AN INTERIOR-POINT SUBSPACE MINIMIZATION METHOD FOR THE TRUST-REGION STEP[*]

JENNIFER B. ERWAY[†] AND PHILIP E. GILL[‡]

Abstract. We consider methods for large-scale unconstrained minimization based on finding an approximate minimizer of a quadratic function subject to a two-norm trust-region inequality constraint. The Steihaug-Toint method uses the conjugate-gradient algorithm to minimize the quadratic over a sequence of expanding subspaces until the iterates either converge to an interior point or cross the constraint boundary. Recent extensions of the Steihaug-Toint method allow the accuracy of the trust-region step to be specified, thereby allowing the overall cost of computing the problem functions to be balanced against the cost of computing the trust-region steps. However, if a preconditioner is used with the conjugate-gradient algorithm, the Steihaug-Toint method requires the trust-region norm to be defined in terms of the preconditioning matrix. If a different preconditioner is used for each subproblem, the shape of the trust-region can change substantially from one subproblem to the next, which invalidates many of the assumptions on which standard methods for adjusting the trust-region radius are based. In this paper we propose a method that allows the trust-region norm to be defined independently of the preconditioner. The method solves the inequality constrained trust-region subproblem over a sequence of evolving low-dimensional subspaces. Each subspace includes an accelerator direction obtained from a Newton method applied to an primal-dual interior method. A crucial property of this direction is that it can be computed by applying the preconditioned conjugate-gradient method to a positive-definite system in both the primal and dual variables of the trust-region subproblem.

Key words. Large-scale unconstrained optimization, trust-region methods, conjugate-gradient methods, Krylov methods, preconditioning.

AMS subject classifications. 49J20, 49J15, 49M37, 49D37, 65F05, 65K05, 90C30

1. Introduction.

The jth iteration of a trust-region method for unconstrained minimization involves finding an approximate solution of the trust-region subproblem:

$$\underset{s\in\mathbb{R}^n}{\text{minimize}} \;\; \mathcal{Q}_j(s) \equiv g_j^T s + \tfrac{1}{2} s^T H_j s \quad \text{subject to} \quad \|s\| \le \delta_j, \tag{1.1}$$

where δ_j is a given positive trust-region radius and $\mathcal{Q}_j(s)$ is the quadratic model of a scalar-valued function with gradient g_j and Hessian H_j. The focus of this paper is on the solution of (1.1) when the matrix H_j is best accessed as an *operator* for the definition of matrix-vector products of the form $H_j v$.

In this context, Steihaug [33] and Toint [34] independently proposed methods for solving (1.1) when the trust-region is defined in terms of the two-norm, i.e., the constraint is $\|s\|_2 \le \delta_j$. If H_j is positive definite, the Newton equations $H_j s = -g_j$ define the *unconstrained* minimizer of (1.1). The Steihaug-Toint method begins with the application of the conjugate-gradient (CG) method to the Newton equations. This process is equivalent to minimizing $\mathcal{Q}_j$ over a sequence of expanding subspaces generated by the conjugate-gradient directions. As long as the curvature of $\mathcal{Q}_j$ remains positive on each of these subspaces, the CG iterates steadily increase in norm and the CG iterates either converge inside the trust region or form a piecewise-linear path with a unique intersection-point on the trust-region boundary. When H_j is not positive definite, a solution of (1.1) must lie on the boundary of the trust region and the CG

[*] Research supported by the National Science Foundation grant DMS-0511766.

[†] Department of Mathematics, PO Box 7388, Wake Forest University, Winston-Salem, NC 27109 (erwayjb@wfu.edu).

[‡] Department of Mathematics, University of California, San Diego, La Jolla, CA 92093-0112 (pgill@ucsd.edu).

method may generate a direction p along which $\mathcal{Q}_j$ has zero or negative curvature. In this case, the algorithm is terminated at the point on p that intersects the boundary of the trust region.

If the Steihaug-Toint method is terminated on the boundary of the trust region, the step may bear little relation to an optimal solution of (1.1). This means that, in contrast to line-search methods, it is not possible to choose an approximate solution that balances the cost of computing the problem functions with the cost of computing the trust-region step (see, e.g., [6] for more discussion of this issue). Several extensions of the Steihaug-Toint method have been proposed that allow the accuracy of a constrained solution to be specified. Gould, Lucidi, Roma, and Toint [15] proposed the generalized Lanczos trust-region (GLTR) algorithm, which finds a constrained minimizer of (1.1) over a sequence of expanding subspaces associated with the Lanczos process for reducing H_j to tridiagonal form. Erway, Gill and Griffin [6] continue to optimize on the trust-region boundary using the sequential subspace minimization (SSM) method. This method, first proposed by Hager [17, 18], approximates a constrained minimizer over a sequence of evolving low-dimensional subspaces that do not necessarily form a nested sequence. Erway, Gill and Griffin use a basis for each subspace that includes an accelerator vector defined by a primal-dual augmented Lagrangian method.

In many applications the convergence rate of CG can be significantly improved by using a preconditioner, which is usually available in the form of a positive-definite operator M_j^{-1} that clusters the eigenvalues of $M_j^{-1}H_j$. If a preconditioned CG method is used, the increasing norm property of the iterates holds only in the *weighted* norm $\|x\|_{M_j} = (x^T M_j x)^{1/2}$, which mandates the use of a trust region of the form $\|s\|_{M_j} \le \delta_j$. Unfortunately, if a different preconditioner is used for each trust-region subproblem, the shape of the trust-region may alter dramatically from one subproblem to the next. Since a fundamental tenet of trust-region methods is that the value of δ_j be used to determine the value of δ_{j+1}, the effectiveness of the trust-region strategy may be seriously compromised. We emphasize the distinction between the *constant* weighted trust region $\|Ns\|_2 = (s^T N^T N s)^{1/2} \le \delta_j$ typically associated with a constant nonsingular scaling matrix N, and the *varying* trust region $\|s\|_{M_j} \le \delta_j$ induced by the preconditioner.

In this paper we propose a CG-based method that does not use the Steihaug-Toint expanding subspace idea, and hence may be used safely in conjunction with CG preconditioning. The Steihaug-Toint method and its extensions proceed with the unconstrained minimization of $\mathcal{Q}_j$ and consider the constraint only if the unconstrained solution lies outside the trust-region. In the proposed method, the constrained problem is minimized directly over a sequence of evolving low-dimensional subspaces that include a basis vector defined in terms of a primal-dual interior method applied to (1.1). A crucial property of this direction is that it can be computed by applying the preconditioned CG method to a positive-definite system in both the primal and dual variables of the constrained problem. The subspace also includes a simple, inexpensive estimate of an eigenvector associated with the least eigenvalue of H_j.

Finally, we mention several Krylov-based iterative methods that are intended to find a solution of the problem of minimizing $\mathcal{Q}(s) = g^T s + \frac{1}{2} s^T H s$ subject to the *equality constraint* $\|s\|_2 = \delta$. The methods of Sorensen [32], Rojas and Sorensen [28], Rojas, Santos and Sorensen [27], and Rendl and Wolkowicz [26] approximate the eigenvalues of a matrix obtained by augmenting H by a row and column. In addition, line-search subspace minimization methods for general large-scale unconstrained optimization have been considered by Fenelon [7], Gill and Leonard [13, 14], Nazareth [22], and Siegel [29, 30].

The paper is organized in five sections. In Section 2, we review the properties of the Steihaug-Toint method and its extensions. In Section 3 we formulate the proposed SSM method and consider some properties of the primal-dual interior-point method used to generate the SSM accelerator direction. Further details of the algorithmic components are given in Section 4, which includes numerical comparisons with the Steihaug-Toint method on unconstrained problems from the CUTEr test collection (see Bongartz et al. [1] and Gould, Orban and Toint [16]). Finally, Section 5 includes some concluding remarks and observations.

1.1. Notation and Glossary. Unless explicitly indicated, $\|\cdot\|$ denotes the vector two-norm or its subordinate matrix norm. The symbol e_i denotes the ith column of the identity matrix I, where the dimensions of e_i and I depend on the context. The eigenvalues of a real symmetric matrix H are denoted by $\{\lambda_i\}$, where $\lambda_n \le \lambda_{n-1} \le \cdots \le \lambda_1$. The associated eigenvectors are denoted by $\{u_i\}$. An eigenvalue λ and a corresponding normalized eigenvector u such that $\lambda = \lambda_n$ are known as a *leftmost eigenpair* of H. The Moore-Penrose pseudoinverse of a matrix A is denoted by $A^\dagger$. Some sections include algorithms written in a Matlab-style pseudocode. In these algorithms, brackets will be used to differentiate between computed and stored quantities. For example, the expression $[Ax] := Ax$ signifies that the matrix-vector product of A with x is computed and assigned to the vector $[Ax]$. Similarly, if P is a matrix with columns $p_1, p_2, \ldots, p_m$, then $[AP]$ denotes the matrix with columns $[Ap_1], [Ap_2], \ldots, [Ap_m]$.

2. Background. In this section we drop the suffix j and focus on a typical trust-region subproblem of the form

$$\underset{s\in\mathbb{R}^n}{\text{minimize}}\ \ \mathcal{Q}(s) \equiv g^T s + \tfrac{1}{2} s^T H s \quad \text{subject to} \quad \|s\| \le \delta. \tag{2.1}$$

Toint [34] and Steihaug [33] independently proposed CG-based methods for solving the trust-region problem. The methods begin by applying the CG method to the Newton equations $Hs = -g$ under the assumption that H is positive definite. The CG iterates $\{s_k\}$ have the form

$$s_0 = 0, \qquad s_k = s_{k-1} + \alpha_{k-1} p_{k-1}, \quad k \ge 1,$$

where the CG directions $\{p_k\}$ satisfy the conjugacy conditions $p_\ell^T H p_m = 0$, for all $0 \le \ell \le k$ and $0 \le m \le k$ such that $\ell \ne m$. Each iterate s_k is such that

$$s_k = \underset{s\in\mathcal{S}_k}{\text{argmin}}\ \{g^T s + \tfrac{1}{2} s^T H s,\ \ \|s\| \le \delta\},$$

where $\mathcal{S}_k$ is a k-dimensional subspace spanned by the directions $p_0, p_1, \ldots, p_{k-1}$. The set $\mathcal{S}_k$ is a member of a sequence of expanding subspaces $\{\mathcal{S}_k\}$ such that $\mathcal{S}_{k-1} \subset \mathcal{S}_k$.

If H is positive definite and the Newton step $-H^{-1}g$ lies inside the trust region, then the CG iterations are terminated with the iterate s_k such that $\|g + Hs_k\| \le \tau \|g\|$, where τ is a given positive tolerance. For small values of τ, this s_k approximates the unconstrained step $-H^{-1}g$.

Steihaug establishes the key property that if $p_\ell^T H p_\ell > 0$ for $0 \le \ell \le k-1$, then the norms of the CG iterates $\{s_k\}$ are strictly increasing in the two-norm, i.e., $\|s_k\|_2 >$

$\|s_{k-1}\|_2$. This implies that there is no reason to continue computing CG iterates once they cross the trust-region boundary. In particular, if one of the conditions

$$p_{k-1}^T H p_{k-1} \le 0 \quad \text{or} \quad \|s_{k-1} + \alpha_{k-1} p_{k-1}\|_2 \ge \delta \tag{2.2}$$

holds, then the solution of (2.1) lies on the boundary of the trust region and the CG iterations are terminated. If one of the conditions (2.2) hold, Steihaug's method redefines the final iterate as $s_k = s_{k-1} + \gamma_{k-1} p_{k-1}$, where γ_{k-1} is a solution of the one-dimensional trust-region problem

$$\underset{\gamma}{\text{minimize}} \;\; \mathcal{Q}(s_{k-1} + \gamma p_{k-1}) \quad \text{subject to} \quad \|s_{k-1} + \gamma p_{k-1}\|_2 \le \delta.$$

Toint redefines s_k as the Cauchy point if $p_{k-1}^T H p_{k-1} \le 0$. Either choice gives an approximate solution that is always at least as good as the Cauchy point. As a result, the underlying trust-region algorithm is globally convergent to a first-order point when endowed with an appropriate strategy for adjusting the trust-region radius (see Powell [24, 25]).

The Steihaug-Toint method terminates at the first boundary point, which implies that s_k may be a poor approximate solution of (2.1) in the constrained case. This lack of accuracy control was noted by Gould, Lucidi, Roma and Toint [15], who proposed solving the constrained problem using the *generalized Lanczos trust-region (GLTR) method.* This method solves (2.1) on an expanding sequence of subspaces generated by the vectors v_0, v_1, …, v_{k-1} associated with the Lanczos process for reducing H to tridiagonal form. The subspace minimization problem at the kth step is given by

$$\underset{y \in \mathbb{R}^k}{\text{minimize}} \;\; g^T V_k y + \tfrac{1}{2} y^T T_k y, \quad \text{subject to} \quad \|y\|_2 \le \delta, \tag{2.3}$$

where T_k is tridiagonal and V_k is the matrix of Lanczos vectors. The reduced problem (2.3) is solved using a variant of the Moré-Sorensen algorithm [20] that exploits the tridiagonal structure of the reduced Hessian T_k. Once an optimal y_k for the reduced problem has been found, the solution $s_k = V_k y_k$ is computed by repeating the Lanczos recurrence and regenerating the columns of V_k.

Erway, Gill and Griffin [6] propose the *phased sequential subspace minimization (phased-SSM) method*, which, like the GLTR method, has a constrained second phase if there is no minimizer of $\mathcal{Q}$ inside the trust-region. The iterates of the second phase solve the constrained problem on a sequence of low-dimensional subspaces. At the start of the kth iteration of the second phase, s_{k-1} is the current best estimate of the solution of (2.1), and σ_{k-1} is an approximate Lagrange multiplier associated with the constraint $\|s\|_2 = \delta$. The kth iterate (s_k, σ_k) is a primal-dual solution of the reduced equality-constraint problem

$$\underset{s \in \mathbb{R}^n}{\text{minimize}} \;\; \mathcal{Q}(s) \quad \text{subject to} \quad \|s\|_2 = \delta, \quad s \in \mathcal{S}_k = \text{span}\{s_{k-1}, z_k, s_k^a\},$$

where z_k is the current best estimate of the leftmost eigenvector of H, and s_k^a is an "accelerator" direction that is intended to increase the rate of convergence. The use of the previous iterate s_{k-1} as a generator of $\mathcal{S}_k$ guarantees that $\mathcal{Q}(s_k) < \mathcal{Q}(s_{k-1})$. The accelerator direction is obtained from a regularized Newton method applied to the constrained problem. The method includes a parameter that allows the user to take advantage of the tradeoff between the overall number of function evaluations and matrix-vector products. At one extreme, a low-accuracy solution comparable to the

Steihaug-Toint point is obtained. Overall, trust-region methods using low-accuracy subproblem solutions require fewer matrix-vector products but more function evaluations. At the other extreme, high-accuracy solutions minimize the number of function evaluations at the expense of increasing the number of matrix-vector products.

