the same as that for the corresponding unconstrained 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 problem.

3. Approximation to the Trust Region Solution in Unconstrained Minimization. There are two possible ways to approximate a full-space trust region solution in unconstrained minimization.

Byrd, Schnabel, and Schultz [11] suggest substituting 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. (Our implementation employs a two-dimensional choice for $\mathcal{S}_k$.)

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 either negative curvature is revealed, the current approximate solution reaches the boundary of the trust region, or the Newton system residual is sufficiently reduced.

We believe that a subspace trust region approach better captures the negative curvature information compared to the Steihaug approach [6]. To justify this we have conducted a limited computational study in the unconstrained minimization setting.

We implement the subspace method with the subspace $\mathcal{S}_k$ defined by the gradient direction g_k and the output of a Modified Preconditioned Conjugate Gradient (MPCG) method applied to the linear Newton system: $H_k s = -g_k$. The output is either an inexact Newton step s_k^{IN} defined by,

$$H_k s_k^{IN} = -g_k + r_k \quad \text{such that} \quad \|r_k\|/\|g_k\| \leq \eta_k, \tag{3.2}$$

or a direction of negative curvature, detected by MPCG. Algorithm MPCG is given in greater detail in FIG. 11, Appendix B. Our implementation of the Steihaug method can also be found in Appendix B.

Both the Steihaug and subspace implementations are wrapped in a standard trust region framework for the unconstrained minimization problem. For both methods the preconditioning matrix used is $P_k = G_k^2$ where G_k is the diagonal matrix computed from $G_{kii} = \sqrt{|H_{kii}|}$ for $H_{kii} \neq 0$ and $G_{kii} = 1$ otherwise. The same strategy is used to update Δ_k (see §6 for more details). We let $\Delta_0 = 0.1\|g_0\|$ where the $\|\cdot\|_2$ is used for the subspace method and $\|\cdot\|_G$ for the Steihaug method ([6]).

We used twenty different unconstrained nonlinear test problems. All but four are test problems described in [12], but with all the bound constraints removed. The problems EROSENBROCK and EPOWELL are taken from [13]. The last two problems, molecule problems MOLE1 and MOLE3, are described in [9, 10]. For all problems, the number of variables n is 260. The minimization algorithm terminates when $\|g\|_2 \leq 10^{-6}$. We use the parameter $\eta = 0.0005$ in both FIG. 11 and FIG. 12.

Tables 3 and 4 compare the Steihaug and subspace methods described above in terms of the number of minimization iterations and the total number of conjugate gradient (CG) iterations. Table 3 shows problems for which negative curvature was not detected, and Table 4 shows problems for which negative curvature was detected. Although not included here, the function values and gradient norms (upon termination) were virtually the same for both methods for all problems. Since these values were essentially the same among the two methods, we only discuss the difference in iterations counts. The difference in minimization and CG iteration counts is plotted in FIG. 3 and FIG. 4.

Most notable in Table 3 and the graphs of FIG. 3 is how strikingly similar the results are for the Steihaug and subspace methods; the minimization with each method stops within two iterations of the other in all cases. Furthermore, both methods take an *identical* number of total CG iterations except for the problem BROWN1 where the Steihaug method takes four more iterations. When negative curvature is encountered, shown in Table 4 and in FIG. 4, the iteration counts for each method are again similar for a few problems. For most problems, however, the Steihaug method takes more iterations, and for some problems the difference is substantial. This is particularly true for the problems CHAINWOOD, MOLE1 and MOLE3 (for CHAINWOOD, problem 3 in FIG. 4, the total difference in iteration counts is

	Problem	Minimization Subspace	 Steihaug	CG Subspace	 Steihaug
1.	BROWN1	27	29	39	43
2.	BROWN3	6	6	6	6
3.	BROYDEN1A	11	11	81	81
4.	BROYDEN1B	5	5	34	34
5.	BROYDEN2B	7	7	71	71
6.	CHAINSING	22	22	188	188
7.	CRAGGLEVY	21	21	125	125
8.	DEGENSING	22	22	188	188
9.	EPOWELL	18	18	72	72
10.	GENSING	22	22	83	83
11.	TOINTBROY	7	7	58	58
12.	VAR	43	43	5590	5590

