ITERATIVE METHODS FOR FINDING A TRUST-REGION STEP*

Jennifer B. ERWAY† Philip E. GILL‡ Joshua D. GRIFFIN§

UCSD Department of Mathematics
Technical Report NA-07-02¶
November 2007

Abstract

We consider the problem of finding an approximate minimizer of a general quadratic function subject to a two-norm constraint. The Steihaug-Toint method minimizes the quadratic over a sequence of expanding subspaces until the iterates either converge to an interior point or cross the constraint boundary. The benefit of this approach is that an approximate solution may be obtained with minimal work and storage. However, the method does not allow the accuracy of a constrained solution to be specified. We propose an extension of the Steihaug-Toint method that allows a solution to be calculated to any prescribed accuracy. If the Steihaug-Toint point lies on the boundary, the constrained problem is solved on a sequence of evolving low-dimensional subspaces. Each subspace includes an accelerator direction obtained from a regularized Newton method applied to the constrained problem. A crucial property of this direction is that it can be computed by applying the conjugate-gradient method to a positive-definite system in both the primal and dual variables of 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 associated with the underlying trust-region method. At one extreme, a low-accuracy solution is obtained that is comparable to the Steihaug-Toint point. At the other extreme, a high-accuracy solution can be specified that minimizes the overall number of function evaluations at the expense of more matrix-vector products.

Key words. Large-scale unconstrained optimization, trust-region methods, conjugate-gradient method, Lanczos tridiagonalization process

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

*Research supported by National Science Foundation grants DMS-9973276 and DMS-0511766.

†Department of Mathematics, 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).

§Sandia National Laboratories, Livermore, CA 94551-9217 (jgriffi@sandia.gov).

¶Simultaneously issued as technical reports at the Department of Mathematics, University of California, San Diego; and Sandia National Laboratories, Livermore.

1. Introduction

This paper concerns the formulation of algorithms for finding an approximate solution of the constrained minimization problem:

$$\begin{array}{ll} \underset{s\in\mathbb{R}^n}{\text{minimize}} & \mathcal{Q}(s) \equiv g^T s + \frac{1}{2} s^T H s \\ \text{subject to} & \|s\|_2 \leq \delta, \end{array} \tag{1.1}$$

where g is an n-vector, H is a symmetric matrix, and δ is a given positive scalar. Problem (1.1) is considered in the context of a trust-region method for minimizing a general nonlinear scalar-valued function f. In this setting, g and H are usually the gradient $\nabla f(x)$ and Hessian $\nabla^2 f(x)$ at the current x, and $\mathcal{Q}(s)$ represents a quadratic model of $f(x)-f(x+s)$. In general, H may have an arbitrary distribution of positive, negative and zero eigenvalues. Each iteration of a trust-region method involves finding an approximate solution of problem (1.1) with a given value of the so-called *trust-region radius* δ. Because of its crucial role in the trust-region method, we refer to (1.1) as the *trust-region problem*. The choice of inner-product norm $\|s\|_2$ is critical for the methods described here. Other methods based on the use of the infinity norm are proposed by, e.g., [2,5,25] (See Gould et al. [4] for further discussion of the choice of trust-region norm.)

In the trust-region context it is generally unnecessary (and inefficient) to compute an exact solution of (1.1). The accuracy of the trust-region solution generally determines the number of function evaluations required by the underlying optimization method. Broadly speaking, increasing the accuracy of the trust-region solution, decreases the number of trust-region subproblems that must be solved, but increases the number of evaluations of f and its derivatives. (Notwithstanding this effect on the overall cost of an optimization, an approximate solution must have sufficient accuracy to allow the underlying method to converge.) For a given optimization problem, the optimal accuracy involves a tradeoff between the cost of evaluating the function and its derivatives and the cost of the linear algebra associated with solving problem (1.1). As these costs are problem dependent, an effective general-purpose trust-region solver should allow the accuracy to be varied so that the method may be tailored to suit a particular problem.

A number of methods for solving (1.1) rely on the properties of direct matrix factorizations. For example, the method of Moré and Sorensen [26] makes repeated use of the Cholesky factorization of a positive semidefinite matrix (see also, [3, 11–13, 22, 38]). These methods are designed to solve problems for which the cost of a matrix factorization is not excessive—e.g., if n is sufficiently small or H is sufficiently sparse. However, some problems are sufficiently large that it becomes necessary to exploit structure in H in order to solve equations of the form $Hu = v$ efficiently (this includes, but is not restricted to, the case where H is a large sparse matrix). In these *large-scale* cases it is necessary to use iterative methods for the solution of the constituent linear equations (see, e.g., [2, 17, 32, 33, 40]). A crucial property of such methods is that H is used only as an operator for the definition of matrix-vector products of the form Hv. This means that the linear algebra overhead associated with optimization methods based on iterative solvers is directly proportional to the

average number of matrix-vector products.

The conjugate-gradient (CG) method is one of the most widely used iterative methods for solving symmetric positive-definite linear equations. Toint [30] and Steihaug [40] independently proposed methods based on the properties of the CG method for solving symmetric positive-definite linear equations. If the unconstrained minimizer of $\mathcal{Q}$ lies outside the trust-region, the Steihaug-Toint method terminates with a point on the boundary but does not allow the accuracy of the constrained solution to be specified. This difficulty was observed by Gould, Lucidi, Roma, and Toint [17], who proposed that the Steihaug-Toint procedure be supplemented by the generalized Lanczos trust-region (GLTR) algorithm, which finds a constrained minimizer of (1.1) over a sequence of expanding subspaces defined by the Lanczos vectors.

Hager [20] has proposed a sequential subspace minimization (SSM) method for finding the exact solution of a quadratically constrained quadratic function. In an SSM method, the constrained problem is solved over a sequence of subspaces that does not satisfy an expansion property. SSM methods have been developed in the context of trust-region methods for large-scale unconstrained and constrained optimization by Griffin [19] and Erway [9].

Other Krylov-based iterative methods approximate the eigenvalues of a matrix obtained by augmenting H by a row and column (see, Sorensen [39], Rojas and Sorensen [34], Rojas, Santos and Sorensen [33], and Rendl and Wolkowicz [32]). Subspace minimization methods for general large-scale unconstrained optimization have been considered by Fenelon [10], Gill and Leonard [14, 15], Nazareth [28], and Siegel [36, 37].

Here we consider an extension of the Steihaug-Toint method that allows an approximate solution of (1.1) to be calculated to any prescribed accuracy. The method is designed to exploit the best features of the GLTR and SSM methods. As in the GLTR method, a constrained second phase is activated if the unconstrained minimizer of $\mathcal{Q}$ lies outside the trust-region. However, the iterates of the second phase solve the constrained problem on a sequence of evolving low-dimensional subspaces, as in an SSM method. This "phased-SSM method" has several features that distinguish it from existing methods. First, a simple inexpensive estimate of the smallest eigenvalue of H is computed in both the constrained and unconstrained phases. This estimate extends the Steihaug-Toint method to the case where $g = 0$ and provides a better point on the constraint boundary to start the second phase. In addition, the low-dimensional subspace used in the second phase includes an accelerator direction obtained from a regularized Newton method applied to the constrained problem. A crucial property of this direction is that it can be computed by applying the CG method to a positive-definite system in both the primal and dual variables of 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 is obtained that is comparable to the Steihaug-Toint point. Roughly speaking, this solution will be computed with fewer matrix-vector products, but will give a trust-region method that requires the most evaluations of the function. At the other extreme,

a high-accuracy solution can be specified that minimizes the number of function evaluations at the expense of substantially more matrix-vector products.

The paper is organized in six sections. In Section 2, we discuss the trust-region problem (1.1) and review the characterization of a global solution. In Section 3, we discuss methods based on subspace minimization and review related work, including the Steihaug-Toint method, the GLTR method [17], and Hager's sequential subspace minimization (SSM) method. The phased-SSM method is described in Section 4. Section 5 includes numerical results that compare the phased-SSM method with the Steihaug-Toint method on large unconstrained problems from the CUTEr test collection (see Bongartz et al. [1] and Gould, Orban and Toint [18]). Finally, Section 6 includes some concluding remarks and observations.

Unless explicitly indicated, $\|\cdot\|$ denotes the vector two-norm or its subordinate matrix norm. The symbol e_j denotes the jth column of the identity matrix I, where the dimensions of e_j and I depend on the context. The eigenvalues of a real symmetric matrix H are denoted by $\{\lambda_j\}$, where $\lambda_n \leq \lambda_{n-1} \leq \cdots \leq \lambda_1$. The associated eigenvectors are denoted by $\{u_j\}$. An eigenvalue λ and a corresponding normalized eigenvector u such that $\lambda = \lambda_n$ are known as the *leftmost eigenpair* of H. The matrix $A^\dagger$ denotes the Moore-Penrose pseudoinverse of A. 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 labeled $[Ax]$. Similarly, if P is a matrix with columns p_1, p_2, ..., p_m, then $[AP]$ denotes the matrix of computed columns $[Ap_1]$, $[Ap_2]$, ..., $[Ap_m]$.

2. The Constrained Trust-Region Problem

The optimality conditions for problem (1.1) are summarized in the following result. (For a proof, see, e.g., Gay [11], Sorensen [38], Moré and Sorensen [27] or Conn, Gould and Toint [4].)