The Steihaug-Toint approach is a reliable and efficient way of applying the CG method to large-scale optimization. In addition, recent extensions to the Steihaug-Toint method add the ability to increase the accuracy of the trust-region solution when needed. However, there are some situations where the Steihaug-Toint method may not be efficient.

Preconditioning the conjugate-gradient method. In the final iterations of a trust-region method, it is usually the case that δ is relatively large and the trust-region constraint is not active. For these iterations, a large number of conjugate-gradient steps may be needed to find an approximate minimizer of $\mathcal{Q}$. The rate of convergence of the CG iterates can be significantly improved by using a preconditioner. In many applications there is a natural preconditioner available in the form of a positive-definite operator M^{-1} such that the eigenvalues of $M^{-1}H$ are clustered into a small number of groups. However, the increasing norm property of the CG iterates holds only in the *weighted* norm $\|x\|_M = (x^TMx)^{1/2}$, which imposes a trust region of the form $\|s\|_M \le \delta$. If a different preconditioner is used for each trust-region subproblem, the shape of the trust-region may change dramatically from one subproblem to the next. Since a fundamental tenet of trust-region methods is that the value of δ for one subproblem is a good estimate of δ for the next, the efficiency of the strategy for updating δ may be seriously compromised.

Convergence to second-order points. The Steihaug-Toint method and its extensions are *first-order methods*, in the sense that they are guaranteed to converge to points that satisfy the first-order necessary conditions for optimality (i.e., $g = 0$). If direct matrix factorizations are used, it is possible to approximate a global minimizer of the trust-region subproblem and thereby guarantee convergence to points that satisfy the second-order conditions for optimality, i.e., points at which the gradient is zero and the Hessian is positive semidefinite (see, e.g., Moré and Sorensen [20]). We know of no method based on the conjugate-gradient method that is guaranteed to find a global solution of (1.1) in finite-precision. For example, the Steihaug-Toint method is not guaranteed to compute a solution on the boundary when $\mathcal{Q}$ is unbounded below. Suppose that H is indefinite and $\mathcal{Q}(s)$ has a stationary point $\hat{s}$ such that $\|\hat{s}\| < \delta$. If H is positive definite on the Krylov subspace spanned by g, Hg, H^2g, $\ldots$, then CG will terminate at the interior point $\hat{s}$. Notwithstanding these theoretical difficulties, it seems worthwhile devising strategies that have the potential of providing convergence to a global solution in "most cases".

Efficiency for repeated constrained subproblems. When solving a difficult problem, it is often the case that a sequence of problems of the form (1.1) must be solved in which only the trust-region radius δ is changing. However, the Steihaug-Toint method is unable to exploit this information during the generation of the expanding sequence of subspaces.

In the next section, we propose a CG-based method that is designed to mitigate these ill-effects. (i) The method allows the use of CG preconditioning in conjunction with a standard method for updating the trust-region radius. (ii) The likelihood of approximating the global minimizer of (1.1) is increased by the computation of an approximate left-most eigenpair of H that is not based on the CG Krylov subspace. In particular, it allows the computation of a nonzero step when $g = 0$ and H is indefinite. (iii) Information garnered during the solution of one subproblem may be used to expedite the solution of the next.

3. A SSM Method with Interior-Point Acceleration.

The Steihaug-Toint method and its extensions start with the unconstrained minimization of $\mathcal{Q}$ and consider the constraint only if the unconstrained solution lies outside the trust-region. In the proposed *interior-point sequential subspace minimization (IP-SSM) method*, the inequality constrained problem (2.1) is minimized directly over a sequence of low-dimensional subspaces, giving a sequence of reduced inequality constraint problems of the form

$$\underset{s\in\mathbb{R}^n}{\text{minimize}}\ \ \mathcal{Q}(s) \quad \text{subject to} \quad \|s\| \le \delta, \quad s \in \mathcal{S}_k = \text{span}\{s_{k-1}, z_k, s_k^a\}, \tag{3.1}$$

where s_{k-1} is the current best estimate of the subproblem solution, z_k is the current best estimate of u_n (the leftmost eigenvector of H), and s_k^a is an interior-point accelerator direction. The Lanczos-CG algorithm is used to define the accelerator direction, and the Lanczos vectors also provide independent vectors for the definition of low-dimensional subspaces associated with the reduced versions of the leftmost eigenvalue problem.

3.1. Definition of the accelerator direction.

The accelerator direction s_k^a is an approximate Newton step for minimizing a primal-dual barrier function associated with the problem

$$\underset{s\in\mathbb{R}^n}{\text{minimize}}\ \ \mathcal{Q}(s) = g^Ts + \tfrac{1}{2}s^THs \quad \text{subject to} \quad \tfrac{1}{2}\delta^2 - \tfrac{1}{2}s^Ts \ge 0. \tag{3.2}$$

This is an inequality constrained optimization problem with Lagrange multiplier σ and Lagrangian function

$$L(s,\sigma) = \mathcal{Q}(s) - \sigma(\tfrac{1}{2}\delta^2 - \tfrac{1}{2}s^Ts) = \mathcal{Q}(s) - \sigma c(s),$$

where $c(s)$ denotes the constraint residual $c(s) = \frac{1}{2}\delta^2 - \frac{1}{2}s^Ts$. The necessary and sufficient conditions for a global solution of (3.2) imply the existence of a vector s and scalar σ such that

$$\begin{aligned} (H+\sigma I)s &= -g, \quad \text{with } H+\sigma I \text{ positive semidefinite,} \\ c(s)\sigma &= \ \ 0, \quad \text{with } \sigma \ge 0 \text{ and } c(s) \ge 0. \end{aligned} \tag{3.3}$$

(For a proof, see, e.g., Gay [11], Sorensen [31], Moré and Sorensen [21], or Conn, Gould and Toint [3].) The subproblem (3.2) is said to be *degenerate* if g is orthogonal to the invariant subspace associated with λ_n and $\|s_{\scriptscriptstyle L}\| < \delta$, where $s_{\scriptscriptstyle L}$ is the least-length solution of the (necessarily compatible) system $(H-\lambda_n I)s = -g$ (i.e., $s_{\scriptscriptstyle L} = -(H-\lambda_n I)^\dagger g$). In the degenerate case, there are two situations to consider. If λ_n is positive, the quantities $\sigma = 0$ and $s = -H^{-1}g$ satisfy the optimality conditions (3.3) because $\|s\| < \|s_{\scriptscriptstyle L}\| < \delta$. Alternatively, if λ_n is negative or zero, the equations $(H+\sigma I)s = -g$ cannot be used alone to determine the optimal s. However, the leftmost eigenvector u_n is a null vector of $H - \lambda_n I$, and there exists a scalar τ such that

$$(H-\lambda_n I)(s_{\scriptscriptstyle L} + \tau u_n) = -g \ \text{ and } \ \|s_{\scriptscriptstyle L} + \tau u_n\| = \delta.$$

In this case, $\sigma = -\lambda_n$ and $s = s_{\scriptscriptstyle L} + \tau u_n$ satisfy the optimality conditions (3.3) and thereby constitute a global solution of (3.2).

The accelerator direction is derived from the properties of an interior method for the solution of (3.2). The logarithmic barrier function associated with (3.2) is given by

$$\mathcal{B}^w(s) = \mathcal{Q}(s) - w \ln c(s), \tag{3.4}$$

where w is a positive weight (i.e., barrier parameter) such that $w \to 0$. The function $\mathcal{B}^w(s)$ is well-defined for all s such that $c(s) > 0$.

The modified barrier function is a generalization of the logarithmic barrier function (3.4) that allows prior knowledge of σ to be introduced into the formulation (see, e.g., Polyak [23], Conn, Gould and Toint [2]). Consider the shifted problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; \mathcal{Q}(s) \quad \text{subject to} \;\; c(s) + \mu \geq 0, \tag{3.5}$$

where μ is a positive parameter. The modified barrier function is then the conventional logarithmic barrier function associated with the shifted problem, i.e.,

$$\mathcal{B}^w(s) = \mathcal{Q}(s) - w \ln\big(c(s) + \mu\big). \tag{3.6}$$

For the weight, we choose $w = \mu\sigma_e$, where σ_e is a nonnegative estimate of the optimal σ. Differentiating $\mathcal{B}^w(s)$ with respect to s yields

$$\nabla\mathcal{B}^w(s) = g + Hs + \frac{\mu\sigma_e}{c+\mu}s = g + (H + \hat{\sigma}I)s,$$

where $\hat{\sigma} = \hat{\sigma}(s) = \mu\sigma_e/(c(s)+\mu)$. Similarly, the Hessian of $\mathcal{B}^w(s)$ is given by

$$\nabla^2\mathcal{B}^w(s) = H + \hat{\sigma}I + \omega s s^T,$$

where $\omega = \hat{\sigma}/(c+\mu) = \mu\sigma_e/(c+\mu)^2$.

The next result shows that if $\mathcal{B}^w(s)$ is defined with σ_e equal to the optimal σ of (3.3), then the optimal s may be computed using one unconstrained minimization of $\mathcal{B}^w(s)$.

Theorem 3.1. *Assume that s^* and σ^* satisfy the optimality conditions* (3.3). *Then there exists a positive $\bar{\mu}$ such that for all $\mu < \bar{\mu}$, the point s^* is an unconstrained minimizer of $\mathcal{B}^w(s) = \mathcal{Q}(s) - w\ln(c(s)+\mu)$ with $w = \mu\sigma^*$.* □

In Fig. 1 we illustrate the problem of minimizing $\mathcal{Q}(s) = g^Ts + \frac{1}{2}s^THs$ subject to $\|s\|_2 \leq \delta$, where

$$g = \begin{pmatrix} 2 \\ 4 \end{pmatrix}, \qquad H = \begin{pmatrix} 1 & 0 \\ 0 & -2 \end{pmatrix} \quad \text{and} \quad \delta = 4. \tag{3.7}$$

The function $\mathcal{Q}(s)$ is unbounded below and the optimal s lies on the boundary of the trust region. The unique global solution is given by $s = (-0.49902, -3.96875)$ and $\sigma^* = 3.00787$, with minimum $\mathcal{Q}(s^*) = -32.49962$ (all numbers are given to six figures). There is another local minimum $\mathcal{Q}(\hat{s}) = -1.0082$ at $\hat{s} = (-1.0173, 3.8684)^T$ corresponding to the multiplier $\hat{\sigma} = 0.9660$ for which $H + \hat{\sigma}I$ is indefinite. The leftmost plot in Fig. 1 gives the level curves of $\mathcal{Q}(s)$ and the position of the local solutions $\hat{s}$ and s^*. The center and rightmost plots give the modified barrier function for the values $\sigma_e = \sigma^*$ and $\sigma_e = 5$. In both cases the shift was $\mu = 1$ in order to emphasize the shifted trust-region constraint.

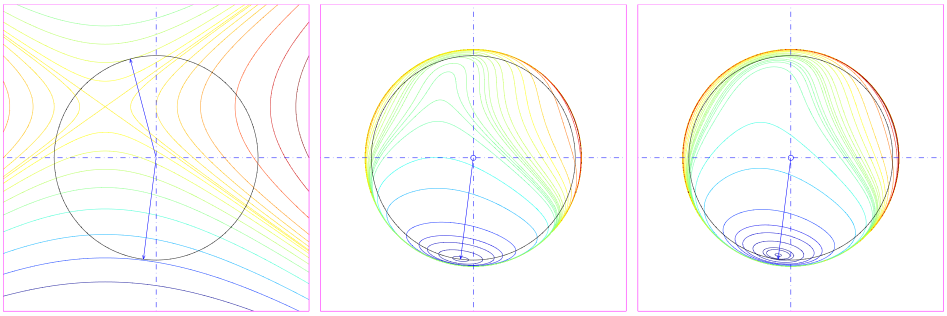

FIG. 1. *The leftmost figure depicts the level curves of the subproblem* (3.7). *The objective* $\mathcal{Q}$ *has a saddle point at* $(-2, 2)$ *and decreases along directions parallel to the* s_2 *axis that start at* $(-2, 2)$. *The interior of the circle depicts the set of points satisfying the trust-region constraint* $\|s\|_2 \leq 4$. *There are two local minimizers of* $\mathcal{Q}$*, but the direction* $s^* = (-0.49902,\ -3.96875)$ *(to six figures) corresponding to* $\sigma^* = 3.00787$ *is the global solution. The center and rightmost figures depict the level curves of the modified barrier function with* $\sigma_e = \sigma^*$ *and* $\sigma_e = 5$*, respectively.*

In FIG. 2 we illustrate a degenerate subproblem with

$$g = \begin{pmatrix} 2 \\ 0 \end{pmatrix}, \qquad H = \begin{pmatrix} 1 & 0 \\ 0 & -2 \end{pmatrix} \quad \text{and} \quad \delta = 4. \tag{3.8}$$

Both $(-\frac{2}{3},\, 3.94405)^T$ and $(-\frac{2}{3},\, -3.94405)^T$ give the same global minimum of $\mathcal{Q}$. The leftmost plot in FIG. 2 gives the level curves of $\mathcal{Q}(s)$ and the position of the two global minimizers. The center and rightmost plots give the modified barrier function for the values $\sigma_e = \sigma^*$ and $\sigma_e = 5$. As in example (3.7), the shift was $\mu = 1$.