TABLE 3

Comparison when only positive curvature is encountered: number of iterations

	Problem	Minimization Subspace	 Steihaug	CG Subspace	 Steihaug
1.	AUGMLAGN	36	29	267	228
2.	BROYDEN2A	22	19	247	196
3.	CHAINWOOD	156	988	3905	3878
4.	EROSENBROCK	44	46	52	86
5.	GEROSE	23	33	166	165
6.	GENWOOD	58	63	304	275
7.	MOLE1	46	119	460	376
8.	MOLE3	125	186	6311	5356

TABLE 4

Comparison when negative curvature is encountered: number of iterations

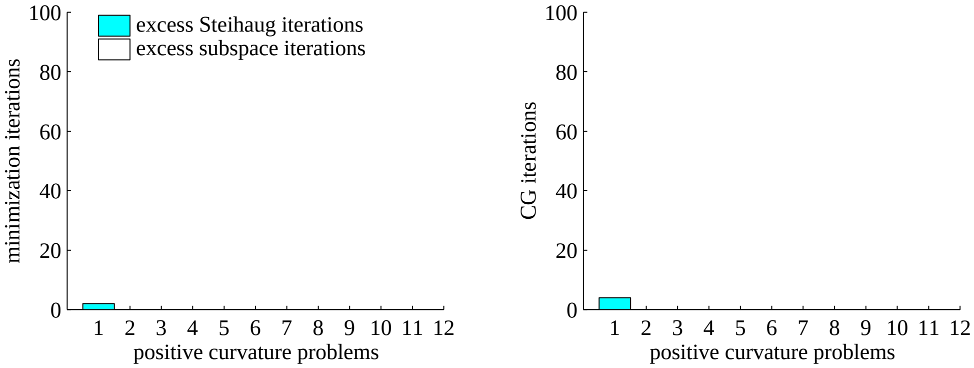

FIG. 3. *Comparison of subspace and Steihaug trust region methods for unconstrained problems*

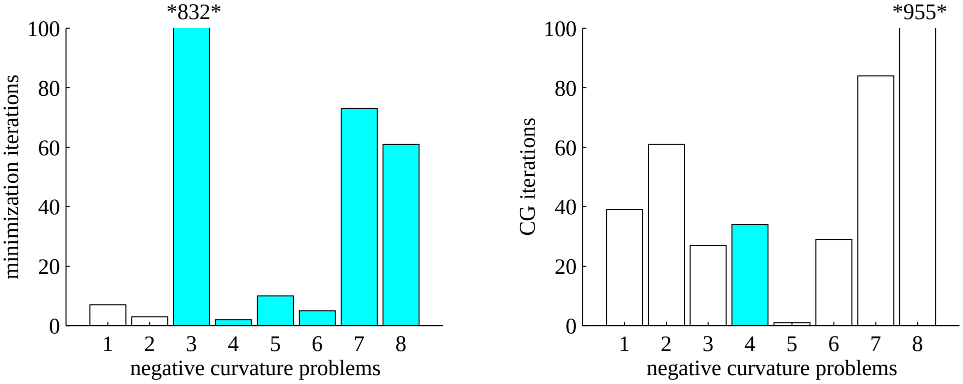

FIG. 4. *Comparison of subspace and Steihaug trust region methods for unconstrained problems*

explicitly noted as it is beyond the scale of the graph). In general the subspace method does take more CG iterations on problems with negative curvature, but it is these extra relatively inexpensive CG iterations that reduce the total number of minimization iterations. (Again, for the problem MOLE3 the difference in CG iterations is explicitly noted in FIG. 4 as it is beyond the scale of the graph.)

A closer examination of the behavior of the two algorithms indeed shows that when negative curvature is *not* encountered, both methods take similar steps. (In this case, if the trust region is large enough, both methods in FIG. 11 and FIG. 12 will stop under the same conditions after the same number of CG iterations, as displayed in Table 3.) By the nature of the algorithms, if the Steihaug method detects negative curvature, then so will the subspace approach. However if the subspace algorithm detects negative curvature, the Steihaug method may terminate before it finds negative curvature; and then it does not converge (to a local minimizer) as quickly as the subspace method. The important role that negative curvature plays is supported by the fact that the subspace method often moves in a substantial negative curvature direction when the Steihaug method overlooks negative curvature. Furthermore, it is when the trust region radius Δ_k is small that the Steihaug method is most likely to stop early and miss negative curvature. Thus it appears that the effectiveness of the Steihaug idea decreases as nonlinearity increases.