Theorem 2.1. *Let δ be a given positive constant. A vector s^* is a global solution of the trust-region problem* (1.1) *if and only if $\|s^*\| \leq \delta$ and there exists a unique $\sigma^* \geq 0$ such that $H + \sigma^* I$ is positive semidefinite with*

$$(H + \sigma^* I)s^* = -g, \quad \text{and} \quad \sigma^*(\delta - \|s^*\|) = 0. \tag{2.1}$$

Moreover, if $H + \sigma^* I$ is positive definite, then the global minimizer is unique. ∎

The trust-region problem is said to be *degenerate* if $\|s_L\| < \delta$, where s_L is the least-length solution of the (necessarily compatible) system $(H - \lambda_n I)s = -g$, (i.e., $s_L = -(H - \lambda_n I)^\dagger g$). 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 condition (2.1) because $\|s\| < \|s_L\| < \delta$. Alternatively, if λ_n is negative or zero, the system $(H + \sigma I)s = -g$ cannot be used alone to determine s^*. However, the generalized

eigenvector u_n is a null vector of $H - \lambda_n I$, and there exists a scalar τ such that

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

In this case, $\sigma = -\lambda_n$ and $s = s_L + \tau u_n$ satisfy the optimality conditions (2.1) and thereby constitute a global solution of (1.1).

The conditions of Theorem 2.1 imply that s^* depends on the size of δ and the eigenvalue distribution of H. If H is positive definite and $\|H^{-1}g\| \leq \delta$, then $\sigma^* = 0$ and s^* satisfies the equations $Hs = -g$ and is the unique unconstrained global solution of (1.1). Otherwise, s^* is a solution of the equality-constraint problem

$$\underset{s\in\mathbb{R}^n}{\text{minimize}}\;\; g^T s + \tfrac{1}{2} s^T H s \;\;\text{subject to}\;\; \|s\| = \delta. \tag{2.2}$$

Broadly speaking, there are two approaches to finding an approximate solution of (1.1). The first is to proceed with the solution of the unconstrained problem and consider the constraint only if the unconstrained solution appears to lie outside the trust-region. The class of dog-leg methods are of this type (see, e.g., Dennis and Schnabel [7], Shultz, Schnabel and Byrd [35], and Byrd, Schnabel and Shultz [3]), as are the methods considered in this paper. The second approach is to start with the equality-constraint problem (2.2) and switch to the unconstrained Newton direction if it appears that σ^* is zero. Methods of this type include those of Gay [11] and Moré and Sorensen [26].

Methods based on first solving (2.2) attempt to find a root of the nonlinear equation $\varphi(\sigma) = \|(H+\sigma I)^{-1}g\| - \delta = 0$ that lies in the interval $(-\lambda_n, \infty)$ defined by the leftmost generalized eigenvalue. Each iteration of the root finder involves solving the positive-semidefinite system $(H+\sigma I)s = -g$ associated with the current estimate of σ^*. If the σ values appear to be converging to a negative root in $(-\lambda_n, \infty)$, which implies that the Newton step lies inside the trust-region, the value $\sigma = 0$ is selected.

The preferred method using direct linear solvers is the Moré-Sorensen method [26]. The accuracy of an approximate solution is specified by the tolerances c_1, $c_2 \in (0, 1)$. At each iteration, the Cholesky factorization of $H+\sigma I$ is used to compute the vector q such that

$$(H + \sigma I)q = -g. \tag{2.3}$$

If $\|q\| < -(1-c_1)\delta$, then an approximate null vector z is also computed. A safeguarding scheme is used to ensure that σ remains within $(-\lambda_n, \infty)$. The Moré-Sorensen algorithm gives an approximate solution s that satisfies

$$\mathcal{Q}(s) - \mathcal{Q}^* \leq c_1(2 - c_1)\max(|\mathcal{Q}^*|, c_2), \;\text{ and }\; \|s\| \leq (1 + c_1)\delta. \tag{2.4}$$

where $\mathcal{Q}^*$ denotes the global minimum of (1.1). In the context of a standard underlying trust-region method, the Moré-Sorensen algorithm gives convergence to points that satisfy both the first and second-order necessary conditions for optimality.

The majority of methods for solving (1.1) using iterative linear solvers are based on the conjugate-gradient method. In this situation, the close relationship between the CG method for linear equations and the CG method for unconstrained optimization leads naturally to methods that consider the trust-region constraint after first determining that the unconstrained solution is infeasible.

3. Subspace Minimization Methods

We start by considering methods that approximate s^* by solving a trust-region problem restricted to a *subspace* $\mathcal{S}$ of $\mathbb{R}^n$. In the simplest case, $\mathcal{S} = \text{span}\{g\}$, which gives the step as solution of

$$\underset{s\in\mathbb{R}^n}{\text{minimize}} \;\; \mathcal{Q}(s) \quad \text{subject to} \;\; \|s\| \leq \delta, \quad s \in \text{span}\{g\}. \tag{3.1}$$

If $g \neq 0$, the solution of this problem has the form $s_c = -\alpha_0 g$ ($\alpha_0 > 0$), which is called the *Cauchy point* [31]. An important property of the Cauchy point is that convergence to first-order points is guaranteed for any approximate trust-region solution s such that $\mathcal{Q}(s) \leq \mathcal{Q}(s_c)$ (see Powell [31]).

More generally, $\mathcal{S}$ is the last of a *sequence* of subspaces $\{\mathcal{S}_k\}$, where each $\mathcal{S}_k$ is the span of at most m independent vectors $p_0^{(k)}$, $p_1^{(k)}$, …, $p_{m-1}^{(k)}$. Such algorithms generate a sequence $\{s_k\}$ of approximations to s^*, where s_k is a solution of

$$\underset{s\in\mathbb{R}^n}{\text{minimize}} \;\; \mathcal{Q}(s) \quad \text{subject to} \;\; \|s\| \leq \delta, \quad s \in \mathcal{S}_k. \tag{3.2}$$

If P_k denotes the matrix with columns $p_0^{(k)}$, $p_1^{(k)}$, …, $p_{m-1}^{(k)}$, then $s_k = P_k y_k$, where y_k is the solution of the *reduced problem*

$$\underset{y\in\mathbb{R}^m}{\text{minimize}} \;\; g^T P_k y + \tfrac{1}{2} y^T P_k^T H P_k y, \quad \text{subject to} \;\; \|y\|_C \leq \delta, \tag{3.3}$$

with $C = P_k^T P_k$. The vector $P_k^T g$ and matrix $P_k^T H P_k$ are known as the *reduced gradient* and *reduced Hessian*, respectively.

The conjugate-gradient (CG) method for solving the positive-definite symmetric system $Hs = -g$ may be interpreted as a subspace minimization method for finding the *unconstrained* minimizer of $\mathcal{Q}(s)$. This method implicitly defines a subspace basis that increases in dimension at each step, with $P_k = \begin{pmatrix} P_{k-1} & p_{k-1} \end{pmatrix}$. The directions $\{p_j\}$ are chosen to be conjugate with respect to H, i.e.,

$$p_i^T H p_j = 0, \;\; \text{for} \;\; i \neq j \leq k-1. \tag{3.4}$$

The conjugacy property gives a positive-definite *diagonal* reduced Hessian. This diagonal structure guarantees that the directions $\{p_j\}$ are linearly independent and thereby provides a sequence of *expanding* subspaces $\mathcal{S}_{k-1} \subset \mathcal{S}_k$, with $\mathcal{Q}(s_{k-1}) > \mathcal{Q}(s_k)$. Conjugacy also allows the directions to be generated using a simple two-term recurrence relation and gives the subspace minimizer as $s_k = s_{k-1} + \alpha_{k-1} p_{k-1}$, where α_{k-1} is the minimizer of the univariate function $\mathcal{Q}(s_{k-1} + \alpha p_{k-1})$.

The focus of this paper is on the "Lanczos-CG" variant of the CG method, which defines the conjugate directions in terms of quantities used to transform H to tridiagonal form (see Paige and Saunders [29]). At the kth step ($k \geq 1$), the new conjugate direction p_{k-1} is computed in terms of p_{k-2} and the last column of the matrix V_k such that

$$HV_k = V_k T_k + \beta_k v_k e_k^T, \tag{3.5}$$

where $V_k = \begin{pmatrix} v_0 & v_1 & \cdots & v_{k-1} \end{pmatrix}$ with $V_k^T V_k = I$, and T_k is the tridiagonal matrix

$$T_k = \begin{pmatrix} \gamma_0 & \beta_1 & & & \\ \beta_1 & \gamma_1 & \beta_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_{k-1} \\ & & & \beta_{k-1} & \gamma_{k-1} \end{pmatrix}. \tag{3.6}$$

Given $v_0 = -g/\|g\|$, the Lanczos vectors v_1, v_2, ..., v_k are generated using a two-term recurrence relation that requires one matrix-vector product at each step. The LDL^T factorization $T_k = L_k D_k L_k^T$ provides the conjugate directions by means of the identity $V_k = P_k L_k^T$. Paige and Saunders establish the identity $g + Hs_k = \alpha_{k-1}\beta_k v_k$, which implies that s_k will be an exact solution of $Hs = -g$ if $\alpha_{k-1}\beta_k = 0$. (If $\beta_k = 0$ then T_k is *reducible* and the Lanczos vectors form an invariant subspace of H. The matrix H is *irreducible* if $\beta_k \neq 0$ for all k.)

Algorithm 3.1 below defines the Lanczos-CG method for finding an approximate solution of a positive-definite linear system. If τ is a preassigned scalar tolerance, the calculation ***Lanczos-CG***$(H, -g, \tau\|g\|)$ defines an approximate solution of $Hs = -g$ such that $\|g + Hs\| \leq \tau\|g\|$. At each iteration, the vector v is the most recently computed Lanczos vector and $\bar{v}$ is the previous value of v. The scalars γ and β are the diagonals and off-diagonals of the tridiagonal matrix (3.6). The Lanczos vectors are scaled so that the off-diagonal elements β are nonpositive. This makes the step α_{k-1} and direction p_{k-1} identical to those of the standard conjugate-gradient method of Hestenes and Steifel [23].

