

On Modified Newton Methods for Unconstrained Optimization Based on Lanczos Tridiagonalization*

Aban ANSARI-ÖNNESTAM[†] Anders FORSGREN[‡] Philip E. GILL[§]

October 22, 2026

Abstract

For a strongly convex unconstrained minimization problem, Newton’s method with a line search offers rapid convergence to a minimizer by incorporating second-order information. However, for a general unconstrained problem the Hessian need not be positive definite, which can lead to the inability to find a search direction or to search directions that fail to produce descent. Modified Newton methods are designed to handle this situation. This paper considers modifications to a Newton-conjugate-gradient approach based on Lanczos tridiagonalization. Existing methods are extended using dynamic modification schemes that provide a descent direction while keeping the modification bounded. A Lanczos-based scheme and a conjugate-gradient (CG)-based scheme are formulated and analyzed. These methods are mathematically equivalent in exact arithmetic, but their practical behavior differs in finite precision. Finally, it is shown how the convergence and computational cost are affected by varying the amount of additional curvature introduced and by allowing the early termination of the inner iterations.

Key words. Unconstrained optimization, conjugate-gradient methods, line-search methods, Lanczos tridiagonalization, modified Newton methods.

1. Introduction

This paper concerns the formulation and analysis of modified Newton methods for the unconstrained optimization problem

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x),$$

where f is a twice-continuously differentiable scalar-valued function. The focus is on line-search methods, which generate a sequence $\{x_k\}$ with $x_{k+1} = x_k + \alpha_k p_k$, where α_k is a positive scalar and p_k is a direction of decrease for f at x_k .

If the Hessian matrix $\nabla^2 f(x_k)$ is sufficiently positive definite (i.e., the least eigenvalue is sufficiently positive), the search direction p_k may be computed by solving the Newton equations $\nabla^2 f(x_k) p_k = -\nabla f(x_k)$, which constitute the first-order optimality conditions for the minimizer of the quadratic model

$$\varphi_k(p) = f(x_k) + p^T \nabla f(x_k) + \frac{1}{2} p^T \nabla^2 f(x_k) p. \quad (1.1)$$

*The work of the first two authors was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

[†]Division of Applied Mathematics, Department of Mathematics, Linköping University, SE-581 83 Linköping, Sweden (aban.ansari-onnestam@liu.se).

[‡]Division of Numerical Analysis, Optimization and Systems Theory, Department of Mathematics, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden (andersf@kth.se).

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

The Newton equations may be solved directly using conventional matrix factorization methods. For example, an efficient method is based on the LDL^T factorization $\nabla^2 f(x_k) = L_k D_k L_k^T$ in which L_k is unit lower triangular and D_k is diagonal. If the number of variables is large and $\nabla^2 f(x_k)$ is sparse, then there are very efficient sparse-matrix methods for computing the factors of a symmetrically permuted version of $\nabla^2 f(x_k)$. The efficiency of these methods follows from the fact that there exist permutations such that L_k and D_k are also sparse. However, factorization methods may be impractical if n is large and $\nabla^2 f(x_k)$ is not sparse. Such matrices arise in methods for large-scale constrained minimization that involve the solution of equations with matrices of the form $Z^T \nabla^2 f(x) Z$, where $\nabla^2 f(x_k)$ and Z may be sparse but the product is dense. These cases are examples of problems for which iterative methods for the solution of the relevant equations are the only option. In this context the matrix associated with the linear equations is used only in the computation of matrix-vector products.

For a nonconvex objective function, the Hessian may not be positive definite. As a result, the conventional Newton direction may fail to be a descent direction or, if the Hessian is singular, the Newton equations may not have a solution. Modified Newton methods handle this problem by changing the Newton equations to obtain a more suitable search direction. The aim is to improve the behavior of the method when the Hessian is not positive definite, while retaining the fast local convergence of Newton's method when the Hessian is sufficiently positive definite.

A common approach is to replace the Hessian by a related positive-definite matrix. Early Newton-type methods based on modified Cholesky factorizations were developed by Gill and Murray [8]. Kaniel and Dax [12] proposed a related approach based on a symmetric decomposition of the Hessian, which is used to construct a descent direction when the Hessian is not positive definite. This idea was later developed further through alternative ways of modifying the factorization. Examples include the methods of Schnabel and Eskow [19, 20] and Cheng and Higham [3]. Fang and O'Leary [6] give a broad overview and comparison of modified Cholesky methods.

Other methods retain and use information about negative curvature instead of removing it completely. For example, Forsgren, Gill, and Murray [7] use a partial Cholesky factorization to compute both a descent direction and a direction of negative curvature. Moré and Sorensen [14] use a symmetric indefinite factorization to obtain directions of negative curvature. For large-scale problems, similar ideas appear in Newton-CG methods, where conjugate gradients are used to find an approximate solution of a damped or regularized Newton system while also detecting nonpositive curvature. The method may then use either an approximate Newton direction or a direction of negative curvature. Such approaches can also provide guarantees for approaching points that satisfy approximate second-order optimality conditions (see Royer, O'Neill and Wright [18] and Yao et al. [24]).

Regularized and hybrid methods provide other ways of handling an indefinite Hessian. These include adaptively shifting the Hessian and combining regularized Newton directions with directions of negative curvature (see, e.g., Shen, Chen and Liang [21], Yang [23], Zhu and Xiao [26], and Zhou et al. [25]). Although these methods construct their search directions in different ways, they all aim to produce reliable steps when the Hessian is indefinite and to recover Newton-like behavior near a sufficiently regular local minimizer.

This work builds on the modified Newton method proposed by Nash [15], which applies a two-stage modification to the Lanczos tridiagonalization of the Hessian. The modification is constructed so that the resulting matrix retains sufficient positive curvature while the magnitude of the applied modifications remains bounded. As the method operates on the

tridiagonal matrix generated by the Lanczos process, it requires only quantities that become available during the Lanczos iterations.

In Nash's original method, a fixed positive parameter δ is used both to determine whether a modification is required and to specify a lower bound on the diagonal entries of the LDL^T factorization of the modified tridiagonal matrix. Choosing a small value of δ avoids unnecessary modifications, but may produce a modified matrix with very small positive curvature in several directions. This observation motivates the separation of the threshold used to determine whether the curvature is sufficiently large from the target curvature imposed when a modification is required.

In this paper, Nash's two-stage modification of the tridiagonal matrix is extended by introducing iteration-dependent target curvatures that are distinct from the fixed curvature threshold. Bounds on the resulting modifications are established when the sequence of target curvatures is uniformly bounded. Two choices of the iteration-dependent target curvature are proposed, both of which use information from the current tridiagonal matrix. In addition, a gradient-based CG formulation of the modification is derived and shown to be equivalent to the Lanczos formulation in exact arithmetic. Finally, numerical results are presented to facilitate a comparison of the proposed methods and demonstrate the effects of using several criteria for the early termination of the inner iterations.

The remainder of the paper is organized as follows. Section 2 introduces the notation used throughout the paper. Section 3 gives a brief review of the gradient-based method of conjugate gradients (CG) and its Lanczos-based formulation, and describes the modification proposed by Nash. Section 4 introduces the iteration-dependent target curvatures and establishes a bound on the resulting modifications. Section 5 derives the equivalent gradient-based CG formulation. Section 6 presents the numerical results, and Section 7 provides some concluding remarks.

2. Notation

In order to simplify the notation, the discussion will focus on the generic quadratic function

$$\varphi(p) = g^T p + \frac{1}{2} p^T H p,$$

where g and H denote the gradient $\nabla f(x_k)$ and Hessian $\nabla^2 f(x_k)$, respectively (cf. (1.1)). The methods are iterative, and given $p_0 \in \mathbb{R}^n$ generate a finite sequence of inner iterates

$$p_{j+1} = p_j + \omega_j q_j, \text{ for } j = 0, 1, \dots, \text{ with } q_j \in \mathbb{R}^n \text{ and } \omega_j \in \mathbb{R}.$$

The vector $r(p)$ is used to denote $g + Hp$, the gradient of φ , with $r_j = r(p_j)$. Under the assumption that H is positive definite, if $r(p_j) = 0$ then p_j is the minimizer p^* and the iterative method terminates. Unless otherwise stated, the scalar ω denotes the step length in the inner (Lanczos) iterations. For clarity, the index used for p is j for the inner iterations and k for the outer iterations.

Unless explicitly indicated otherwise, $\|\cdot\|$ denotes the vector two-norm or its induced matrix norm. The least eigenvalue of a symmetric matrix H will be denoted by $\lambda_{\min}(H)$. The vectors e and e_j denote, respectively, the column vector of ones and the j th column of the identity matrix I ; the dimensions of e , e_j and I are defined by the context.

3. Preliminaries

This section starts with a review of the class of conjugate-gradient iterative methods for the solution of the positive-definite Newton equations $\nabla^2 f(x_k) p_k = -\nabla f(x_k)$. This is

followed by a review of iterative methods based on the Lanczos process (Lanczos [13]) and it is shown how these methods may be used to formulate iterative methods in which an indefinite $\nabla^2 f(x_k)$ is modified implicitly to be positive definite. In all the methods to be considered, the matrix associated with the linear equations is used only in the computation of matrix-vector products.

3.1. The method of conjugate gradients.

All the methods considered are either explicitly or implicitly related to the method of conjugate gradients proposed by Hestenes and Stiefel [11]. The method minimizes the quadratic function $\varphi(p)$ by generating a sequence of search directions $\{q_j\}$ that are conjugate with respect to H , i.e., $q_i^T H q_j = 0$ for $i \neq j$. Starting from an initial point p_0 , the iterates are generated according to

$$p_j = p_{j-1} + \omega_{j-1} q_{j-1},$$

where ω_{j-1} is the step length obtained from an exact line search along q_{j-1} . The search directions are constructed iteratively from the gradient vectors $r_0, r_1, \dots, r_{j-1}$ so that the resulting iterates minimize $\varphi(p)$ over a sequence of expanding subspaces. In exact arithmetic, the method terminates at the minimizer after at most m iterations, where $m \leq n$. The method is summarized in Algorithm 1. To distinguish this formulation from the Lanczos-based method of conjugate gradients, the procedure in Algorithm 1 will be referred to as **grad.CG**.