4. The STIR Method. Supported by the discussion in §3, we propose a large-scale subspace adaptation of the TIR method [1] for the bound constrained problem (1.1).

In moving from the unconstrained subspace approach to the box-constrained setting, it seems natural to 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. A two-dimensional subspace for the trust region subproblem (2.5) can be selected from the span of the two vectors $\{D_k^{-1} g_k, \hat{s}_k^N\}$ and a negative curvature direction $\hat{w}_k$ for $\hat{M}_k$. This suggests that we form $\mathcal{S}_k$ from the directions $\{D_k^{-2} g_k, D_k^{-1}\hat{s}_k^N, D_k^{-1}\hat{w}_k\}$. Will such subspace formulations succeed in achieving optimality? We examine this issue in more detail.

It is clear that the including the scaled gradient direction $D_k^{-2} g_k$ 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 the following conditions must be met.

Firstly, it is clear that 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 it is indefinite, i.e., $\mathcal{S}_k$ must contain a direction $w_k = D_k^{-1}\hat{w}_k$ such that

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

Secondly, 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:

DEFINITION 4.1. *A direction sequence* $\{s_k\}$ *has* large-step-size *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 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:

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

Can we select two-dimensional subspaces satisfying all three properties and thus guarantee quadratic (superlinear) convergence to a second-order point? The answer, in theory, is yes — the subspace adaptation of TIR algorithm (STIR) in FIG.5 is an example of a subspace method capable of achieving the desired properties.

To ensure convergence to a solution, the solution sequence of the subspace trust region subproblems (4.1) need to have large-step-size. Lemma 1 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 two sequences of uniformly independent vectors in the sense that $\liminf\{\|z_k - w_k\|\} > 0$, each with large-step-size.

LEMMA 1. *Assume that* $\{w_k\}$ *and* $\{z_k\}$ *have large-step-size with* $\|D_k w_k\| = 1$ *and* $\|D_k z_k\| = 1$. *Moreover,* $\liminf_{k\to\infty}\{\|z_k - w_k\|\} > 0$. *Then the solution sequence* $\{p_k\}$ *to the subproblem (4.1) with* $\mathcal{S}_k = \text{span}\{z_k, w_k\}$ *has large-step-size.*

Proof. The proof is very straightforward and is omitted here. ■

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

(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 $x_k + s_k \in int(\mathcal{F})$.

Theorem 2 below, with the proof provided in the Appendix, formalizes the convergence properties of STIR.

THEOREM 2. *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.5. Then*

1. *If (AS.3) is satisfied, then the Kuhn-Tucker condition is satisfied at every limit point.*

The STIR Method

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

For $k = 0, 1, \cdots$

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 $x_k + s_k \in int(\mathcal{F})$, based on the subspace subproblem,
$$\min_s \{\psi_k(s) : \|D_k s\|_2 \le \Delta_k, \; s \in \mathcal{S}_k\},$$
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 large-step-size. 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. 5. *The STIR Method for Minimization Subject to Bound Constraints*

2. *Assume that both (AS.3) and (AS.5) are satisfied and $\hat{w}_k$ in FIG. 5 contains sufficient negative curvature information whenever $\hat{M}_k$ is indefinite, 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, then 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, then 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, then $\hat{M}_*$ 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 the same as in [1].

DEFINITION 4.2. *A point $x \in \mathcal{F}$ is* **nondegenerate** *if, for each index i:*

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

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 hinges on obtaining (guaranteed) sufficient negative curvature direction with large-step-size. We discuss this next.

5. Computing Negative Curvature Directions with Large-Step-Size.

Is it possible, in principle, to satisfy both the sufficient negative curvature requirement (4.2) and the large-step-size property? The answer is yes: 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 $\lim_{k\to\infty} \lambda_{\min}(\hat{M}_k) < 0$, the sequence $\{D_k^{-1} u_k\}$ has large-step-size.