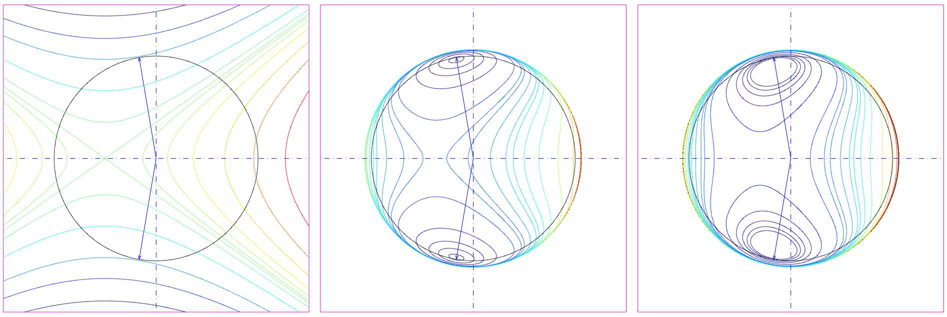

FIG. 2. *The leftmost figure depicts the level curves of the degenerate subproblem* (3.8). *The trust-region solution is not unique—both of the vectors shown is a constrained global minimizer of* $\mathcal{Q}(s)$. *The center and rightmost figures depict the level curves of the modified barrier function with* $\sigma_e = \sigma^*$ *and* $\sigma_e = 5$*, respectively.*

Forsgren and Gill [8] describe a primal-dual barrier function that is minimized simultaneously with respect to both the primal *and* dual variables. A benefit of this approach is that the quality of the dual variables may be monitored explicitly during the solution of the subproblem. In the case of trust-region subproblem, this allows more flexibility in compelling the iterates to converge to a solution with $\sigma \in (-\lambda_n, \infty)$. The primal-dual formulation of the conventional barrier function (3.4) is given by

$$\mathcal{M}^{w,\nu}(s,\sigma) = \mathcal{Q}(s) - w \ln c(s) - \nu\Big(w \ln\big(c(s)\sigma/w\big) + \big(w - c(s)\sigma\big)\Big),$$

where ν is a positive scalar (usually chosen to be one). If we consider this function for the shifted problem (3.5), we obtain

$$\mathcal{Q}(s) - w \ln\big(c(s)+\mu\big) - \nu\Big(w \ln\big((c(s)+\mu)\sigma/w\big) + \big(w - (c(s)+\mu)\sigma\big)\Big),$$

which has the same minimizers as the function

$$\mathcal{Q}(s) - w \ln\big(c(s)+\mu\big) - \nu\Big(w \ln\big((c(s)+\mu)\sigma\big) + \big(w - (c(s)+\mu)\sigma\big)\Big).$$

If we define $w = \mu\sigma_e$, then the primal-dual modified barrier function is given by

$$\mathcal{M}^{\sigma_e,\nu}(s,\sigma) = \mathcal{Q}(s) - \mu\sigma_e \ln\big(c(s)+\mu\big) - \nu\mu\sigma_e \ln\big((c(s)+\mu)\sigma\big) - \nu\big(\mu(\sigma_e - \sigma) - c(s)\sigma\big),$$

which may be written more concisely as

$$\mathcal{M}^{\sigma_e,\nu}(s,\sigma) = \mathcal{Q}(s) - \mu\sigma_e \ln\big((c(s)+\mu)^{\nu+1}\sigma^{\nu}\big) - \nu\big(\mu(\sigma_e - \sigma) - c(s)\sigma\big). \tag{3.9}$$

Differentiating $\mathcal{M}^{\sigma_e,\nu}(s,\sigma)$ with respect to both s and σ gives

$$\nabla\mathcal{M}^{w,\nu}(s,\sigma) = \begin{pmatrix} g + (H+\sigma I)s - (1+\nu)\big(\sigma - \hat{\sigma}\big)s \\ \nu d(\sigma - \hat{\sigma}) \end{pmatrix},$$

where $d = (c+\mu)/\sigma$ and, as above, $\hat{\sigma} = \mu\sigma_e/(c(s)+\mu)$. Similarly, the Hessian is given by

$$\nabla^2\mathcal{M}^{w,\nu}(s,\sigma) = \begin{pmatrix} H_M & -\nu s \\ -\nu s^T & \nu d\hat{\sigma}/\sigma \end{pmatrix},$$

where H_M is the Hessian of $\mathcal{M}^{w,\nu}$ with respect to s, i.e.,

$$H_M = H + \big(\sigma + (1+\nu)(\hat{\sigma} - \sigma)\big)I + (1+\nu)\frac{\hat{\sigma}}{\sigma d}ss^T.$$

The next result is a primal-dual version of Theorem 3.1. It shows that if σ_e is equal to the optimal multiplier, then the optimal values of both s and σ may be computed using one unconstrained minimization of $\mathcal{M}^{w,\nu}$.

THEOREM 3.2. *If s^* and σ^* satisfy the optimality conditions* (3.3)*, then (s^*, σ^*) is a stationary point of the function $\mathcal{M}^{\sigma_e,\nu}(s,\sigma)$ with $\sigma_e = \sigma^*$. Moreover, if $\nu > 0$, then there exists a positive scalar $\bar{\mu}$ such that for all $\mu < \bar{\mu}$, the point (s^*, σ^*) is an unconstrained minimizer of $\mathcal{M}^{\sigma_e,\nu}(s,\sigma)$ with $\sigma_e = \sigma^*$.*

Proof. The result follows by evaluating $\nabla\mathcal{M}^{\sigma_e}(s,\sigma)$ and $\nabla^2\mathcal{M}^{\sigma_e}(s,\sigma)$ with $\sigma_e = \sigma^*$ at the point (s^*, σ^*) and using the optimality conditions (3.3). □

The practical significance of this result is that a good estimate σ_e of the optimal multiplier σ^* will define a primal-dual function that can be minimized for an improved estimate of (s^*, σ^*).

In the remaining discussion we restrict our attention to the case $\nu = 1$. The idea is to use the optimal multiplier σ from the subspace minimization problem (3.1) to define the multiplier estimate σ_e for the next primal-dual function (3.9). The accelerator direction is then an approximate Newton direction associated with minimizing the function

$$\mathcal{M}^{\sigma_e}(s,\sigma) = \mathcal{Q}(s) - \mu\sigma_e \ln\bigl((c(s)+\mu)^2\sigma\bigr) - \bigl(\mu(\sigma_e - \sigma) - c(s)\sigma\bigr), \tag{3.10}$$

which is defined for all (s,σ) such that $c(s) > -\mu$ and $\sigma > 0$.

3.2. Calculation of the accelerator direction. Given an approximate minimizer (s,σ) of $\mathcal{M}^{\sigma_e}$, the Newton equations for the next iterate $(s+p, \sigma+q)$ are:

$$\begin{pmatrix} H\bigl(\sigma + 2(\hat\sigma - \sigma)\bigr) + 2\frac{\hat\sigma}{\sigma d} ss^T & -s \\ -s^T & \hat\sigma d/\sigma \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix} = -\begin{pmatrix} g + H(\sigma)s - 2\bigl(\sigma - \hat\sigma\bigr)s \\ d(\sigma - \hat\sigma) \end{pmatrix}, \tag{3.11}$$

where $H(\sigma) = H + \sigma I$. An *approximate* Newton method is defined by replacing $\hat\sigma$ by σ everywhere in the matrix $\nabla^2\mathcal{M}^{\sigma_e}(s,\sigma)$. This gives the equations

$$\begin{pmatrix} H(\sigma) + (2/d)ss^T & -s \\ -s^T & d \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix} = -\begin{pmatrix} g + H(\sigma)s - 2(\sigma - \hat\sigma)s \\ d(\sigma - \hat\sigma) \end{pmatrix}. \tag{3.12}$$

As the iterates converge, it must hold that $\hat\sigma \to \sigma$ and the solution of (3.12) approaches the solution of (3.11).

Finally, we multiply the last equation and last variable by $d^{-\frac{1}{2}}$ and $d^{\frac{1}{2}}$, respectively, to improve the scaling when $\sigma \to 0$. This gives

$$\begin{pmatrix} H(\sigma) + 2\bar s\bar s^T & -\bar s \\ -\bar s^T & 1 \end{pmatrix} \begin{pmatrix} p \\ \bar q \end{pmatrix} = -\begin{pmatrix} g + H(\sigma)s - 2d^{\frac{1}{2}}\bigl(\sigma - \hat\sigma\bigr)\bar s \\ d^{\frac{1}{2}}(\sigma - \hat\sigma) \end{pmatrix}, \tag{3.13}$$

where $\bar s = d^{-\frac{1}{2}}s$ and $q = d^{-\frac{1}{2}}\bar q$. These equations are positive definite in a neighborhood of a minimizer (s,σ) such that $\sigma \in (-\lambda_n, \infty)$, and they may be solved using the CG method. If a direction of negative or zero curvature is detected, the direction is used to update a lower bound on the best estimate of σ (see Section 3.5). It is not necessary to minimize $\mathcal{M}^{\sigma_e}(s,\sigma)$ to high accuracy because the accuracy of the accelerator step affects only the *rate* of convergence of the SSM method. In most cases, only one Newton iteration need be performed (see Section 4).

The CG method may be used in conjunction with a preconditioner of the form

$$P = \begin{pmatrix} M(\sigma) + 2\bar s\bar s^T & -\bar s \\ -\bar s^T & 1 \end{pmatrix},$$

where $M(\sigma)$ is a positive-definite approximation to $H(\sigma)$. The equations $Pv = u$ used to apply the preconditioner are solved by exploiting the equivalence of the systems:

$$\begin{pmatrix} M(\sigma) + 2\bar s\bar s^T & -\bar s \\ -\bar s^T & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \tag{3.14a}$$

and

$$\begin{pmatrix} M(\sigma) & \bar s \\ \bar s^T & -1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} u_1 + 2u_2\bar s \\ -u_2 \end{pmatrix} \tag{3.14b}$$

(see Forsgren, Gill and Griffin [9]). Equations (3.14b) are solved analytically if $M(\sigma)$ is diagonal. Alternatively, if $M(\sigma)$ is defined using an incomplete Cholesky factorization of $H(\sigma)$ we solve (3.14b) using the block factorization:

$$\begin{pmatrix} M(\sigma) & \bar{s} \\ \bar{s}^T & -1 \end{pmatrix} = \begin{pmatrix} I & \\ w^T & 1 \end{pmatrix} \begin{pmatrix} M(\sigma) & \\ & -\bigl(1 + w^T\bar{s}\bigr) \end{pmatrix} \begin{pmatrix} I & w \\ & 1 \end{pmatrix},$$

where w satisfies $M(\sigma)w = \bar{s}$. Thus, the preconditioned CG computations may be arranged so that only solves with $M(\sigma)$ are required.

3.3. Properties of the accelerator direction. The conventional primal-dual interior-point approach to solving the trust-region subproblem is based on finding s and σ that satisfy the perturbed optimality conditions

$$\begin{aligned} (H + \sigma I)s &= -g, & \sigma &> 0, \\ c(s)\sigma &= \mu, & c(s) &> 0 \end{aligned} \tag{3.15}$$

for a sequence of decreasing values of the positive parameter μ. Let $F(s,\sigma)$ denote the vector-valued function whose components are the residuals $(H + \sigma I)s + g$ and $c(s)\sigma - \mu$. Given an approximate zero (s,σ) of F such that $c(s) > 0$ and $\sigma > 0$, the Newton equations for the next iterate $(s + p, \sigma + q)$ are:

$$\begin{pmatrix} H + \sigma I & s \\ -\sigma s^T & c(s) \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix} = -\begin{pmatrix} g + (H + \sigma I)s \\ c(s)\sigma - \mu \end{pmatrix}.$$

The assumption that $\sigma > 0$ implies that it is safe to divide the last equation by $-\sigma$ to give the symmetrized equations:

$$\begin{pmatrix} H + \sigma I & s \\ s^T & -d \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix} = -\begin{pmatrix} g + (H + \sigma I)s \\ d(\hat{\sigma} - \sigma) \end{pmatrix},$$

where $d = c(s)/\sigma$ and $\hat{\sigma} = \mu/c(s)$. The presence of the nonzero $(2,2)$ block implies that the conventional interior-point approach defines a *regularization* of Newton's method for a solution of the optimality conditions (3.3). The regularized solution lies on the central path of solutions $\bigl(s(\mu), \sigma(\mu)\bigr)$ of (3.15) (see, e.g., [10]). This implies that the regularized solution $\bigl(s(\mu), \sigma(\mu)\bigr)$ will be different from (s^*, σ^*) for a given nonzero μ. Moreover, the influence of the regularization on the Newton equations diminishes as $\mu \to 0$.

These considerations suggest that we seek an alternative "exact" regularization that allows the use of a fixed value of μ, but does not perturb the regularized solution. Consider the perturbed optimality conditions

$$\begin{aligned} (H + \sigma I)s &= -g, & \sigma &> 0, \\ c(s)\sigma &= \mu(\sigma_e - \sigma), & c(s) &> -\mu, \end{aligned} \tag{3.16}$$

where σ_e is a nonnegative estimate of σ^*. If $\sigma_e = \sigma^*$, these conditions are satisfied by (s^*, σ^*) for any positive μ. The symmetrized Newton equations associated with conditions (3.16) are

$$\begin{pmatrix} H + \sigma I & s \\ s^T & -d \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix} = -\begin{pmatrix} g + (H + \sigma I)s \\ d(\hat{\sigma} - \sigma) \end{pmatrix},$$

where now, $d = (c(s) + \mu)/\sigma$ and $\hat{\sigma} = \mu\sigma_e/(c(s) + \mu)$. Forsgren, Gill and Griffin [9] show that these equations are equivalent to the so-called *doubly-augmented system*:

$$\begin{pmatrix} H + \sigma I + \frac{2}{d} s s^T & -s \\ -s^T & d \end{pmatrix} \begin{pmatrix} p \\ q \end{pmatrix} = -\begin{pmatrix} g + (H + \sigma I)s - 2(\sigma - \hat{\sigma})s \\ d(\sigma - \hat{\sigma}) \end{pmatrix}.$$

which is identical to the approximate Newton system (3.12). It follows that the equations for the accelerator direction also define a regularized form of Newton's method applied to the optimality conditions.

