A GLOBALLY CONVERGENT MODIFIED CONJUGATE-GRADIENT LINE-SEARCH ALGORITHM WITH INERTIA CONTROLLING

WENWEN ZHOU[†], JOSHUA D. GRIFFIN[†], AND IOANNIS G. AKROTIRIANAKIS [†]

Abstract. In this paper we have addressed the problem of unboundedness in the search direction when the Hessian is indefinite or near singular. A new algorithm has been proposed which naturally handles singular Hessian matrices, and is theoretically equivalent to the trust-region approach. This is accomplished by performing explicit matrix modifications adaptively that mimic the implicit modifications used by trust-region methods. Further, we provide a new variant of modified conjugate gradient algorithms which implements this strategy in a robust and efficient way. Numerical results are provided demonstrating the effectiveness of this approach in the context of a line-search method for large-scale unconstrained nonconvex optimization.

Key words. nonlinear programming, unconstrained optimization, trust region methods, conjugate gradient method

AMS subject classifications. 90C06, 90C30, 90C26, 65K05, 49M37, 49M30

1. Introduction.

In this paper we consider the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x) \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is assumed to be a twice continuously differentiable and possible nonconvex. Two popular approaches for handling nonconvexity are line-search and trust-region methods. Both begin with a second-order Taylor expansion modeling changes in $f(x_k)$ near the current point x_k:

$$m_k(s) = s^T g_k + \frac{1}{2} s^T H_k s \approx f(x_k + s) - f(x_k) \tag{1.2}$$

where $g_k = \nabla f(x_k)$ and $H_k = \nabla^2 f(x_k)$. When H_k is positive-definite, the unique global minimizer of the quadratic model is given by $s_k = -H_k^{-1} g_k$. If we define λ_1 to be the smallest eigenvalue of H_k, then it can be shown that $\|s_k\| \to \infty$ as $\lambda_1 \to 0$. Thus, as H_k approaches a singular system, the corresponding minimizer s_k of $m_k(s)$ drifts arbitrarily far away; this increases the probability that the predicted decrease in the objective, given $m_k(s_k)$, is irrelevant, and the chance of $f(x)$ being decreased at such a point, degrading to random; further, if H_k is indefinite, $m_k(s)$ is no longer bounded below, and more sophisticated machinery must be used in conjunction with the quadratic model, to ensure the resultant trial-step remains sufficiently bounded. Line-search methods, and trust-region methods were designed to handle this issue [6, 14, 9, 21]. The primary difference is that line-search methods explicitly modify H_k, while trust-region methods implicitly modify H_k using an explicit constraint on the step-size.

Mathematically, we can compare the two approaches as follows. Line-search algorithms seek to find a small perturbation E_k of H_k, forming an approximate Hessian

[†]Advanced Analytics Devision, Operations Research and Management Science R&D, SAS Institute Inc., 100 SAS Campus Drive, Cary, NC 27513, USA.
Email: {Wenwen.Zhou,Joshua.Griffin,Ioannis.Akrotirianakis}@sas.com.

$\hat{H}_k = H_k + E_k$ where $\hat{H}_k \succ 0$. The search direction s_k is then determined by solving:

$$\underset{s\in\mathbb{R}^n}{\text{minimize}} \;\; m_k(s) + P_k(s) \triangleq \hat{m}(s) \tag{1.3}$$

where $P_k(s) = (s^T E_k s)/2$. Trust-region algorithms, on the other hand, minimize the original quadratic model, subject to an explicit step-length constraint, for example:

$$\begin{array}{ll} \underset{s\in\mathbb{R}^n}{\text{minimize}} & m_k(s) \\ \text{subject to} & \|s\|_2 \le \delta_k. \end{array} \tag{1.4}$$

The global minimum for (1.4) can be determined by finding a solution pair (s_k, σ_k) satisfying

$$\begin{array}{ll} (H_k + \sigma_k I)s_k = -g_k, & (H_k + \sigma_k I) \succeq 0, \quad \sigma_k \ge 0, \\ \|s_k\| \le \delta, & \sigma_k(\|s_k\| - \delta) = 0. \end{array}$$

Thus the trust-region solution has a more complex form, and is arguably more difficult to solve.

Line-search methods are attractive in that they avoid this added complexity and permit a simpler subproblem to be solved, once an appropriate Hessian modification has been determined. Common strategies for modifying the indefinite Hessian matrix can be found in standard texts for optimization such as Nocedal and Wright [21], Gill, Murray, and Wright [14], and Fletcher [10]; however, it is well-known that these strategies may be problematic when the Hessian has eigenvalues near zero. For this reason, many turn to trust-region algorithms where the necessary matrix modification is naturally and implicitly defined via a step-size constraint. Extensive discussions on trust-region algorithms can be found in Conn et al. [6] and Nocedal and Wright [21]. More recently, hybrid trust-search methods have also been proposed that either perform a line-search on the trust-region subproblem solution [22, 11, 12, 18, 13], or use an explicit trust-region to determine a suitable weight for near singular vectors with respect to the Hessian [18, 13] .

For large-scale problems, matrix factorizations may be prohibitive making iterative approaches attractive. Iterative approaches for solving (1.3) and (1.4) are often based upon applying modified variants of either PCG or Lanczos method to the system $Hs = -g$ as suggested in [23, 25, 20, 26, 2, 15, 5, 19, 17, 8, 7]. The following works [26, 15, 5, 19, 17, 8, 7] all fall into the iterative trust-region, or trust-search category; while [23, 20, 2] are dedicated to developing iterative matrix modification strategies for line-search methods.

In this paper we will focus on iterative strategies for constructing valid line-search directions. Previous mentioned iterative matrix modification strategies are all related in that the matrix E_k from (1.3) has form

$$E_k = \sum_j \alpha_j q_j q_j^T \tag{1.5}$$

where q_j denotes a sequence of Lanczos vectors (O'Leary [23], Nash [20] and Arioli et al. [2]). The Lanczos vectors are advantageous for iterative matrix modification as they avoids restarts and are guaranteed to exist as long as the current CG residual vector is nonzero. These strategies thus differ primarily in how they select α_j. The unifying feature of all three approaches [23, 20, 2] is their strong motivation to keep $\|E_k\|$ small, while avoiding singularity in $\hat{H}_k$ using a fixed nonzero lower bound σ.

O'Leary [23] and Nash [20] proposed iterative line-search methods based upon classical matrix modification strategies, that, as mentioned earlier, can be problematic near singularity. A more dynamic approach was proposed by Arioli et al. [2] where an adaptive bound was proposed for choosing the lower bound on the modified Hessian. However, the authors mentioned that it was unclear how best to choose the sequence of lower bounds σ_k, and also required that this parameter always be larger than the fixed nonzero lower bound σ. It is worth noting at this point that all three approaches face a dilemma in selecting σ appropriately: (1) if chosen too small, the resulting search direction may be quite poor and dramatically slow convergence, (2) if chosen too large, then $\hat{H}_k$ cannot converge to H^* in the limit.

In this paper we present a new inertia controlling matrix-modification strategy that naturally selects appropriate modifications for the Hessian matrix. The inertia of the Hessian is controlled (like a trust-region algorithm) based upon the quality of the previous search direction; for this reason, it is quite possible that a dramatic modification to the Hessian will be made, even if the smallest eigenvalue of the current Hessian matrix is large and positive. Similarly the current Hessian may be numerically singular, but left unmodified. We show that this strategy creates an implicit trust-region radius that we control in a similar manner to a trust-region. However, unlike iterative trust-region methods, there is no need to accommodate an explicit step-size constraint. The end results is that we are free to construct a line-search direction from a simple modified CG algorithm that handles singularity in the Hessian matrix as naturally and robustly as corresponding trust-region approaches. Thus more complex conjugate-gradient based strategies for optimizing a constrained subproblem, such as those in [26, 15, 5, 19, 17, 8, 7] are unnecessary.