Algorithm 1. The method of conjugate gradients.

```

procedure  $[p] = \text{grad.CG}(H, g, p_0, \tau_{tol})$ 
  Restrictions:  $\tau_{tol} > 0$ ,  $H$  positive definite;
   $r_0 = g + H p_0$ ;  $j = 0$ ;
  while  $\|r_j\| > \tau_{tol}$  do
    if  $j = 0$  then
       $q_j = -r_j$ ;
    else
       $q_j = -r_j + (r_j^T H q_{j-1} / q_{j-1}^T H q_{j-1}) q_{j-1}$ ;
    end if
     $\omega_j = -r_j^T q_j / q_j^T H q_j$ ;  $p_{j+1} = p_j + \omega_j q_j$ ;  $r_{j+1} = r_j + \omega_j H q_j$ ;
     $j \leftarrow j + 1$ ;
  end while
   $p = p_j$ ;
end procedure

```

The subspace spanned by the search directions $q_0, q_1, \dots, q_{j-1}$ is precisely the Krylov subspace generated by H and the initial gradient r_0 , namely

$$\text{span} \{ q_0, q_1, \dots, q_{j-1} \} = \text{span} \{ r_0, H r_0, \dots, H^{j-1} r_0 \}.$$

It follows that the quadratic function is minimized over a sequence of expanding Krylov subspaces.

3.2. A method of conjugate gradients based on Lanczos tridiagonalization.

An alternative implementation of the method of conjugate gradients may be derived using Lanczos tridiagonalization (see Paige and Saunders [17]). Given $v_1 = -g/\beta_1$ with $\beta_1 = \|g\|$,

the Lanczos vectors are computed from the sequence

$$\beta_2 v_2 = H v_1 - \gamma_1 v_1, \quad \beta_{j+1} v_{j+1} = H v_j - \gamma_j v_j - \beta_j v_{j-1}, \quad j = 2, 3, \dots,$$

where β_{j+1} is chosen to give $\|v_{j+1}\| = 1$, and $\gamma_j = v_j^T H v_j$. The Lanczos process generates a sequence of tridiagonal matrices T_j , $j = 1, 2, \dots, m$, such that

$$H V_j = V_j T_j + \beta_{j+1} v_{j+1} e_j^T,$$

with

$$V_j = (v_1 \ v_2 \ \dots \ v_j) \quad \text{and} \quad T_j = \begin{pmatrix} \gamma_1 & \beta_2 & & & \\ \beta_2 & \gamma_2 & \beta_3 & & \\ & \beta_3 & \gamma_3 & \beta_4 & \\ & & \ddots & \ddots & \ddots \\ & & & \ddots & \ddots & \beta_j \\ & & & & \beta_j & \gamma_j \end{pmatrix}.$$

In exact arithmetic, the Lanczos iteration will terminate with $\beta_{m+1} = 0$ for some integer $m \leq n$, which implies that $T_m = V_m^T H V_m$ with $V_m^T V_m = I_m$.

A set of conjugate vectors may be computed using a suitable factorization of T_j . If H is positive definite, then so is T_j , which has a unique LDL^T factorization $T_j = L_j D_j L_j^T$, with D_j diagonal and L_j unit lower triangular. Given T_j , the matrix T_{j+1} has the form

$$T_{j+1} = \begin{pmatrix} L_j & 0 \\ l_{j+1} e_j^T & 1 \end{pmatrix} \begin{pmatrix} D_j & 0 \\ 0 & d_{j+1} \end{pmatrix} \begin{pmatrix} L_j^T & l_{j+1} e_j \\ 0 & 1 \end{pmatrix},$$

where $l_{j+1} = \beta_{j+1}/d_j$ and $d_{j+1} = \gamma_{j+1} - \beta_{j+1}^2/d_j$. If $j \geq 1$ then l_{j+1} and d_{j+1} are associated with the LDL^T factors of a 2×2 matrix, i.e.,

$$\begin{pmatrix} d_j & \beta_{j+1} \\ \beta_{j+1} & \gamma_{j+1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ l_{j+1} & 1 \end{pmatrix} \begin{pmatrix} d_j & 0 \\ 0 & d_{j+1} \end{pmatrix} \begin{pmatrix} 1 & l_{j+1} \\ 0 & 1 \end{pmatrix}.$$

It remains to compute the conjugate vectors from the Lanczos vectors and LDL^T factors. With the definition $Q_j = V_j L_j^{-T}$, it holds that

$$Q_j^T H Q_j = L_j^{-1} V_j^T H V_j L_j^{-T} = L_j^{-1} T_j L_j^{-T} = D_j,$$

and the columns of $Q_j = V_j L_j^{-T}$ are mutually conjugate with respect to H . An important property of Q_j is that its columns may be computed using a simple two-term recurrence involving the Lanczos vectors. If the columns of Q_j are denoted by $q_0, q_1, \dots, q_{j-1}$, then

$$(v_1 \ v_2 \ \dots \ v_j) = V_j = Q_j L_j^T = (q_0 \ l_2 q_0 + q_1 \ \dots \ l_j q_{j-2} + q_{j-1}),$$

which gives the two-term recurrence

$$q_0 = v_1, \quad q_i = v_{i+1} - l_{i+1} q_{i-1}, \quad i = 1, 2, \dots, j-1.$$

When an exact line search is used, the step length may be calculated iteratively as

$$\omega_{j-1} = \begin{cases} \beta_1/d_1 & \text{if } j = 1; \\ -(\beta_j/d_j)\omega_{j-2} & \text{otherwise.} \end{cases} \quad (3.1)$$

With this method, the iterates $p_j = p_{j-1} + \omega_{j-1}q_{j-1}$ minimize φ over the same sequence of expanding subspaces as the method of conjugate gradients, i.e., the iterates are identical. The scalars β_j and d_j are positive, and the expression (3.1) implies that the members of the sequence $\{\omega_i\}$ alternate in sign and are *not* the same as the step lengths of the conjugate-gradient method. A simple change to the Lanczos process can be used to provide positive step lengths at every iteration. In the computation of β_{j+1} as $\|Hv_j - \gamma_j v_j - \beta_j v_{j-1}\|$ there is an implicit choice of sign for the calculation of the norm. If β_{j+1} is computed as $-\|Hv_j - \gamma_j v_j - \beta_j v_{j-1}\|$ then $\omega_{j-1} > 0$ for all $j > 0$. A Lanczos-based method of conjugate gradients is summarized in Algorithm 2.

Algorithm 2. Lanczos-based method of conjugate gradients.

```

procedure  $[p] = \text{lanczos\_CG}(H, g, \tau_{tol}, j_{\max})$ 
  Restrictions:  $g \neq 0$ ,  $H$  positive definite  $\tau_{tol} > 0$ ,  $j_{\max} > 0$ ;
   $p_0 = 0$ ;  $v_0 = 0$ ;
   $\beta_1 = \|g\|$ ;  $v_1 = -g/\beta_1$ ;  $\gamma_1 = v_1^T H v_1$ ;  $q_0 = v_1$ ;
   $d_1 = \gamma_1$ ;
   $j = 1$ ;
  while  $j < j_{\max}$  do
     $s_{j+1} = H v_j - \gamma_j v_j - \beta_j v_{j-1}$ ;  $\beta_{j+1} = -\|s_{j+1}\|$ ;
    if  $|\beta_{j+1}| > \tau_{tol}$  then
       $v_{j+1} = s_{j+1}/\beta_{j+1}$ ;  $\gamma_{j+1} = v_{j+1}^T H v_{j+1}$ ;
       $d_{j+1} = \gamma_{j+1} - \beta_{j+1}^2/d_j$ ;  $l_{j+1} = \beta_{j+1}/d_j$ ;
       $q_j = v_{j+1} - l_{j+1}q_{j-1}$ ;
    end if
    if  $j = 1$  then  $\omega_0 = \beta_1/d_1$  else  $\omega_{j-1} = -\omega_{j-2}\beta_j/d_j$ ;
     $p_j = p_{j-1} + \omega_{j-1}q_{j-1}$ ; [The gradient norm at  $p_j$  is  $|\beta_{j+1}|\omega_{j-1}$ ]
    if  $|\beta_{j+1}| \leq \tau_{tol}$  break;
     $j \leftarrow j + 1$ ;
  end while
   $p = p_j$ ;
end procedure

```

3.3. Nash's modified Lanczos method

An alternative to the conventional modified LDL^T factorization was proposed by Nash [15]. The method computes a diagonal modification in a sequential two-stage procedure, using only the information from the rows and columns of T that have been computed thus far. Each diagonal entry is modified twice – the first modification ensures that the resulting matrix is sufficiently positive definite, while the second controls the magnitude of subsequent modifications by bounding the correction required at the next iteration.

Initially, the value of γ_1 , the leading diagonal of T , is computed. If H is not positive definite, this value may be negative or zero. To allow for this, the least nonnegative scalar $\bar{\sigma}_1$ is defined such that $\bar{\gamma}_1 = \gamma_1 + \bar{\sigma}_1 \geq \delta$, i.e.,

$$\bar{\sigma}_1 = \max \{ \gamma_1, \delta \} - \gamma_1,$$

where $\delta > 0$ is chosen to be the smallest allowable positive value of the modified diagonal entry. (The value of δ determines the minimum distance of the modified matrix to singularity.)