The next result defines the precise effect of the regularization parameter μ in the degenerate case.

THEOREM 3.3 (Regularization of the degenerate case). *Let (s,σ) be a solution of the trust-region subproblem such that:* (i) $\|s\| = \delta$*;* (ii) $H+\sigma I$ *is positive semidefinite and singular;* (ii) $g \in \operatorname{null}(H+\sigma I)^{\perp}$*; and* (iii) $\|(H+\sigma I)^{\dagger} g\| < \delta$. *If the leftmost eigenvalue of H has algebraic multiplicity 1, then the augmented system matrix*

$$\begin{pmatrix} H+\sigma I + (2/d)ss^T & -s \\ -s^T & d \end{pmatrix} \tag{3.17}$$

is positive definite.

Proof. Assumptions (i)–(iii) imply that (s,σ) is a constrained degenerate solution. In particular, it holds that $\sigma = -\lambda_n$, where λ_n is the leftmost eigenvalue of H. A solution s of the trust-region subproblem is given by

$$s = -(H-\lambda_n I)^{\dagger} g + \beta z, \tag{3.18}$$

where z is a unit vector such that $z \in \operatorname{null}(H-\lambda_n I)$ and β is a nonzero scalar such that $\|s\| = \delta$. Consider the following decomposition of (3.17):

$$\begin{pmatrix} H+\sigma I + \frac{2}{d}ss^T & -s \\ -s^T & d \end{pmatrix} = \begin{pmatrix} I & -\frac{1}{d}s \\ 0 & 1 \end{pmatrix} \begin{pmatrix} H-\lambda_n I + \frac{1}{d}ss^T & 0 \\ 0 & d \end{pmatrix} \begin{pmatrix} I & 0 \\ -\frac{1}{d}s^T & 1 \end{pmatrix}.$$

Assume that $H+\sigma I + (2/d)ss^T$ is not positive definite. Then there exists a nonzero p such that $p^T(H-\lambda_n I + (2/d)ss^T)p \le 0$. As $H-\lambda_n I$ is positive semidefinite, it must hold that $p \in \operatorname{null}(H-\lambda_n I)$ and $s^Tp = 0$. Moreover, since $(H-\lambda_n I)^{\dagger} g \in \operatorname{range}(H-\lambda_n I)$, it must hold that $s^Tp = \beta z^Tp = 0$, which implies that $z^Tp = 0$. But this is only possible if $\dim(\operatorname{null}(H-\lambda_n I)) > 1$. Thus $H+\sigma I + (2/d)ss^T$ must be positive definite and the result follows. ☐

3.4. Calculation of the approximate leftmost eigenpair. The approximate Newton equations (3.13) are solved using the Lanczos-CG variant of the preconditioned conjugate-gradient method. During the evaluation of the Lanczos process, the Lanczos vectors are used to generate the subspace associated with an SSM method for an estimate of the leftmost eigenpair of H. The estimate is computed by solving the reduced generalized eigenproblem

$$\underset{z\in\mathbb{R}^n}{\text{minimize}}\;\; z^THz \quad \text{subject to} \;\; \|z\|_2 = 1, \;\; z \in \mathcal{Z}_k = \operatorname{span}\{z_{k-1}, \bar{v}_k, \bar{v}_{k-1}\}, \tag{3.19}$$

where z_{k-1} is the leftmost eigenvector estimate from the previous CG iteration, and $\bar{v}_k$ and $\bar{v}_{k-1}$ are the first $n-1$ components of the two most recently computed Lanczos vectors. Given the matrix Z_k whose columns form a maximally linearly independent subset of $\{z_{k-1}, \bar{v}_k, \bar{v}_{k-1}\}$, the solution z_k of (3.19) is defined as $z_k = Z_k w_k$, where w_k solves the reduced problem

$$\underset{y}{\text{minimize}}\;\; y^TZ_k^THZ_ky \quad \text{subject to} \;\; \|Z_ky\|_2 = 1.$$

This problem has at most three dimensions, and is solved in closed form. Once z_k has been determined, the leftmost eigenvalue is estimated by the Rayleigh quotient

$\zeta_k = z_k^T H z_k$. The inclusion of z_{k-1} as a generator of $\mathcal{Z}_k$ ensures that the Rayleigh quotients decrease monotonically.

The calculation of $Z_k^T H Z_k$ requires the vectors Hz_{k-1}, $H\bar{v}_k$ and $H\bar{v}_{k-1}$. The vector Hz_{k-1} is the solution of the previous reduced eigenproblem. The vectors $H\bar{v}_k$ and $H\bar{v}_{k-1}$ are available as part of the two-term Lanczos recurrence. For the next step, the vector Hz_k is defined in terms of the identity $Hz_k = HZ_k w_k$, which involves a simple linear combination of Hz_{k-1}, $H\bar{v}_k$ and $H\bar{v}_{k-1}$. It follows that once Hz_0 is calculated, no additional matrix-vector products are needed. The calculation of the eigenpair is summarized in Algorithm ***ssmEig*** below.

For the first outer iteration (i.e., $j = 0$) a random vector is used for z_0, the initial estimate of the leftmost eigenvector of H_0. In subsequent iterations, z_0 is defined as the eigenvector estimate from the previous trust-region subproblem. As the sequence $\{H_j\}$ converges, z_0 should be a good estimate of the leftmost eigenvector for each subproblem.

Algorithm *ssmEig*.

```
$\big(z, \zeta, [Hz], \sigma_\ell\big) = \boldsymbol{ssmEig}(z, \bar{v}_k, \bar{v}_{k-1}, [Hz], [H\bar{v}_k], [H\bar{v}_{k-1}], \sigma_\ell)$;
    Define $Z$ from a maximally linearly independent subset of $\bar{v}_k$, $\bar{v}_{k-1}$, and $z$;
    Form $Z^T HZ$ and $Z^T Z$ from $z$, $\bar{v}_k$, $\bar{v}_{k-1}$, $[H\bar{v}_k]$, $[H\bar{v}_{k-1}]$ and $[Hz]$;
    $w := \operatorname{argmin}_y \big\{ y^T Z^T HZy \ : \ \|Zy\|_2 = 1 \big\}$;
    $z := Zw$;   $\zeta = z^T Hz$;
    $[Hz] := [HZ]w$;
    $\sigma_\ell = \max\{|\zeta|, \sigma_\ell\}$;
```

3.5. Approximate safeguarding. The algorithm maintains two approximate solutions: (s_e, σ_e) and (s_a, σ_a). The pair (s_e, σ_e) is the solution of the subspace minimization problem (3.1). The accelerator pair (s_a, σ_a) is the most recently computed estimate of a minimizer of $\mathcal{M}^{\sigma_e}(s, \sigma)$.

At each iteration, a safeguarding algorithm ensures that both σ_a and σ_e are strictly positive and not less than σ_ℓ, a current greatest lower bound on $-\lambda_n$. The algorithm also attempts to adjust σ_a so that matrix $H + \sigma_a I$ of (3.13) is positive definite. In order to maintain the monotonicity of $\mathcal{Q}$, the value of s_e is always the solution of the subspace minimization problem. However, σ_e may be overwritten by σ_a if $\sigma_e < \sigma_\ell < \sigma_a$. In addition, (s_a, σ_a) is replaced by (s_e, σ_e) if $\sigma_a < \sigma_\ell < \sigma_e$. In the event that both σ_a and σ_e are less than σ_ℓ, the leftmost eigenpair is used to update σ_a and σ_e.

Finally, in order that the matrix of (3.13) is positive definite, the IP-SSM method also ensures $c(s_a) + \mu > 0$. If $c(s_a) \le -\mu$, the accelerator direction is replaced by s_e, with a suitable rescaling (if necessary) to guarantee that s_a lies exactly on the boundary.

If $\sigma \in (-\lambda_n, \infty)$ and $d > 0$, the Newton system (3.13) is positive definite and can be solved using CG. The following result shows that if CG computes a conjugate direction p of negative curvature, then the approximate Newton equations (3.12) are indefinite, and p is used to update the estimate of the leftmost eigenvector.

THEOREM 3.4. *If p is a direction of negative curvature for the matrix*

$$B = \begin{pmatrix} H + \sigma I + (2/d)ss^T & -s \\ -s^T & d \end{pmatrix},$$

```
Algorithm safeguard.
$(s_a, \sigma_a, \sigma_e, [Hs_a]) = \textbf{\textit{safeguard}}(s_a, \sigma_a, s_e, \sigma_e, \zeta, z, \sigma_\ell, [Hs_a], [Hs_e], [Hz])$;
    Choose $\sigma_{\min} > 0$;
    if $\sigma_a < \sigma_\ell$ and $\sigma_\ell < \sigma_e$ then
        $\sigma_a := \max\{\sigma_e, \sigma_{\min}\}$;   $s_a := s$;   $[Hs_a] := [Hs]$
    else if $\sigma_e < \sigma_\ell$ and $\sigma_\ell < \sigma_a$ then
        $\sigma_e := \sigma_a$;
    else if $\sigma_a < \sigma_\ell$ and $\sigma_e < \sigma_\ell$ then
        $\sigma_e := -\zeta$;   $\sigma_a := -\zeta$;   $s_a := \delta \times z$;   $[Hs_a] := [Hz]$;
    end
```

with d a positive scalar, then the vector of first n elements of p is a direction of negative curvature for $H + \sigma I$.

Proof. As p is a direction of negative curvature for B, we have

$$p^T B p = p^T \begin{pmatrix} H + \sigma I + (2/d) s s^T & -s \\ -s^T & d \end{pmatrix} p < 0.$$

Let $\hat{p}$ and ρ denote the first n elements and the last element of p respectively. A simple calculation yields

$$\begin{aligned} p^T B p &= \hat{p}^T (H + \sigma I)\hat{p} + \frac{2}{d}(s^T\hat{p})^2 - 2\rho s^T \hat{p} + \rho^2 d \\ &= \hat{p}^T (H + \sigma I)\hat{p} + \frac{1}{d}(s^T\hat{p})^2 + \frac{1}{d}\big((s^T\hat{p})^2 - 2(s^T\hat{p})(\rho d) + (\rho d)^2\big) \\ &= \hat{p}^T (H + \sigma I)\hat{p} + \frac{1}{d}\big((s^T\hat{p})^2 + (s^T\hat{p} - \rho d)^2\big) < 0. \end{aligned}$$

It follows that $\hat{p}^T(H + \sigma I)\hat{p} < -\big((s^T\hat{p})^2 + (s^T\hat{p} - \rho d)^2\big)/d < 0$, as required. □

4. Numerical Results.

The Steihaug-Toint and IP-SSM methods were implemented and run in MATLAB. Numerical results are given for unconstrained problems from the CUTEr test collection (see Bongartz et al. [1] and Gould, Orban and Toint [16]). The test set was constructed using the CUTEr interactive `select` tool, which allows the identification of groups of problems with certain characteristics. In our case, the `select` tool was used to identify the twice-continuously differentiable unconstrained problems for which the number of variables can be varied.

For all problems, the dimension n was chosen so that $n \geq 1000$, where $n = 1000$ was the default unless otherwise recommended in the CUTEr documentation. A combination line-search trust-region method was used to define the update to the trust-region radius. For all problems, the initial trust-region radius was chosen to be one.

The trust-region method is considered to have solved a CUTEr problem successfully when a trust-region iterate x_j satisfies

$$\|g(x_j)\|_2 \leq \max\{\epsilon \|g(x_0)\|_2, \epsilon |f(x_0)|, 10^{-5}\}, \tag{4.1}$$

where $\epsilon = 10^{-6}$. If x_0 is a non-optimal stationary point, the presence of the term $f(x_0)$ prevents the trust-region algorithm from terminating at x_0. If a solution is not found within $2n$ iterations, the iterations are terminated and the algorithm is considered to have failed. Throughout this section we refer to s_j as the approximate solution of the jth trust-region problem.

4.1. Solving the reduced subproblem. At the core of the algorithm is a reduced trust-region subproblem (3.1) with at most three dimensions. Given the matrix P_k whose columns form a maximally linearly independent subset of $\{s_{k-1}, z_k, s_k^a\}$, the solution s_k of (3.1) may be written as $s_k = P_k w_k$, where w_k solves the reduced problem

$$\underset{y}{\text{minimize}} \;\; \mathcal{Q}(P_k y) \equiv g^T P_k y + \tfrac{1}{2} y^T P_k^T H P_k y, \quad \text{subject to} \;\; \|P_k y\|_2 \le \delta. \tag{4.2}$$

A maximally linearly independent subset of the vectors $\{s_{k-1}, z_k, s_k^a\}$ is found using a QR decomposition with column interchanges. As in algorithm ***ssmEig***, the matrices $P_k^T H P_k$ and $P_k^T P_k$ can be formed with no additional matrix-vector products. The vector Hs_k is defined in terms of the identity $Hs_k = HP_k w_k$, which involves a simple linear combination of Hs_{k-1}, Hz_k, and Hs_k^a. The matrices $P_k^T H P_k$ and $P_k^T P_k$ are symmetrized in each case.

The reduced problem is solved using a modified version of the Moré-Sorensen algorithm [20] that computes an exact left-most eigenpair of the 3×3 shifted Hessian. At each iteration, the Cholesky factorization of $P_k^T H P_k + \sigma P_k^T P_k$ is used to compute a vector w_R such that

$$(P_k^T H P_k + \sigma P_k^T P_k) w_R = -P_k^T g.$$

The accuracy of an approximate solution of (4.2) is determined by preassigned tolerances γ_1, $\gamma_2 \in (0,1)$. On termination, the approximate solution of (3.1) is $s = s_R + s_N$, where $s_R = P_k w_R$ and $s_N = P_k w_N$, with w_N defined as the zero vector if $(1-\gamma_1)\delta \le \|s_R\| \le (1+\gamma_1)\delta$, or a leftmost eigenvector of $P_k^T H P_k + \sigma P_k^T P_k$ if $\|s_R\| < (1-\gamma_1)\delta$. The resulting value of s satisfies

$$\mathcal{Q}(s) - \mathcal{Q}^* \le \gamma_1(2-\gamma_1)\max(|\mathcal{Q}^*|, \gamma_2), \quad \text{and} \quad \|s\| \le (1+\gamma_1)\delta, \tag{4.3}$$

where $\mathcal{Q}^*$ denotes the global minimum of (4.2) (see Moré and Sorensen [20]).