Although we perform a modification in the same space of Lanczos vectors, the α_j's are selected in a manner that ensure equivalence convergence properties to that of trust-region methods. In a perhaps dramatic departure from recent approaches, we make no effort to bound the size of the modification matrix E_k; in fact, E_k may be infinitely large whenever H_k is indefinite without effecting the convergence properties of the algorithm. Further, we make no effort to bound the smallest singular value of $H_k + E_k$ directly; rather modification to H_k occur whenever to not do so would adversely affect the resulting search direction. A key feature of our algorithm is that the modified Hessian can approach a singular system only in as much as the current corresponding gradient also approaches zero. Thus we ensure that $\hat{H}_k^{-1} g_k$ remains bounded.

Proceeding in this manner a new matrix-free algorithm is created that naturally handles nonconvexity. It combines the concepts of trust-regions, without using an actual trust-region, to avoid weaknesses of past matrix-modifications strategies in the presence of singularity. This strategy in practice appears to be much more adaptive to the local geometry of the problem defined by both first and second order information. By dynamically controlling inertia in this manner, a natural way to select the lower bound on the smallest eigenvalue of H_k at each iteration is provided (an issue described as "unclear" in Arioli et al. [2]). We further emphasize that the approach we present is stable with minimal memory requirements in that it does not require explicit vector storage nor rely on (easily lost) Lanczos vectors orthogonality. Numerical results demonstrates the effectiveness of this approach in the context of a line-search method for large-scale unconstrained nonconvex optimization.

Before closing this section, we should note that the concept of implicit trust-regions is used in a separate context by the very interesting work of Baker et. al [3], where the trust-region ratio ”predicted versus actual reduction” is used within the trust-region subproblem solver to decide when to halt optimization of the quadratic model. We thus caution readers familiar with this paper that we will make occasional use the same term to describe our algorithm, however, in a context that is quite different.

The paper is organized as follows. In Section 1.1 we provide basic notation and definitions that will be used throughout the paper. In Section 1.2 a short discussion is given to motivate the new matrix modification strategy in the context of eigenvectors and eigenvalues. In Section 2 we describe the new globally convergent modified conjugate gradient line-search algorithm. Section 3 contains global and local convergence theory for this algorithm that is equivalent in strength to existing theory for trust-region algorithms. It will also explain why the proposed algorithm can be seen as using an implicitly defined trust-region. Section 4 contains numerical results on a suite of test problems. The final conclusion and some possible future work are given in Section 5. In the appendix we provide a theorem in the context of the modified conjugate-gradient algorithm that may be used to prove convergence for our approach even if the modification matrix E_k has infinite norm.

1.1. Notation. To reduce notational complexity, for the remainder of the paper, we will drop the k suffix whenever discussing a single subproblem; thus, we will use g for g_k and H for H_k. The ratio of the predicted reduction and actual reduction is defined by

$$\rho_k(s) = \frac{f(x_k + s) - f(x_k)}{m_k(s) - m_k(0)} \tag{1.6}$$

where $m_k(s)$ is defined in (1.2). Finally, when referring to the Modified Conjugate Gradient (MCG) algorithm in this paper we imply any conjugate-gradient algorithm that uses E_k of the form (1.5) to ensure positive-definiteness.

1.2. Motivating a new matrix modification strategy. In this section, we will illustrate the effectiveness of this matrix modification approach by exploring the spectral decomposition of H. For notational simplicity in this section, we drop the suffix k. Let $H = V\Sigma V^T$ where $V = [v_1, \ldots, v_n]$ denote the matrix of the normalized eigenvectors of H, and Σ the corresponding diagonal matrix of eigenvalues, $\text{diag}(\Sigma) = (\sigma_1, \ldots, \sigma_n)$. Then, given a bound δ and setting $s = Vy$, the corresponding trust-region subproblem transformed into the following problem:

$$\begin{array}{ll} \underset{y \in \mathbb{R}^n}{\text{minimize}} & m_k(Vy) = y^T \hat{g} + y^T \Sigma y \\ \text{subject to} & \|y\|_2 \le \delta. \end{array}$$

where $\hat{g} = V^T g$ and we have made use of the property that $\|Vy\|_2 = \|y\|_2$. Because the objective is now completely decoupled, the transformed subproblem would completely decouple into a series of n one-dimensional problems if the trust-region constraint were similarly decoupled. For this reason we may think of replacing the two-norm with the infinity norm of y, as the p-vector norms are equivalent.

Thus, instead solve the following related subproblem:

$$\begin{array}{ll} \underset{y \in \mathbb{R}^n}{\text{minimize}} & y^T \hat{g} + \frac{1}{2} y^T \Sigma y \\ \text{subject to} & -\delta \le y \le \delta, \end{array}$$

which may be solved analytically as the solution now completely decouples into a sequence of n one-dimensional trust-region subproblems:

$$\min_{y_i} \; y_i \hat{g}_i + \frac{\sigma_i y_i^2}{2}, \text{ subject to } |y_i| \leq \delta, \text{ for } i = 1, \ldots, n.$$

Thus we see that

$$s^* = V \hat{\Sigma}^{-1} V^T g,$$

where

$$\hat{\sigma}_i = \begin{cases} \sigma_i & \text{if } \sigma_i > 0 \text{ and } |\hat{g}_i / \sigma_i| \leq \delta \\ 1/\delta & \text{if } \sigma_i \leq 0 \text{ and if } g_i = 0, \\ |\hat{g}_i|/\delta & \text{otherwise.} \end{cases}$$

We therefore see that the motivation for modification of Σ is highly dependent upon the size of $g^T v_i$ and inversely related to the desired step-length. A similar discussions may be found in [17]. The approach used in this paper will incrementally build the solution vector in a similar fashion to the above discussion, however, we will substitute the conjugate vectors for the eigenvectors, and the normalized CG diagonal for σ_i. That is, as with the eigenvectors, we set $s = Py$, where P denotes a matrix of H-conjugate vectors. Then we can analytically define our implicit trust-region as

$$\{s = Py : \|y\|_\infty \leq \delta\}.$$

Because the CG vectors are not orthogonal, we do not have the equivalent condition $\|s\|_2 = \|Py\|_2$. Thus, a well-defined trust-region algorithm cannot be directly applied, as the size and direction of vectors in the matrix P will change from iteration to iteration. For this reason we develop an upper bound on the diameter of this region in terms of the inertia of the modified H and the current gradient, that will encapsulate this region in a natural way. (Note that the actual implicit trust-region used at each iteration will have a more general form:

$$\{s = Py : y_\ell \leq y \leq y_u\}$$

where y_ℓ and y_u are implicitly defined.) Proceeding in this manner we are able to show that the resultant method (as best we can tell) is one-to-one equivalent with similar iterative trust-region methods in theoretical strength; for both global and local convergence properties of (1.1).