Once v_2 has been computed the new tridiagonal matrix may be modified if necessary as

$$\begin{pmatrix} \bar{d}_1 & \beta_2 \\ \beta_2 & \gamma_2 \end{pmatrix} + \begin{pmatrix} \sigma_1 & 0 \\ 0 & \bar{\sigma}_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ l_2 & 1 \end{pmatrix} \begin{pmatrix} d_1 & 0 \\ 0 & \bar{d}_2 \end{pmatrix} \begin{pmatrix} 1 & l_2 \\ 0 & 1 \end{pmatrix},$$

where σ_1 and $\bar{\sigma}_2$ are nonnegative scalars computed so that the quantity $\sigma_1 + \bar{\sigma}_2$ is minimized. As $\bar{d}_1 = \gamma_1 + \bar{\sigma}_1$ (with $\bar{d}_1 \geq \delta$), this implies that γ_1 may undergo two modifications during the factorization, which is the reason for the use of bars to denote *intermediate* values that may be modified again at the next stage of the factorization. The j th stage ($j \geq 2$) involves the computation of the factorization

$$\begin{pmatrix} \bar{d}_j & \beta_{j+1} \\ \beta_{j+1} & \gamma_{j+1} \end{pmatrix} + \begin{pmatrix} \sigma_j & 0 \\ 0 & \bar{\sigma}_{j+1} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ l_{j+1} & 1 \end{pmatrix} \begin{pmatrix} d_j & 0 \\ 0 & \bar{d}_{j+1} \end{pmatrix} \begin{pmatrix} 1 & l_{j+1} \\ 0 & 1 \end{pmatrix},$$

where σ_j and $\bar{\sigma}_{j+1}$ are nonnegative scalars defined so that the quantity $\sigma_j + \bar{\sigma}_{j+1}$ is minimized. If the quantity $\bar{d}_{j+1}$ is denoted by $\phi_{j+1}(\sigma_j, \bar{\sigma}_{j+1})$ to indicate the dependence on σ_j and $\bar{\sigma}_{j+1}$, then

$$\phi_{j+1}(\sigma_j, \bar{\sigma}_{j+1}) \triangleq \bar{d}_{j+1} = \gamma_{j+1} + \bar{\sigma}_{j+1} - \beta_{j+1}^2 / (\bar{d}_j + \sigma_j).$$

Note that $\phi_{j+1}(0, 0) = \gamma_{j+1} - \beta_{j+1}^2 / \bar{d}_j$ is the Schur complement computed by a conventional Cholesky factorization. If $\phi_{j+1}(0, 0) \geq \delta$, then $\phi_{j+1}(0, 0)$ is sufficiently positive, $\sigma_j = \bar{\sigma}_{j+1} = 0$ and the factorization is not modified. Otherwise, the factors are modified by finding σ_j and $\bar{\sigma}_{j+1}$ that solve the subproblem

$$\begin{aligned} & \underset{\sigma_j, \bar{\sigma}_{j+1}}{\text{minimize}} && \sigma_j + \bar{\sigma}_{j+1} \\ & \text{subject to} && \sigma_j \geq 0, \quad \bar{\sigma}_{j+1} \geq 0, \quad \phi_{j+1}(\sigma_j, \bar{\sigma}_{j+1}) \geq \delta. \end{aligned} \tag{3.2}$$

The search directions and iterates computed with the modified tridiagonal matrix will then coincide with the iterates generated by Algorithm 2 using the modified Hessian

$$\tilde{H}_{m-1} = H + \sum_{i=1}^{m-1} (\bar{\sigma}_i + \sigma_i) v_i v_i^T + \bar{\sigma}_m v_m v_m^T,$$

where m denotes the index such that $\beta_{m+1} = 0$.

4. Extensions of Nash's Method

Nash's two-stage modification ensures that the pivots of the modified LDL^T factorization are bounded below by a prescribed constant $\delta > 0$, while also bounding the magnitude of the diagonal modification. However, if δ is fixed at a small value, the resulting matrix may have very small positive curvature in the direction of several Lanczos vectors. The solution of the modified Newton equations may then have large components in these directions, i.e., directions in which the original Hessian has small or negative curvature. Moreover, the resulting solution may have an excessively large norm. The line search in the outer iterations may then require many function evaluations before an acceptable step size is obtained.

Choosing a uniformly larger value of δ could reduce this effect, but may impose unnecessary modifications. This justifies the separation of the threshold used to detect insufficient curvature from the target curvature imposed after modification. A fixed small value of δ is

retained to determine whether the current Schur complement is sufficiently positive. If this test fails, the target curvature is replaced by an iteration-dependent value $\delta_{j+1} \geq \delta$. Apart from the choice of δ_{j+1} , the two-stage modification remains unchanged. It is shown that, provided the values δ_{j+1} are uniformly bounded above, the resulting diagonal modifications retain a corresponding uniform bound.

At the j th stage of Nash's method, a modified LDL^T factorization of the 2×2 matrix

$$B_j = \begin{pmatrix} \bar{d}_j & \beta_{j+1} \\ \beta_{j+1} & \gamma_{j+1} \end{pmatrix} \quad (4.1)$$

is computed, and the unmodified Schur complement $\phi_{j+1}(0, 0) = \gamma_{j+1} - \beta_{j+1}^2 / \bar{d}_j$ is compared to the fixed detection threshold δ . If $\phi_{j+1}(0, 0) \geq \delta$, then the current curvature is considered sufficient and no modification is made. In this case, no target curvature is required. Conversely, when $\phi_{j+1}(0, 0) < \delta$, an iteration-dependent target curvature $\delta_{j+1} \geq \delta$ is selected and used as the lower bound for the Schur complement in the modification subproblem.

If a modification is necessary, the subproblem analogous to (3.2) is solved, where δ is replaced by δ_{j+1} , i.e.,

$$\begin{aligned} & \underset{\sigma_j, \bar{\sigma}_{j+1}}{\text{minimize}} && \sigma_j + \bar{\sigma}_{j+1} \\ & \text{subject to} && \sigma_j \geq 0, \quad \bar{\sigma}_{j+1} \geq 0, \quad \phi_{j+1}(\sigma_j, \bar{\sigma}_{j+1}) \geq \delta_{j+1} \end{aligned} \quad (4.2)$$

where $\phi_{j+1}(\sigma_j, \bar{\sigma}_{j+1}) = \gamma_{j+1} + \bar{\sigma}_{j+1} - \beta_{j+1}^2 / (\bar{d}_j + \sigma_j)$. Therefore the fixed detection threshold δ is used only to determine whether the current curvature is sufficient, whereas the target curvature δ_{j+1} specifies the curvature added after the need for a modification has already been established.

Nash's approach begins by testing the minimizer with respect to the nonlinear constraint for feasibility. If this point is infeasible, the solution is selected from the two remaining candidates by choosing the one that minimizes the objective, noting that at least one candidate is always feasible. Here a slightly different approach is used, where first-order information from the nonlinear constraint is utilized and a greedy selection strategy using its partial derivatives is employed. An advantage of this derivation is that the inequalities needed to establish the bounds on the modifications are immediately apparent. The method is summarized below (for a thorough discussion of greedy selection methods see Bretthauer and Shetty [2]). As the objective function is linear in both variables with equal coefficients, the solution is dependent on the partial derivatives

$$\nabla_{\sigma_j} \phi_{j+1}(\sigma_j, \bar{\sigma}_{j+1}) = \frac{\beta_{j+1}^2}{(\bar{d}_j + \sigma_j)^2}, \quad \text{and} \quad \nabla_{\bar{\sigma}_{j+1}} \phi_{j+1}(\sigma_j, \bar{\sigma}_{j+1}) = 1.$$

The partial derivatives are equal for the values

$$\sigma_j = \beta_{j+1} - \bar{d}_j, \quad \bar{\sigma}_{j+1} = \delta_{j+1} - \gamma_{j+1} + \beta_{j+1},$$

which is the required solution of (4.2) if the nonnegativity constraints are satisfied. If $\beta_{j+1} - \bar{d}_j < 0$, then for any feasible $\sigma_j \geq 0$ it holds that $\nabla_{\sigma_j} \phi_{j+1}(\sigma_j, \bar{\sigma}_{j+1}) < 1$ and the solution is given by

$$\sigma_j = 0, \quad \bar{\sigma}_{j+1} = \delta_{j+1} - \gamma_{j+1} + \beta_{j+1}^2 / \bar{d}_j.$$

Otherwise the nonnegativity constraint is violated by $\delta_{j+1} - \gamma_{j+1} + \beta_{j+1} < 0$, in which case for any feasible $\bar{\sigma}_{j+1}$, it holds that $\nabla_{\bar{\sigma}_{j+1}} \phi_{j+1}(\sigma_j, \bar{\sigma}_{j+1}) > 1$ and the solution is given by

$$\sigma_j = \beta_{j+1}^2 / (\gamma_{j+1} - \delta_{j+1}) - \bar{d}_j, \quad \bar{\sigma}_{j+1} = 0.$$

The resulting procedure for finding modifications based on the bound δ_{j+1} is described in Algorithm 3.

Algorithm 3. The tridiagonal modification.

```

procedure  $[\bar{d}_{j+1}, \sigma_j^*, \bar{\sigma}_{j+1}^*] = \text{modT}(\bar{d}_j, \gamma_{j+1}, \beta_{j+1})$ 
  Choose  $\delta > 0$ ;
   $\phi = \gamma_{j+1} - \beta_{j+1}^2 / \bar{d}_j$ ;  $[\phi = \phi_{j+1}(0, 0)]$ 
  if  $\phi \geq \delta$  then
     $\sigma_j^* = 0$ ;  $\bar{\sigma}_{j+1}^* = 0$ ;  $\bar{d}_{j+1} = \phi$ ;
  else
    Choose the target curvature  $\delta_{j+1} \geq \delta$ ; [See Algorithm 5]
     $\bar{d}_{j+1} = \delta_{j+1}$ ;
    Set  $\sigma = |\beta_{j+1}| - \bar{d}_j$ ;  $\bar{\sigma} = \delta_{j+1} + |\beta_{j+1}| - \gamma_{j+1}$ ;
    if  $\sigma \geq 0$  and  $\bar{\sigma} \geq 0$  then
       $\sigma_j^* = \sigma$ ;  $\bar{\sigma}_{j+1}^* = \bar{\sigma}$ ;
    else if  $\bar{\sigma} < 0$  then
       $\sigma_j^* = \bar{d}_j(\delta_{j+1} - \phi) / (\gamma_{j+1} - \delta_{j+1})$ ;  $\bar{\sigma}_{j+1}^* = 0$ ;
    else
       $\sigma_j^* = 0$ ;  $\bar{\sigma}_{j+1}^* = \delta_{j+1} - \phi$ ;
    end if
  end if
end procedure

```

A modified Lanczos method based on the two-stage modification with distinct bounds on the curvature detection threshold δ and the iteration-dependent target curvature δ_{j+1} is given in Algorithm 4. When a modification is necessary, the value δ_{j+1} may be chosen freely, provided that it is bounded below by δ . To keep the modifications bounded, a safeguard $\delta_{\max}$ is imposed so that the target curvature cannot grow arbitrarily large. This implies that the target curvature satisfies the bounds $\delta \leq \delta_{j+1} \leq \delta_{\max}$.