The reduced trust-region subproblem must be solved to an accuracy that is at least as good as that required for the full problem. Suitable values for the constants γ_1 and γ_2 of (4.3) are $\gamma_1 = \min\{10^{-1}\tau, 10^{-6}\}$ and $\gamma_2 = 0$, where τ denotes the accuracy required in the full space.

The calculations associated with the solution of the reduced problem are given in algorithm ***ssmSolve***, with $s_e = s_{k-1}$, $z = z_k$, and $s_a = s_k^a$. The inclusion of the best approximation s_{k-1} in $\text{span}\{s_{k-1}, z_k, s_k^a\}$ guarantees that $\mathcal{Q}$ decreases at each step. Care must be taken to separate the nullspace components of the Moré-Sorensen solution to test the optimality conditions correctly, i.e., both s_e and s_R are stored. The Moré-Sorensen algorithm also returns the optimal σ for the reduced problem, which is denoted by σ_e in ***ssmSolve***.

The accelerator direction s_k^a is defined as an approximate solution of the Newton equations (3.13). The relevant calculations are described in algorithm ***ipAccelerator*** below.

4.2. Solving the trust-region subproblem. At the start of each subspace minimization, the regularization parameter μ is initialized at 10^{-1}. At the end of the

Algorithm *ssmSolve*.

```
$\big(s_e, s_R, \sigma_e, [Hs_e], [Hs_R]\big) = \boldsymbol{ssmSolve}\big(s_e, s_a, z, [Hs_e], [Hz], [Hs_a]]\big)$;
    Define $P$ from a maximally linearly independent subset of $s_e$, $z$ and $s_a$;
    Form $P^THP$, $P^TP$ and $P^Tg$ from $s_e$, $z$, $s_a$, $[Hs]$, $[Hz]$, and $[Hs_a]$;
    $y := \text{argmin}\,\big\{\, g^TPy + \frac{1}{2}y^TP^THPy \ :\ \|Py\|_2 \le \delta \,\big\}$;
    $s_e := Py$;   $[Hs_e] := [HP]y$;
```

Algorithm *ipAccelerator*.

```
$\big(s_a, [Hs_a], \sigma_a, \sigma_\ell\big) = \boldsymbol{ipAccelerator}\big(s_a, [Hs_a], \sigma_a, \sigma_\ell\big)$;
    $\hat{\sigma} := \mu\sigma_e / \big(c(s_a) + \mu\big)$;
    Find an approximate solution $(p, q)$ of (3.13) with $(s, \sigma) = (s_a, \sigma_a)$;
    Update $\sigma_\ell$ if a direction of negative curvature was found by Lanczos-CG;
    $\alpha_\sigma :=$ if $q < 0$ then $(\sigma_a + q - \sigma_\ell)/q$ else $+\infty$;
    $\alpha_s :=$ the positive root of $c(\alpha s_a) + \mu = 0$;
    $\alpha_M := \min\{1, (1-\mu)\alpha_\sigma, (1-\mu)\alpha_s\}$;
    Compute $\alpha$ $(0 < \alpha \le \alpha_M)$ satisfying the Wolfe line search conditions for $\mathcal{M}^{\sigma_e}$;
    $s_a := s_a + \alpha p$;   $\sigma_a := \sigma_a + \alpha q$;   $[Hs_a] := [Hs_a] + \alpha[Hp]$;
```

kth step a new μ is computed such that

$$\mu = \begin{cases} \min\{10^{-1}, \frac{1}{2}\bar{\mu}\} & \text{if } \zeta < 0; \\ 10^{-1}/(1+k) & \text{otherwise,} \end{cases} \tag{4.4}$$

where ζ is the best estimate of the left-most eigenvalue and $\bar{\mu} = -2\|s_a\|^2/\zeta$. Including the iteration index allows for μ to be gradually decreased in order to obtain an improved accelerator direction.

More details of proposed method are presented in Algorithm ***IP-SSM*** below. An important feature of the method is that σ_e and the approximate leftmost eigenpair from one outer iteration are used to initialize the next. At the start of each subproblem, the initial value of the interior-point accelerator variable σ_a is set to σ_e, as long as σ_e is larger than $\sigma_{\min}$, a preassigned constant that specifies the smallest allowable value of σ_a. (In the final iterations of the trust-region method the trust-region constraint will be inactive and $\sigma_e = 0$.)

Algorithm *IP-SSM*.

```
$\big(s_e, [Hs_e], \sigma_e, z\big) = \boldsymbol{IP\text{-}SSM}\big(g, \delta, \sigma_e, z\big)$;
    Specify $\tau > 0$;   $\sigma_{\min} > 0$;   $\mu := 10^{-1}$;
    $\sigma_\ell := 0$;   $\sigma_a := \max\{\sigma_e, \sigma_{\min}\}$;
    $s_e := -g$;   $[Hs_e] := Hs_e$;   $s_a := 0$;   $[Hs_a] := 0$;
    $r_e := \|g + (H + \sigma_e I)s_e\|_{M^{-1}} + \sigma_e |c(s_e)|$;
    while $k < k_{\max}$ and $r_e > \tau$ do
        $\big(s_a, [Hs_a], \sigma_a, \sigma_\ell\big) := \boldsymbol{ipAccelerator}\big(s_a, [Hs_a], \sigma_a, \sigma_\ell\big)$;
        $\big(\tilde{s}, s_R, \tilde{\sigma}, [H\tilde{s}], [Hs_R]\big) := \boldsymbol{ssmSolve}\big(s_e, s_a, z, [Hs_e], [Hz], [Hs_a]\big)$;
    $\tilde{r} := \|g + (H + \tilde{\sigma} I)s_R\|_{M^{-1}} + \tilde{\sigma}|c(\tilde{s})|$;
    if $\tilde{r} \le r_e$ and $\tilde{\sigma} > \sigma_\ell$ then $s_e := \tilde{s}$;   $\sigma_e := \tilde{\sigma}$;   $[Hs_e] := [H\tilde{s}]$;   $r_e := \tilde{r}$;
    $\big(s_a, \sigma_a, \sigma_e, [Hs_a]\big) :=$ safeguard$(s_a, \sigma_a, s_e, \sigma_e, \zeta, z, \sigma_\ell, [Hs_a], [Hs_e], [Hz], \delta)$;
    if $r_a < r_e/10$ then $\sigma_e := \sigma_a$;
    Update $\mu$ using (4.4);   Compute $c(s_a)$;
    if $c(s_a) + \mu \le 0$ then
        $s_a := s_e$;   $\sigma_a := \sigma_e$;   $[Hs_a] := [Hs_e]$;
        Update $\mu$ using (4.4);   Compute $c(s_a)$;
        if $c(s_a) + \mu \le 0$ then $s_a := \delta \times s_a/\|s_a\|_2$;   $[Hs_a] := [Hs_a]/\|s_a\|_2$;
    end
    if $r_a < \tau$ and $r_e > \tau$ then
        if $\mathcal{Q}(s_a) \le \mathcal{Q}(s_e)$ then
            $s_e := s_a$;   $\sigma_e := \sigma_a$;   break;
        end
    end
    $\sigma_a := \max\{\sigma_a, \sigma_{\min}\}$;   $k := k + 1$;
end
```

Algorithm ***IP-SSM*** is terminated with final iterate (s, σ) given by either (s_e, σ_e) or (s_a, σ_a), depending on the values of the residuals r_e and r_a such that

$$r_e = \|g + (H + \sigma_e I)s_R\|_{M^{-1}} + \sigma_e|c(s_e)|, \quad \text{and} \tag{4.5a}$$

$$r_a = \|g + (H + \sigma_a I)s_a\|_{M^{-1}} + \sigma_a|c(s_a)|. \tag{4.5b}$$

The idea is to choose the iterate with the least residual, subject to the requirement that s improves on the Cauchy step. Given a positive tolerance τ, the final iterate is $(s, \sigma) = (s_e, \sigma_e)$ if $r_e \le \tau$, or $(s, \sigma) = (s_a, \sigma_a)$ if $r_a \le \tau < r_e$ and $\mathcal{Q}(s_a) \le \mathcal{Q}(s_e)$. The initial value $s_e = -g$ guarantees that every subspace minimizer improves on the Cauchy step. The condition $\mathcal{Q}(s_a) \le \mathcal{Q}(s_e)$ ensures that this improvement is inherited by the final point.

4.3. Termination of the trust-region subproblem. The principal termination condition for the trust-region subproblem is based on the Dembo-Eisenstat-Steihaug criterion [4]. As above, let s_j denote the approximate solution of the jth trust-region subproblem. In our implementation of the Steihaug-Toint method, the Lanczos-CG method terminates successfully with a point s_j inside the trust region if

$$\|g_j + H_j s_j\|_{M_j^{-1}} \le \tau_1 \|g_j\|_{M_j^{-1}}, \quad \text{where} \quad \tau_1 = \min\big\{10^{-1}, \|g_j\|_{M_j^{-1}}^{0.1}\big\}. \tag{4.6}$$

This condition forces a relative decrease in the residual comparable to that required by Gould et al. [15]. The IP-SSM method terminates successfully with direction s_j if

$$\min\{r_e, r_a\} < \tau_1 \|g_j\|_{M_j^{-1}} \tag{4.7}$$

(see (4.5) for the definitions of r_e and r_a). A limit of 20 Lanczos vectors is imposed on all calculations involving the Lanczos-CG process. If this limit is reached during the accelerator calculation, the Lanczos-CG iterate with the smallest residual is returned. In addition, the number of subspace minimization steps in ***IP-SSM*** is limited to 10. These limits are designed to avoid excessive computation when far from the solution.

4.4. The trust-region algorithm. The approximate solution s_j of the jth trust-region subproblem is used to update the trust-region iterate as $x_{j+1} = x_j + \alpha_j s_j$, where α_j is obtained using the biased Wolfe line search proposed by Gertz [12]. In the combination line-search trust-region algorithm given below, $\mathcal{Q}_j^-(s) = g_j^T s + \frac{1}{2}\big[s^T H_j s\big]_-$, where $[\,c\,]_-$ denotes the negative part of c, i.e., $[\,c\,]_- = \min\{\,0, c\,\}$. With this choice of quadratic model, the sufficient decrease condition on α_j is

$$\frac{f\big(x_j + s_j(\alpha_j)\big) - f(x_j)}{\mathcal{Q}_j^-\big(s_j(\alpha_j)\big)} > \eta_1, \tag{4.8}$$

where η_1 is a preassigned scalar such that $0 < \eta_1 < \frac{1}{2}$. The line-search parameters used for the experiments were $\eta_1 = 10^{-4}$, $\eta_2 = \frac{1}{4}$, $\omega = \frac{9}{10}$, and $\gamma_3 = \frac{3}{2}$.

```
**Combination Line-Search/Trust-Region Algorithm.**
Specify constants $0 < \eta_1 < \eta_2 < 1$;   $0 < \eta_1 < \frac{1}{2}$;   $0 < \eta_1 < \omega < 1$;   $1 < \gamma_3$;
Choose $x_0$;   $\delta_0 := 1$;   $j := 0$;
while not converged do
    Find an approximate solution $s_j$ for $\min\big\{\mathcal{Q}_j(s) : \|s\|_2 \le \delta_j\big\}$;
    Find $\alpha_j$ satisfying the Wolfe conditions:
        $f(x_j + \alpha_j s_j) \le f(x_j) + \eta_1 \mathcal{Q}_j^-(\alpha_j s_j)$ and $|g(x_j + \alpha_j s_j)^T s_j| \le -\omega \mathcal{Q}_j^{-\prime}(\alpha_j s_j)$;
    $x_{j+1} := x_j + \alpha_j s_j$;
    if $\big(f(x_{j+1} - f(x_j)\big)/\mathcal{Q}_j^-(s_j) \ge \eta_2$ then
        if $\|s_j\|_2 = \delta_j$ and $\alpha_j = 1$ then
            $\delta_{j+1} := \gamma_3 \delta_j$;
        else if $\|s_j\|_2 < \delta_j$ and $\alpha_j = 1$ then
            $\delta_{j+1} := \max\{\delta_j, \gamma_3 \|s_j\|_2\}$;
        else
            $\delta_{j+1} := \alpha_j \|s_j\|_2$;
        end if
    else
        $\delta_{j+1} := \min\{\alpha_j \|s_j\|_2, \alpha_j \delta_j\}$;
    end if
    $j := j + 1$;
end do
```

A key feature of the combination line-search trust-region method is that the trust-region radius is updated as a function of α_j. The term "biased" is used by Gertz to refer to a deliberate bias against reducing the trust-region radius when α_j is small. The line search used here is a modification of Gertz's line that may reduce the trust-region radius even when α_j is small. Nevertheless, our line-search retains a natural bias against decreasing the trust-region radius; in particular, the trust-region radius is not decreased if $\|s_j\|_2 < \delta_j$ and $\alpha_j = 1$. Numerical results reported in [6] for the CUTEr unconstrained problems show that this combination trust-region/line-search method is generally more efficient than a trust-region method based on the conventional rules for updating δ_j.

Tables 1–2 give the results of applying the Steihaug-Toint and IP-SSM methods to 52 problems from the CUTEr test set. The limit on the number of Lanczos vectors computed in the Steihaug-Toint method was 100. Problems that require more than 100 Lanczos iterations for a subproblem were considered to be unsuitable for testing with an unpreconditioned method. For each method, column "Fe" gives the total number of function evaluations and column "prods" gives the total number of matrix-vector products. The final value of f and the final relative decrease in $\|g\|_2$ (i.e., $\|g(x_k)\|_2/\|g(x_0)\|_2$) are also provided. The final function values are given to help identify local solutions and to identify cases where the converged gradient does not correspond to a stationary point. (For problems with a large value of $\|g_0\|_2$, it is unreasonable to require $\|g_j\|_2$ to satisfy a small absolute tolerance.)

Tables 1–2 indicate that on many problems the number of function evaluations required for the Steihaug-Toint and IP-SSM methods are comparable. This is clearly indicated in the associated bar graphs of Figs. 3–4. In these cases, the approximate solution of every subproblem lies inside the trust region and both methods are able to solve the subproblem to high accuracy.