2. Algorithm. We can divide the line-search algorithm into outer and inner iterations. The outer iteration, described in Algorithm 1, performs a line-search and checks for convergence of 1.1. An inertia controlling parameter λ_k is modified each iteration based upon the quality of the search direction and whether or not the inertia of H_k was modified. In Section 3 we will show that λ_k is inversely related to an upper bound on an implicitly defined trust region. The inner iteration, described in Algorithm 2, is a variant of the modified conjugate-algorithm applied to system $H_k s = -g_k$ that controls the inertia of H_k by taking into account the following factors: (i) the resultant effect on the growth of $\hat{s}_j$, (ii) the size of the current gradient, and (iii) the quality of the last search direction. To avoid confusion with other modified

CG methods, for brevity we will refer to this variant as Inertia Controlling Modified Conjugate Gradients (ICMCG). The outer and inner algorithms together build a line-search method that incorporates a new matrix-modification strategy possessing the theoretical strength of a trust-region algoritm; this is shown by demonstrating that Algorithm 2 is actually modifying H_k according to an implicit trust region.

As in Arioli et al. [2] we enforce a lower bound on the modified Hessian in terms of

Algorithm 1 Line-search with ICMCG

Require: Choose x_0, and a sequence $\{\eta_k\} > 0$ satisfying $\eta_k \to 0$;
Require: Set $\epsilon > 0$, $\lambda_0 > 0$, and $k = 0$;
1: **while** $(\|g_k\| > \epsilon)$ **do**
2: cgtol $= \eta_k \|g_k\|$;
3: $[s_k, \mathrm{isMod}] = \textbf{ICMCG}(\mathrm{H_k}, \mathrm{g_k}, \lambda_\mathrm{k}, \mathrm{cgtol})$;
4: $\gamma_k = 1$;
5: **while** $(\rho_k(\gamma_k s_k) < 0.25)$ **do**
6: $\gamma_k = 0.5\gamma_k$;
7: **end while**
8: $x_{k+1} = x_k + \gamma_k s_k$;
9: **if** $(\gamma_k < 1)$ **then**
10: $\lambda_{k+1} = 2\lambda_k$;
11: **else if** $(\rho_k > 0.75)$ and isMod $= 1$ **then**
12: $\lambda_{k+1} = 0.5\lambda_k$;
13: **end if**
14: k=k+1;
15: **end while**

the conjugate vectors of the form:

$$\frac{p_i^T \hat{H} p_i}{p_i^T p_i} \geq \sigma_k.$$

The first distinction of the strategy described in this paper from existing strategies is that we make σ_k proportional to $\|g_k\|$ via the relation $\sigma_k = \lambda_k \|g_k\|$; this allows σ_k to approach 0 in the limit, so long as λ_k is bounded. This can be seen in Steps (4)–(10) in Algorithm 2. Second, the scale term λ_k is used to refine the rate at which σ_k goes to zero according to progress made during the previous iteration of the outer algorithm. This helps tailor the choice of σ_k to the specific problem being solved.

Thus in Steps (9)–(13) of Algorithm 1, λ_k is modified in a similar manner to the trust-region radius in a trust-region algorithm. In Section 3 we show that the parameter λ_k has an inverse relationship with the implicit trust-region used to prove convergence of the outer iterations. Essentially, when the predicted ratio is good, λ_k is decreased, and conversely, when the predicted ratio is bad, λ_k is subsequently increased. The inner iterations of Algorithm 2 solve the $\hat{H}_k s_k = -g_k$ within a scale term η_k of the current norm of the objective gradient. We later prove that convergence is at least linear if η_k is bounded away from 0, and superlinear if η_k converges to 0.

As a result of the following inequality

$$\frac{p_i^T H p_i}{p_i^T p_i} > \lambda_k \|g_k\| \tag{2.1}$$

we see that the modified Hessian can approach a singular system only in as much as the current corresponding gradient also approaches zero. This ensures that even if

$\|H_k^{-1}\|$ approach infinity, the step s_k must still converge to 0 in the limit (which is necessary for fast convergence). Note that Algorithm 2 can easily be adapted to use a preconditioner if available, as in regular PCG (precondioned conjugate gradient) methods. To permit the algorithm to be as general as possible, we only require that the modification term δ satisfy the bound

$$\delta \geq (\lambda_k \|g_k\| \|p_i\|^2 - p_i^T \hat{H} p_i)/r_i^T r_i.$$

in Step 7 of Algorithm 2 with equality on the very first iteration. Thereafter, the modification matrix $\delta r_i r_i^T$ may be as large as desired. (In Appendix we provide a theorem demonstrating that if $\delta = \infty$ whenever indefiniteness is detected and $i > 0$, then ICMCG will terminate with $s_k = \hat{s}_i$ and $r_{i+1} = 0$. Though we do not recommend such an extreme variant of ICMCG in practice, we do emphasize that all the convergence properties stated and proved in Section 3 will continue to hold.

Algorithm 2 Inertia Controlling Modified Conjugate Gradients (ICMCG)

1: **function**$[s_k, \text{isMod}] = \textbf{ICMCG}(\text{H}, \text{g}, \lambda, \text{cgtol})$
2: $\quad \hat{H} = H$, $p_0 = -g$;
3: $\quad$ Set $\hat{s}_0 = 0$, $r_0 = p_0$, $\text{isMod} = 0$, and $i = 0$;
4: $\quad$ **while**$(\|r_i\| > \text{cgtol})$ **do**
5: $\quad\quad$ **if** $(p_i^T \hat{H} p_i \leq \lambda_k \|g_k\| \|p_i\|^2)$ **then**
6: $\quad\quad\quad$ Set $\delta_{\text{low}} = (\lambda_k \|g_k\| \|p_i\|^2 - p_i^T \hat{H} p_i)/r_i^T r_i$;
7: $\quad\quad\quad$ **if** $i = 0$ **then** choose $\delta = \delta_{\text{low}}$ **else** choose $\delta \geq \delta_{\text{low}}$ **end**
8: $\quad\quad\quad$ $\hat{H} = \hat{H} + \delta r_i r_i^T$;
9: $\quad\quad\quad$ $\text{isMod} = 1$;
10: $\quad\quad$ **end**
11: $\quad\quad$ $\alpha_i = r_i^T r_i / p_i^T \hat{H}_i p_i$;
12: $\quad\quad$ $\hat{s}_{i+1} = \hat{s}_i + \alpha_i p_i$; $\quad r_{i+1} = r_i + \alpha_i \hat{H} s_i$;
13: $\quad\quad$ $\beta_{i+1} = r_{i+1}^T r_{i+1} / r_i^T r_i$; $\quad p_{i+1} = -r_{i+1} + \beta_{i+1} p_i$;
14: $\quad\quad$ $i = i + 1$;
15: $\quad$ **end**
16: $\quad$ Set $s_k = \hat{s}_i$
17: **endfunction**

3. Convergence results. In this section we show that the update strategy for λ_k in Algorithm 1 facilitates an implicit trust-region and can be used to adaptively control the size of s_k based upon the progress made during the previous line-search. As best as we can tell, there is a one-one correspondence match existing convergence theorems for trust-region methods with that of Algorithm 1. In this section, we have mainly high-lighted for completeness, way to modify existing proofs to obtain the same convergence properties. Before stating the results, we first state some useful properties of the conjugate gradient algorithm that are needed for later proofs. In Lemma 3.1, we provided some known properties of modified conjugate-gradient algorithms that use the space of Lanczos vectors to perform the matrix modifications.