Algorithm 3.1. $[x] :=$ ***Lanczos-CG***$(A, b, \tau_{\text{tol}})$
$x := 0$; $q := b$; $\beta := -\|q\|$; $\alpha := 1$; $\tau := -\beta$; $j := -1$;
while $\tau > \tau_{\text{tol}}$ **do**
  $v := q/\beta$; $j := j + 1$;
  $[Av] := Av$; $\gamma := v^T[Av]$;
  if $j = 0$ **then**
    $l := 0$; $p := v_j$;
    $q := [Av] - \gamma v$;
  else
    $l := \beta/d$; $p := v - lp$;
    $q := [Av] - \gamma v - \beta\bar{v}$;
  end
  $d := \gamma - \beta l$; $\alpha := -\beta\alpha/d$;
  $x := x + \alpha p$;
  $\beta := -\|q\|$; $\tau := -\beta\alpha$; $\bar{v} := v$;
end do

3.1. The Steihaug-Toint method

Toint [30] and Steihaug [40] independently proposed CG-based methods for solving the trust-region problem. Their methods start by computing the CG iterates for the system $Hs = -g$ (or, equivalently, for the unconstrained minimizer of $\mathcal{Q}(s)$). In the ideal situation, H is positive definite and the Newton step $-H^{-1}g$ lies inside the trust region. In this case the CG iterations are terminated when the condition $\|g + Hs_k\| \leq \tau\|g\|$ is satisfied, in which case s_k approximates the unconstrained step $-H^{-1}g$.

Steihaug establishes the key property that if $p_j^T H p_j > 0$ for $0 \leq j \leq k-1$, then the norms of the CG iterates $\{s_k\}$ are strictly increasing, i.e., $\|s_k\| > \|s_{k-1}\|$. In the context of solving the trust-region problem, this implies that there is no reason to continue computing CG iterates once they cross the trust-region boundary. In particular, if the condition $\|Hs_{k-1} + g\| \leq \tau\|g\|$ is not satisfied and either

$$p_{k-1}^T H p_{k-1} \leq 0 \quad \text{or} \quad \|s_{k-1} + \alpha_{k-1} p_{k-1}\| \geq \delta, \tag{3.7}$$

then the solution of (1.1) lies on the boundary of the trust region and the CG iterations are terminated. If one of the conditions (3.7) 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

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

(Toint redefines s_k as the Cauchy point if $p_{k-1}^T H p_{k-1} \leq 0$.) An important property of both the Toint and Steihaug methods is that the approximate solution is always at least as good as the Cauchy point. As a result, the underlying trust-region algorithms is globally convergent to a first-order point when endowed with an appropriate strategy for adjusting the trust-region radius.

3.2. The generalized Lanczos trust-region method

The Steihaug-Toint method accepts the first point computed on the boundary, regardless of its accuracy as a solution of (1.1). This implies that s_k may be a poor approximate solution of (1.1) in the constrained case. This lack of accuracy control was noted by Gould, Lucidi, Roma and Toint [17], who proposed solving the constrained problem using the generalized Lanczos trust-region (GLTR) method, which is a subspace minimization method defined on an expanding sequence of subspaces generated by the Lanczos vectors. The subspace minimization problem (3.2) is solved with the columns of P_k defined using the Lanczos process. This gives the reduced problem

$$\underset{y \in \mathbb{R}^k}{\text{minimize}} \;\; g^T V_k y + \tfrac{1}{2} y^T T_k y, \quad \text{subject to} \;\; \|y\| \leq \delta, \tag{3.8}$$

which may be solved using a variant of the Moré-Sorensen algorithm 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 $s_k = V_k y_k$ in the full space must

be computed. This implies that the columns of V_k must be stored explicitly or regenerated by repeating the Lanczos recurrence.

If the Lanczos process is always restarted when T_k is reducible, then, in theory, the GLTR method may be used to solve the trust-region problem to arbitrary accuracy. However, the need to regenerate V_k in the constrained case substantially increases the number of matrix-vector products. Another, more serious difficulty is that rounding errors quickly lead to a loss of orthogonality of the Lanczos vectors. This loss of orthogonality implies that the solution of reduced problem (3.8) rapidly diverges from the required solution, which is based on the problem

$$\underset{y\in\mathbb{R}^k}{\text{minimize}}\ \ g^TV_ky + \tfrac{1}{2}y^TV_k^THV_ky \quad \text{subject to}\ \ \|V_ky\| \leq \delta.$$

These considerations prompt Gould et al. to impose a modest limit on the number Lanczos iterations in the constrained case. (When GLTR is used as part of a trust-region method for unconstrained optimization, the Steihaug point is accepted if is within 90% of the best value found so far.)

3.3. Sequential subspace minimization (SSM) methods

In [20], Hager considers subspace minimization methods for finding an exact solution of the equality constrained problem:

$$\underset{s\in\mathbb{R}^n}{\text{minimize}}\ \ \mathcal{Q}(s) = g^Ts + \tfrac{1}{2}s^THs \quad \text{subject to}\ \ s^Ts = \delta^2. \tag{3.9}$$

In contrast to the Steihaug-Toint and GLTR methods, which generate a sequence of *expanding* subspaces, Hager's method relies on generating good quality low-dimensional subspaces. At the start of the kth iteration, values (s_{k-1}, σ_{k-1}) are known such that $s_{k-1}^Ts_{k-1} = \delta^2$ and $\sigma_{k-1} \in (-\lambda_n, \infty)$ (cf. Theorem 2.1). The kth iterate (s_k, σ_k) is a solution of the subspace minimization problem

$$\underset{s\in\mathbb{R}^n}{\text{minimize}}\ \ \mathcal{Q}(s) \quad \text{subject to}\ \ s^Ts = \delta^2, \quad s \in \mathcal{S}_k, \tag{3.10}$$

where $\mathcal{S}_k = \text{span}\{s_{k-1}, \nabla\mathcal{Q}(s_{k-1}), z_0, s_{\text{SQP}}\}$. The use of the previous iterate s_{k-1} in $\mathcal{S}_k$ guarantees that $\mathcal{Q}(s_k) < \mathcal{Q}(s_{k-1})$. The vector z_0 is the best estimate of the leftmost eigenvector computed as part of a startup phase that solves a reduced problem of dimension $\ell = \max\{10, n/100\}$. (The startup problem may be solved a number of times.) The vector s_{SQP} is computed from one step of Newton's method applied to (3.9). As the Newton equations are not positive definite, Hager uses a projected method to ensure that the CG iterates are well defined. This method is equivalent to applying the CG method with constraint preconditioning (see, e.g., [16,24]). These methods require that the initial iterate satisfies the constraint, which implies that the reduced problem (3.10) must be solved to high accuracy.

Hager and Park [21] show that any SSM method based on a subspace $\mathcal{S}_k$ containing the vectors s_{k-1}, $\nabla\mathcal{Q}(s_{k-1})$ and u_n is globally convergent to a solution of the trust-region problem. This result provides a justification of the composition of $\mathcal{S}_k$, but it does not constitute a convergence proof for the SSM method because u_n is unknown in general.

4. Phased Sequential Subspace Minimization

The proposed method combines three basic components: (i) the Lanczos variant of the CG method for solving positive-definite linear equations, (ii) a simple inexpensive method for estimating the leftmost eigenpair of H; and (iii) a sequential subspace minimization method in which the low-dimensional subspace includes a regularized Newton accelerator direction. The method generates a sequence $\{s_k\}$ such that $\|s_k\| \leq \delta$ and $\mathcal{Q}(s_k)$ is the best value of $\mathcal{Q}$ found so far. On termination, s_k satisfies $\mathcal{Q}(s_k) < \mathcal{Q}(s_c)$, where s_c is the Cauchy point defined in (3.1).

The phased-SSM method has two phases. The first is comprised of an extended version of the Steihaug-Toint method in which two additional features are provided: (a) the use of an inexpensive estimate of the leftmost eigenvector; and (b) the use of a low-dimensional subspace minimization a better exit point on the boundary in the constrained case. Property (a) extends the Steihaug-Toint method to the case where the starting point is a stationary point but not a local minimizer. Property (b) allows the calculation of a substantially better estimate of a trust-region solution on the boundary.

The second phase is activated if Phase 1 is terminated at a point on the trust region boundary that has insufficient accuracy. In Phase 2, a CG-based SSM method is used to solve the constrained problem over a sequence of evolving low-dimensional subspaces. Each subspace is spanned by three vectors: the current best approximate solution; an estimate of the leftmost eigenvector; and the regularized Newton accelerator direction.

The Lanczos process is the "driving mechanism" for both phases. The Lanczos vectors not only generate the conjugate directions for solving the positive-definite equations of both phases, but also provide independent vectors for the definition of the evolving low-dimensional subspaces associated with the reduced versions of the trust-region and leftmost eigenvector problems. As these processes require a steady stream of Lanczos vectors within each phase, the Lanczos process is restarted with a random initial vector if the tridiagonal matrix is reducible (i.e., if an off-diagonal element β_k of T_k is zero).

To allow for the case $g = 0$ with H indefinite, a preassigned positive scalar tolerance τ_0 is used to define a "zero" vector g. If $\|g\| > \tau_0$, the Lanczos process is initialized with $v_0 = -g/\|g\|$. Otherwise, the vector g is assumed to be negligible and v_0 is set to be a normalized random vector.

4.1. Phase 1 overview

In the first phase, standard CG iterates are generated until a sufficiently accurate solution of $Hs = -g$ is found inside the trust region or it becomes evident that the solution lies on the boundary. The Lanczos-CG method of Algorithm 3.1 is used to generate the CG iterates (see Section 3). The principal cost of each CG iteration is the matrix-vector product associated with the Lanczos two-term recurrence relation.

Embedded in the Lanczos-CG algorithm is the calculation of an estimate of a

leftmost eigenpair. The estimate is computed by solving the reduced eigenproblem

$$\underset{z\in\mathbb{R}^n}{\text{minimize}}\ z^THz \quad \text{subject to } \|z\| = 1,\ \ z \in \mathcal{Z}_k, \tag{4.1}$$

where $\mathcal{Z}_k = \text{span}\{v_k, z_{k-1}\}$, with v_k the most recently computed Lanczos vector, and z_{k-1} the leftmost eigenvector estimate from the previous CG iteration.