We would expect the Steihaug-Toint method to require more function evaluations when solutions of the trust-region subproblem frequently occur on the boundary. In these cases, the difference in function evaluations is sometimes significant (see, e.g., *broydn7*, *genrose*, *fminsurf*, and *fminsrf2*). In a few cases, IP-SSM performed slightly worse than the Steihaug-Toint method. In one case (*ncb20*), IP-SSM was significantly worse. The superiority of the Steihaug-Toint method in these cases appears to be good fortune rather than the result of a consistently better solution of the subproblem.

The results of Tables 1–2 are summarized in Table 3. In general, IP-SSM required 23% fewer function evaluations than Steihaug-Toint. By comparison, Gould et al. [15] report that GLTR solved 16 out of a set of 17 CUTEr problems and, for those solved by both GLTR and Steihaug-Toint, GLTR required 12.5% fewer function evaluations.

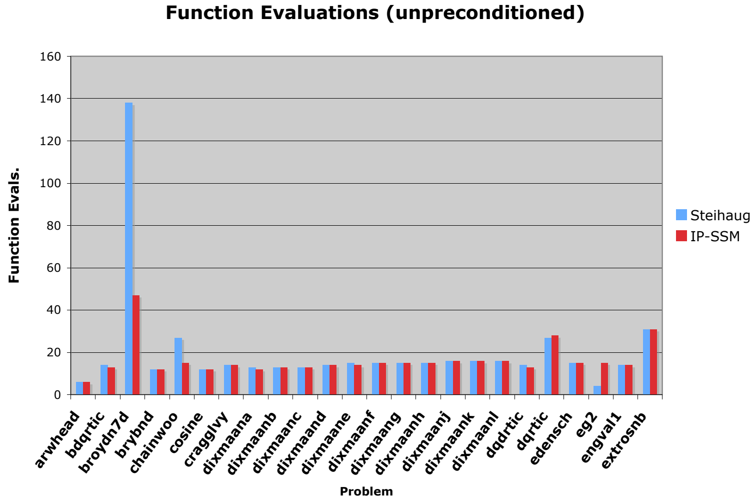

Fig. 3. *Steihaug-Toint and IP-SSM with no preconditioning. CUTEr problems A–E.*

The Steihaug-Toint and IP-SSM methods were compared for a diagonal preconditioner and an incomplete Cholesky preconditioner. The nonzero elements of the

Table 1

Steihaug-Toint and IP-SSM with no preconditioning. CUTEr problems A–E

	Steihaug				IP-SSM			
Problem	fe	prods	$f(x)$	$\mathtt{rel}(\|g\|_2)$	fe	prods	$f(x)$	$\mathtt{rel}(\|g\|_2)$
arwhead	6	6	1.69e-10	7.97e-09	6	22	4.39e-11	4.39e-09
bdqrtic	14	41	3.98e+03	2.65e-07	13	66	3.98e+03	5.39e-07
broydn7d	138	407	3.75e+02	3.77e-07	47	808	3.38e+02	3.34e-06
brybnd	12	46	6.65e-07	1.66e-06	12	81	9.93e-07	2.33e-06
chainwoo	27	57	1.31e+01	9.76e-06	15	83	3.93e+03	1.54e-05
cosine	12	10	-9.99e+02	1.33e-05	12	36	-9.99e+02	7.16e-06
cragglvy	14	36	3.36e+02	2.84e-06	14	75	3.36e+02	1.50e-06
dixmaana	13	11	1.00e+00	1.75e-06	12	36	1.00e+00	1.64e-05
dixmaanb	13	11	1.00e+00	7.81e-07	13	39	1.00e+00	6.27e-09
dixmaanc	13	11	1.00e+00	6.58e-06	13	38	1.00e+00	6.68e-06
dixmaand	14	12	1.00e+00	4.91e-06	14	42	1.00e+00	4.90e-06
dixmaane	15	82	1.00e+00	8.65e-06	14	127	1.00e+00	3.80e-06
dixmaanf	15	30	1.00e+00	7.48e-06	15	60	1.00e+00	7.57e-06
dixmaang	15	24	1.01e+00	9.44e-06	15	55	1.01e+00	9.73e-06
dixmaanh	15	19	1.03e+00	1.36e-05	15	51	1.03e+00	1.37e-05
dixmaanj	16	40	1.00e+00	4.39e-06	16	74	1.00e+00	4.36e-06
dixmaank	16	30	1.00e+00	5.53e-06	16	63	1.00e+00	5.57e-06
dixmaanl	16	25	1.01e+00	6.19e-06	16	60	1.02e+00	6.95e-06
dqdrtic	14	11	1.90e-03	3.17e-05	13	28	5.20e-04	1.53e-05
dqrtic	27	20	1.63e+11	4.17e-03	28	61	3.29e+10	1.27e-03
edensch	15	25	2.19e+02	3.33e-06	15	59	2.19e+02	4.17e-06
eg2	4	3	-9.99e+02	1.10e-11	15	223	-9.99e+02	7.45e-07
engval1	14	17	1.11e+03	6.72e-06	14	45	1.11e+03	4.54e-06
extrosnb	31	70	2.24e-02	6.48e-06	31	132	2.35e-02	6.86e-06

Steihaug-Toint preconditioner M were:

$$M_{ll} = \max\{|H_{ll}|, 10^{-3}\}.$$

The elements of the IP-SSM preconditioner were based on the diagonals of $H + \sigma_a I$, i.e.,

$$M_{ll} = \max\{|H_{ll} + \sigma_a|, 10^{-3}\}.$$

For consistency, the stopping criteria and error terms (r_a and r_e) for both methods were defined in terms of the M^{-1} norm. Although this choice of norm is more natural for the Steihaug-Toint method, results were similar using the two-norm. That is, when preconditioning was successful, Lanczos-CG converged rapidly—usually satisfying the stopping criteria in both norms. Problems that require more than 100 Lanczos vectors for a subproblem were considered to be unsuitable for testing with a diagonal preconditioner and were removed from the test set.

Tables 4–5 give the results obtained using diagonal preconditioning. An asterisk indicates that a solver failed to converge after $2n$ function evaluations. The function evaluations are also compared in Figures 5–6, with an overall summary given in the first half of Table 8. The results of Table 8 include only those problems on which both the Steihaug-Toint and IP-SSM methods converged. For both methods, diagonal preconditioning improved the convergence rate of the Lanczos-CG algorithm on many

Table 2
Steihaug-Toint and IP-SSM with no preconditioning. CUTEr problems F–Z.

	Steihaug				IP-SSM			
Problem	fe	prods	$f(x)$	rel($\|g\|_2$)	fe	prods	$f(x)$	rel($\|g\|_2$)
fletchcr	1986	16396	1.65e-10	7.75e-06	1747	21133	1.62e-07	1.36e-05
fminsrf2	*	*	*	*	132	11076	1.00e+00	4.11e-05
fminsurf	352	989	1.00e+00	3.32e-05	65	1282	1.00e+00	1.11e-05
freuroth	16	19	1.21e+05	2.19e-06	16	48	1.21e+05	2.26e-06
genrose	1180	5736	1.00e+00	1.24e-06	844	17566	1.00e+00	5.63e-07
liarwhd	19	27	4.03e-07	1.30e-08	17	62	2.96e-04	5.21e-06
ncb20	111	1414	9.10e+02	1.50e-05	75	2351	9.18e+02	1.06e-05
ncb20b	10	61	1.68e+03	8.28e-06	9	105	1.68e+03	7.86e-06
noncvxu2	36	26	1.15e+06	7.40e-03	44	270	9.75e+05	7.90e-03
noncvxun	37	28	6.80e+05	7.40e-03	42	328	6.97e+05	7.75e-03
nondia	4	3	6.27e-03	2.46e-07	4	11	6.24e-03	5.67e-07
nondquar	23	115	5.51e-04	9.53e-07	20	189	7.07e-04	9.24e-07
penalty1	28	17	3.01e+13	2.11e-03	28	53	3.01e+13	2.11e-03
powellsg	16	40	4.63e-03	4.88e-06	15	101	7.01e-03	6.86e-06
power	15	33	3.56e+04	3.41e-06	15	61	3.51e+04	3.39e-06
quartc	27	20	1.63e+11	4.17e-03	28	61	3.29e+10	1.27e-03
schmvett	9	37	-2.99e+03	2.04e-05	8	62	-2.99e+03	8.85e-06
sinquad	19	27	-2.94e+05	3.55e-07	18	84	-2.94e+05	7.57e-08
sparsqur	14	23	4.24e-03	1.83e-06	14	66	4.19e-03	1.91e-06
spmsrtls	18	117	5.61e-08	1.15e-05	20	299	4.05e-08	1.16e-05
srosenbr	9	10	1.61e-09	6.99e-09	9	42	4.48e-09	1.46e-08
testquad	72	1259	2.57e+02	1.24e-05	12	130	6.66e+02	8.17e-06
tointgss	15	13	1.00e+01	2.47e-05	20	85	1.00e+01	1.91e-05
tquartic	18	23	3.44e-17	2.06e-10	15	53	2.66e-15	3.04e-06
tridia	50	842	2.25e-03	1.33e-05	18	1388	2.84e-03	5.48e-06
vardim	13	12	6.87e+08	1.75e-06	13	40	6.87e+08	1.75e-06
vareigvl	14	26	3.54e-04	7.23e-06	14	56	3.55e-04	7.46e-06
woods	12	14	1.97e+03	9.62e-06	13	39	1.97e+03	2.92e-06

Table 3
Comparison of unpreconditioned methods

	Steihaug	IP-SSM
Problems solved	52	53
Function evals (fe)	4612	3537
Matrix mults (prods)	28371	48329
Improvement in fe	—	23%

problems—as seen in the decrease in the number of matrix-vector products. However, in the case of Steihaug's method, even with faster convergence of Lanczos-CG, more function evaluations are required by the overall solver (see, e.g., *broydn7*, *genrose*, *ncb20*, *noncvxu2*, and *noncvxun*).

The two methods were also tested using an incomplete Cholesky preconditioner. For these tests, computed incomplete Cholesky factorizations using (`icfs`), a factorization proposed by Lin and Moré [19]. In these runs, the preconditioner for the Steihaug-Toint method was the incomplete Cholesky factorization of H_j. The precon-

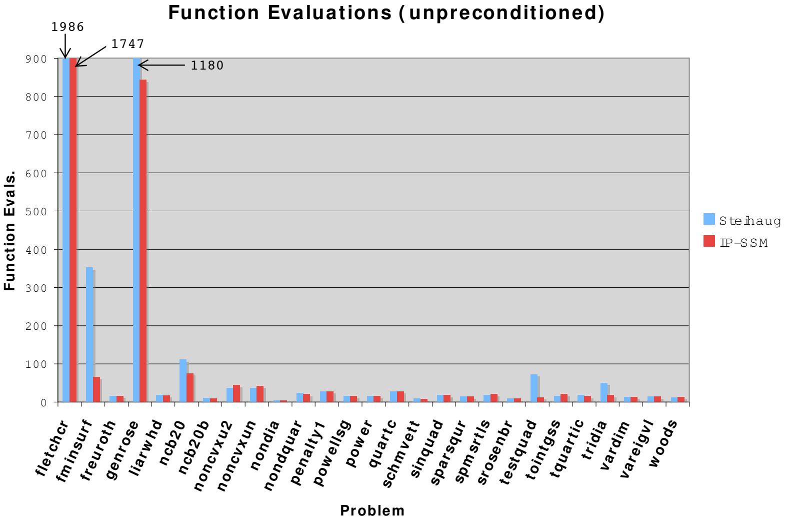

FIG. 4. *Steihaug-Toint and IP-SSM with no preconditioning. CUTEr problems F–Z.*

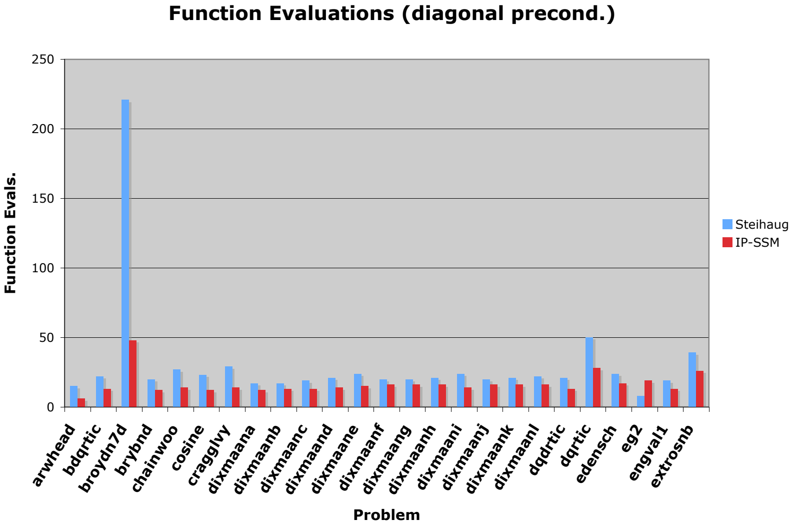

FIG. 5. *Steihaug-Toint and IP-SSM with diagonal preconditioning. CUTEr problems A–E.*

ditioner for IP-SSM was the incomplete Cholesky factorization of the positive-definite matrix $H_j + \sigma_a I$. Problems requiring more than 100 Lanczos vectors for a subproblem were considered to be unsuitable for testing with this type of preconditioner and were removed from the test set. The M^{-1} norm was used to test convergence for both Steihaug's method and the IP-SSM method. Also, problems with a dense Hessian were removed from the test set.

Tables 6–7 contains the results from these tests; these results are also displayed in

Table 4

Steihaug-Toint and IP-SSM with diagonal preconditioning. CUTEr problems A–E

.