However, it is not computationally feasible to compute the (exact) eigenvector u_k. Therefore, approximations, and short cuts, are in order. Can we compute approximate eigenvectors with large-step-size?

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 j-th step are computed in the Krylov space

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

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 the following. If a sequence $\{D_k^{-1}\hat{q}_k\}$ has large-step-size then each sequence in $D_k^{-1}\{\hat{q}_k, \hat{M}_k\hat{q}_k, \cdots, \hat{M}_k^{j-1}\hat{q}_k\}$ 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 $\{D_k^{-1}\hat{q}_k^1, \cdots, D_k^{-1}\hat{q}_k^j\}$ all have large-step-size if the Lanczos vectors $\{\hat{q}_k^1 \cdots, \hat{q}_k^j\}$ 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 large-step-size. In other words, in order to generate a negative curvature direction sequence with large-step-size, orthogonality needs to be maintained in the Lanczos process. Fortunately, as discussed in [16], it is quite reasonable to assume that until all of the distinct eigenvalues of the original matrix have been approximated well, orthogonality of the Lanczos vectors are well maintained. Since we are only interested in a direction with sufficient negative curvature, we expect that it can be computed before loss of orthogonality occurs.

A second (and cheaper) strategy is to employ a modified preconditioned conjugate gradient scheme, e.g., MPCG in FIG.12. Unfortunately, this process is not guaranteed to generate sufficient negative curvature; nonetheless, as indicated in [17], the MPCG output will satisfy the large-step-size property.

Finally we consider a modified Cholesky factorization, e.g., [18], to obtain a negative curvature direction. Assume that $\{\hat{M}_k\}$ is indefinite and $\{d_k\}$ is obtained from the modified Cholesky method. We demonstrate below that $\{d_k = D_k^{-1}\hat{d}_k\}$ has large-step-size under a nondegeneracy assumption.

The negative curvature direction $\hat{d}_k = D_k d_k$ computed from the modified Cholesky method (see [18], page 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 non-negative diagonal matrix. Without loss of generality, we assume that $P_k = I$.

We argue, by contradiction, that $\{d_k\}$ has the large-step-size property. Assume that $\{d_k\}$ does not have this property. From $L_k^T\hat{d}_k = e_{j_k}$ and that L_k is a lower triangular matrix with unit diagonals, it is clear that $\hat{d}_k[\mathrm{j_k}{+}1 : \mathrm{n}] = 0$. Moreover, from $\hat{M}_k\hat{d}_k + E_k\hat{d}_k = \delta_{kj_k}L_k e_{j_k}$, $\delta_k > \delta$ 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.

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

$$\begin{bmatrix} \hat{M}_k[1\!:\!\mathrm{j_k}{-}1, 1\!:\!\mathrm{j_k}{-}1] & 0 \\ 0 & \eta_k \end{bmatrix}$$

where η_k is positive (because of the nondegeneracy assumption). Therefore, we conclude that $\{d_k\}$ has large-step-size.