Lemma 3.1 (Arioli et al. [2]). *Suppose a MCG is applied to the linear system $H_k s = -g_k$ using the Lanczos (or residual) vectors to ensure $p_k \hat{H} p_k > 0$. Then in exact arithmetic the algorithm converges to a point $\hat{s}$ satisfying*

$$\hat{H} \hat{s} = -g_k$$

in less than n iteration. Further, the following properties hold for $0 \le j < i$:

$$p_i^T \hat{H} p_j = 0, \tag{3.1}$$
$$\hat{s}_i^T r_i = 0, \tag{3.2}$$
$$p_j^T r_i = 0, \tag{3.3}$$
$$r_i^T g_k = -r_i^T r_i. \tag{3.4}$$

Finally, if H_k is not modified, then

$$m(\hat{s}_i) \le m(\hat{s}_{i-1}) \tag{3.5}$$

Because of Lemma 3.1 we are assured that all the nice properties of CG will naturally hold for the modified system $\hat{H}s = -g_k$; essentially, it says that applying regular CG to $\hat{H}s = -g_k$, we would generate the same sequence of vectors. We also state the following additional properties considering the modified conjugate-gradient algorithm.

Lemma 3.2. *Suppose that Algorithm 2 is applied to the sytem $H_k s = -g_k$ with inertia monitoring parameter λ. Then the following properties hold at each iteration:*

$$p_i^T \hat{H} p_i \ge \lambda \|p_i\|^2 \|g_k\|, \tag{3.6}$$
$$\|r_i\| \le \|p_i\|, \tag{3.7}$$
$$m(\hat{s}_i) \le m(\hat{s}_{i-1}) \tag{3.8}$$

where $m_k(s) = s^T g_k + \dfrac{1}{2} s^T H_k s$. Further $\hat{s}_i$ solves the subspace subproblem

$$\begin{array}{ll} \underset{\hat{s} \in \mathbb{R}^n}{\text{minimize}} & \hat{m}(\hat{s}) = \hat{s}^T g_k + \dfrac{1}{2} \hat{s}^T \hat{H} \hat{s} \\ \text{subject to} & \hat{s} \in \text{span}(p_0, \ldots, p_{i-1}). \end{array} \tag{3.9}$$

That is, $\hat{s}_i$ denotes the unconstrained minimizer of the modified quadratic $\hat{m}(\hat{s})$ within the subspace spanned by the conjugate-vectors, and where $\hat{m}(s)$ is a special form of (1.3) and $\sum_j \delta_j r_j r_j^T$ is used as $E_k(s)$.

Before proving the lemma, note that only (3.6) is unique to the approach proposed in this paper. All other properties stated in Lemma 3.2 are common properties shared by all modified-conjugate gradient algorithms that use Lanczos (or equivalently residual vectors) to shift toward positive-definiteness. We now give the proof of the lemma.

Proof. That $p_i \hat{H} p_i$ is bounded in (3.6) follows by construction of Algorithm 2. We next observe that by (3.4),

$$\|p_j\|_2^2 = (-r_j + \beta_j p_{j-1})^T (-r_j + \beta_j p_{j-1}) = \|r_j\|^2 + \beta_j^2 \|p_{j-1}\|^2 \ge \|r_j\|_2^2,$$

proving (3.7). The proof of (3.8) can be shown as follows.

By (3.5) we know that $\hat{m}(\hat{s}_{i+1}) \le \hat{m}(\hat{s}_i)$ and since

$$\hat{m}(\hat{s}_{i+1}) = \hat{m}(\hat{s}_i + \alpha_i p_i) = \hat{m}(\hat{s}_i) + \alpha_i (\hat{H}\hat{s}_i + g_k)^T p_i + \alpha_i^2 \frac{1}{2} p_i^T \hat{H} p_i,$$

Thus, we obtain the quantity

$$\alpha_i (\hat{H}\hat{s}_i + g_k)^T p_i + \frac{1}{2} \alpha_i^2 p_i^T \hat{H} p_i \le 0.$$

Note similarly that

$$m(\hat{s}_{i+1}) = m(\hat{s}_i + \alpha_i p_i) = m(\hat{s}_i) + \alpha_i (H_k \hat{s}_i + g_k)^T p_i + \alpha_i^2 \frac{1}{2} p_i^T H_k p_i,$$

Hence we only need show that

$$\alpha_i (H_k \hat{s}_i + g_k)^T p_i + \alpha_i^2 \frac{1}{2} p_i^T H_k p_i \leq \alpha_i (\hat{H} \hat{s}_i + g_k)^T p_i + \frac{1}{2} \alpha_i^2 p_i^T \hat{H} p_i.$$

Trivially $p_i^T H_k p_i \leq p_i^T \hat{H} p_i$. Hence, since

$$\alpha_i (\hat{H} \hat{s}_i + g_k)^T p_i = \alpha_i (H_k \hat{s}_i + g_k)^T p_i + \alpha_i (E \hat{s}_i)^T p_i$$

we need only show that $\hat{s}_i^T E p_i \geq 0$, where $E = \sum_{j=1}^{i} \delta_j r_j r_j^T$. By (3.3) we know

$$\hat{s}_i^T E p_i = \delta_i (p_i^T r_i) \hat{s}_i^T r_i = 0,$$

since $\hat{s}_i^T r_i = 0$ as CG always keeps the current solution vector and current residual vector orthogonal. The proof of (3.9) is a well-known property of the conjugate-gradient algorithm (for example see [6]) which necessarily holds for the modified system $\hat{H} s = -g_k$, as unmodified CG applied to this system would generate the same sequence of conjugate vectors. ☐

Lemma 3.1 illustrates several properties of modified conjugate gradient algorithms that use the Lanczos vector (which is always a multiple of the CG residual r_k) to ensure positive-definiteness. Properties (3.1)–(3.4) ensure that conjugacy is not lost, even if $\hat{H}$ were modified at each CG iteration. Second, the corresponding sequence of modified CG solution vectors s_i decrease the modified quadratic model monotonically at each iteration. Further, because ${p_i}^T \hat{H} p_i$ is bounded below by $\lambda_k \|p_i\|^2 \|g_k\|$ for all i, we can show that the CG vector s_k is bounded in terms of λ_k as stated in Theorem 3.3.

THEOREM 3.3. *Let s_k denote the search direction obtained by Algorithm 2. Then*

$$\|s_k\| \leq \frac{n}{\lambda_k} \tag{3.10}$$

Proof. Let $\hat{s}_i$ denote the corresponding i_{th} iteration of Algorithm 2. We will begin by showing that

$$\|\hat{s}_i\| \leq \frac{i}{\lambda_k}, \tag{3.11}$$

and the result then follows from Lemma 3.1. That is, the magnitude of the i_{th} iterate of Algorithm 2 is less than $1/\lambda_k$ times the number of Algorithm 2 iterations completed thus far. Equation (3.11) is shown by induction on i. It is obvious when $i = 0$; now assume that it is true at iteration j, that is,

$$\|\hat{s}_j\| \leq \frac{j}{\lambda_k} \tag{3.12}$$

From (3.4), we have:

$$\hat{s}_{j+1} = \hat{s}_j + \alpha_j p_j = \hat{s}_j + \frac{{r_j}^T r_j}{{p_j}^T \hat{H} p_j} p_j = \hat{s}_j - \frac{{r_j}^T g_k}{{p_j}^T \hat{H} p_j} p_j$$

And from (2.1) and (3.12), and (3.7), we have

$$\|\hat{s}_{j+1}\| \leq \|\hat{s}_j\| + \frac{\|p_j\|\|r_j\|\|g_k\|}{\lambda_k\|p_j\|^2\|g_k\|} = \|\hat{s}_j\| + \frac{\|r_j\|}{\lambda_k\|p_j\|} \leq \frac{j}{\lambda_k} + \frac{1}{\lambda_k} = \frac{j+1}{\lambda_k}$$

This completes the proof by inductions. □

Note that what the proof Theorem 3.3 actually showed is that at each iteration of Algorithm 2 the solution vector, $\hat{s}_i$, can grow by at most a factor of $1/\lambda_k$. Notice further that this is not a strict upper bound, and hence only loosely defines where the true trust-region lives. This important result together with several desirable properties of the algorithm given later that are very similar to those in trust region methods show that the algorithm proposed in this paper is theoretically equivalent to the trust-region approach. Nevertheless, this new approach does not have to explicitly deal with the issues associated with the bounds of trust regions

Before we consider the global convergence of Algorithm 1, we give one more lemma. This useful lemma is very similar to the result from a trust region method that the total predicted decrease is at least a fraction of the that obtained at the Cauchy point [6, 21].