	Steihaug				IP-SSM			
Problem	fe	prods	$f(x)$	$\mathtt{rel}(\|g\|_2)$	fe	prods	$f(x)$	$\mathtt{rel}(\|g\|_2)$
arwhead	15	11	0.00e+00	1.03e-11	6	21	6.81e-09	3.44e-07
bdqrtic	22	19	3.98e+03	1.24e-07	13	53	3.98e+03	6.59e-08
broydn7d	221	341	3.67e+02	3.50e-06	48	700	3.22e+02	1.81e-06
brybnd	20	19	4.92e-06	4.03e-06	12	45	1.36e-07	2.16e-06
chainwoo	27	33	3.93e+03	3.91e-06	14	59	3.93e+03	8.33e-06
cosine	23	20	-9.99e+02	5.36e-06	12	28	-9.99e+02	1.01e-05
cragglvy	29	40	3.36e+02	2.16e-06	14	60	3.36e+02	1.48e-06
dixmaana	17	13	1.00e+00	1.08e-06	12	38	1.00e+00	1.40e-05
dixmaanb	17	12	1.00e+00	3.83e-06	13	40	1.00e+00	8.34e-07
dixmaanc	19	14	1.00e+00	2.80e-06	13	40	1.00e+00	8.10e-06
dixmaand	21	16	1.00e+00	1.36e-07	14	44	1.00e+00	1.45e-06
dixmaane	24	26	1.00e+00	1.63e-06	15	52	1.00e+00	8.32e-08
dixmaanf	20	22	1.00e+00	4.80e-06	16	59	1.00e+00	5.41e-06
dixmaang	20	20	1.01e+00	1.17e-05	16	60	1.00e+00	5.70e-06
dixmaanh	21	19	1.03e+00	1.39e-05	16	59	1.01e+00	6.09e-06
dixmaani	24	29	1.00e+00	2.40e-06	14	69	1.00e+00	3.02e-06
dixmaanj	20	22	1.00e+00	8.86e-06	16	60	1.00e+00	8.44e-06
dixmaank	21	22	1.01e+00	6.21e-06	16	60	1.01e+00	8.65e-06
dixmaanl	22	21	1.02e+00	7.16e-06	16	59	1.02e+00	9.15e-06
dqdrtic	21	12	2.23e-26	1.11e-16	13	26	1.75e-14	7.04e-11
dqrtic	50	28	6.28e+10	2.37e-03	28	64	3.17e+10	1.28e-03
edensch	24	21	2.19e+02	7.09e-06	17	61	2.19e+02	5.20e-07
eg2	8	6	-9.99e+02	2.90e-09	19	151	-9.99e+02	1.38e-06
engval1	19	15	1.11e+03	5.71e-06	13	37	1.11e+03	1.35e-05
extrosnb	39	48	4.94e-02	5.28e-06	26	92	6.72e-02	5.52e-06

Figures 5–6, and they summarized in the second half of Table 8. As before, an asterisk in Tables 6–7 denotes when a solver failed to converge after $2n$ function evaluations. And, the asterisk in Table 8 denotes that the computation only included problems on which both the Steihaug-Toint and IP-SSM methods converged. The reduction in the overall number of matrix-vector products implies that the incomplete Cholesky preconditioner improved the rate of convergence of Lanczos-CG for both methods. However, in the case of the Steihaug-Toint method, preconditioning did not improve the overall performance in terms of function evaluations; in fact, the unpreconditioned method performed better in many cases (see, e.g., *chainwoo*, *genrose*, *ncb20*, *noncvxu2*, or *noncvxun*).

We summarize results from Tables 1–7 using performance profiles (in $\log_2$ scale) proposed by Dolan and Moré [5]. Let $\text{card}(\mathcal{S})$ denote the number of elements in a finite set $\mathcal{S}$. Let $\mathcal{P}$ denote the set of problems used for a given numerical experiment. For each method s we define the function $\pi_s : [0, r_M] \mapsto \mathbb{R}^+$ such that

$$\pi_s(\tau) = \frac{1}{\text{card}(\mathcal{P})}\,\text{card}(\{p \in \mathcal{P} : \log_2(r_{p,s}) \leq \tau\}),$$

where $r_{p,s}$ denotes the ratio of number of function evaluations needed to solve problem p with method s and the least number of function evaluations needed to solve problem

Table 5

Steihaug-Toint and IP-SSM with diagonal preconditioning. CUTEr problems F–Z.

	Steihaug				IP-SSM			
Problem	fe	prods	$f(x)$	rel($\|g\|_2$)	fe	prods	$f(x)$	rel($\|g\|_2$)
fletchcr	1665	13552	1.06e-09	1.26e-05	1624	17914	1.10e-10	3.22e-06
fminsrf2	83	190	1.00e+00	2.27e-05	72	3429	1.00e+00	3.52e-06
freuroth	23	22	1.21e+05	3.89e-06	14	38	1.21e+05	2.12e-05
genhumps	1346	1315	1.02e+03	8.90e-03	*	*	*	*
genrose	1930	3786	1.00e+00	1.15e-06	839	10499	1.00e+00	5.23e-06
indef	*	*	*	*	81	797	-4.77e+09	1.17e+00
liarwhd	26	27	2.09e-05	3.38e-06	17	75	6.13e-08	1.85e-07
ncb20	451	462	8.93e+02	1.28e-06	94	1648	9.17e+02	1.45e-06
ncb20b	10	60	1.68e+03	1.50e-05	9	78	1.68e+03	1.13e-05
noncvxu2	753	504	5.10e+06	8.63e-03	44	322	1.23e+06	8.15e-03
noncvxun	569	399	4.00e+06	8.26e-03	43	371	1.18e+06	8.06e-03
nondia	83	63	9.91e-01	7.33e-07	5	18	1.02e-07	4.76e-07
nondquar	26	98	5.53e-04	9.14e-07	16	114	6.71e-04	6.92e-07
powellsg	23	40	4.47e-03	4.83e-06	15	90	7.20e-03	7.16e-06
quartc	50	28	6.28e+10	2.37e-03	28	64	3.17e+10	1.28e-03
scosine	24	22	-9.99e+02	3.96e-07	8	44	-2.99e+03	1.64e-05
schmvett	14	25	-2.99e+03	1.35e-05	84	2525	7.11e+02	2.24e+00
sinquad	35	31	-2.94e+05	3.47e-10	18	86	-2.94e+05	9.20e-07
sparsine	30	158	9.28e-03	6.41e-06	11	217	9.82e-03	1.60e-06
sparsqur	25	26	1.31e-03	1.56e-06	14	59	6.24e-03	2.82e-06
spmsrtls	69	101	6.91e-08	1.62e-05	22	375	3.30e-09	1.63e-06
srosenbr	20	17	3.25e-05	1.22e-06	9	34	6.30e-07	1.39e-07
testquad	28	16	5.52e-26	1.49e-17	10	28	1.28e+01	1.84e-06
tointgss	21	17	1.00e+01	1.13e-05	18	46	1.00e+01	4.38e-05
tquartic	16	24	6.46e-13	4.47e-07	11	46	5.98e-17	1.98e-07
tridia	24	27	4.41e-06	3.54e-06	12	54	7.50e-06	7.58e-06
vardim	56	34	4.53e+08	1.28e-06	13	44	6.87e+08	1.75e-06
woods	27	26	1.97e+03	3.04e-06	13	43	1.97e+03	3.16e-06

p. The number r_M denotes the maximum value of $\log_2(r_{p,s})$. Figures 9–11 depict the functions π_s for each of the methods tested.

5. Concluding Remarks. The numerical results suggest that when the solution of the trust-region subproblem lies on the boundary of the trust region, solving the subproblem more accurately can decrease the overall number of function evaluations. The numerical results indicate that when no preconditioning is used, IP-SSM outperforms Steihaug's method in terms of function evaluations and may be more efficient when the cost of a function evaluation is expensive relative to the cost of a matrix-vector product.

Gould et al. [15] found that preconditioners similar to those used here did not significantly improve the convergence rate for either the GLTR or Steihaug-Toint method. However, our results indicate that there can be benefits when preconditioning IP-SSM. In particular, preconditioning may provide a substantial decrease in the number of matrix-vector products.

Our experiments indicate that the increase in function evaluations observed with the preconditioned Steihaug-Toint method is the result of using a different trust-region

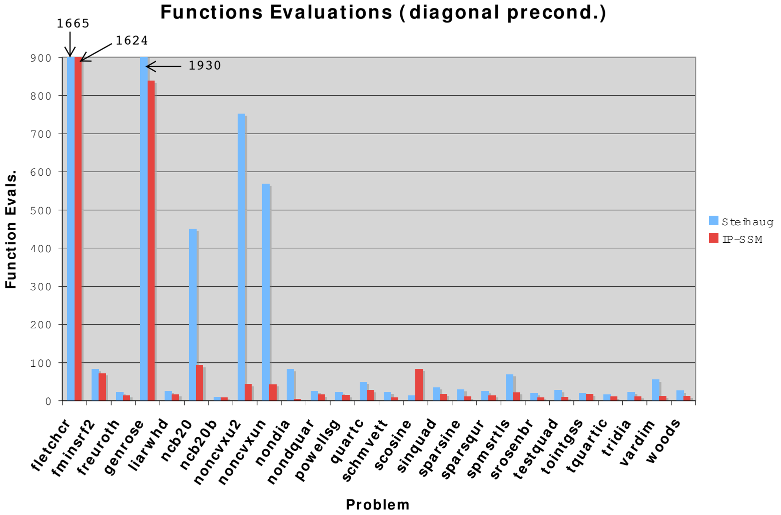

FIG. 6. *Steihaug-Toint and IP-SSM with diagonal preconditioning. CUTEr problems F–Z.*

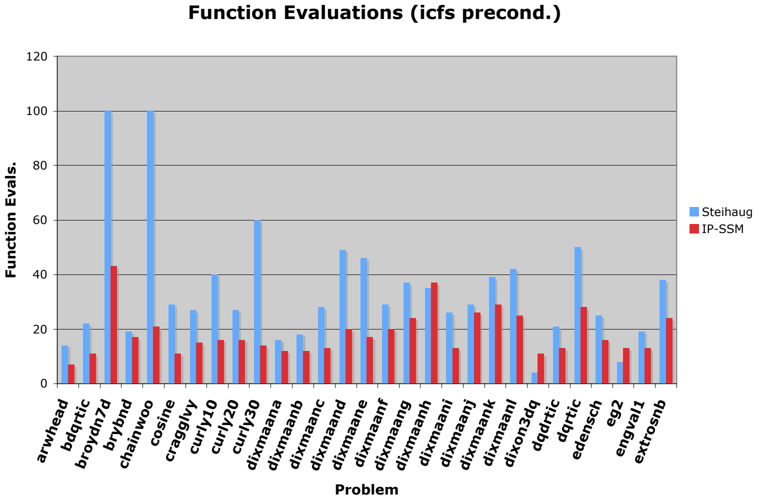

FIG. 7. *Steihaug-Toint and IP-SSM with* `icfs` *preconditioning. CUTEr problems A–E.*

norm at each iteration. There are two potential difficulties associated with altering the shape of the trust region: (i) the preconditioner that defines the trust-region norm may become ill-conditioned; and (ii) there may be no simple relationship between the trust-region radii of consecutive iterations.

More generally, the results of Section 4 indicate that it is possible to solve the trust-region subproblem efficiently when the trust-region scaling is chosen independently of the preconditioner.

TABLE 6
Steihaug-Toint and IP-SSM with ICFS preconditioning. CUTEr problems A–E.

	Steihaug				IP-SSM			
Problem	fe	prods	$f(x)$	rel($\\|g\\|_2$)	fe	prods	$f(x)$	rel($\\|g\\|_2$)
arwhead	14	10	2.32e-10	1.21e-07	7	22	8.52e-11	6.15e-08
bdqrtic	22	15	3.98e+03	5.20e-07	11	41	3.98e+03	6.35e-07
broydn7d	100	63	3.65e+02	4.92e-06	43	176	3.70e+02	1.56e-06
brybnd	19	14	3.59e-12	4.35e-09	17	51	6.43e-08	5.17e-06
chainwoo	100	71	6.36e+01	5.38e-06	21	64	3.93e+03	7.32e-06
cosine	29	20	-9.99e+02	8.75e-08	11	22	-9.99e+02	1.03e-05
cragglvy	27	19	3.36e+02	7.04e-07	15	54	3.36e+02	6.35e-07
curly10	40	22	-1.00e+05	8.75e-07	16	114	-1.00e+05	3.17e-09
curly20	27	19	-1.00e+05	1.82e-10	16	108	-1.00e+05	1.38e-12
curly30	60	32	-1.00e+05	2.96e-12	14	112	-1.00e+05	2.82e-10
dixmaana	16	11	1.00e+00	6.65e-06	12	34	1.00e+00	1.58e-05
dixmaanb	18	12	1.00e+00	1.76e-06	12	39	1.00e+00	6.03e-06
dixmaanc	28	19	1.00e+00	1.12e-05	13	39	1.00e+00	1.94e-08
dixmaand	49	28	1.00e+00	7.43e-06	20	122	1.00e+00	3.93e-06
dixmaane	46	28	1.00e+00	3.97e-07	17	64	1.00e+00	7.67e-08
dixmaanf	29	20	1.00e+00	3.85e-06	20	64	1.00e+00	8.40e-06
dixmaang	37	22	1.00e+00	9.28e-06	24	93	1.02e+00	1.29e-05
dixmaanh	35	21	1.01e+00	1.06e-05	37	202	1.07e+00	9.56e-06
dixmaani	26	19	1.00e+00	1.13e-06	13	44	1.00e+00	3.93e-08
dixmaanj	29	20	1.01e+00	1.15e-05	26	152	1.00e+00	4.21e-06
dixmaank	39	26	1.01e+00	1.08e-05	29	169	1.00e+00	4.59e-06
dixmaanl	42	28	1.01e+00	6.38e-06	25	146	1.02e+00	7.85e-06
dixon3dq	4	3	9.42e-29	2.87e-15	11	70	2.45e-09	4.49e-07
dqdrtic	21	12	3.03e-26	1.30e-16	13	26	4.12e-14	1.09e-10
dqrtic	50	28	6.28e+10	2.37e-03	28	69	3.18e+10	1.26e-03
edensch	25	18	2.19e+02	1.28e-07	16	52	2.19e+02	1.32e-06
eg2	8	6	-9.99e+02	2.90e-09	13	60	-9.99e+02	1.47e-06
engval1	19	13	1.11e+03	6.42e-06	13	32	1.11e+03	1.85e-06
extrosnb	38	27	1.96e-02	6.28e-06	24	69	2.96e-02	8.99e-06

REFERENCES

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

[2] A. R. Conn, N. I. M. Gould, and P. L. Toint, *A globally convergent Lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds*, Math. Comp., 66 (1997), pp. 261–288, S1–S11.