Given the matrix Z_k whose columns form a maximally linearly independent subset of $\{v_k, z_{k-1}\}$, the solution z_k of (4.1) may be written as $z_k = Z_k w_k$, where w_k solves the reduced problem

$$\underset{w}{\text{minimize}}\ w^TZ_k^THZ_kw \quad \text{subject to } w^TZ_k^TZ_kw = 1.$$

This problem is at most two dimensional, and may be solved in closed form. Once z_k has been determined, the leftmost eigenvalue is estimated by the Rayleigh quotient $\zeta_k = z_k^THz_k$. The inclusion of z_{k-1} in the reduced space $\mathcal{Z}_k$ ensures that the Rayleigh quotients decrease monotonically.

The eigenpair estimate is available at almost no additional cost. Apart from the calculation of Hz_0, the estimation of the leftmost eigenpair involves no additional matrix-vector products. To see this, note that the calculation of $Z_k^THZ_k$ requires the vectors Hz_{k-1} and Hv_k. The vector Hv_k is available as part of the two-term Lanczos recurrence, and Hz_{k-1} is available as part of the previous reduced eigenproblem. For the next step, the vector Hz_k is defined in terms of the identity $Hz_k = HZ_kw_k$, which involves a simple linear combination of Hv_k and Hz_{k-1}. The calculation of the eigenpair is summarized in Algorithm 4.1 below.

In the context of the ith iteration of a method for unconstrained minimization, the vector z_0 used to start the first trust-region problem (i.e., $i = 0$) is a normalized random vector. In subsequent iterations, z_0 is the final eigenvector estimate associated with the previous trust-region problem. Thus, the initial generalized eigenvalue problem is solved over the subspace $\mathcal{Z}_0 = \text{span}\{v_0, z_0\} = \text{span}\{-g, z_0\}$. As the unconstrained solver converges, the sequence of Hessians $\{H_i\}$ converges, and z_0 should be a good estimate of the leftmost eigenvector for the current Hessian.

Algorithm 4.1. $\big[z, \zeta, [Hz]\big] = \textbf{\textit{subspaceEig}}\big(z, v, [Hz], [Hv]\big)$
Define Z from a maximally linearly independent subset of v and z;
Form Z^THZ and Z^TZ from z, v, $[Hv]$ and $[Hz]$;
$w := \text{argmin}\ \big\{\ z^TZ^THZz\ :\ z^TZ^TZz = 1\ \big\}$;
$z := Zw$; $\quad \zeta = z^THz$;
$[Hz] := [HZ]w$;

4.2. Phase 1 termination

In the first phase, Lanczos-CG iterates s_k and leftmost eigenpair $(z_k, \zeta_k) = (z_k, z_k^THz_k)$ until one of several termination conditions are satisfied. Termination may occur at an interior or boundary point.

Termination inside the trust region The Lanczos-CG iterates are terminated inside the trust-region if the following conditions hold.

$$\begin{aligned} &\big(\|g\| > \tau_0 \ \textbf{and} \quad (\|g + Hs_k\| \leq \tau_1\|g\| \ \textbf{or} \ |\beta_k| \leq \max\{1, \gamma_{\max}\}\sqrt{\epsilon_M}\,)\big) \\ \textbf{or} \ &\big(\|g\| \leq \tau_0 \ \textbf{and} \ \|\zeta_k z_k - Hz_k\| \leq \tau_1\|\zeta_0 z_0 - Hz_0\|\big), \end{aligned} \tag{4.2}$$

where τ_0 and τ_1 are preassigned scalars and$\gamma_{\max}$ is the diagonal of T_k with largest magnitude, i.e., $\gamma_{\max} = \max_{0\leq i\leq k-1} |\gamma_i|$. (See Section 5 for more details concerning the definition of τ_0 and τ_1 in the context of a trust-region method for unconstrained minimization.) When $\|g\| > \tau_0$, the first condition of (4.2) ensures that the size of the final residual is sufficiently reduced relative to the initial residual g. The condition $|\beta_k| \leq \max\{1, \gamma_{\max}\}\sqrt{\epsilon_M}\,)$ is used to detect the case where T_k is badly scaled and close to being reducible. If either of these tests is satisfied, the point s_k is considered to be an acceptable approximate solution of (1.1).

When $\|g\| \leq \tau_0$, the second condition of (4.2) is intended to provide an approximate leftmost eigenvector of H. If termination occurs with $\zeta_k > 0$ then it is likely that $s_k = 0$ is the solution of (1.1).

If the condition $|\beta_k| \leq \max\{1, \gamma_{\max}\}\sqrt{\epsilon_M}\,)$ holds when $\|g\| \leq \tau_0$, the matrix T_k is assumed to be reducible and the Lanczos process is restarted with a random vector.

Termination on the boundary Phase 1 is terminated if any one of the following events occur:

(i) Lanczos-CG generates an iterate s_k that lies outside of the trust region.

(ii) Lanczos-CG computes a direction p_{k-1} such that $p_{k-1}^T H p_{k-1} \leq 0$.

(iii) The Rayleigh quotient $\zeta_k = z_k^T H z_k$ is negative, where z_k is the estimate of the leftmost eigenvector.

The occurrence of any one of these events implies that the solution of (1.1) must lie on the constraint boundary. A final point on the boundary is defined by solving the trust-region subproblem over the subspace

$$\mathcal{S}_k = \text{span}\{s_{k-1}, p_{k-1}, z_k\},$$

where s_{k-1} is the last CG iterate inside the trust region, p_{k-1} is the last CG direction, and z_k is the approximation to the leftmost eigenvector. The reduced problem has the form

$$\underset{y}{\text{minimize}} \ \ g^T P_k y + \tfrac{1}{2} y^T P_k^T H P_k y, \quad \text{subject to} \ \ \|P_k y\| \leq \delta, \tag{4.3}$$

where P_k is a matrix whose columns span $\mathcal{S}_k$. The QR decomposition with column interchanges may be used to determine a maximally linearly independent subset of the vectors $\{s_{k-1}, p_{k-1}, z_k\}$. The calculations associated with the solution of the reduced problem are given in Algorithm 4.2. As in Algorithm 4.1, the quantities $P_k^T H P_k$ and $P_k^T P_k$ may be formed with no additional matrix-vector products. On exit, the vectors s_k and Hs_k are defined in readiness for the start of Phase 2.

Algorithm 4.2. $\big[s, \sigma, [Hs]\big] := \textbf{\textit{subspaceSolve}}\big(s, p, z, [Hs], [Hp], [Hz]\big)$
Define P from a maximally linearly independent subset of s, p and z;
Form P^THP, P^TP and P^Tg from s, p and z, $[Hs]$, $[Hp]$ and $[Hz]$;
$y := \operatorname{argmin}\big\{\, g^TPy + \frac{1}{2}y^TP^THPy \;:\; y^TP^TPy = \delta \,\big\}$;
$s := Py$; $\quad [Hs] := [HP]y$;

The reduced problem has at most three dimensions, and may be solved efficiently by a method that exploits direct matrix factorizations (the Moré-Sorensen algorithm was used to obtain the results of Section 5).

4.3. Phase 2 overview

For Phase 2 to be initiated, the solution of the trust-region problem must lie on the boundary of the trust region constraint, which implies that the Phase 2 iterations must minimize $\mathcal{Q}(s)$ subject to the equality constraint $\|s\| = \delta$. Without loss of generality, we consider the equivalent problem

$$\underset{s\in\mathbb{R}^n}{\text{minimize}} \;\; \mathcal{Q}(s) = g^Ts + \tfrac{1}{2}s^THs \;\; \text{subject to} \;\; \tfrac{1}{2}\delta^2 - \tfrac{1}{2}s^Ts = 0. \tag{4.4}$$

The Phase 2 algorithm refines the solution on the boundary by solving the subspace constrained minimization problem

$$\underset{s\in\mathbb{R}^n}{\text{minimize}} \;\; \mathcal{Q}(s) \;\; \text{subject to} \;\; \|s\| \le \delta, \quad s \in \mathcal{S}_k = \operatorname{span}\{s_{k-1}, p_k, z_k\}, \tag{4.5}$$

where s_{k-1} is the current best approximate solution, p_k is a Newton "accelerator" direction defined below, and z_k is the current best estimate of the leftmost eigenvector of H. The inclusion of the best approximation s_{k-1} in $\operatorname{span}\{s_{k-1}, p_k, z_k\}$ guarantees that $\mathcal{Q}(s)$ decreases at each step. The reduced problem has at most three dimensions, and may be solved using Algorithm 4.2. with $s = s_{k-1}$, $p = p_k$ and $z = z_k$.

4.4. Definition of the Newton accelerator direction

The accelerator direction p_k is defined as one step of a regularized Newton method applied to the equality constrained problem (4.4). Given a nonnegative scalar Lagrange multiplier σ, the Lagrangian function associated with (4.4) is

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

where $c(s)$ denotes the value of the constraint residual

$$c(s) = \tfrac{1}{2}s^Ts - \tfrac{1}{2}\delta^2. \tag{4.6}$$

The gradient of the Lagrangian with respect to s and σ is given by

$$\nabla L(s, \sigma) = \begin{pmatrix} \nabla\mathcal{Q}(s) + \sigma s \\ \frac{1}{2}s^Ts - \frac{1}{2}\delta^2 \end{pmatrix} = \begin{pmatrix} g + (H + \sigma I)s \\ c(s) \end{pmatrix}.$$

Similarly, the Hessian matrix of second derivatives is

$$\nabla^2 L(s,\sigma) = \begin{pmatrix} H+\sigma I & s \\ s^T & 0 \end{pmatrix}.$$

Optimal values of s and σ may be found by applying Newton's method to find a zero of the function $F(s,\sigma) \triangleq \nabla L(s,\sigma)$. Given an estimate $w = (s,\sigma)$ of a zero, Newton's method defines a new estimate $w + \Delta w$, where $\Delta w = (\Delta s, \Delta\sigma)$ is a solution of the Newton equations $F'(w)\Delta w = -F(w)$. This system may be written in terms of σ and s as:

$$\begin{pmatrix} H+\sigma I & s \\ s^T & 0 \end{pmatrix} \begin{pmatrix} \Delta s \\ \Delta\sigma \end{pmatrix} = -\begin{pmatrix} g + (H+\sigma I)s \\ c(s) \end{pmatrix}. \tag{4.7}$$

The Newton equations are indefinite and cannot be solved directly using the Lanczos-CG method. Instead, we solve a related system that is positive semidefinite in the neighborhood of (s^*,σ^*). This alternative system may be viewed as a *regularized* Newton system.