Lemma 3.4. *Let s_k be the search direction calculated by Algorithm 2. Then we have:*

$$m(0) - m(s_k) \geq \frac{\|g_k\|}{2} \min\left(\frac{1}{\lambda_k}, \frac{\|g_k\|}{\|H_k\|}\right) \tag{3.13}$$

Proof. Because $s_k = \hat{s}_i$ and $m(\hat{s}_i) \leq m(\hat{s}_{i-1})$ by Lemma 3.1, it suffices to show this bound for $m(\hat{s}_1)$.

Since $\alpha_0 = g_k^T g_k / g_k^T \hat{H} g_k$ we have

$$\begin{aligned} m(\hat{s}_1) = m(-\alpha_0 g_k) &= -\alpha_0 g_k^T g_k + \frac{1}{2}\alpha_0^2 g_k^T H_k g_k \\ &= \frac{(g_k^T g_k)^2}{g_k^T \hat{H} g_k}\left[-1 + \frac{(g_k^T H_k g_k)}{2 g_k^T \hat{H} g_k}\right] \end{aligned} \tag{3.14}$$

Now either H_k is modified on the first iteration, or remains the same. If H_k is unmodified, then, from (3.14) we have

$$m(\hat{s}_1) \leq -\frac{\|g_k\|^4}{2 g_k^T H_k g_k} \leq -\frac{\|g_k\|^4}{2\|H_k\|\|g_k\|^2} = -\frac{\|g_k\|^2}{2\|H_k\|} \leq -\frac{\|g_k\|}{2} \min\left(\frac{1}{\lambda_k}, \frac{\|g_k\|}{\|H_k\|}\right),$$

implying (3.13) holds. If H_k is modified, then the following must hold:

$$g_k^T H_k g_k / g_k^T g_k \leq \lambda_k \|g_k\| \text{ and } g_k^T \hat{H} g_k / g_k^T g_k = \lambda_k \|g_k\|.$$

And hence, because of (3.14), we have

$$m(\hat{s}_1) = \frac{\|g_k\|}{\lambda_k}\left[-1 + \frac{g_k^T H_k g_k}{2\|g_k\|^3 \lambda_k}\right] \leq -\frac{\|g_k\|}{2\lambda_k},$$

which again implies (3.13) holds.□

The following two lemmas ensure that at each iteration s_k is a valid line-search direction and thus sufficient decrease is guaranteed in a finite number of line-search iterations.

LEMMA 3.5. *Suppose that s_k is obtained from Algorithm 2, and g_k is not 0. Then*

$$s_k^T g_k < 0.$$

Proof. This follows naturally from (3.9) in Lemma 3.2 which states that s_k is the unconstrained minimizer of the $\hat{m}(s)$ in the space of computed conjugate gradient vectors. If $s_k^T g_k > 0$, then

$$\hat{m}(-s_k) < \hat{m}(s_k),$$

contradicting (3.9). If $s_k^T g_k = 0$ then $\hat{m}(s_k) = s_k^T \hat{H} s_k \geq 0$. However, this is a contradiction since CG applied to $\hat{H}s = -g$ generates the identical sequence $\{s_k\}$ satisfying

$$\hat{m}(s_k) \leq \hat{m}(s_{k-1}) \leq \ldots \leq \hat{m}(s_1) = \min_{\alpha} \hat{m}(\alpha g) < 0,$$

as g is nonzero. □

It is known that a trust-region method does not strictly decrease $f(x)$ at each iteration; however, by adding a line search we ensure $f(x)$ is decreased sufficiently at every iteration. The following lemma is used to show that the line-search will converge in a finite number of iterations.

LEMMA 3.6. *Assume that s_k is obtained from Algorithm 2. Then the line-search in Algorithm 1 converges in a finite number of iterations. That is, there exists an α_k such that*

$$\rho(\alpha_k s_k) \geq 0.25 \tag{3.15}$$

Furthermore, we have:

$$m(\alpha_k s_k) \leq \alpha_k \left[s_k^T g_k + \frac{1}{2} \max(0, s_k^T H_k s_k) \right] < 0. \tag{3.16}$$

Proof. Because $m(s)$ denotes the second-order Taylor expansion of $f(x)$ at x_k, we necessarily have the following

$$\lim_{\alpha_k \to 0} \rho_k(\alpha_k s_k) = 1.$$

for any direction s_k. Thus the line-search in Algorithm 1 will find an α_k satisfying (3.15) in a finite number of iterations.
Furthermore we have

$$\begin{aligned} m(\alpha_k s_k) = \alpha_k \left[s_k^T g_k + \frac{\alpha_k}{2} s_k^T H_k s_k \right] &\leq \alpha_k \left[s_k^T g_k + \frac{\alpha_k}{2} \max(0, s_k^T H_k s_k) \right] \\ &\leq \alpha_k \left[s_k^T g_k + \frac{1}{2} \max(0, s_k^T H_k s_k) \right] \end{aligned}$$

since $\alpha_k \in (0, 1]$. From Lemma 3.5 we know that $s_k^T g_k < 0$. If $s_k^T H_k s_k \leq 0$, then

$$\alpha_k \left[s_k^T g_k + \frac{1}{2} \max(0, s_k^T H_k s_k) \right] = \alpha_k s_k^T g_k < 0.$$

If $s_k^T H_k s_k > 0$, then

$$\alpha_k \left[s_k^T g_k + \frac{1}{2} \max(0, s_k^T H_k s_k) \right] = \alpha_k m(s_k) < 0,$$

by (3.13).
Thus, (3.16) holds.□

Lemma 3.6 implies that the line search used is well defined. Now we are able to give our convergence theorem and its proof. The proof is derived using minor modifications to the proof of the trust-region convergence proof given in [21].

THEOREM 3.7. *Assume that* $\|H_k\| \le \beta$ *for some constant* β*,* f *is continuously differentiable and that that* $\mathcal{L}(x) = \{x : f(x) < f(x_0)\}$ *is bounded. Then we have*

$$\lim_{k\to\infty} \inf \|g_k\| = 0 \tag{3.17}$$

Proof. Suppose for contradiction that there is a $\epsilon > 0$ and integer K such that

$$\|g_k\| \ge \epsilon \quad \text{for all } k > K. \tag{3.18}$$

The inertia controlling parameter λ_k is either bounded or unbounded. First suppose that λ_k is unbounded. Then there exists an infinite convergent subsequence k_i satisfying $\rho(s_{k_i}) < .25$. We will show that this cannot happen. To simplify the proof, the subindex i is omitted. First note that

$$\begin{aligned} |\rho(s_k) - 1| &= \left| \frac{(f(x_k) - f(x_k + s_k) - (m_k(0) - m_k(s_k))}{m_k(0) - m_k(s_k)} \right| \\ &= \left| \frac{f(x_k + s_k) - f(x_k) - m(s_k)}{m_k(0) - m_k(s_k)} \right|. \end{aligned} \tag{3.19}$$