The iterates generated by Algorithm 4 depend on the entries of the modified tridiagonal matrix. At iteration j , the two-stage modification changes the diagonal entries $\bar{d}_j$ and $\bar{d}_{j+1}$, producing $d_j = \bar{d}_j + \sigma_j$ and a new value of $\bar{d}_{j+1}$. The search directions $\{q_0, q_1, \dots, q_{j-1}\}$ and the step lengths $\omega_0, \omega_1, \dots, \omega_{j-2}$ have already been determined and are not affected by this modification. The modified value d_j determines the current step length ω_{j-1} , whereas $\bar{d}_{j+1}$ does not occur in the update

$$p_j = p_{j-1} + \omega_{j-1} q_{j-1}.$$

Suppose that the Lanczos process terminates at iteration m , i.e., $\beta_{m+1} < \tau_{tol}$. On completion of iteration j ($j < m$) the modified Hessian is given by

$$\tilde{H}_j = H + \sum_{i=1}^j (\bar{\sigma}_i + \sigma_i) v_i v_i^T + \bar{\sigma}_{j+1} v_{j+1} v_{j+1}^T. \quad (4.3)$$

Then, in exact arithmetic, the iterates $\{p_1, p_2, \dots, p_j\}$ generated by Algorithm 4 coincide with the first j iterates generated by Algorithm 2 applied to $\tilde{H}_j$. Thus, through iteration

Algorithm 4. Lanczos-based modified conjugate-gradient method.

```

procedure  $[p] = \text{lanczos\_mod\_CG}(H, g, \tau_{tol}, j_{\max})$ 
  Restrictions:  $g \neq 0, \tau_{tol} > 0, j_{\max} > 0;$ 
   $p_0 = 0; v_0 = 0;$ 
   $\beta_1 = \|g\|; v_1 = -g/\beta_1; \gamma_1 = v_1^T H v_1; q_0 = v_1;$ 
  Choose  $\delta > 0;$ 
   $\bar{\sigma}_1 = \max\{\gamma_1, \delta\} - \gamma_1; \bar{d}_1 = \gamma_1 + \bar{\sigma}_1;$ 
   $j = 1;$ 
  while  $j < j_{\max}$  do
     $s_{j+1} = H v_j - \gamma_j v_j - \beta_j v_{j-1}; \beta_{j+1} = -\|s_{j+1}\|;$ 
    if  $|\beta_{j+1}| > \tau_{tol}$  then
       $v_{j+1} = s_{j+1}/\beta_{j+1}; \gamma_{j+1} = v_{j+1}^T H v_{j+1};$ 
       $[d_{j+1}, \sigma_j, \bar{\sigma}_{j+1}] = \text{modT}(\bar{d}_j, \gamma_{j+1}, \beta_{j+1});$  [The tridiagonal modification]
       $d_j = \bar{d}_j + \sigma_j; l_{j+1} = \beta_{j+1}/d_j;$ 
       $q_j = v_{j+1} - l_{j+1} q_{j-1};$ 
    else
       $d_j = \bar{d}_j;$ 
    end if
    if  $j = 1$  then  $\omega_0 = \beta_1/d_1$  else  $\omega_{j-1} = -\omega_{j-2}\beta_j/d_j;$ 
     $p_j = p_{j-1} + \omega_{j-1} q_{j-1};$  [The gradient norm at  $p_j$  is  $|\beta_{j+1}|\omega_{j-1}$ ]
    if  $|\beta_{j+1}| \leq \tau_{tol}$  break;
     $j \leftarrow j + 1;$ 
  end while
   $p = p_j;$ 
end procedure

```

j , the modified Lanczos-based method may be interpreted as applying the conventional Lanczos-based method of conjugate gradients to the current modified Hessian $\tilde{H}_j$.

The modifications generated by Nash's original method are bounded (Nash [15, Theorem 2]). The next step is to extend this result to iteration-dependent curvature targets. In the implementation, the target curvatures are safeguarded by the prescribed upper bound $\delta_{\max}$, so that

$$0 < \delta \leq \delta_{j+1} \leq \delta_{\max}$$

at every stage at which a modification is required. The lower bound ensures that the modified pivots remain sufficiently positive, while the upper bound makes it possible to control the modifications uniformly over the iterations. No additional assumption is made on how the values δ_{j+1} are selected. Under these conditions, the modifications generated by Algorithm 3 satisfy the following bound.

Proposition 4.1. *Let $T \in \mathbb{R}^{m \times m}$ denote the symmetric tridiagonal matrix generated by the Lanczos process applied to a symmetric matrix H . Assume that $T + E = LDL^T$ is a modified Cholesky factorization with modification computed using Algorithm 3. In addition, assume that $\delta \leq \delta_{j+1} \leq \delta_{\max}$ for all $0 \leq j < m$. Then*

$$\begin{aligned} \bar{\sigma}_{j+1} &\leq \delta_{j+1} + |\gamma_{j+1}| + |\beta_{j+1}|, & j = 0, 1, \dots, m-1, \\ \sigma_j &\leq |\beta_{j+1}|, & j = 1, 2, \dots, m-1, \end{aligned}$$

so that

$$\|E\| \leq \delta_{\max} + \gamma + 2\beta, \quad \text{where } \gamma = \max_{1 \leq j \leq m} \{|\gamma_j|\} \quad \text{and} \quad \beta = \max_{1 \leq j \leq m} \{|\beta_j|\}.$$

Proof. For the initial modification, if $\gamma_1 \geq \delta$ then $\bar{\sigma}_1 = 0$. Otherwise $\bar{\sigma}_1 = \delta_1 - \gamma_1$, giving the bound $\bar{\sigma}_1 \leq \delta_1 + |\gamma_1|$.

For $j \geq 1$, if $\gamma_{j+1} - \beta_{j+1}^2/\bar{d}_j \geq \delta$, then $\sigma_j = \bar{\sigma}_{j+1} = 0$, and the bounds on σ_j and $\bar{\sigma}_{j+1}$ hold trivially. Otherwise the solution $(\sigma_j, \bar{\sigma}_{j+1})$ is one of the following

$$\sigma_j = |\beta_{j+1}| - \bar{d}_j, \quad \bar{\sigma}_{j+1} = \delta_{j+1} - \gamma_{j+1} + |\beta_{j+1}| \quad (4.4a)$$

$$\sigma_j = 0, \quad \bar{\sigma}_{j+1} = \delta_{j+1} - \gamma_{j+1} + \beta_{j+1}^2/\bar{d}_j \quad (4.4b)$$

$$\sigma_j = \beta_{j+1}^2/(\gamma_{j+1} - \delta_{j+1}) - \bar{d}_j, \quad \bar{\sigma}_{j+1} = 0. \quad (4.4c)$$

If the $(\sigma_j, \bar{\sigma}_{j+1})$ of (4.4a) is feasible, then the nonnegativity of $\bar{d}_j$ implies that

$$\sigma_j \leq |\beta_{j+1}| \quad \text{and} \quad \bar{\sigma}_{j+1} \leq \delta_{j+1} + |\gamma_{j+1}| + |\beta_{j+1}|.$$

Alternatively, if $(\sigma_j, \bar{\sigma}_{j+1})$ of (4.4a) is infeasible, then either $|\beta_{j+1}| \leq \bar{d}_j$ or $|\beta_{j+1}| \leq \gamma_{j+1} - \delta_{j+1}$. If $|\beta_{j+1}| \leq \bar{d}_j$, then the solution is given by (4.4b), with

$$\bar{\sigma}_{j+1} = \delta_{j+1} - \gamma_{j+1} + \beta_{j+1}^2/\bar{d}_j \leq \delta_{j+1} - \gamma_{j+1} + |\beta_{j+1}|.$$

As $\gamma_{j+1} - \beta_{j+1}^2/\bar{d}_j < \delta < \delta_{j+1}$ and $\bar{\sigma}_{j+1}$ is nonnegative, the required bound on $\bar{\sigma}_{j+1}$ follows directly.

If $|\beta_{j+1}| \leq \gamma_{j+1} - \delta_{j+1}$, then the solution is given by (4.4c), with

$$\sigma_j = \beta_{j+1}^2/(\gamma_{j+1} - \delta_{j+1}) - \bar{d}_j \leq |\beta_{j+1}| - \bar{d}_j,$$

in which case $\sigma_j \leq |\beta_{j+1}|$. It follows that $\|E\| \leq \delta_{\max} + \gamma + 2\beta$, where $\gamma = \max_{1 \leq j \leq m} \{|\gamma_j|\}$ and $\beta = \max_{1 \leq j \leq m} \{|\beta_j|\}$. ■

4.1. Iteration-dependent target curvature

The fixed value δ is used to determine whether the current Schur complement is sufficiently positive. If $\gamma_{j+1} - \beta_{j+1}^2/\bar{d}_j \geq \delta$ then no modification is necessary. When the detection test fails, an iteration-dependent target curvature is computed. In order to allow the imposed curvature to reflect the local properties of the matrix being modified, the target curvatures are based on information available from the current matrix being modified. This allows the target curvature to vary throughout the factorization.

Two choices of δ_{j+1} are proposed. The first candidate, denoted by **Schur**, is the magnitude of the unmodified Schur complement $|\gamma_{j+1} - \beta_{j+1}^2/\bar{d}_j|$. The second, denoted by **Eigen**, is the minimum absolute eigenvalue of B_j (cf. (4.1)). In both cases, the choice of δ_{j+1} is safeguarded below by δ and above by $\delta_{\max}$ before it is used in Algorithm 3. The corresponding definitions are summarized in Algorithm 5.

Algorithm 5. Two alternative choices of δ_{j+1} .

procedure $[\delta_{j+1}] = \text{posCurve}(\bar{d}_j, \gamma_{j+1}, \beta_{j+1}, \delta, \delta_{\max})$ [Schur]
 $\hat{\delta}_{j+1} = |\gamma_{j+1} - \beta_{j+1}^2/\bar{d}_j|$; $\delta_{j+1} = \min \{ \max \{ \hat{\delta}_{j+1}, \delta \}, \delta_{\max} \}$;
end procedure

procedure $[\delta_{j+1}] = \text{posCurve}(\bar{d}_j, \gamma_{j+1}, \beta_{j+1}, \delta, \delta_{\max})$ [Eigen]
 $\tau = (\gamma_{j+1} - \bar{d}_j)/(2\beta_{j+1})$;
 $t = \text{sign}(\tau)/(|\tau| + \sqrt{1 + \tau^2})$;
 $\lambda_1 = \bar{d}_j - t\beta_{j+1}$; $\lambda_2 = \gamma_{j+1} + t\beta_{j+1}$;
 $\hat{\delta}_{j+1} = \min \{ |\lambda_1|, |\lambda_2| \}$; $\delta_{j+1} = \min \{ \max \{ \hat{\delta}_{j+1}, \delta \}, \delta_{\max} \}$;
end procedure

When the test in Algorithm 3 detects insufficient curvature, Algorithm 5 can be used to determine the curvature target δ_{j+1} used in the modification subproblem.