6. Computational Experience. We demonstrate the computational performance of our STIR method given in FIG.5. 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 comparisons with SBMIN of LANCELOT [8] 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 below in FIG.6.

```
**Updating Trust Region Size** $\Delta_k$
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. 6. *Updating Trust Region Size*

Our experiments were carried out on a Sun Sparc workstation using the Matlab environment.

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* $\|D_k g_k\|_\infty \leq \tau_1$.

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 iterations.

We first report the results of the STIR method using the modified Cholesky factorization. Table 5 lists the number of iterations required for some standard testing problems (for details of these problems see [12]). (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 5 indicate that, for these testing problems at least, the number of iterations increases only slightly, if at all, with the problem size. Moreover, in comparison to the unconstrained problems, the presence of the bound restrictions does not seem to increase the number of iterations. This is depicted pictorially in FIG. 7. 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 but using a conjugate gradient implementation. We use the algorithm MPCG in FIG.12 to find the directions 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 6 and FIG. 8. 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

	n				
Problem	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

TABLE 5

STIR method with exact Newton steps: number of iterations

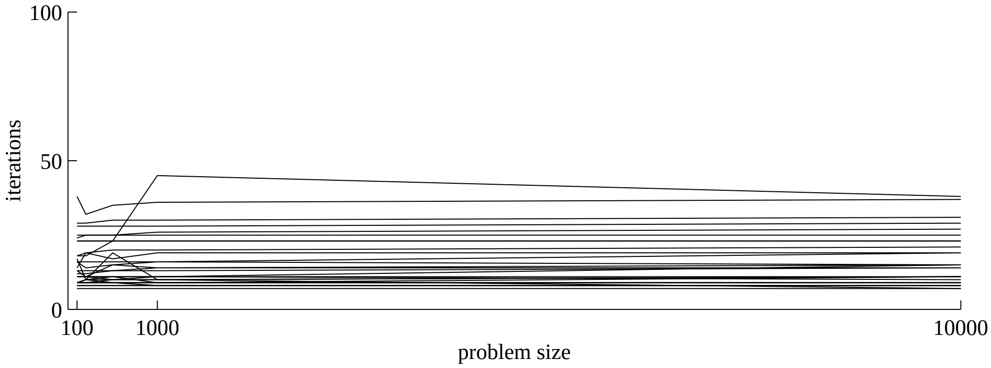

FIG. 7. *STIR performance with exact Newton steps*

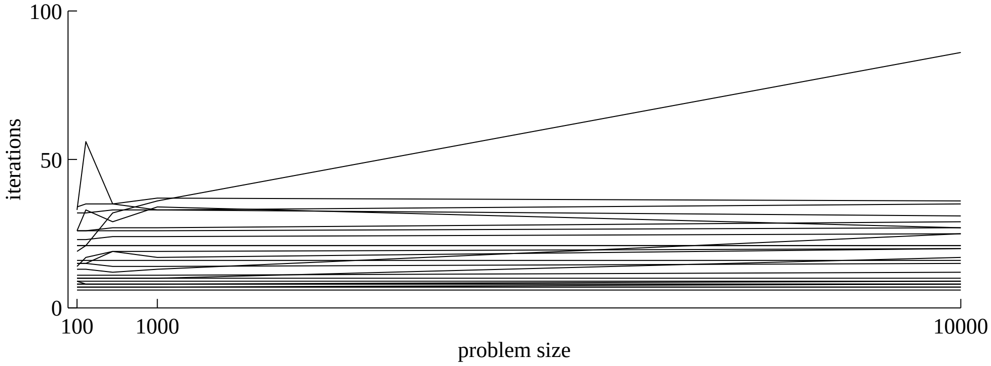

FIG. 8. *STIR method with inexact Newton steps*

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 (note that the STIR with exact Newton steps only takes 38 iterations). However, this change would also increase the amount of computation (conjugate gradient iterations).

Next we include some results which indicate that our STIR method is fairly insensitive to the starting point. The results in Table 7 were obtained using exact Newton steps on problems of dimension 1000. The results in Table 8 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 [12]; *upper* starts all variables at upper bounds; *lower* 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 points. These results are also shown graphically in FIG. 9 and FIG. 10. From these graphs it is clear that both implementations of STIR are fairly robust when it comes to starting

	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

TABLE 6

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

Problem	Starting Point 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
DEGENSING C	28	47	39	52	42	39	36
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
CRAGGLEVY C	30	38	33	26	26	34	37
AUGMLAGN C	36	40	26	36	15	23	37
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 7

STIR method with exact Newton steps for $n = 1000$: number of iterations

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 [8]. In particular, we choose problems where negative curvature is present or where it appears that the "active set" at the solution may be difficult 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 use the default settings for LANCELOT and adjusted our STIR stopping conditions to be comparable if not more stringent.

First consider a constrained convex quadratic problem. The results, given in Table 9, show that our proposed STIR method is markedly superior (by an order of magnitude) to SBMIN on this problem (*c.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 mis-identifies 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.

Table 10 summarizes the performances of STIR and SBMIN, on a set of constrained problems exhibiting negative curvature. (Again the problems are from [12] except the last two have been constrained differently to display negative curvature.) STIR is significantly better on these problems — this is probably due to the fact that negative curvature is better exploited in our subspace trust region approach than in

	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
DEGENSING C	33	43	37	42	37	37	44
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