Given a positive scalar μ and a nonnegative scalar σ_e, consider the function of both s and σ given by

$$L_\mu(s,\sigma) = \mathcal{Q}(s) + \sigma_e c(s) + \frac{1}{2\mu}c(s)^2 + \frac{1}{2\mu}\big(\mu(\sigma-\sigma_e) - c(s)\big)^2.$$

The gradient and Hessian of $L_\mu(s,\sigma)$ with respect to (s,σ) are

$$\nabla L_\mu(s,\sigma) = \begin{pmatrix} g + (H+\sigma I)s + 2(\hat\sigma-\sigma)s \\ \mu(\sigma-\sigma_e) - c(s) \end{pmatrix},$$

and

$$\nabla^2 L_\mu(s,\sigma) = \begin{pmatrix} H+\sigma I + 2(\hat\sigma-\sigma)I + \frac{2}{\mu}ss^T & -s \\ -s^T & \mu \end{pmatrix},$$

where $\hat\sigma = \hat\sigma(s) = \sigma_e + c(s)/\mu$.

Theorem 4.1. *Let (s^*,σ^*) be a solution of (4.4), then there exists a $\bar\mu$ such that for all $\mu < \bar\mu$, the point (s^*,σ^*) minimizes the function*

$$\mathcal{Q}(s) + \sigma^* c(s) + \frac{1}{2\mu}c(s)^2 + \frac{1}{2\mu}\big(\mu(\sigma-\sigma^*) - c(s)\big)^2. \qquad \blacksquare$$

This result suggests that, given a nonnegative σ_e such that $\sigma_e \approx \sigma^*$, we may obtain a better estimate of (s^*,σ^*) by minimizing

$$L_\mu(s,\sigma) = \mathcal{Q}(s) + \sigma_e c(s) + \frac{1}{2\mu}c(s)^2 + \frac{1}{2\mu}\big(\mu(\sigma-\sigma_e) - c(s)\big)^2$$

with respect to both s and σ. The Newton equations for minimizing $L_\mu(s,\sigma)$ are:

$$\begin{pmatrix} H + (\sigma + 2(\hat\sigma-\sigma))I + \frac{2}{\mu}ss^T & -s \\ -s^T & \mu \end{pmatrix} \begin{pmatrix} \Delta s \\ \Delta\sigma \end{pmatrix} = -\begin{pmatrix} g + (H + (\sigma+2(\hat\sigma-\sigma)I)s \\ \mu(\sigma-\sigma_e) - c(s) \end{pmatrix},$$

or, equivalently,

$$\begin{pmatrix} H + \bar{\sigma} I + \frac{2}{\mu} s s^T & -s \\ -s^T & \mu \end{pmatrix} \begin{pmatrix} \Delta s \\ \Delta \sigma \end{pmatrix} = - \begin{pmatrix} g + (H + \bar{\sigma} I) s \\ \mu(\sigma - \sigma_e) - c(s) \end{pmatrix}, \tag{4.8}$$

where $\bar{\sigma} = \sigma + 2(\hat{\sigma} - \sigma) = 2\hat{\sigma} - \sigma$.

The Moré-Sorensen algorithm applied to the reduced problem provides estimates of both σ^* and s^*. This suggests that the optimal σ from the reduced problem (4.5) is a good choice for σ_e.

The linear system (4.8) has only one row and column more than the equations associated with the unconstrained case. The Lanczos-CG method may be used to compute an approximate Newton step. The accuracy of the accelerator step effects only the rate of convergence to the constrained solution and does not effect the convergence properties of the SSM method. It follows that it is not necessary to minimize $L_\mu(s, \sigma)$ to high accuracy. In practice we define p_k as an approximate solution of the first Newton system (4.8). In addition, for reasons of efficiency, the number of Lanczos-CG iterations used to find an approximate solution of (4.8) is limited. In the results of Section 5 the iterations were limited to 10.

A benefit of using the Lanczos-CG method for solving (4.8) is that s need not satisfy $c(s) = 0$, i.e., s need not lie exactly on the boundary of the trust region.

The calculations associated with the definition of the Newton accelerator direction are given in Algorithm 4.3 below.

Algorithm 4.3. $[p, \sigma_p] := \textbf{\textit{NewtonAccelerator}}(p, \sigma_p, \sigma_e)$
Set $\hat{\sigma} = \sigma_e + c(p)/\mu$; $\bar{\sigma} = \sigma_p + 2(\hat{\sigma} - \sigma_p)$;
Set $[x] = \textbf{\textit{Lanczos-CG}}(A, b, \tau_{\text{tol}})$; where

$$A = \begin{pmatrix} H + \bar{\sigma} I + \frac{2}{\mu} p p^T & -p \\ -p^T & \mu \end{pmatrix}, \qquad b = - \begin{pmatrix} g + (H + \bar{\sigma} I) p \\ \mu(\sigma_p - \sigma_e) - c(p) \end{pmatrix};$$

$\Delta p := x_{1:n}$; $\Delta \sigma_p := x_{n+1}$;
$\alpha_M =$ **if** $\Delta \sigma_p < 0$ **then** $(\sigma_p + \Delta \sigma_p - \sigma_\ell)/\Delta \sigma_p$ **else** $+\infty$;
$\alpha_M := \min\{1, \eta \alpha_M\}$;
Compute α $(0 < \alpha \leq \alpha_M)$ satisfying the Wolfe line search conditions for $L_{\mu, \sigma_e}(s, \sigma)$;
$p := p + \alpha \Delta p$; $\sigma_p := \sigma_p + \alpha \Delta \sigma_p$;

4.5. Phase 2 termination

Given a positive tolerance τ_2, the Phase 2 iterations are terminated if $r_{\mathcal{S}} \leq \tau_2 \|g\|$, where $r_{\mathcal{S}}$ is the residual associated with the current best estimate (s, σ_e), i.e.,

$$r_{\mathcal{S}} = \|g + (H + \sigma_e I) q\| + \sigma_e |c(s)|, \tag{4.9}$$

where $q = P\bar{q}$ with $\bar{q}$ the solution of the reduced system analogous to (2.3), i.e.,

$$(P^T H P + \sigma_e P^T P)\bar{q} = P^T g.$$

The condition (4.9) takes into account that the Moré-Sorensen algorithm gives only an *approximate* solution of the reduced problem. The reduced trust-region problem must be solved to an accuracy that is at least as good as that required for the full problem. Suitable values for the constants c_1 and c_2 of (2.4) are $c_1 = \min\{10^{-1}\tau_2, 10^{-6}\}$ and $c_2 = 0$.

Similarly, we define the error in the optimality conditions for the Newton accelerator (p, σ_p) (the approximate minimizer of L_{μ,σ_e}). In this case, the residual is:

$$r_A = \|g + (H + \sigma_p I)p\| + \sigma_p |c(p)|. \tag{4.10}$$

In practice, the residual associated with the accelerator is generally larger than the residual associated with the reduced-problem solution. To see this, consider the value of $|c(s)|$, the error in the "constraint" part of the optimality conditions. As the Newton system is not being solve accurately, the value of $|c(s)|$ at a typical Newton iterate may be large even when the solution lies on the boundary. By contrast, every solution of the reduced subproblem will have $|c(s)|$ of the order of the Moré-Sorensen tolerance c_1. (cf. (2.4)).

4.6. Properties of the Newton accelerator

The method above may be regarded as a regularization of Newton's method. If both sides of the system (4.8) are multiplied by the nonsingular matrix

$$\begin{pmatrix} I & \frac{2}{\mu}s \\ 0 & 1 \end{pmatrix},$$

and the last row is scaled by -1 we obtain

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

If $c(s) \approx 0$ and $\sigma_e \approx \sigma$, these equations are a perturbation of the Newton equations (4.7). The following theorem shows that the perturbation μ serves as a *regularization* parameter in the degenerate case.

Theorem 4.2. (Regularization of the degenerate case) *Suppose that* (s, σ) *denotes a solution of the trust-region subproblem and that* (i) $\|s\| = \delta$; (ii) $H + \sigma I$ *is positive semidefinite and singular;* (ii) $g \in \text{null}(H+\sigma I)^{\perp}$; *and* (iii) $\|(H+\sigma I)^{\dagger} g\| < \delta$. *If the leftmost eigenvalue of* H *has algebraic multiplicity 1, then the augmented system matrix*

$$\begin{pmatrix} H + \sigma I + \frac{2}{\mu} s s^T & -s \\ -s^T & \mu \end{pmatrix} \tag{4.11}$$

is positive definite.