The developments of this section have been presented as extensions of Nash's modified Lanczos method because the curvature information required by the two-stage modification arises naturally from the tridiagonal matrix generated by the Lanczos process. Nevertheless, the Lanczos and conjugate-gradient methods are closely related and generate the same sequence of Krylov subspaces in exact arithmetic. This observation suggests that the proposed modifications need not be restricted to a Lanczos-based implementation. In the next section, it is shown that the quantities required by the two-stage modification may be recovered directly from the gradient-based method of conjugate gradients. The resulting formulation provides a gradient-based implementation that is equivalent to the modified Lanczos method of Algorithm 4.

5. A Modified Method of Conjugate Gradients

The modifications introduced in Section 4 are formulated using the tridiagonal matrices generated by the Lanczos process as an extension of the original method proposed by Nash. Nevertheless, the Lanczos and gradient-based methods of conjugate gradients are closely related and, in exact arithmetic, generate the same sequence of expanding Krylov subspaces. This suggests that the quantities required by the two-stage modification may also be recovered directly from the gradient-based CG.

A direct gradient-based implementation is not immediate. In the Lanczos formulation, the next Lanczos vector may be generated independently of finding the next iterate. Consequently, the curvature information required to complete the two-stage modification is available before the step length and the next iterate are computed. In the gradient-based CG formulation, however, the next gradient needed to complete the modification is available only after a step has been taken. Moreover, the second modification in the direction of the current gradient changes the curvature along the current search direction and hence changes the step length used to generate the next gradient. This gives an apparent circular dependence: the completed modification is required to determine the step length, whereas the next gradient required to compute that modification is available only after the step has been taken.

In this section it is shown that this dependence may be resolved by first computing a provisional gradient. A rank-one modification in the direction of the current gradient

changes the scale, but not the direction, of the gradient obtained following an exact line search. This ensures that the provisional gradient provides the direction of the next gradient and may be used to recover the curvature information needed to complete the modification. It is then shown that the resulting gradient-based formulation generates the same modified Hessians as the Lanczos formulation in exact arithmetic. Finally, an implementation is derived that requires only one additional vector and one additional scalar relative to the conventional gradient-based method of conjugate gradients.

Throughout this section, p_j , r_j , and q_j denote the current iterate, gradient, and search direction, respectively, with the initial iterate $p_0 = 0$.

5.1. Recovery of curvature information

In exact arithmetic, the gradients generated by CG are mutually orthogonal. The normalized gradients therefore define the orthonormal matrix

$$R_{j+1} = \begin{pmatrix} \frac{r_0}{\|r_0\|} & \frac{r_1}{\|r_1\|} & \cdots & \frac{r_j}{\|r_j\|} \end{pmatrix},$$

such that the matrix $R_{j+1}^T H R_{j+1}$ is tridiagonal. This follows from the mutual orthogonality of the gradients, which successively form a basis for the expanding Krylov subspaces. This implies that the diagonal and off-diagonal entries of $R_{j+1}^T H R_{j+1}$ may be recovered directly from the CG gradients. In particular,

$$\gamma_{j+1} = \frac{r_j^T H r_j}{\|r_j\|^2}, \quad \text{and} \quad \beta_{j+1} = \frac{r_j^T H r_{j-1}}{\|r_j\| \|r_{j-1}\|} \quad \text{for } j \geq 1.$$

It follows that the diagonal modifications generated by the two-stage procedure may be expressed in terms of the normalized CG gradients. After the first modification associated with r_j has been introduced, the provisional modified Hessian has the form

$$\tilde{H}_j = H + \sum_{i=1}^j \frac{\bar{\sigma}_i + \sigma_i}{r_{i-1}^T r_{i-1}} r_{i-1} r_{i-1}^T + \frac{\bar{\sigma}_{j+1}}{r_j^T r_j} r_j r_j^T.$$

The next stage of the modification will add a rank-one update in the direction of r_j with curvature σ_{j+1} , and hence changes the curvature along q_j and the step length taken in that direction. In order to determine σ_{j+1} , the two-stage procedure also requires curvature information associated with the next normalized gradient direction. However, this direction would ordinarily be available only *after* the completed step has been taken. The following result shows that it is sufficient to compute a provisional gradient using $\tilde{H}_j$.

Proposition 5.1. *Assume that $p \in \mathbb{R}^n$ such that $r = g + Hp$ is orthogonal to p , and consider search direction q such that $q^T r = -r^T r$. Suppose that curvature is added to H via the rank-one update*

$$\tilde{H} = H + \frac{\sigma}{r^T r} r r^T,$$

where both $q^T H q$ and $q^T \tilde{H} q$ are nonzero. Then the gradient r_+ found using an exact line search with $\tilde{H}$ is parallel to the gradient $\tilde{r}_+$ found using $\tilde{H}$. In particular,

$$\tilde{r}_+ = \frac{q^T H q}{q^T H q + \sigma r^T r} r_+.$$

Proof. Using $\tilde{H}$ to calculate the step size in the exact line search yields the gradient

$$\tilde{r}_+ = r + \left(\frac{r^T r}{q^T H q + \sigma r^T r} \right) (H q - \sigma r)$$

whereas using H yields

$$r_+ = r + \frac{r^T r}{q^T H q} H q.$$

Therefore

$$\begin{aligned} \frac{q^T H q}{q^T H q + \sigma r^T r} r_+ &= \frac{q^T H q}{q^T H q + \sigma r^T r} \left(r + \frac{r^T r}{q^T H q} H q \right) \\ &= \left(\frac{q^T H q + \sigma r^T r - \sigma r^T r}{q^T H q + \sigma r^T r} \right) r + \left(\frac{r^T r}{q^T H q + \sigma r^T r} \right) H q \\ &= r + \left(\frac{r^T r}{q^T H q + \sigma r^T r} \right) (H q - \sigma r) = \tilde{r}_+. \end{aligned}$$

It follows that $\tilde{r}_+ = \frac{q^T H q}{q^T H q + \sigma r^T r} r_+$, as required. ■

Proposition 5.1 resolves the apparent circular dependence. An exact line search generates a new gradient orthogonal to the previous search space, which implies that the action of $\tilde{H}_{j+1}$ on the iterate p_j is identical to that of

$$\tilde{H}_j + \frac{\sigma_{j+1}}{r_j^T r_j} r_j r_j^T.$$

Therefore a provisional step $\bar{\omega}_j$ may be computed first using $\tilde{H}_j$, yielding

$$\bar{r}_{j+1} = r_j + \bar{\omega}_j \tilde{H}_j q_j.$$

The role of $\bar{\omega}_j$ is to generate a vector parallel to the gradient that would be obtained using the subsequent modified Hessian $\tilde{H}_{j+1}$. Consequently, the normalized provisional gradient may be used to compute the quantities

$$\gamma_{j+2} = \frac{\bar{r}_{j+1}^T H \bar{r}_{j+1}}{\|\bar{r}_{j+1}\|^2}, \quad \beta_{j+2} = \frac{\bar{r}_{j+1}^T H r_j}{\|\bar{r}_{j+1}\| \|r_j\|}.$$

These quantities determine σ_{j+1} and $\bar{\sigma}_{j+2}$ as in Algorithm 3.

5.2. Calculating the modified gradient information

A direct implementation of the gradient-based formulation would require all previous gradients to be stored because the gradient, search direction and step length are all calculated using the action of the modified Hessian on the current search direction. However, the action of the accumulated modifications on the current search direction may be represented instead using one additional vector and one additional scalar. To show this, let $q_0, q_1, \dots, q_j$ denote the search directions generated by the CG algorithm. It holds that

$$q_j^T r_i = -r_j^T r_j \quad \text{for } i \leq j.$$

It is straightforward to establish that for any rank-one modification in the direction of r_i for $i \leq j$, the action on q_j may be recovered as

$$\frac{\sigma}{\|r_i\|^2} r_i r_i^T q_j = -\|r_j\|^2 \frac{\sigma}{\|r_i\|^2} r_i.$$

It follows that the information from the modifications may be stored in a vector ψ_j and scalar χ_j such that $\psi_0 = \mathbf{0}$, $\chi_0 = 0$ and may be updated iteratively as

$$\begin{aligned} \bar{\psi}_{j+1} &= \psi_j + \bar{\sigma}_{j+1} r_j / \|r_j\|^2, & \psi_{j+1} &= \bar{\psi}_{j+1} + \sigma_{j+1} r_j / \|r_j\|^2, \\ \bar{\chi}_{j+1} &= \chi_j + \bar{\sigma}_{j+1} / \|r_j\|^2, & \chi_{j+1} &= \bar{\chi}_{j+1} + \sigma_{j+1} / \|r_j\|^2. \end{aligned}$$

This implies that the search directions and step lengths generated for the modified Hessians $\tilde{H}_j$ and $\tilde{H}_{j+1}$ may be calculated using the expressions

$$q_j^T \tilde{H}_j q_j = q_j^T H q_j + (r_j^T r_j)^2 \bar{\chi}_{j+1}, \quad q_j^T \tilde{H}_{j+1} q_j = q_j^T H q_j + (r_j^T r_j)^2 \chi_{j+1},$$

while the gradients may be generated iteratively using the Hessian-vector product

$$\tilde{H}_j q_j = H q_j - \|r_j\|^2 \bar{\psi}_{j+1}, \quad \tilde{H}_{j+1} q_j = H q_j - \|r_j\|^2 \psi_{j+1}.$$

It follows that the modified Hessian need not be formed explicitly. Relative to the conventional gradient-based method of conjugate gradients, generating the necessary information from modification requires one additional vector, ψ_j , and one additional scalar, χ_j . These identities give a gradient-based implementation of the two-stage modification. At each iteration, the first-stage modification is incorporated into the provisional quantities $\bar{\psi}_{j+1}$ and $\bar{\chi}_{j+1}$. A provisional gradient is then generated and used to compute the additional curvature information required by Algorithm 3. Once the second-stage modification has been determined the step length, gradient, and search direction are updated using the completed modified Hessian. The resulting procedure is given in Algorithm 6.