[3] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[4] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[5] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with COPS*, Technical Memorandum ANL/MCS-TM-246, Argonne National Laboratory, Argonne, IL, 2000.

[6] J. B. Erway, P. E. Gill, and J. D. Griffin, *Iterative methods for finding a trust-region step*, Numerical Analysis Report 07-2, Department of Mathematics, University of California, San Diego, La Jolla, CA, 2007. To appear in SIAM J. Optim.

[7] M. C. Fenelon, *Preconditioned Conjugate-Gradient-Type Methods for Large-Scale Unconstrained Optimization*, PhD thesis, Department of Operations Research, Stanford University, Stanford, CA, 1981.

Table 7
Steihaug-Toint and IP-SSM with ICFS preconditioning. CUTEr problems F–Z.

	Steihaug				IP-SSM			
Problem	fe	prods	$f(x)$	rel($\|g\|_2$)	fe	prods	$f(x)$	rel($\|g\|_2$)
fletchcr	*	*	*	*	1739	5713	2.62e-08	1.21e-05
fletcbv2	2	1	-5.01e-01	3.30e-03	6	24	-5.01e-01	2.31e-02
fminsrf2	24	21	1.00e+00	1.92e-06	47	140	1.00e+00	5.26e-05
freuroth	22	14	1.21e+05	9.44e-08	14	28	1.21e+05	7.89e-06
genhumps	944	927	1.36e+02	9.36e-03	*	*	*	*
genrose	1350	860	1.00e+00	1.84e-07	823	2589	1.00e+00	9.34e-08
liarwhd	27	28	3.63e-05	4.44e-06	17	66	4.89e-08	1.64e-07
morebv	2	1	7.33e-13	1.51e-08	3	24	8.66e-10	4.84e-05
ncb20	133	81	9.11e+02	6.13e-08	90	456	9.22e+02	4.29e-06
ncb20b	18	11	1.68e+03	4.38e-07	15	61	1.68e+03	4.52e-06
noncvxu2	61	39	2.29e+06	8.04e-03	44	215	1.30e+06	6.50e-03
noncvxun	80	55	1.83e+06	7.75e-03	47	232	8.04e+05	7.56e-03
nondia	21	15	4.98e-07	2.34e-09	6	19	4.59e-08	2.15e-08
nondquar	20	17	2.09e-06	3.09e-07	18	136	6.11e-05	7.48e-07
powellsg	24	18	1.19e-03	2.23e-06	15	52	6.43e-03	6.37e-06
quartc	50	28	6.28e+10	2.37e-03	28	69	3.18e+10	1.26e-03
sbrybnd	19	14	3.56e-12	3.16e-09	78	3810	1.72e+01	4.08e-07
scurly10	40	23	-1.00e+05	4.30e-09	98	1327	-1.00e+05	1.36e-03
scurly20	25	16	-1.00e+05	1.09e-07	75	1043	-9.45e+04	1.29e-01
scurly30	28	18	-1.00e+05	1.12e-07	60	889	-1.00e+05	4.16e-03
schmvett	8	6	-2.99e+03	1.32e-09	7	19	-2.99e+03	3.43e-08
sinquad	28	17	-2.94e+05	7.59e-07	16	42	-2.94e+05	9.97e-11
sparsqur	26	19	2.13e-03	1.45e-06	14	46	3.43e-03	1.73e-06
spmsrtls	23	15	2.11e-07	2.26e-05	24	134	1.73e-10	9.36e-07
srosenbr	20	14	1.62e-09	1.37e-07	9	28	7.84e-11	7.29e-09
testquad	28	16	3.94e-26	1.29e-17	10	30	1.27e+01	4.86e-06
tointgss	13	8	1.00e+01	1.07e-17	16	38	1.00e+01	7.88e-06
tquartic	14	24	5.16e-15	1.67e-07	11	53	2.19e-14	6.42e-07
tridia	19	11	1.78e-27	1.07e-16	10	28	4.75e-13	2.44e-10
woods	27	18	1.97e+03	1.17e-05	13	41	1.97e+03	8.38e-06

Table 8
Comparison of preconditioned methods on problems for which all methods converged.

	Steihaug-diag	IP-SSM-diag	Steihaug-ICFS	IP-SSM-ICFS
Problems solved	52	52	58	58
Function evals (fe)	6845	3475	3149	2151
Matrix mults (prods)	20604	40298	2054	13949
Improvement in fe	—	49%	—	32%

[8] A. Forsgren and P. E. Gill, *Primal-dual interior methods for nonconvex nonlinear programming*, SIAM J. Optim., 8 (1998), pp. 1132–1152.

[9] A. Forsgren, P. E. Gill, and J. D. Griffin, *Iterative methods for systems arising in interior-point methods*, SIAM J. Optim., 18 (2007), pp. 666–690.

[10] A. Forsgren, P. E. Gill, and M. H. Wright, *Interior methods for nonlinear optimization*, SIAM Rev., 44 (2002), pp. 525–597.

[11] D. M. Gay, *Computing optimal locally constrained steps*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 186–197.

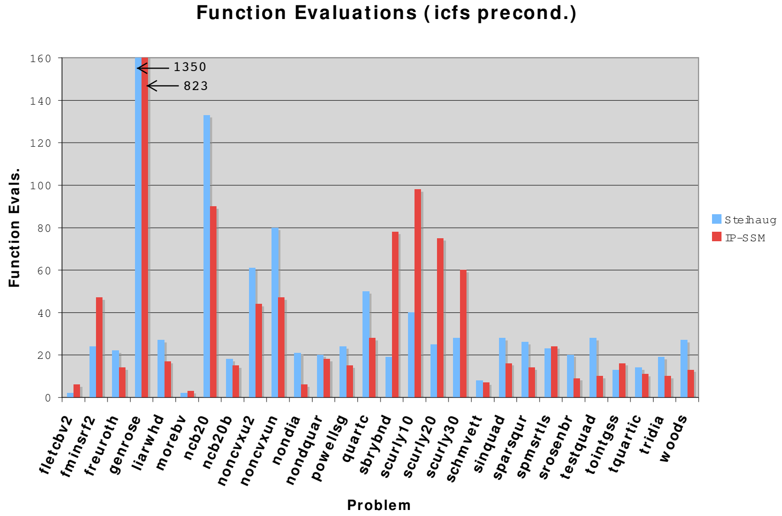

FIG. 8. *Steihaug-Toint and IP-SSM with* `icfs` *preconditioning. CUTEr problems F–Z.*

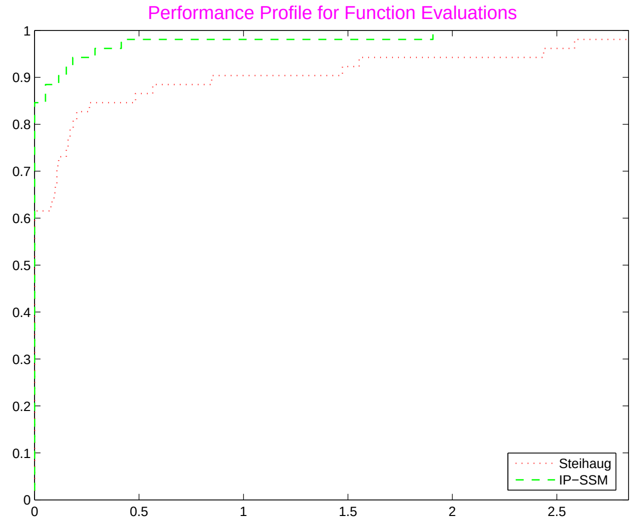

FIG. 9. $\log_2$*-scale performance profile comparing function evaluations without preconditioning.*

[12] E. M. GERTZ, *Combination Trust-Region Line-Search Methods for Unconstrained Optimization*, PhD thesis, Department of Mathematics, University of California, San Diego, 1999.

[13] P. E. GILL AND M. W. LEONARD, *Reduced-Hessian quasi-Newton methods for unconstrained optimization*, SIAM J. Optim., 12 (2001), pp. 209–237.

[14] ———, *Limited-memory reduced-Hessian methods for large-scale unconstrained optimization*, SIAM J. Optim., 14 (2003), pp. 380–401.

[15] N. I. M. GOULD, S. LUCIDI, M. ROMA, AND P. L. TOINT, *Solving the trust-region subproblem using the Lanczos method*, SIAM J. Optim., 9 (1999), pp. 504–525.

[16] N. I. M. GOULD, D. ORBAN, AND P. L. TOINT, *CUTEr and SifDec: A constrained and uncon-*

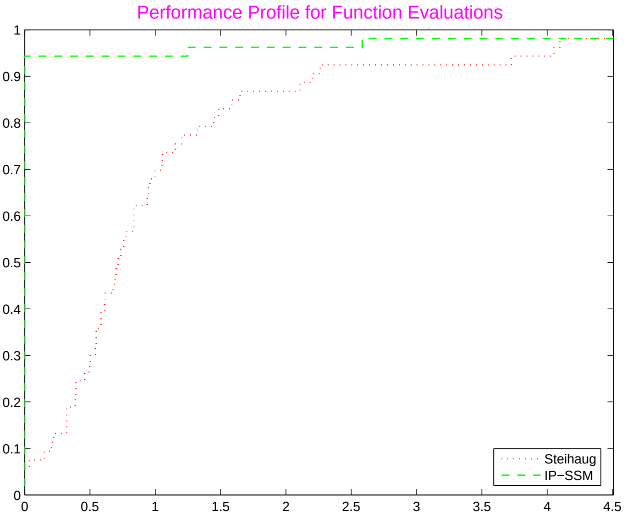

FIG. 10. $\log_2$*-scale performance profile comparing function evaluations using diagonal preconditioning.*

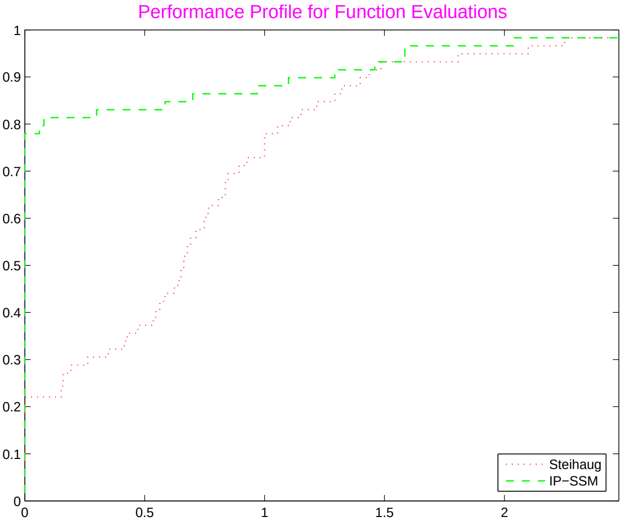

FIG. 11. $\log_2$*-scale performance profile comparing function evaluations using* `icfs`.

strained testing environment, revisited, ACM Trans. Math. Software, 29 (2003), pp. 373–394.

[17] W. W. HAGER, *Minimizing a quadratic over a sphere*, SIAM J. Optim., 12 (2001), pp. 188–208.

[18] W. W. HAGER AND S. PARK, *Global convergence of SSM for minimizing a quadratic over a sphere*, Math. Comp., 74 (2004), pp. 1413–1423.

[19] C.-J. LIN AND J. J. MORÉ, *Incomplete Cholesky factorizations with limited memory*, SIAM J. Sci. Comput., 21 (1999), pp. 24–45.

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

[21] ———, *Newton's method*, in Studies in Mathematics, Volume 24. Studies in Numerical Analysis, G. H. Golub, ed., Math. Assoc. America, Washington, DC, 1984, pp. 29–82.

[22] J. L. Nazareth, *The method of successive affine reduction for nonlinear minimization*, Math. Program., 35 (1986), pp. 97–109.

[23] R. Polyak, *Modified barrier functions (theory and methods)*, Math. Program., 54 (1992), pp. 177–222.

[24] M. J. D. Powell, *A new algorithm for unconstrained optimization*, in Nonlinear Programming (Proc. Sympos., Univ. of Wisconsin, Madison, Wis., 1970), J. B. Rosen, O. L. Mangasarian, and K. Ritter, eds., Academic Press, New York, 1970, pp. 31–65.

[25] ———, *Convergence properties of a class of minimization algorithms*, in Nonlinear Programming, 2 (Proc. Sympos. Special Interest Group on Math. Programming, Univ. Wisconsin, Madison, Wis., 1974), O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, New York, 1974, pp. 1–27.

[26] F. Rendl and H. Wolkowicz, *A semidefinite framework for trust region subproblems with applications to large scale minimization*, Math. Programming, 77 (1997), pp. 273–299.

[27] M. Rojas, S. A. Santos, and D. C. Sorensen, *A new matrix-free algorithm for the large-scale trust-region subproblem*, SIAM J. Optim., 11 (2001), pp. 611–646.

[28] M. Rojas and D. C. Sorensen, *A trust-region approach to the regularization of large-scale discrete forms of ill-posed problems*, SIAM J. Sci. Comput., 23 (2002), pp. 1842–1860.

[29] D. Siegel, *Updating of conjugate direction matrices using members of Broyden's family*, Math. Program., 60 (1993), pp. 167–185.

[30] ———, *Modifying the BFGS update by a new column scaling technique*, Mathematical Programming, 66 (1994), pp. 45–78. Ser. A.

[31] D. C. Sorensen, *Newton's method with a model trust region modification*, SIAM J. Numer. Anal., 19 (1982), pp. 409–426.

[32] ———, *Minimization of a large-scale quadratic function subject to a spherical constraint*, SIAM J. Optim., 7 (1997), pp. 141–161.

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

[34] P. L. Toint, *Towards an efficient sparsity exploiting Newton method for minimization*, in Sparse Matrices and Their Uses, I. S. Duff, ed., London and New York, 1981, Academic Press, pp. 57–88.