Proof. Assumptions (i)–(iii) imply that (s, σ) is a 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

$$s = -(H - \lambda_n I)^\dagger g + \beta z, \tag{4.12}$$

where z is a unit vector such that $z \in \text{null}(H - \lambda_n I)$ and β is a nonzero scalar such that $\|s\| = \delta$. Consider the following decomposition of (4.11):

$$\begin{pmatrix} H + \sigma I + \frac{2}{\mu} ss^T & -s \\ -s^T & \mu \end{pmatrix} = \begin{pmatrix} I & -\frac{1}{\mu} s \\ 0 & 1 \end{pmatrix} \begin{pmatrix} H - \lambda_n I + \frac{1}{\mu} ss^T & 0 \\ 0 & \mu \end{pmatrix} \begin{pmatrix} I & 0 \\ -\frac{1}{\mu} s^T & 1 \end{pmatrix}. \tag{4.13}$$

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

A safeguarded Newton method may be used to minimize $L_\mu(s, \sigma)$ with respect to both s and σ. Algorithm 4.4 is an approximate safeguarding scheme that attempts to ensure the system in (4.8) is positive definite based on the current values of σ_e, σ_ℓ, σ_p, $c(p) = \frac{1}{2} p^T p - \delta^2$, and the error in the optimality conditions. At all times, the safeguarding algorithm ensures that both σ_p and σ_e are greater than the current estimate σ_ℓ of $\max\{-\lambda_n, 0\}$. The algorithm adjusts σ_e so that the matrix $H + \bar{\sigma} I$ of (4.8) is positive definite. In this algorithm, the iterates $\{s_k\}$ are never overwritten, i.e., the solution to the subspace solve is preserved. Also, the Newton iterates (p, σ_p) are only overwritten with the subspace solve iterates if $\sigma_p < \sigma_\ell$ and $\sigma_e > \sigma_\ell$. In the event that both σ_p and σ_e are less than σ_ℓ, the leftmost eigenpair is used to update the iterates.

Provided $\sigma \in (-\lambda_n, \infty)$, then with safeguarding, the system (4.8) is positive definite, and thus, can be solved using CG. The following theorem shows that if CG detects that the matrix in (4.8) is indefinite, the conjugate direction can be used to safeguard the system in subsequent iterations, as well as to update the leftmost eigenvector estimate.

Theorem 4.3. *Assume that p is a direction of negative curvature for the matrix*

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

where μ is a positive scalar. Then the vector of first n elements of p is a direction of negative curvature for $H + \sigma I$.

Proof. The result follows trivially from the identity

$$B = \begin{pmatrix} H + \sigma I & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} I & -\frac{1}{\mu} s \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{\mu} ss^T & 0 \\ 0 & \mu \end{pmatrix} \begin{pmatrix} I & 0 \\ -\frac{1}{\mu} s^T & 1 \end{pmatrix},$$

Algorithm 4.4. $[p, \sigma_e, \sigma_\ell] := \textbf{\textit{safeguard}}(p, s, \sigma_e, \sigma_\ell)$

if $\sigma_p < \sigma_\ell$ **and** $\sigma_\ell < \sigma_e$ **then** $\sigma_p := \sigma_e$; $p := s$; **end**
$\hat{\sigma} = \sigma_e + c(p)/\mu$; $\bar{\sigma} = \sigma_p + 2(\hat{\sigma} - \sigma_p)$;
if $\bar{\sigma} < \sigma_\ell$ **then**
  if $\sigma_p < \sigma_\ell$ **and** $\sigma_\ell < \sigma_e$ **then**
    $\sigma_p := \sigma_e$; $p := s$; $\hat{\sigma} = \sigma_e + c(p)/\mu$; $\bar{\sigma} = \sigma_p + 2(\hat{\sigma} - \sigma_p)$;
    if $\bar{\sigma} < \sigma_\ell$ **then** $\sigma_e := \sigma_p + |c(p)|/\mu$; **end**
  else if $\sigma_p > \sigma_\ell$ **and** $\sigma_e < \sigma_\ell$ **then**
    if $c(p) > 0$ **then** $\sigma_e := \sigma_p$; **else** $\sigma_e := \sigma_p + |c(p)|/\mu$; **end**
  else if $\sigma_p > \sigma_\ell$ **and** $\sigma_e > \sigma_\ell$ **then**
    $r_{\mathcal{S}} = \|g + (H + \sigma_e I)q\| + \sigma_e |c(s)|$; $r_A = \|g + (H + \sigma_p I)p\| + \sigma_p |c(p)|$;
    if $r_{\mathcal{S}} < r_A$ **then**
      $p := s$; $\sigma_p := \sigma_e$;
      $\hat{\sigma} = \sigma_e + c(p)/\mu$; $\bar{\sigma} = \sigma_p + 2(\hat{\sigma} - \sigma_p)$;
      if $\bar{\sigma} < \sigma_\ell$ **then** $\sigma_e := \sigma_p + |c(p)|/\mu$; **end**
    else
      $\sigma_e := \sigma_p$; $\hat{\sigma} = \sigma_e + c(p)/\mu$; $\bar{\sigma} = \sigma_p + 2(\hat{\sigma} - \sigma_p)$;
      if $\bar{\sigma} < \sigma_\ell$ **then** $\sigma_e := \sigma_p + |c(p)|/\mu$; **end**
    end
  else
    $\sigma_p := |\zeta|$; $\sigma_e := |\zeta|$; $p := \delta \times z$;
    $\hat{\sigma} = \sigma_e + c(p)/\mu$; $\bar{\sigma} = \sigma_p + 2(\hat{\sigma} - \sigma_p)$;
    if $\bar{\sigma} < \sigma_\ell$ **then** $\sigma_e := \sigma_p + |c(p)|/\mu$; **end**
  end
end

and the fact that the second term in the matrix sum is positive semidefinite. ∎

In each iteration of Phase 2, the regularization parameter μ was defined as

$$\mu = \begin{cases} \min\{10^{-2}, \frac{1}{2}\bar{\mu}\} & \text{if } \zeta < 0; \\ 10^{-2} & \text{otherwise,} \end{cases}$$

where $\bar{\mu} = -2\|p\|^2/(\zeta + \sigma_p)$.

5. Numerical Results

The Steihaug-Toint and phased-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 [18]). 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 locate the twice-continuously differentiable unconstrained problems for which the number of variables in the data file can be varied. The final test set consisted of 49 problems. For all problems, the dimension was $n = 1000$ unless otherwise recommended in the CUTEr documentation. In all cases, $n \geq 1000$. A combination line-search trust-region method was used to define the update to the trust-region radius.

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

$$\|g(x_j)\| \leq \max\{\epsilon\|g(x_0)\|, \epsilon|f(x_0)|, \sqrt{\epsilon_M}\}, \tag{5.1}$$

where $\epsilon = 10^{-6}$ and ϵ_M denotes machine precision. 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.

5.1. Termination of Phase one and Lanczos-CG

In the Steihaug-Toint method and in phase one of the phased-SSM method, the principal termination condition is based on the Dembo-Eisenstat-Steihaug criterion [6]. In particular, if s_j denotes the approximate solution of the jth trust-region problem, then the Lanczos-CG method terminates successfully with a point s_j inside the trust region if

$$\|g_j + H_j s_j\| \leq \tau_{1j}\|g_j\|, \quad \text{where} \quad \tau_{1j} = \min\left\{10^{-1}, \|g_j\|^{0.1}\right\}. \tag{5.2}$$

This condition forces a relative decrease in the residual comparable to that required by Gould et al. [17].

5.2. Termination of phase two

In Phase 2 the test for convergence immediately follows the solution of the reduced problem. A user-specified parameter ϵ_s ($0 < \epsilon_s \leq 1$) allows control over the accuracy of the trust-region solution in the constrained case. The parameter τ_2 for the jth step is:

$$\tau_{2j} = \frac{1}{\epsilon_s} \min\left\{10^{-1}, \|g_j\|^{0.1}\right\}. \tag{5.3}$$

The value $\epsilon_s = 1$ corresponds to solving the constrained problem to the same accuracy as the unconstrained problem. The value $\epsilon_s \approx \epsilon_M$ corresponds to accepting the Steihaug point as the Phase 2 solution.

The iteration limit imposed in Phase 2 is smaller than that imposed on Phase 1. If Phase 2 is not converging well, this usually implies that the estimate of the leftmost eigenpair is poor. In this case, it is sensible to terminate the solution of the subproblem. In all the runs reported here, a limit of 10 iterations was enforced during the second phase. In all runs, a limit of 50 Lanczos vectors was imposed for the calculation of the Newton accelerator direction. If this iteration limit is reached, the Lanczos-CG iterate with the smallest residual is returned as the accelerator direction.

5.3. 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 a line search based on Gertz's "biased" Wolfe line search (see Gertz [12]). In Algorithm 5.1 below,

$$\mathcal{Q}_j^-(s) = g_j^T s + \tfrac{1}{2}\big[s^T H_j s\big]_- \tag{5.4}$$

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

$$\frac{f\big(x_j + s_j(\alpha_j)\big) - f(x_j)}{\mathcal{Q}_j^-\big(s_j(\alpha_j)\big)} > \eta_1, \tag{5.5}$$

where η_1 is a preassigned scalar such that $0 < \eta_1 < \frac{1}{2}$. The line search parameters used for the experiments were $\eta_1 = 10^{-4}$, $\eta_2 = 0.25$, $\omega = 0.9$, and $\gamma_3 = 1.5$.

Algorithm 5.1. Combination Line Search/Trust-Region Method.
Specify constants $0 < \eta_1 < \eta_2 < 1$; $0 < \eta_1 < \frac{1}{2}$; $0 < \eta_1 < \omega < 1$; $1 < \gamma_3$;
Find α_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 $\|s_j\| = \delta_j$ **and** $\alpha_j = 1$ **then**
    $\delta_{j+1} = \gamma_3 \delta_j$;
  else if $\|s_j\| < \delta_j$ **and** $\alpha_j = 1$ **then**
    $\delta_{j+1} = \max\{\delta_j, \gamma_3 \|s_j\|\}$;
  else
    $\delta_{j+1} = \alpha_j \|s_j\|$;
  end if
else
  $\delta_{j+1} = \min\{\alpha_j \|s_j\|, \alpha_j \delta_j\}$;
end if

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. Algorithm 5.1 above differs from Gertz's line search in that it is possible for the trust-region radius to be reduced even when α_j is small. Nevertheless, Algorithm 5.1 still retains a natural bias against decreasing the trust-region radius; in particular, the trust-region radius is not decreased if $\|s_j\| < \delta_j$ and $\alpha_j = 1$.