5.3. Equivalence of the modified Lanczos and gradient-based formulations

It has been shown that Nash's two-stage modification may be integrated into a gradient-based CG framework. It remains to show that the quantities recovered from the provisional gradients produce the same modifications as those generated by the Lanczos-based method.

The iterates generated by Algorithm 6 depend on the accumulated modification of the Hessian. At iteration j , the two-stage procedure introduces additional curvature only in the current gradient direction r_{j-1} and in the direction of the provisional gradient $\bar{r}_j$. Consequently, the search directions $q_0, q_1, \dots, q_{j-2}$, the step lengths $\omega_0, \omega_1, \dots, \omega_{j-2}$, and the iterates $p_0, p_1, \dots, p_{j-1}$, remain unchanged. The completed modification in the direction r_{j-1} determines the current step length ω_{j-1} , and hence the next iterate

$$p_j = p_{j-1} + \omega_{j-1} q_{j-1}.$$

If the algorithm terminates at iteration m , i.e., $\|r_m\| < \tau_{tol}$, then the modified Hessian on completion of iteration j ($j < m$) is

$$\tilde{H}_j = H + \sum_{i=1}^j \frac{\bar{\sigma}_i + \sigma_i}{r_{i-1}^T r_{i-1}} r_{i-1} r_{i-1}^T + \frac{\bar{\sigma}_{j+1}}{r_j^T r_j} r_j r_j^T. \quad (5.1)$$

Algorithm 6. A gradient-based modified conjugate-gradient method.

```

procedure  $[p] = \text{grad\_mod\_CG}(H, g, \tau_{tol}, j_{\max})$ 
  Restrictions:  $g \neq 0, \tau_{tol} > 0, j_{\max} > 0;$ 
   $p_0 = 0; r_0 = g; \chi_0 = 0; \psi_0 = \mathbf{0};$ 
   $\gamma_1 = r_0^T H r_0 / \|r_0\|^2; q_0 = -r_0;$ 
  Choose  $\delta > 0;$ 
   $\bar{\sigma}_1 = \max \{ \gamma_1, \delta \} - \gamma_1; \bar{d}_1 = \gamma_1 + \bar{\sigma}_1;$ 
   $j = 1;$ 
  while  $j \leq j_{\max}$  do
     $\bar{\chi}_j = \chi_{j-1} + \bar{\sigma}_j / \|r_{j-1}\|^2; \bar{\psi}_j = \psi_{j-1} + \bar{\sigma}_j r_{j-1} / \|r_{j-1}\|^2;$ 
     $\bar{\omega}_{j-1} = -r_{j-1}^T q_{j-1} / (q_{j-1}^T H q_{j-1} + \bar{\chi}_j \|r_{j-1}\|^4);$ 
     $\bar{r}_j = r_{j-1} + \bar{\omega}_{j-1} (H q_{j-1} - \|r_{j-1}\|^2 \bar{\psi}_j)$ 
    if  $\|\bar{r}_j\| > \tau_{tol}$  then
       $\gamma_{j+1} = \bar{r}_j^T H \bar{r}_j / \|\bar{r}_j\|^2; \beta_{j+1} = \bar{r}_j^T H r_{j-1} / (\|\bar{r}_j\| \|r_{j-1}\|);$ 
       $[d_{j+1}, \sigma_j, \bar{\sigma}_{j+1}] = \text{modT}(\bar{d}_j, \gamma_{j+1}, \beta_{j+1});$  [The tridiagonal modification]
       $\chi_j = \bar{\chi}_j + \sigma_j / \|r_{j-1}\|^2; \psi_j = \bar{\psi}_j + \sigma_j r_{j-1} / \|r_{j-1}\|^2;$ 
       $\omega_{j-1} = -r_{j-1}^T q_{j-1} / (q_{j-1}^T H q_{j-1} + \chi_j \|r_{j-1}\|^4);$ 
       $r_j = r_{j-1} + \omega_{j-1} (H q_{j-1} - \|r_{j-1}\|^2 \psi_j)$ 
       $q_j = -r_j + (q_{j-1}^T H r_j / (q_{j-1}^T H q_{j-1} + \chi_j \|r_{j-1}\|^4)) q_{j-1};$ 
    else
       $\omega_{j-1} = \bar{\omega}_{j-1}; r_j = \bar{r}_j;$ 
    end if
     $p_j = p_{j-1} + \omega_{j-1} q_{j-1};$ 
    if  $\|r_j\| \leq \tau_{tol}$  break;
     $j \leftarrow j + 1;$ 
  end while
   $p = p_j;$ 
end procedure

```

It follows that the iterates $p_0, p_1, \dots, p_j$, generated by Algorithm 6 coincide with the iterates generated through iteration j by Algorithm 1 applied to $\tilde{H}_j$. Therefore, equivalence of the Lanczos and gradient-based modifications may be established by showing the equivalence of the modified Hessian, which is done in the following proposition.

Proposition 5.2. *Suppose that Algorithms 4 and 6 are applied in exact arithmetic to the same symmetric matrix H and vector g , and assume that the algorithms use the same rule for selecting the target curvatures. Then, at every completed inner iteration, the two algorithms generate the same modifications and hence the same modified Hessian. Consequently, the corresponding search directions are parallel and the iterates are identical.*

Proof. The proof is by induction on the inner iteration. At the initial iteration, the first Lanczos vector satisfies

$$v_1 = -\frac{1}{\|g\|} g = -\frac{1}{\|r_0\|} r_0.$$

Thus, v_1 and the first normalized CG gradient are parallel. The initial curvature is the same in both formulations, i.e.,

$$\gamma_1 = v_1^T H v_1 = \frac{r_0^T H r_0}{\|r_0\|^2}.$$

The value of $\bar{\sigma}_1$ and the corresponding rank-one modification are therefore identical. Assume that the two formulations have generated identical modifications through the completed modified Hessian at iteration j , and that the normalized CG gradients are parallel to the corresponding Lanczos vectors. The provisional gradient $\bar{r}_{j+1}$ is computed using $\tilde{H}_j$. By equivalence of the Lanczos and gradient-based method of conjugate gradients, $\bar{r}_{j+1}$ is parallel to the Lanczos vector v_{j+2} generated using H . It follows that the quantities computed from the provisional gradient agree with the corresponding Lanczos quantities:

$$\gamma_{j+2} = \frac{\bar{r}_{j+1}^T H \bar{r}_{j+1}}{\|\bar{r}_{j+1}\|^2} = v_{j+2}^T H v_{j+2}, \quad \beta_{j+2} = \frac{\bar{r}_{j+1}^T H r_j}{\|\bar{r}_{j+1}\| \|r_j\|} = v_{j+2}^T H v_{j+1}.$$

Hence the two formulations apply Algorithm 3 to the same tridiagonal data and select the same target curvature. They therefore compute the same values of σ_{j+1} and $\bar{\sigma}_{j+2}$, implying that the modifications introduced by the two algorithms are identical. This proves that the completed modified Hessians agree at iteration $j+1$. The result follows by induction. As the two algorithms generate identical modified Hessians, their search directions are parallel and their exact line searches generate identical iterates. ■

The developments in this section show that Nash's two-stage modification may be integrated into a gradient-based CG framework with the use of provisional gradients to compute curvature information and the storage of one extra vector and scalar to represent the accumulated modifications. In exact arithmetic, the resulting modification generates an identical modified Hessian as the Lanczos-based formulation in Algorithm 4, as well as parallel search directions and identical iterates. In finite precision, the quantities are recovered using different recurrence relations. The behavior of the two formulations in finite precision is compared in the following section.

6. Numerical Results

The results were obtained for optimization problems from the CUTEst test collection (see Bongartz et al. [1] and Gould, Orban and Toint [10]). As of August 1, 2026, there are 291 unconstrained problems in the test set. A subset of 286 problems was selected based on the number of variables. In particular, a problem was chosen if the number of variables was of the order of 5000 or less. The same criterion was used to set the dimension of those problems for which the problem size can be specified. The 5 problems BA-L16LS, BA-L21LS, BA-L49LS, BA-L52LS, and BA-L73LS have more than 23K variables and were excluded from the test set. A total of 200 of the 286 problems exhibited an indefinite Hessian.

The runs were performed using MATLAB version R2024a on a System76 Prime Elite Linux machine running Ubuntu 22.04 with a Ryzen 9 9950X (16 Cores, 32 Threads) and 48 GB DDR5 RAM (5600 MHz). The results were obtained using a MATLAB implementation of Algorithm 7, which implements a conventional line-search modified Newton method with a first-order termination condition. The method uses a backtracking line search based on the affine model $\Gamma_k(x) = f(x_k) + \nabla f(x_k)^T(x - x_k)$.

For the proposed methods to be computationally viable for large-scale problems, it is important to limit the number of inner iterations. Carrying the Lanczos process to completion at every outer iteration may require a large number of Hessian-vector products and may therefore incur a substantial computational cost. Nevertheless, the curvature information accumulated during the initial Lanczos iterations may be sufficient to produce a more effective search direction than the negative gradient. Early termination may thus reduce

Algorithm 7. Modified Newton method.

Constants: $j_{\max}, k_{\max}; 0 < \eta_A < \frac{1}{2}; 0 < \gamma_C < 1;$
 $k = 0;$
Choose x_0 ; Evaluate $f(x_0)$ and $\nabla f(x_0);$
while $k \leq k_{\max}$ **and not converged do**
 Use an iterative method to compute a search direction p_k ;
 $\alpha_k = 1; \ell_k = 0;$
 $\rho_k = (f(x_k) - f(x_k + \alpha_k p_k)) / (\Gamma_k(x_k) - \Gamma_k(x_k + \alpha_k p_k));$
 while $\rho_k < \eta_A$ **do**
 $\alpha_k \leftarrow \alpha_k \gamma_C; \ell_k \leftarrow \ell_k + 1;$
 $\rho_k = (f(x_k) - f(x_k + \alpha_k p_k)) / (\Gamma_k(x_k) - \Gamma_k(x_k + \alpha_k p_k));$
 end while
 $x_{k+1} = x_k + \alpha_k p_k;$ Evaluate $\nabla f(x_{k+1});$
 $k \leftarrow k + 1;$
end while

the cost of computing the search direction while retaining some of the potential advantages of a Krylov-subspace method. In the following, we compare three criteria for terminating the Lanczos process before completion.