CRAGGLEVY C	26	39	35	27	24	35	45
AUGMLAGN C	34	60	45	32	10	24	46
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

TABLE 8
STIR method with inexact Newton steps for $n = 1000$: number of iterations

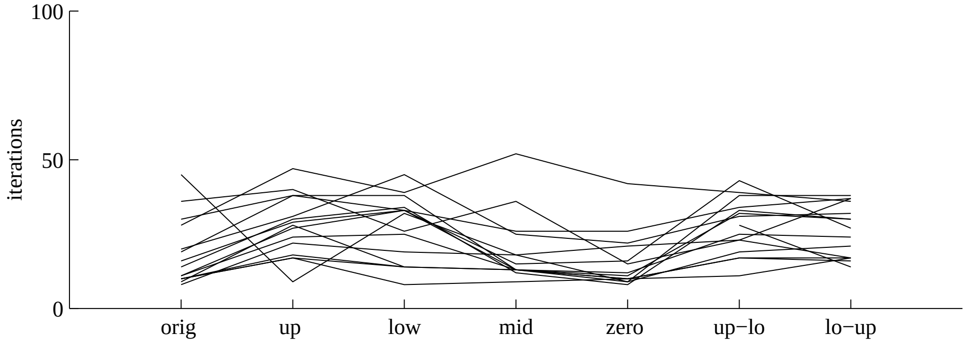

FIG. 9. *STIR method with exact Newton steps at varied starting points*

	inexact STIR		SBMIN	
$n = 800$	iteration	c.g. it	iteration	c.g. it
BIGGSB2	16	5551	281	53157

TABLE 9
STIR with inexact Newton steps vs. LANCELOT SBMIN on a convex quadratic: number of iterations

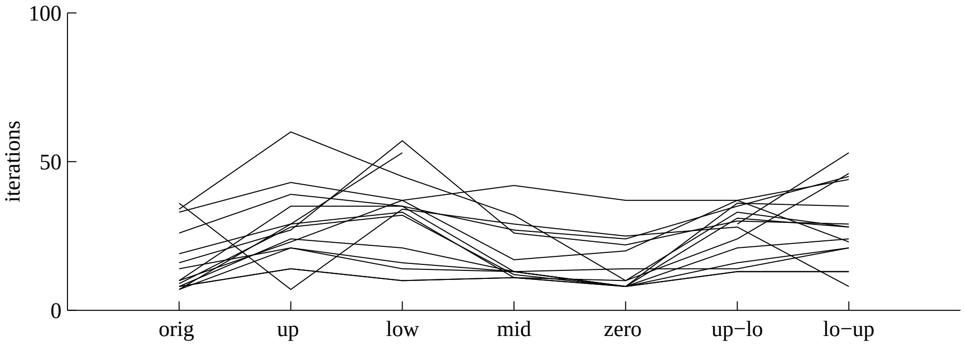

FIG. 10. *STIR method with inexact Newton steps at varied starting points*

	inexact STIR			SBMIN		
	n			n		
Problem	100	1000	10000	100	1000	10000
AUGMLAGN U	34	30	37	29	38	46
CHAINWOOD U	122	1004	8953	6594	$>$ 10000	$>$ 10000
GENWOOD U	62	67	63	439	952	554
GENROSE U	29	29	29	76	76	76
CHAINWOOD NC	17	31	16	54	48	61
GENWOOD NC	16	24	23	47	$>$ 1000	60

TABLE 10
STIR with inexact Newton steps vs. LANCELOT SBMIN when negative curvature exists: number of iterations

the Steihaug trust region method, which SBMIN employs. This is consistent with results presented in §3, e.g., see Table 4.

7. Conclusion. Based on the trust-region interior reflective (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 direction.

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 use of our 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 [8], indicate that our proposed STIR method can significantly outperform an active-set approach for some large-scale problems.

REFERENCES

[1] Thomas F. Coleman and Yuying Li. An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, to appear.

[2] Thomas F. Coleman and Yuying Li. On the convergence of reflective Newton methods for large-scale nonlinear minimization subject to bounds. *Mathematical Programming*, 67:189–224, 1994.

[3] Thomas F. Coleman and Yuying Li. A reflective Newton method for minimizing a quadratic function subject to bounds on the variables. *SIAM Journal on Optimization*, to appear.

[4] Jorge J. Moré and D.C. Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4:553–572, 1983.

[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, 1994.

[6] T. Steihaug. The conjugate gradient methods and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20:626–637, 1983.

[7] I.I. Dikin. Iterative solution of problems of linear and quadratic programming. *Doklady Akademiia Nauk SSSR*, 174:747–748, 1967.

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

[9] Bruce A. Hendrickson. *The molecule problem: Determining conformation from pairwise distances*. Cornell University Ph.D thesis, Computer Science, 1991.

[10] Thomas F. Coleman. Large-scale numerical optimization: Introduction and overview. In Allen Kent and James G. Williams, editors, *Encyclopedia of Computer Science and Technology*, pages 167–195. Marcel Dekker, INC., 1993.

[11] Gerald A. Shultz, R. B. Schnabel, and Richard H. Byrd. A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties. *SIAM Journal on Numerical Analysis*, 22(1):47–67, 1985.

[12] Andrew R. Conn, N. I. M. Gould, and Ph. L. Toint. Testing a class of methods for solving minimization problems with simple bounds on the variables. *Mathematics of Computation*, 50(182):399–430, 1988.

[13] Jorge J. Moré, Burton S. Garbow, and Kenneth. E. Hillstrom. Testing unconstrained optimization software. *ACM Transactions on Mathematical Software*, 7:17–41, 1981.

[14] Richard H. Byrd, Robert B. Schnabel, and Gerald A. Shultz. Approximate solution of the trust region problem by minimization over two-dimensional subspaces. *Mathematical Programming*, 40:247–263, 1988.

[15] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1989.

[16] Jane K. Cullum and Ralph A. Willoughby. *Lanczos Algorithms for Large symmetric eigenvalue computations, Vol. 1 Theory*. Birkhaüser, Boston, 1985.

[17] Mary Ann Branch. *Inexact Reflective Newton Methods for Large-Scale Optimization Subject to Bound Constraints*. PhD thesis, Department of Computer Science, Cornell University, 1995.

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

A. Proofs for THEOREM 2. The convergence results (THEOREM 2) for the STIR algorithm can be obtained in a similar manner 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 large-step-size if the corresponding $\{\hat{M}_*\}$ were indefinite at a limit point (see Lemma 3 below). Moreover, if $\hat{M}_k$ is positive definite at a limit point then we prove that the step size along the subspace trust region solution is sufficiently large in the following sense:

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

Here α_k is the stepsize, along p_k, to the boundary of the feasible region (see Lemma 4). Based on 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 $\mathcal{S}_k = \text{span}\{w_k, z_k\}$. 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

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

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.5. has large-step-size if the corresponding sequence $\{\hat{M}_k\}$ satisfies that $\lim_{k\to\infty} \lambda_{\text{min}}(\hat{M}_k) < 0$.

LEMMA 3. *If* $\limsup \lambda_{\text{min}}(\hat{M}_k) < 0$ *for a subsequence, then the corresponding solution subsequence* $\{p_k\}$ *of the subspace trust region subproblem (4.1) has large-step-size.*

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

For the subsequence with $|\mathcal{S}_k| = 1$, the corresponding trust region solution sequence clearly has large-step-size.

For the subsequence with $|\mathcal{S}_k| = 2$, it is clear that $\liminf\{\|\frac{D_k^{-2}\text{sgn}(g_k)}{\|D_k^{-2}\text{sgn}(g_k)\|} - \frac{w_k}{\|w_k\|}\|\} > 0$. Since $\{z_k\} = \{D_k^{-2}\text{sgn}(g_k)\}$ and $\{w_k\}$ have large-step-size, we have that $\{p_k\}$ has large-step-size following Lemma 1. ■

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

LEMMA 4. *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

$$(A.3) \qquad \text{diag}(g_k) J_k^v D_k s_k^N = -D_k^{-1} g_k - D_k^{-1} H_k D_k^{-1} D_k s_k^N.$$

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

LEMMA 5. *Assume that* p_k *is a solution to the subspace subproblem (4.1) with* $\mathcal{S}_k = \text{span}\{D_k^{-2} g_k, s_k^N\}$. *If* $\{x_k\}$ *converges to a nondegenerate point* x_* *where the second order sufficiency conditions are satisfied,* *then*

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

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

Proof. By definition,

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

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

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

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

$$p_k = \gamma_k(-D_k^{-2} g_k) + \beta_k s_k^N \quad \text{where } \beta_k \geq 0 \text{ and } \gamma_k \geq 0, \tag{A.5}$$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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 $0 < \overline{\lambda} < \lambda^*$ so that $\gamma(\overline{\lambda}) = 0$. This implies that

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

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

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

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

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

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 2. *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.5. Then*

1. *If (AS.3) is satisfied, then 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. 5 contains sufficient negative curvature information whenever* $\hat{M}_k$ *is indefinite, 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 is nondegenerate, then 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, then 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, then $\hat{M}_*$ is positive definite, $\{x_k\}$ converges to x_*, all iterations are eventually successful, and $\{\Delta_k\}$ is bounded away from zero.

Proof. Using Lemma 3, for any subsequence with $\lim_{k\to\infty} \lambda_{\min}(\hat{M}_k) < 0$, the corresponding $\{p_k\}$ has large-step-size. Therefore, there exists $\epsilon_0 > 0$ such that $\alpha_k > \epsilon_0,$ for k sufficiently large. Hence, using Lemma 4, for some $\epsilon_1 > 0$,

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

Condition (AS.5) 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 4, 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 5, 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). ∎

B. Implementation Details. We present details of our Steihaug method implementation in FIG.11 and the modified preconditioned conjugate gradient (MPCG) in FIG.12.

For more details on the large-step-size property of the steps computed from FIG.12, see [17].