Tables 1–2 give the results of applying the Steihaug-Toint method and the phased-SSM method with $\epsilon_s = 1$ on the 49 problems from the CUTEr test set. For each solver, the columns give the total number of function evaluation ("Fe"), the

Table 1: Steihaug and phased-SSM on CUTEr problems a–e.

	Steihaug				phased-SSM ($\epsilon_s = 1$)			
Problem	fe	prods	$f(x)$	$\|g(x)\|$	fe	prods	$f(x)$	$\|g(x)\|$
arwhead	6	6	1.69e-10	6.37e-05	6	10	1.69e-10	6.37e-05
bdqrtic	14	41	3.98e+03	7.95e-02	13	44	3.98e+03	1.65e-01
broydn7d	139	404	3.75e+02	3.70e-04	75	1805	3.45e+02	6.33e-04
brybnd	12	46	6.65e-07	5.76e-03	12	57	3.93e-07	4.56e-03
chainwoo	27	57	1.31e+01	2.08e+00	25	99	3.93e+03	9.74e-01
cosine	12	10	-9.99e+02	3.01e-04	12	17	-9.99e+02	3.32e-04
cragglvy	14	36	3.36e+02	3.60e-01	14	48	3.36e+02	3.17e-01
dixmaana	13	11	1.00e+00	1.43e-03	13	21	1.00e+00	2.27e-03
dixmaanb	13	11	1.00e+00	1.10e-03	13	21	1.00e+00	1.10e-03
dixmaanc	13	11	1.00e+00	1.74e-02	13	21	1.00e+00	1.78e-02
dixmaand	14	12	1.00e+00	2.63e-02	14	23	1.00e+00	2.63e-02
dixmaane	14	93	1.00e+00	2.72e-03	14	93	1.00e+00	4.48e-03
dixmaanf	15	30	1.00e+00	9.92e-03	15	42	1.00e+00	9.95e-03
dixmaang	15	24	1.01e+00	2.43e-02	15	36	1.01e+00	2.43e-02
dixmaanh	15	19	1.03e+00	7.15e-02	15	31	1.03e+00	7.17e-02
dixmaanj	16	40	1.00e+00	5.71e-03	16	53	1.00e+00	5.72e-03
dixmaank	16	30	1.00e+00	1.41e-02	16	43	1.00e+00	1.41e-02
dixmaanl	16	25	1.01e+00	3.24e-02	16	38	1.01e+00	3.25e-02
dqdrtic	14	11	1.90e-03	1.21e+00	14	20	3.46e-05	1.19e-01
dqrtic	27	20	1.63e+11	1.98e+08	28	39	5.52e+10	8.83e+07
edensch	15	25	2.19e+02	4.36e-02	15	37	2.19e+02	4.41e-02
eg2	4	3	-9.99e+02	5.96e-09	4	5	-9.99e+02	5.96e-09
engval1	14	17	1.11e+03	2.63e-02	14	27	1.11e+03	2.50e-02
extrosnb	31	70	2.24e-02	2.46e-01	29	75	4.42e-02	1.50e-01

total number of matrix-vector products ("prods"), and the final values of f and $\|g\|$. The final values are listed to help identify local solutions and to identify cases where the converged gradient does not correspond to a local minimizer. (For problems with large $\|g_0\|$, requiring g_j to satisfy a small absolute tolerance is unreasonable).

Tables 1–2 show that Steihaug's method and phased-SSM method behave very similarly on many problems. These problems correspond to situations when the approximate solution of every subproblem is in the interior of the trust region. In these cases, phased-SSM method never enters the second phase and is as efficient as Steihaug's method. (Note that the extra matrix-vector products are associated with computing the initial estimate of the leftmost eigenvector for each Hessian).

We would expect the Steihaug-Toint method not to perform well in terms of the number of function evaluations when solutions of the trust-region subproblem frequently occur on the boundary. In these cases, the number of function evaluations required by the methods are sometimes significantly different, (e.g., see *broydn7*, *genrose*, *fminsurf*, or *fminsrf2*). On a few problems, the performance of phased-SSM

Table 2: Steihaug and phased-SSM on CUTEr problems f–z.

	Steihaug				phased-SSM ($\epsilon_s = 1$)			
Problem	fe	prods	$f(x)$	$\|g(x)\|$	fe	prods	$f(x)$	$\|g(x)\|$
fminsrf2	363	1515	1.00e+00	2.47e-05	60	2080	1.00e+00	2.05e-05
fminsurf	334	750	1.00e+00	2.47e-06	60	1270	1.00e+00	3.31e-06
freuroth	16	19	1.21e+05	5.41e-02	16	29	1.21e+05	5.41e-02
genrose	1218	5940	1.00e+00	3.04e-03	802	19004	1.00e+00	2.09e-03
liarwhd	19	27	4.03e-07	1.28e-03	19	41	1.51e-07	7.83e-04
ncb20	61	452	9.10e+02	2.92e-04	104	2836	9.18e+02	1.01e-03
ncb20b	10	61	1.68e+03	9.99e-04	9	79	1.68e+03	2.02e-04
noncvxu2	36	26	1.15e+06	2.21e+03	50	371	8.59e+05	1.69e+03
noncvxun	37	28	6.60e+05	2.19e+03	48	298	1.14e+06	2.32e+03
nondia	4	3	6.27e-03	9.86e-02	4	5	6.27e-03	9.86e-02
nondquar	23	115	5.52e-04	3.84e-03	18	151	5.26e-04	3.83e-03
penalty1	28	17	3.01e+13	5.14e+10	28	33	3.01e+13	5.14e+10
penalty2	2	1	1.45e+83	4.94e+38	2	1	1.45e+83	4.94e+38
powellsg	16	40	4.63e-03	3.54e-02	16	55	1.78e-03	1.73e-02
power	15	33	3.56e+04	1.25e+05	15	46	3.63e+04	1.28e+05
quartc	27	20	1.63e+11	1.98e+08	28	39	5.52e+10	8.83e+07
schmvett	9	37	-2.99e+03	6.79e-04	8	36	-2.99e+03	1.01e-03
sparsqur	14	23	4.24e-03	7.20e-02	14	47	4.27e-03	7.71e-02
spmsrtls	18	126	2.83e-09	9.33e-05	29	566	1.42e-08	2.45e-04
srosenbr	9	10	1.61e-09	3.58e-05	9	17	2.25e-09	4.24e-05
testquad	15	168	1.87e+01	1.09e+02	13	122	7.48e+01	2.44e+02
tointgss	15	13	1.00e+01	4.68e-03	15	22	1.00e+01	2.22e-03
vardim	13	12	6.87e+08	2.78e+10	13	23	6.87e+08	2.78e+10
vareigvl	14	26	3.54e-04	1.54e-02	14	37	3.54e-04	1.53e-02
woods	12	14	1.97e+03	2.49e+00	13	27	1.97e+03	2.66e-01

was slightly inferior to that of Steihaug's method. And, in one case (problem *ncb20*), phased-SSM performed significantly worse. The superiority of Steihaug's method in these cases appears to be the effect of good fortune rather than a consistently better subproblem solution.

As noted by Gould et al. [17], it is sometimes better not to solve the subproblem to high accuracy when the solution lies on the boundary. This may be especially true when the trust-region iterates are far from a minimizer of f. Tables 3–4 give results for different values of the tolerance ϵ_s in Phase 2, (i.e., when the subproblem solution lies on the boundary). In particular, the table gives the number of function evaluations and matrix-vector products required by phased-SSM for several values of ϵ_s in (4.9). (The recommended value is $\epsilon_s = 1$). The value $\epsilon_s = \epsilon_M$ has the effect of forcing phased-SSM to terminate before entering Phase 2. In this case, the results indicate that phased-SSM gives a significant reduction in the number of function evaluations for little or no sacrifice in computation time.

Table 3: Inexact phased-SSM on CUTEr problems a–e.

ϵ_s	ϵ_M		0.05		0.1		0.5		1.0	
Problem	fe	prods	fe	prods	fe	prods	fe	prods	fe	prods
arwhead	6	10	6	10	6	10	6	10	6	10
bdqrtic	13	44	13	44	13	44	13	44	13	44
broydn7d	134	487	75	859	67	737	78	1449	75	1805
brybnd	12	57	12	57	12	57	12	57	12	57
chainwoo	23	107	25	84	25	84	28	256	25	99
cosine	12	17	12	17	12	17	12	17	12	17
cragglvy	14	48	14	48	14	48	14	48	14	48
dixmaana	13	21	13	21	13	21	13	21	13	21
dixmaanb	13	21	13	21	13	21	13	21	13	21
dixmaanc	13	21	13	21	13	21	13	21	13	21
dixmaand	14	23	14	23	14	23	14	23	14	23
dixmaane	14	93	14	93	14	93	14	93	14	93
dixmaanf	15	42	15	42	15	42	15	42	15	42
dixmaang	15	36	15	36	15	36	15	36	15	36
dixmaanh	15	31	15	31	15	31	15	31	15	31
dixmaanj	16	53	16	53	16	53	16	53	16	53
dixmaank	16	43	16	43	16	43	16	43	16	43
dixmaanl	16	38	16	38	16	38	16	38	16	38
dqdrtic	14	20	14	20	14	20	14	20	14	20
dqrtic	28	36	28	39	28	39	28	39	28	39
edensch	15	37	15	37	15	37	15	37	15	37
eg2	4	5	4	5	4	5	4	5	4	5
engval1	14	27	14	27	14	27	14	27	14	27
extrosnb	29	73	29	75	29	75	29	75	29	75

The results highlight the trade-off between the accuracy of the subproblem solutions and the computational effort. Table 5 compares Steihaug's method and phased-SSM for various values of ϵ_s. Generally speaking, as $\epsilon_s \to 1$, the required number of function evaluations for the test set decreases and the number of matrix-vector products increases. Depending on the application and cost of a matrix-vector product relative to the cost of a function evaluation, a less stringent stopping criteria (e.g., $\epsilon_s \ll 1$ may result in a more efficient algorithm.