Nash [15] and Nash and Sofer [16] propose a criterion based on the reduction in the quadratic objective function:

$$j(\tilde{\varphi}_j - \tilde{\varphi}_{j-1}) / \tilde{\varphi}_j \leq \eta_{\text{obj}}. \quad (6.1)$$

(In Appendix A it is shown how the objective function $\tilde{\varphi}$ can be computed without explicit knowledge of the modified Hessian). Dembo, Eisenstat, and Steihaug [5] propose alternative criteria based on the relative residual norm

$$\|r_j\| \leq \eta_{\text{res}} \|g\|, \quad (6.2)$$

and the dynamic reduction of the relative norm of the residual

$$\|r_j\| \leq \min \left\{ \frac{1}{2}, \|g\| \right\} \|g\|. \quad (6.3)$$

This last criterion becomes progressively more stringent as $\|g\|$ decreases. The termination conditions (6.1), (6.2) and (6.3) provide alternative mechanisms for controlling the number of Lanczos iterations and, consequently, the number of Hessian-vector products used to compute the search direction.

In order to satisfy the conditions required for global convergence of the modified Newton method, two additional early-termination conditions are imposed. The first is based on the magnitude of $\|p_j\|$, and the second is based on the value of the normalized directional derivative $-g^T p_j / (\|g\| \|p_j\|)$. These conditions utilize two fundamental properties of the conventional CG method, namely, that the norm of p_j is monotonically increasing (Steihaug [22]), and that the normalized directional derivative is monotonically decreasing (Dai, Martínez and Yuan [4]). These results apply to the *modified* Lanczos and gradient-based formulations because the associated search directions are identical to those generated by the Lanczos and CG methods applied to a positive-definite modified Hessian (cf. (4.3) and (5.1)). The specific termination criteria are

$$\|p_j\| \geq p_{\max} \quad \text{and} \quad -\frac{g^T p_j}{\|g\| \|p_j\|} \leq c_{\min}.$$

Figure 1 gives the performance profiles for the Lanczos-based and CG-based methods based on Nash's original method and the two proposed extensions. The profiles also include results from the Lanczos trust-region method GLTR (Gould et al. [9]). The left figure gives the profiles for the number of function evaluations. The right figure gives the profiles for the number of Hessian-vector products. The iteration limit is $n + 1$ in all cases.

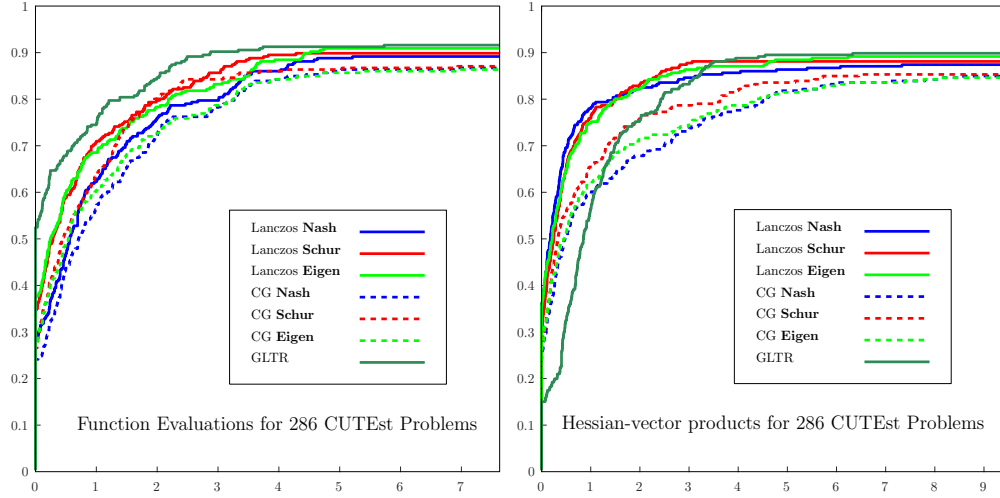


Figure 1: Performance profiles for the Lanczos-based and CG-based methods based on Nash's original method and the two proposed extensions. The profiles also include results from the Lanczos trust-region method GLTR. The iteration limit is $n + 1$ in all cases. The methods are applied to 286 unconstrained problems from the CUTEst test set. The left figure gives the profiles for the number of function evaluations. The right figure gives the profiles for the number of Hessian-vector products.

A run was considered to be a success based on two criteria. The first was

$$\|\nabla f(x_k)\| \leq \varepsilon_S(1 + |f(x_k)|) \quad \text{or} \quad \|\nabla f(x_k)\| \leq \epsilon_M^{1/2},$$

where ε_S is a preassigned stationarity tolerance and ϵ_M is the relative machine precision. If the problem is badly scaled, a line search failure may occur before these conditions can be satisfied. In this case the run was considered a success if

$$\|\nabla f(x_k)\| \leq \epsilon_M^{1/2} \|\nabla f(x_0)\| \quad \text{or} \quad |f(x_k)| \leq \epsilon_M^{2/3}(1 + |f(x_0)|).$$

The specific values of the parameters used in the runs are given in Table 1.

Parameter	Value	Parameter	Value
Positive curvature threshold, δ	1.0e-10	Lanczos tolerance, τ_{tol}	1.0e-07
Stationarity tolerance, ε_S	1.0e-05	Newton iteration limit, $k_{\max}$	3000
Lanczos iteration limit, $j_{\max}$	6000	Armijo step tolerance, η_A	1.0e-04
Armijo step reduction factor, γ_C	0.5	Early termination factor, η_{res}	0.5
Early termination factor, η_{obj}	0.01	Upper bound on $\ p_j\ $, $p_{\max}$	1.0e+20
Lower bound on $-g^T p_j / (\ g\ \ p_j\)$, $c_{\min}$	0.01		

Table 1: Constants and tolerances used across all problems and solvers.

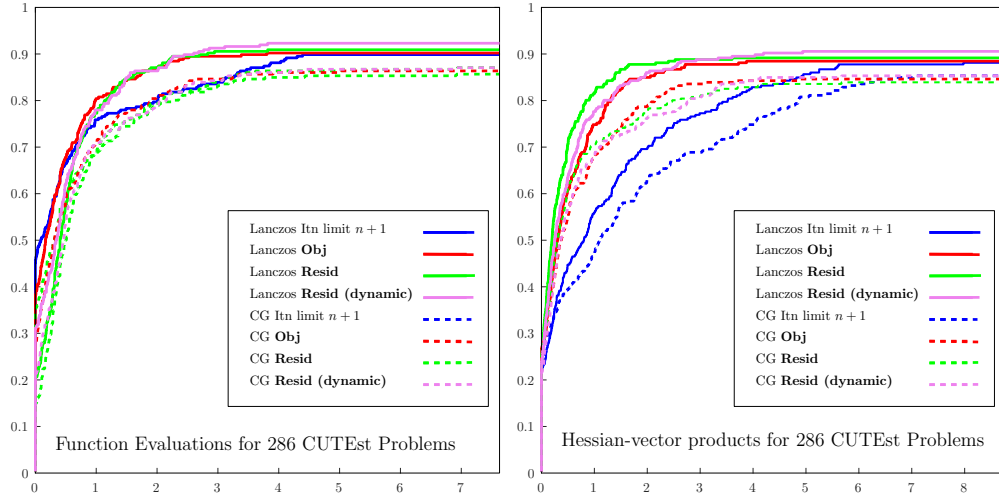


Figure 2: Performance profiles for the Lanczos-based and CG-based methods using the Schur-complement strategy with three early-termination strategies (6.1)–(6.3), labeled **Obj**, **Resid** and **Resid (dynamic)**, respectively. The parameters and tolerances for the methods with early termination are given in Table 1. The methods are applied to 286 unconstrained problems from the CUTEst test set. The left figure gives the profiles for the number of function evaluations. The right figure gives the profiles for the number of Hessian-vector products.

First, the Lanczos implementation exhibits a small but consistent advantage over the conjugate-gradient implementation. Although the performance profiles are similar, the Lanczos-Nash method solves slightly more problems and achieves slightly better overall efficiency. Second, the trust-region method GLTR appears to be the most robust method in terms of the total number of problems solved. However, this increased robustness is generally accompanied by a larger number of Hessian-vector products, indicating a higher cost per solve.

The results also suggest that the choice of target curvature has a noticeable effect on performance. Both adaptive strategies outperform the fixed-curvature approach proposed by Nash over a substantial portion of the test set. Among the adaptive variants, the Schur-complement strategy (**Schur**) provides the strongest overall performance profile, whereas the eigenvalue-based strategy (**Eigen**) solves a few additional problems and appears to be slightly more robust. Overall, the results indicate that exploiting iteration-dependent curvature information can improve efficiency without sacrificing robustness.

Finally, the performance profiles in Figure 2 show that early termination of the inner iteration is beneficial for all the modification strategies considered. In each case, the use of an early-termination criterion improves the balance between function evaluations and Hessian-vector products when compared with carrying the inner iteration to completion. This suggests that the curvature information obtained during the initial Lanczos iterations is often sufficient to compute an effective search direction, and that additional inner iterations may provide only limited practical benefit.

7. Conclusions

A modified Newton method for large-scale unconstrained optimization has been proposed in which two distinct curvature parameters are used. The first determines when modification of the Hessian is required, whereas the second specifies the target curvature imposed by the modification. Separating these roles provides greater flexibility than using a single parameter for both detection and modification. At the same time, the proposed method retains a bound on the difference between the true Hessian and the modified Hessian, thereby controlling the magnitude of the modification.

Two iteration-dependent choices of the target curvature have been proposed. These choices allow the imposed curvature to vary over the course of the optimization process, rather than remaining fixed at all iterations. We have also derived a conjugate-gradient-type implementation of the modification that is theoretically equivalent to its Lanczos formulation. The conjugate-gradient formulation avoids the explicit construction of the Lanczos basis and provides an alternative implementation using Hessian-vector products.

The different modification strategies have been evaluated numerically on a collection of unconstrained optimization problems. The two proposed iteration-dependent choices of the target curvature are compared with the fixed target-curvature choice proposed by Nash. These comparisons are used to examine the computational performance and robustness of the resulting methods, and to investigate the effect of allowing the target curvature to vary between iterations.

Three strategies for the early termination of the Lanczos iterations and conjugate-gradient iterations have been considered. Carrying the inner iteration to completion may require a large number of Hessian-vector products on large-scale problems, whereas the curvature information obtained during the initial iterations may be sufficient to produce an effective search direction. Numerical experiments indicate that the use of early-termination criteria can limit the cost of the inner iteration while retaining the potential benefits of the modified Krylov-subspace direction.