```
function $[s]$ = Steihaug($H$, $g$, $C$, $\eta$, $\Delta$)
% Note: $C$ is some preconditioning matrix for H. $C$ must be positive definite.
$n = length(g);\ kmax = n/2;$
$k = 0;\ p_0 = 0;\ Q_0 = 0;\ r_0 = -g;$

while $k < kmax$
    Step 1:    Solve $Cz_k = r_k;$
    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 $\gamma_k \leq 0$, compute $\tau > 0$ so that $\|p_k + \tau d_k\|_C = \Delta$, $s = p_k + \tau d_k$, 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 $\|p_k\| \geq \Delta$
                 compute $\tau > 0$ so that $\|p_k + \tau d_k\|_C = \Delta$, $s = p_k + \tau d_k$, return, end
    Step 7:    if $\|C^{-1}\| \cdot \|r_k\| \leq \eta \cdot \|C^{-1} r_0\|$, $s = p_k$, return, end
end
$s = p_k$, return
```

FIG. 11. *The Steihaug algorithm*

```
Step 0: $j = 0$; $p_0 = 0$; $d_0 = 0$; $r_0 = -\hat{g}$;
% Note: $\eta$ and $\epsilon$ are positive constants.
while $j <$ maximum iteration
    Step 1:  Solve $Pz_j = r_j$
    Step 2:  $j = j + 1$
    Step 3:  if $j = 1$
                 $d_1 = z_0$
             else
                 $\beta_j = r_{j-1}^T z_{j-1} / r_{j-2}^T z_{j-2}$
                 $d_j = z_{j-1} + \beta_j d_{j-1}$
             end
    Step 4:  $\gamma_j = d_j^T \hat{M} d_j$
    Step 5:  if $\gamma_j \leq 0$, exit: $d = d_j, p = p_j$                    (B.1)
             else if $\gamma_j \leq \epsilon(d_j^T P d_j)$, exit: $d = 0, p = p_j$   (B.2)
             else
                 $\alpha_j = r_{j-1}^T z_{j-1} / \gamma_j$
                 $p_j = p_{j-1} + \alpha_j d_j$
                 $r_j = r_{j-1} - \alpha_j \hat{M} d_j$
             end
    Step 6:  if $\|C^{-1}\| \cdot \|r_j\| \leq \eta \cdot \|C^{-1} r_0\|$, exit: $p = p_j$, $d = 0$   (B.3)
end
```

FIG. 12. *MPCG*