The results of Tables 1–2 and 3–4 are summarized in Table 5. In general, the phased-SSM method required between 24% and 35% fewer function evaluations than Steihaug's method. By comparison, Gould et al. [17] report that GLTR solved 16 of 17 problems and, for those solved by both GLTR and Steihaug's method, GLTR obtained a 12.5% fewer function evaluations than Steihaug's method.

Table 6 summarizes the results of using different trust-region algorithms. The column with heading "Steihaug-Basic" gives the results obtained using Steihaug's method in conjunction with a "standard" trust-region algorithm (see, e.g., Conn,

Table 4: Inexact phased-SSM on CUTEr problems f–z.

ϵ_s	ϵ_M		0.05		0.1		0.5		1.0	
Problem	fe	prods	fe	prods	fe	prods	fe	prods	fe	prods
fminsrf2	133	1293	102	1912	96	1804	55	1604	60	2080
fminsurf	104	506	89	877	54	713	74	1267	60	1270
freuroth	16	29	16	29	16	29	16	29	16	29
genrose	986	5798	849	13799	819	14061	805	17307	802	19004
liarwhd	19	41	19	41	19	41	19	41	19	41
ncb20	85	652	95	1118	82	1220	94	2020	104	2836
ncb20b	11	64	9	79	9	79	9	79	9	79
noncvxu2	41	60	47	157	47	157	47	262	50	371
noncvxun	35	44	50	172	49	174	48	314	48	298
nondia	4	5	4	5	4	5	4	5	4	5
nondquar	25	121	18	151	18	151	18	151	18	151
penalty1	28	33	28	33	28	33	28	33	28	33
penalty2	2	1	2	1	2	1	2	1	2	1
powellsg	16	55	16	55	16	55	16	55	16	55
power	15	46	15	46	15	46	15	46	15	46
quartc	28	36	28	39	28	39	28	39	28	39
schmvett	8	36	8	36	8	36	8	36	8	36
sparsqur	14	47	14	47	14	47	14	47	14	47
spmsrtls	17	128	24	258	24	258	28	503	29	566
srosenbr	9	17	9	17	9	17	9	17	9	17
testquad	15	183	13	122	13	122	13	122	13	122
tointgss	15	22	15	22	15	22	15	22	15	22
vardim	13	23	13	23	13	23	13	23	13	23
vareigvl	14	37	14	37	14	37	14	37	14	37
woods	13	27	13	27	13	27	13	27	13	27

Gould and Toint [4]). The other columns give results obtained using the recommended "biased" line-search (Algorithm 5.1) for several values of ϵ_s. The improvement in function evaluations is calculated based on the improvement compared to those of "Steihaug-Basic".

We summarize results from Tables 1–2 and 3–4 in Figs 1 and 2 respectively using performance profiles (in $\log_2$ scale) proposed by Dolan and Moré [8]. Fig. 1 plots the function $\pi_s : [0, r_M] \to \mathbb{R}^+$ defined by

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

where $\mathcal{P}$ denotes the set of test problems, and $r_{p,s}$ denotes the ratio of number of function evaluations needed to solve problem p with method s with the least number of function evaluations needed to solve problem p. Here r_M denotes the maximum value of $\log_2(r_{p,s})$. Fig. 2 gives an equivalent plot in terms of matrix-vector products.

Table 5: Comparison of methods. $\delta_0 = 1$.

	Steihaug	$\epsilon_s = \epsilon_M$	$\epsilon_s = 0.05$	$\epsilon_s = 0.1$	$\epsilon_s = 0.5$	$\epsilon_s = 1$
Problems solved	49	49	49	49	49	49
Function evals (fe)	2817	2144	1931	1838	1832	1828
Matrix mults (prods)	10528	10694	20847	20819	26593	29940
Improvement in fe	—	24%	31%	35%	35%	35%

Table 6: Comparison of methods and line searches. $\delta_0 = 1$.

	Steihaug-Basic	Steihaug-Biased	$\epsilon_s = \epsilon_M$	$\epsilon_s = 1$
Problems solved	49	49	49	49
Function evals (fe)	3180	2817	1931	1828
Matrix mults (prods)	31060	10528	20847	29940
Improvement in fe	—	11%	39%	43%

In order for phased-SSM to start the jth problem it is necessary to form the product Hz_0 for the current H and the best leftmost estimate ("z_0") from the previous subproblem. This implies that every subproblem—even those whose solution lies in the interior of the trust region—costs at least one more matrix-vector product than that of Steihaug's method. Nevertheless, the results of Table 5 indicates that Steihaug's method and phased-SSM with $\epsilon_s \approx \epsilon_M$ require comparable numbers of matrix-vector products—a fact that is obscured by the performance profile.

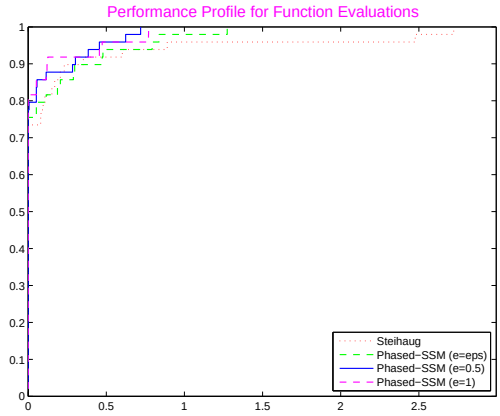

Figure 1: $\log_2$-scale performance profile comparing function evaluations on 49 CUTEr test problems.

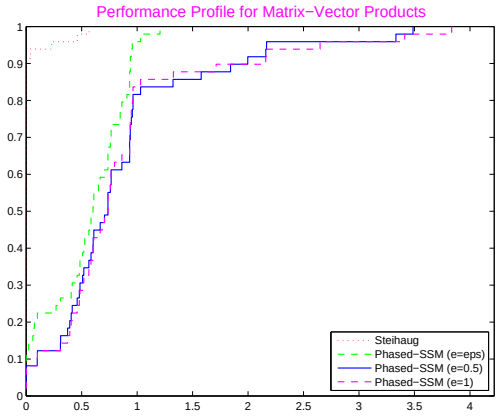

Figure 2: $\log_2$-scale performance profile comparing matrix-vector products on 49 CUTEr test problems.

6. 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 than Steihaug's method appears to decrease the overall number of function evaluations. Based on the results, it appears that phased-SSM outperforms Steihaug's method in terms of function evaluations, and would be a better solver when the cost of a function evaluation is expensive relative to the cost of a matrix-vector product.

The accuracy to choose for the approximate solution in the constrained case is very problem dependent (see, e.g., Gould et al. [17]). The results of Section 5 indicate that it is possible to solve the trust-region problem to less accuracy in the constrained case without detracting from the efficiency of the method. Future research will consider other termination criteria for the trust-region problem.

References

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

[2] M. A. Branch, T. F. Coleman, and Y. Li, *A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems*, SIAM J. Sci. Comput., 21 (1999), pp. 1–23.

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

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

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

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

[7] J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

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

[9] J. B. Erway, *Iterative Methods for Large-Scale Unconstrained Optimization*, PhD thesis, Department of Mathematics, University of California, San Diego, August 2006.

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

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

[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] E. M. Gertz, *A quasi-Newton trust-region method*, Math. Program., 100 (2004), pp. 447–470.

[14] P. E. Gill and M. W. Leonard, *Reduced-Hessian quasi-Newton methods for unconstrained optimization*, SIAM J. Optim., 12 (2001), pp. 209–237.

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

[16] N. I. M. Gould, M. E. Hribar, and J. Nocedal, *On the solution of equality constrained quadratic programming problems arising in optimization*, SIAM J. Sci. Comput., 23 (2001), pp. 1376–1395 (electronic).

[17] N. I. M. Gould, S. Lucidi, M. Roma, and Ph. L. Toint, *Solving the trust-region subproblem using the Lanczos method*, SIAM J. Optim., 9 (1999), pp. 504–525.

[18] N. I. M. Gould, D. Orban, and Ph. L. Toint, *CUTEr and SifDec: A constrained and unconstrained testing environment, revisited*, ACM Trans. Math. Softw., 29 (2003), pp. 373–394.

[19] J. D. Griffin, *Interior-point methods for large-scale nonconvex optimization*, PhD thesis, Department of Mathematics, University of California, San Diego, March 2005.

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

[21] W. W. Hager and S. Park, *Global convergence of SSM for minimizing a quadratic over a sphere*, Math. Comp., 74 (2004), pp. 1413–1423.

[22] M. D. Hebden, *An algorithm for minimization using exact second derivatives*, Tech. Report T.P. 515, Atomic Energy Research Establishment, Harwell, England, 1973.

[23] M. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 409–436.

[24] C. Keller, N. I. M. Gould, and A. J. Wathen, *Constraint preconditioning for indefinite linear systems*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1300–1317 (electronic).

[25] C.-J. Lin and J. J. Moré, *Newton's method for large bound-constrained optimization problems*, SIAM J. Optim., 9 (1999), pp. 1100–1127.

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

[27] J. J. Moré and D. C. Sorensen, *Newton's method*, in Studies in Mathematics, Volume 24. Studies in Numerical Analysis, Math. Assoc. America, Washington, DC, 1984, pp. 29–82.

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

[29] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975), pp. 617–629.

[30] Ph. 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.

[31] M. J. D. Powell, *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), Academic Press, New York, 1974, pp. 1–27.

[32] 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.

[33] 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 (2000/01), pp. 611–646 (electronic).

[34] 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 (electronic).

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

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

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

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

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

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