References

- [1] Ingrid Bongartz, Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. CUTE: Constrained and unconstrained testing environment. *ACM Trans. Math. Software*, 21(1):123–160, 1995. [17](#)
- [2] Kurt M. Bretthauer and Bala Shetty. The nonlinear knapsack problem—algorithms and applications. *European Journal of Operational Research*, 138(3):459–472, 2002. [8](#)
- [3] Sheung Hun Cheng and Nicholas J. Higham. A modified Cholesky algorithm based on a symmetric indefinite factorization. *SIAM J. Matrix Anal. Appl.*, 19(4):1097–1110, 1998. [2](#)
- [4] Yu-Hong Dai, José Mario Martínez, and Jin-Yun Yuan. An increasing-angle property of the conjugate gradient method and the implementation of large-scale minimization algorithms with line searches. *Numer. Linear Algebra Appl.*, 10(4):323–334, 2003. [18](#)
- [5] Ron S. Dembo, Stanley C. Eisenstat, and Trond Steihaug. Inexact Newton methods. *SIAM J. Numer. Anal.*, 19(2):400–408, 1982. [18](#)
- [6] Haw-ren Fang and Dianne P. O’Leary. Modified Cholesky algorithms: a catalog with new approaches. *Math. Program.*, 115(2):319–349, 2008. [2](#)
- [7] Anders Forsgren, Philip E. Gill, and Walter Murray. Computing modified Newton directions using a partial Cholesky factorization. *SIAM J. Sci. Comput.*, 16:139–150, 1995. [2](#)
- [8] Philip E. Gill and Walter Murray. Newton-type methods for unconstrained and linearly constrained optimization. *Math. Program.*, 7:311–350, 1974. [2](#)
- [9] Nicholas I. M. Gould, Stefano Lucidi, Massimo Roma, and Philippe L. Toint. Solving the trust-region subproblem using the Lanczos method. *SIAM J. Optim.*, 9(2):504–525, 1999. [19](#)

- [10] Nicholas I. M. Gould, Dominique Orban, and Philippe L. Toint. CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization. *Comput. Optim. Appl.*, 60(3):545–557, 2015. [17](#)
- [11] Magnus R. Hestenes and Eduard Stiefel. Methods of conjugate gradients for solving linear systems. *J. Res. Nat. Bur. Standards*, 49:409–436, 1952. [4](#)
- [12] Shmuel Kaniel and Achiya Dax. A modified Newton’s method for unconstrained minimization. *SIAM J. Numer. Anal.*, 16:324–331, 1979. [2](#)
- [13] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Nat. Bur. Standards*, 45:255–282, 1950. [4](#)
- [14] Jorge J. Moré and Danny C. Sorensen. On the use of directions of negative curvature in a modified Newton method. *Math. Program.*, 16:1–20, 1979. [2](#)
- [15] Stephen G. Nash. Newton-type minimization via the Lanczos method. *SIAM J. Numer. Anal.*, 21(4):770–788, 1984. [2](#), [6](#), [10](#), [18](#)
- [16] Stephen G. Nash and Ariela Sofer. Assessing a search direction within a truncated-Newton method. *Oper. Res. Lett.*, 9(4):219–221, July 1990. [18](#)
- [17] Christopher C. Paige and Michael A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.*, 12:617–629, 1975. [4](#)
- [18] Clément W. Royer, Michael O’Neill, and Stephen J. Wright. A Newton-CG algorithm with complexity guarantees for smooth unconstrained optimization. *Math. Program.*, 180(1):451–488, 2020. [2](#)
- [19] Robert B. Schnabel and Elizabeth Eskow. A new modified Cholesky factorization. *SIAM J. Sci. and Statist. Comput.*, 11:1136–1158, 1990. [2](#)
- [20] Robert B. Schnabel and Elizabeth Eskow. A revised modified Cholesky factorization algorithm. *SIAM J. Optim.*, 9(4):1135–1148, 1999. [2](#)
- [21] Chungeng Shen, Xiongda Chen, and Yumei Liang. A regularized Newton method for degenerate unconstrained optimization problems. *Optim. Lett.*, 6:1913–1933, 2012. [2](#)
- [22] Trond Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM J. Numer. Anal.*, 20:626–637, 1983. [18](#)
- [23] Yaguang Yang. A globally and quadratically convergent algorithm with efficient implementation for unconstrained optimization. *Computational and Applied Mathematics*, 34(3):1219–1236, 08 2015. [2](#)
- [24] Zhewei Yao, Peng Xu, Fred Roosta, Stephen J. Wright, and Michael W. Mahoney. Inexact Newton-CG algorithms with complexity guarantees. *IMA J. Numer. Anal.*, 43(3):1855–1897, 2023. [2](#)
- [25] Yuhao Zhou, Jintao Xu, Bing-Rui Li, Cheng-Long Bao, Chao Ding, and Jun Zhu. A regularized Newton method for nonconvex optimization with global and local complexity guarantees. *Advances in Neural Information Processing Systems*, 38:73447–73502, 2025. [2](#)
- [26] Hong Zhu and Yunhai Xiao. A hybrid inexact regularized Newton and negative curvature method. *Comput. Optim. Appl.*, 88:849 – 870, 2024. [2](#)

A. Computing the Objective of the Modified Quadratic

If the Lanczos-Nash algorithm terminates at iteration m , the final modified Hessian $\tilde{H}_{m-1}$ may be written in the form

$$\tilde{H}_{m-1} = H + \sum_{i=1}^{m-1} (\sigma_i + \bar{\sigma}_i) v_i v_i^T + \bar{\sigma}_m v_m v_m^T,$$

where $\{\sigma_i\}_{i=1}^{m-1}$ and $\{\bar{\sigma}_i\}_{i=1}^m$ are the diagonal modifications generated by Algorithm 3.

An apparent difficulty of working with $\tilde{\varphi}(p)$ is that the quadratic term involves the final approximate Hessian $\tilde{H}_m$. The following result shows that only the current modified Hessian is needed.

Result A.1. If $j < m$, then the matrix $\tilde{H}_{m-1}$ acts on p_j as

$$H + \sum_{i=1}^j (\sigma_i + \bar{\sigma}_i) v_i v_i^T,$$

i.e., both matrices have the same action on p_j .

Proof. As $p_0 = 0$, it follows that $p_j \in \text{span}\{v_1, v_2, \dots, v_j\}$. The orthogonality of the Lanczos vectors implies that

$$\begin{aligned} \tilde{H}_{m-1} p_j &= H p_j + \sum_{i=1}^{m-1} (\sigma_i + \bar{\sigma}_i) v_i v_i^T p_j + \bar{\sigma}_m v_m v_m^T p_j \\ &= H p_j + \sum_{i=1}^j (\sigma_i + \bar{\sigma}_i) v_i v_i^T p_j = \left(H + \sum_{i=1}^j (\sigma_i + \bar{\sigma}_i) v_i v_i^T \right) p_j, \end{aligned}$$

as required. ■

Result A.2. Given the iterates $p_0, p_1, \dots, p_j$ associated with the Lanczos-Nash algorithm, it holds that

$$\tilde{\varphi}(p_{j+1}) = -\frac{1}{2} p_{j+1}^T \tilde{H}_{m-1} p_{j+1} = \tilde{\varphi}(p_j) - \frac{1}{2} \omega_j \|\tilde{r}_j\|,$$

where $p_{j+1} = p_j + \omega_j q_j$ and $\tilde{r}_j = g + \tilde{H}_{m-1} p_j$.

Proof. If $\tilde{r}_j = \nabla \tilde{\varphi}(p_j) = g + \tilde{H}_{m-1} p_j$, then $\tilde{r}_j = \left(H + \sum_{i=1}^j (\sigma_i + \bar{\sigma}_i) v_i v_i^T \right) p_j$, as the iterates of the Lanczos-Nash algorithm minimize $\tilde{\varphi}(p)$ over a sequence of expanding subspaces, it must hold that p_{j+1} is orthogonal to $\nabla \tilde{\varphi}(p_{j+1})$, i.e.,

$$0 = p_{j+1}^T (g + \tilde{H}_{m-1} p_{j+1}) = \tilde{\varphi}(p_{j+1}) + \frac{1}{2} p_{j+1}^T \tilde{H}_{m-1} p_{j+1},$$

which implies that $\tilde{\varphi}(p_{j+1}) = -\frac{1}{2} p_{j+1}^T \tilde{H}_{m-1} p_{j+1}$. The scalar $p_{j+1}^T \tilde{H}_{m-1} p_{j+1}$ may be written in the form

$$\begin{aligned} p_{j+1}^T \tilde{H}_{m-1} p_{j+1} &= (p_j + \omega_j q_j)^T \tilde{H}_{m-1} (p_j + \omega_j q_j) \\ &= p_j^T \tilde{H}_{m-1} p_j + 2\omega_j q_j^T \tilde{H}_{m-1} p_j + \omega_j^2 q_j^T \tilde{H}_{m-1} q_j. \end{aligned}$$

As $p_j \in \text{span}\{q_0, q_1, \dots, q_{j-1}\}$, the vector q_j is conjugate to p_j and $p_j^T \tilde{H}_{m-1} q_j = 0$. It follows that

$$p_{j+1}^T \tilde{H}_{m-1} p_{j+1} = p_j^T \tilde{H}_{m-1} p_j + \omega_j^2 q_j^T \tilde{H}_{m-1} q_j.$$

Given that $\omega_j \tilde{H}_{m-1} q_j = \tilde{r}_{j+1} - \tilde{r}_j$, the exact line search gives $q_j^T \tilde{r}_{j+1} = 0$ and it follows that $q_j^T \tilde{r}_j = -\|\tilde{r}_j\|$. Therefore we have that

$$\tilde{\varphi}(p_{j+1}) = \tilde{\varphi}(p_j) - \frac{1}{2} \omega_j \|\tilde{r}_j\|,$$

which is what we wanted to show. ■

Result A.2 shows that updating the objective function value requires the norm of the gradient $\tilde{r}_j$. By Result A.1 the norm of the gradient $\tilde{r}_j$ is equivalent to the gradient norm at p_j using the modified Hessian at iteration j , i.e., $\|\tilde{r}_j\| = |\beta_{j+1}| \omega_{j-1}$.