From the Taylor theorem, we have:

$$f(x_k + s_k) - f(x_k) = {g_k}^T s_k + \int_0^1 (\nabla f(x_k + t s_k) - g_k)^T s_k dt$$

Then

$$\begin{aligned} |f(x_k + s_k) - f(x_k) - m(s_k)| &= \left| \frac{1}{2} s_k^T H_k s_k - \int_0^1 (\nabla f(x_k + t s_k) - g_k)^T s_k dt \right| \\ &\le \frac{\beta}{2} \|s_k\|_2^2 + C(s_k)\|s_k\|, \end{aligned}$$

where $\lim_{\|s_k\|\to 0} C(s_k) = 0$. By Theorem 3.3 we then have

$$\frac{\beta}{2} \|s_k\|_2^2 + C(s_k)\|s_k\| \le \frac{n}{2\lambda_k^2} \left(\beta n + 2C(s_k)\lambda_k\right)$$

Thus using Lemma 3.4, (3.19), and (3.18) we have

$$|\rho(s_k) - 1| \le \frac{n(\beta n + 2C(s_k)\lambda_k)}{\|g_k\| \min\left(\lambda_k, \lambda_k^2 \dfrac{\|g_k\|}{\|H_k\|}\right)} \le \frac{n(\beta n + 2C(s_k)\lambda_k)}{\epsilon \min\left(\lambda_k, \lambda_k^2 \dfrac{\epsilon}{\beta}\right)}.$$

Thus

$$\lim_{\lambda_k \to \infty} |\rho(s_k) - 1| = 2nC(s_k)/\epsilon.$$

However, because the subsequence is convergent by assumption, $C(s_k) \to 0$ as $k \to \infty$. Contradicting the assumption that $\rho(s_k) < .25$ for all k in subsequence.

Now suppose that λ_k is bounded. That is, there exists an M such that $\lambda_k \leq M$ for all k. Then by the design of Algorithm 1, there must exist a subsequence $\{x_{k_i}\}$ and an integer K such that $\rho(s_{k_i}) > 0.25$ and $\alpha_{k_i} = 1$ for all $k_i > K$. Therefore, we have:

$$\begin{aligned} f(x_{k_i}) - f(x_{k_i+1}) &= f(x_{k_i}) - f(x_{k_i} + s_{k_i}) \\ &\geq \frac{1}{4}(m(0) - m(s_{k_i})) \\ &\geq \frac{1}{8}\|g_{k_i}\| \min\left(\frac{1}{\lambda_k}, \frac{\|g_{k_i}\|}{\|H_{k_i}\|}\right) \\ &\geq \frac{1}{8}\epsilon \min\left(\frac{1}{M}, \frac{\epsilon}{\beta}\right) > 0. \end{aligned}$$

This implies that $\lim_{k_i \to \infty} f(x_{k_i}) = \infty$ because $f(x_k)$ monotonically decreases. This contradicts the assumption that the level set is bounded. Therefore there cannot exist such an ϵ bound on $\|g_k\|$ and the theorem follows. □

Theorem 3.7 implies that Algorithm 1 will converge after a finite number of iterations. So far we have shown that λ_k serves as an upper bound on an implicit trust-region radius, giving Algorithm 1 convergent properties similar to a trust-region method. As in trust-region algorithm, with slightly stronger assumption we can prove the stronger results, that $\|g_k\| \to 0$ for the entire sequence.

LEMMA 3.8. *Assume that $\|H_k\| \leq \beta$ for some constant β, $\eta_k \in (0, 1/4)$, f is Lipschitz continuously differentiable, and that the level set $\mathcal{L}(x) = \{x : f(x) < f(x_0)\}$ is bounded. Then we have*

$$\lim_{k \to \infty} \|g_k\| = 0 \tag{3.20}$$

Proof. Note that a parallel result in the context of classical trust-region methods is shown in [21] in Theorem 4.8. As in the proof of Theorem 3.7, it is straightforward to perform minor modifications to adapt the proof of Theorem 4.8 to our context (with $\rho = .25$). Therefore, to avoid a nearly redundant proof, we point the interested readers to [21]. □

Next we wish to show that, if x_k is sufficiently near a local minimizer satisfying the second-order sufficient conditions hold, then λ_k in Algorithm 1 is bounded and Algorithm 2 eventually reduces to ordinary CG; that is, for k sufficiently large, condition (2.1) is always satisfied. To do so we will need the following two lemmas that will be used in proving Theorem 3.12. The first lemma shows that whenever x_k is sufficiently close to x^*, the step-size generated by Algorithm 2 is $\mathcal{O}(\|g_k\|)$.

LEMMA 3.9. *Suppose that x^* is an accumulation point of $\{x_k\}$ where x_k is obtained from Algorithm 1. Then, if the second-order sufficient conditions hold at x^*, there exists a $\delta_1 > 0$ and a constant C_1 such that when $\|x_k - x^*\| \leq \delta_1$, we have:*

$$\|s_k\| \leq C_1 \|g_k\| \tag{3.21}$$

Proof. Since the second-order sufficient conditions hold at x^*, there exists $\mu > 0$ bounding the smallest eigenvalue of H_k from below in a neighbourhood of x^*. Further, since $\hat{H}_k = H_k + E_k$ with $E_k \succeq 0$, the smallest eigenvalue of $\hat{H}_k$ is also bounded by μ in this neighborhood. Therefore, when x_k is in this neighbourhood, we have:

$$\begin{aligned} \|s_k\| &\leq \|\hat{H}_k^{-1}\|\|\hat{H}_k s_k\| \\ &\leq \frac{1}{\mu}(\|g_k\| + \|\hat{H}_k s_k + g_k\|) \\ &\leq \frac{1}{\mu}(1+\eta_k)\|g_k\| \end{aligned}$$

Thus, the lemma holds. □

In the second lemma we show that whenever x_k is sufficiently close to x^*, the distance from x_k to x^* is $\mathcal{O}(\|g_k\|_2)$. To prove this lemma we will need a slightly stronger assumption, that $f(x)$ is twice Lipschitz continuous.

LEMMA 3.10. *Suppose that x^* is an accumulation point of $\{x_k\}$ where x_k is obtained from Algorithm 1. Then, if the second-order sufficient conditions hold at x^* and $f(x)$ is twice Lipschitz continuous in an open neighborhood of x^*, there exists a $\delta_2 > 0$ and a constant C_2 such that when $\|x_k - x^*\| \leq \delta_2$, we have:*

$$\|x_k - x^*\| \leq C_2\|g_k\| \tag{3.22}$$

Proof. From the Taylor theorem, we have:

$$g_k - g^* = H^*(x_k - x^*) + \int_0^1 (\nabla^2 f(x^* + t(x_k - x^*)) - H^*)(x_k - x^*)dt \tag{3.23}$$

Similar to the proof of Lemma 3.9, there exists the smallest eigenvalue $\mu > 0$ of H in a neighbourhood of x^*. Thus, from (3.23), when x_k is the neighbourhood, we have: Therefore,

$$\begin{aligned} \|g_k\| &\geq \|H^*(x_k - x^*)\| - \left\|\int_0^1 (\nabla^2 f(x^* + t(x_k - x^*)) - H^*)(x_k - x^*)dt\right\| \\ &\geq \mu\|x_k - x^*\| - \frac{L}{2}\|x_k - x^*\|^2 \end{aligned}$$

where L is a Lipschitz constant. Thus, the above inequality becomes:

$$\|g_k\| \geq (\mu - \frac{L}{2}\|x_k - x^*\|)\|x_k - x^*\|$$

Therefore, the lemma holds for any choice of $\delta_2 < 2\mu/L$. □

For the reader's convenience, we now quote a lemma concerning convergence of an inexact Newton methods.

LEMMA 3.11 (Nocedal et al. [14]). *Consider the iteration $x_{k+1} = x_k + s_k$, where s_k satisfies*

$$\|H_k s_k + g_k\| \leq \eta_k\|g_k\|, \tag{3.24}$$

and x^* is an accumulation point of $\{x_k\}$. Suppose that H_k is positive definite at x^*, and $0 \leq \eta_k < 1$. Then, if the starting x_0 is sufficiently near x^*, the entire sequence

$\{x_k\}$ converges to x^* and the convergence is linear at least, and superlinear if η_k converges to 0.

We now show that the inertia constraint (2.1), like the radius of a trust-region, is asymptotically inactive and that λ_k is bounded from above (note this is similar to showing that the trust-region radius is bounded away from zero). Furthermore, under certain standard assumptions, convergence of Algorithm 1 is superlinear.

THEOREM 3.12. *Suppose that x^* is an accumulation point of $\{x_k\}$ where x_k is obtained from Algorithm 1. Then, if the second-order sufficient conditions hold at x^* and $f(x)$ is twice Lipschitz continuous in an open neighborhood of x^*, the following properties hold:*

- *λ_k in Algorithm 1 is bounded, and there exists an integer K such that Equation (2.1) holds for all $k > K$,*
- *The main sequence $\{x_k\}$ converges at least linearly to x^*, and superlinearly if $\eta_k \to 0$.*
- *The actual to predicted reduction ratio ρ_k converges to 1.*

Proof. The proof is divided into two steps. We will first show that the main sequence $\{x_k\}$ converges to x^* by proving that for k sufficiently large

$$\|x_k - x^*\| \leq C_2\|g_k\|.$$

That is, eventually the bounds in Equation (3.22) holds for the entire sequence. Because $\|g_k\| \to 0$ by Lemma 3.8, this will conclude the first part of the proof.

Let $\delta = \min(\delta_1, \delta_2)$ where δ_1 and δ_2 be defined as in Lemma 3.10 and Lemma 3.9. Then because of Lemma 3.8, there exists an integer K such that for all $k > K$,

$$\|g_k\| \leq \frac{\delta}{C_1 + C_2},$$

where constants C_1 and C_2 are defined from Lemma 3.10 and Lemma 3.9 respectively. Let j denote any iterate in the convergent subsequence such that $\|x_j - x^*\| < \delta$ and $j > K$. Then

$$\|x_{j+1} - x^*\| \leq \|x_j - x^*\| + \|s_j\| \leq (C_1 + C_2)\|g_j\| \leq \delta.$$

This implies two things: (1) we can now apply the theorem recursively to x_{j+1}, and (2) $\|x_{j+1} - x^*\| \leq C_2\|g_{j+1}\|$ concluding the first part of the proof.

We will now prove the remain results. Because $\nabla^2 f(x^*)$ is positive definite at x^*, there exists a constant $\mu > 0$ bounding the eigenvalues of $\nabla^2 f(x)$ from below, in an open neighborhood of x^*. A Taylor series for $f(x)$ about x_k gives

$$f(x_k + s_k) - f(x_k) = m(s_k) + o(\|s_k\|^2) \tag{3.25}$$

By design of Algorithm 2 we have the following bound for each s_k:

$$\|\hat{H}_k s_k + g_k\| \leq \eta_k \|g_k\|. \tag{3.26}$$

Since $\hat{H}_k = H_k + E_k$ with $E_k \succeq 0$, the smallest eigenvalue of $\hat{H}_k$ is always greater than or equal to the smallest eigenvalue of H_k. Thus there must exist an integer K_2 such that for all $k > K_2$ we have that $\|\hat{H}_k s_k\| \geq \mu\|s_k\|$. Therefore, because of (3.26), when k is sufficiently large,

$$\mu\|s_k\| - \|g_k\| \leq \Big|\|Hs_k\| - \|g\|\Big| \leq \|Hs_k + g\| \leq \eta_k\|g_k\|, \tag{3.27}$$

which by reordering gives

$$\|g_k\| \geq \frac{\mu\|s_k\|}{1+\eta_k}. \tag{3.28}$$

From Lemma Lemma 3.4, (3.26), and Theorem 3.3, when k is big, we have:

$$m(0) - m(s_k) \geq 0.5\|g_k\| \min\left(\frac{1}{\lambda_k}, \frac{\|g_k\|}{\|H_k\|}\right) \tag{3.29}$$

$$\geq 0.5\frac{\mu\|s_k\|}{1+\eta_k} \min\left(\frac{\|s_k\|}{n}, \frac{\mu\|s_k\|}{(1+\eta_k)\|H_k\|}\right) \tag{3.30}$$

Therefore, there exists a constant C_3 such that

$$m(0) - m(s_k) \geq C_3\|s_k\|^2 \tag{3.31}$$

Then (3.25) implies that ρ_k converges to 1 which in turn, implies λ_k is bounded, since by construction of Algorithm 1, λ_k can only be increased if $\rho_k < 1/4$. Therefore, since $\|g_k\| \to 0$, for k sufficiently large

$$\|g_k\|\lambda_k < \mu.$$

implying that

$$p^T H p > \mu\|p\|^2 > \lambda_k\|g_k\|\|p\|^2, \tag{3.32}$$

and (2.1) is thus inactive. Hence for k sufficiently large, H_k will cease to be modified, and chosen CG residual tolerance in Step 2 of Algorithm 1 implies we may apply Lemma 3.11 to obtain the remaining assertions of this theorem. □

4. Numerical Results. In this section we report numerical results for ICMCG on unconstrained CUTEr test problems [4, 16]. When using modified CG one may either explicitly store the vectors in the summation of (1.5) corresponding to nonzero α_j or rely on conjugacy and Lanczos orthogonality relations as discussed in [23, 20, 2]. In general, Lanczos orthogonality is quickly lost as soon as an eigenvalue of the Lanczos tridiagonal converges [24], and as stated in [2] we feel caution should be used when relying on such relations; in the Lanczos algorithm it can be shown that the extreme eigenvalues of the Lanczos tridiagonal quickly converge, which is ideal if one is seeking eigenvalues, but problematic if one is relying on conjugacy.

In the Appendix we show that if $\delta = \infty$ whenever $i > 0$, then Algorithm 2 converges immediate with a residual value of 0. However, choosing δ in this manner is permitted by construction in the ICMCG line-search algorithm. This implies that all existing theory still holds if ICMCG terminates the CG process whenever indefiniteness is detected, as long as at least one iteration has been completed. Hence, the ICMCG algorithm never need store any Lanczos vectors other than those necessary for running CG itself. We have found, however, that storing a small number of the Lanczos vectors explicitly (for the numerical results presented in this section a maximum of 5 vectors were stored) and exiting from the ICMCG algorithm early whenever this maximum number of matrix modifications is reached works extremely well. Because a smaller value of δ yields greater predicted decrease in the quadratic model, for these 5 vectors we always set $\delta = \delta_{\text{low}}$ in Algorithm 2. We experimented using a larger number of vectors, up to 40, and found that there was very little in

name	n	status	cpu(s)	name	n	status	cpu(s)
arwhead	5000	S/S	0/0	liarwhd	10000	S/S	0/1
brybnd	5000	S/S	1/1	nondia	9999	S/S	1/1
cosine	10000	S/S	0/1	nondquar	10000	S/S	17/4
cragglvy	5000	S/S	1/1	scosine	10000	S/S	0/1
curly10	10000	M/S	2400/19	scurly10	10000	M/S	190/4
curly20	10000	M/S	3700/20	scurly20	10000	M/S	94/4
curly30	10000	M/S	4500/44	scurly30	10000	M/S	75/4
dqdrtic	5000	S/S	0/0	sinquad	10000	S/S	2/6
dqrtic	5000	S/S	0/1	srosenbr	10000	S/S	0/1
engval1	5000	S/S	0/1	tridia	10000	S/S	0/1
freuroth	5000	S/S	0/1	woods	10000	S/S	1/2

Table 4.1

Numerical results for some a subset of the unconstrained CUTEr problems. The fourth column of each table provides corresponding CPU times (rounded to the nearest whole second) for Algorithm 1 with and without preconditioning. Here "S" stands for solved, while "M" for maximum iterations reached.

the way of performance gain, when more that 5 vectors were used in the Hessian modification per subproblem. We would like to stress that vectors corresponding to Hessian modification are heavy in the components along the small eigenvalue subspaces, and hence, over a small number of subproblems, by exploiting these directions via matrix modifications, we quickly move to a region where the Hessian is nearly positive-definite (or goes unbounded).

We applied the Algorithm 1 to all the unconstrained CUTEr test problems using a SAS translation of the CUTEr test problems. Within the SAS implementation of this algorithm, a preconditioner is used to increase the rate of convergence when near a minimizer. We have found that though a majority of the unconstrained CUTEr problems are quite amenable to CG approaches, and can be solved easily without a preconditioner, a preconditioner can ensure quick rates of convergence near a solution for all test problems. To illustrate the effects of test runs both with and without preconditioning, we provide a sample of the numerical results in Table 4. Note that in order to obtain the fast convergence rates offered by Lemma 3.11 it is necessary to solve the linear system to higher and higher degrees of accuracy. Numerically this means that the CG algorithm can hit the maximum allowed number of CG iterations prior to achieving the desired accuracy, which in turn can substantially slow down convergence. We highlight some of CUTEr problems, to demonstrate that preconditioner is sometimes (but not always) necessary to achieve fast convergence when sufficiently near a minimizer. Overall, with preconditioning turned on, we were able to solve all of the unconstrained CUTEr test problems in less than 7 minutes with the current SAS implementation of this algorithm.

5. Conclusions and Future Work. In this paper we have addressed the problem of unboundedness in the search direction when the Hessian is indefinite or near singular. We have developed a strategy that performs explicit modifications to the Hessian that have the same characteristics as the implicit ones used by classical trust region algorithms. The effect of these modifications is that the search direction is forced to remain within an implicit trust region which is defined by setting an adaptive lower bound on the smallest eigenvalue of H. This lower bound is in similar manner to the trust-region radius within trust region algorithm. We have shown that the success of the proposed approach depends on the fact that the adaptive parameter is directly proportional to the current gradient of the objective function. This is the only matrix modification strategy we know of that ensures that the modified Hessian always converges to H* (whether or not H* is singular), while the resulting intermediate step-sizes always remain bounded. Further we showed the resulting search direction lies within an implicit trust-region. From our numerical experiments we have observed that when H* is indeed singular, faster convergence rates can always be obtained when the modified Hessians are allowed to approach a singular matrix at the limit. We have demonstrated that this near singularity is benign with respect to the corresponding search direction as long as the rate by which the modified Hessians approach singularity is controlled by the rate that the current gradient g_k approaches zero.

An alternate issue, that we have found numerically relevant, though occurring with far less frequency, is the equivalence of the "hard-case" in trust-region algorithms for line-search methods. Note that, in exact arithmetic, no algorithm that constructs it search direction from the Krylov subspace

$$\mathcal{K}(g, H) = \text{span}\{g, Hg, H^2g, \ldots\}$$

can claim to handle the hard-case, when the eigenvector v_1 corresponding to the smallest eigenvalue λ_1 is orthogonal to g, i.e. an extreme example of this case is any stationary point which does not also satisfy the second-order necessary conditions for being minimizer. This is because v_1, a critically needed search direction in this case, lives in an orthogonal subspace to the subspace where the search direction is being constructed: $\mathcal{K}(g, H)$. It is our belief that it is precisely the hard-case which separates numerical performance of trust-region algorithm from what in most case should be nearly equivalent, Levenberg-Marquardt methods. In a second soon to be release sister paper, we will discuss ways to handle this second (far less critical) draw-back of line-search methods by incorporating results from paper [1] for obtaining accurate estimates of v_1 with little additional computational overhead. We have also extended the ICMCG algorithm to constrained optimization and have found that it perform equally well in this context; it our intent to further release a third paper outlining how this may done in a robust and efficient manner.

6. Appendix. In this section, we provide a theorem in the context of the modified conjugate-gradient algorithm to emphasize a key philosophical point behind the algorithm described in this paper: when indefiniteness is corrected along a single dimension, it is preferable to err on the side of the matrix modification being too large, than too small.

THEOREM 6.1. *Suppose a MCG is applied to the system $Hs = -g$ generating the sequence of vectors s_k, p_k (H-conjugate vector), and r_k. Suppose $p_j^T H p_j > 0$ for $j < k$, but $p_k^T H p_k \leq 0$. If we define $H_\delta = H + \delta r_k r_k^T$, then*

$$\lim_{\delta \to \infty} H_\delta(s_k + \alpha(\delta) p_k) \to -g,$$

where $\alpha(\delta) = (r_k^T r_k)/p_k^T H_\delta p_k$ denotes the corresponding CG weight. Then as $\delta \to \infty$

$$\|H_\delta\| \to \infty \text{ while } s_k + \alpha(\delta) p_k \to s_k.$$

Thus the large modification to H has little incremental effect on the current search direction. However as δ approaches its minimum modification value

$$\delta_{\min} = -(p_k^T H p_k)/(r_k^T r_k)^2$$

then

$$\lim_{\delta \to \delta_{\min}} s_k + \alpha(\delta) p_k \to \infty.$$

Thus a small modification to H corresponds to a large modification to s_k.

Proof. First note that CG gives us the following two relations: $p^T r = r^T r$ and $r^T s = 0$ because of Lemma 3.1. Next note that

$$\lim_{\delta \to \infty} \alpha(\delta) = \lim_{\delta \to \infty} \frac{r^T r}{p^T H p + \delta (r^T p)^2} = \lim_{\delta \to \infty} \frac{r^T r}{p^T H p + \delta (r^T r)^2} = 0.$$

Expanding $H_\delta (s + \alpha(\delta) p) + g$ we get

$$\begin{aligned} H_\delta (s + \alpha(\delta) p) + g &= Hs + g + \alpha(\delta) Hp + \delta (r^T s) r + \delta \alpha(\delta) (r^T p) r \\ &= \alpha(\delta) Hp + \left[-r + \delta \alpha(\delta) (r^T r) r \right] \end{aligned}$$

However,

$$\lim_{\delta \to \infty} \delta \alpha(\delta) = \frac{\delta r^T r}{p^T H p + \delta (r^T r)^2} = \frac{1}{r^T r}.$$

And, since $\lim_{\delta \to \infty} \alpha(\delta) Hp = 0$, we have that

$$\lim_{\delta \to \infty} H_\delta (s + \alpha(\delta) p) + g = 0 + [-r + r] = 0.$$

However, when δ approaches its minimum modification value

$$\delta_{\min} = -(p_k^T H p_k)/(r_k^T r_k)^2$$

then, $H_\delta (s_k + \alpha(\delta) p_k)$ approach a singular matrix

$$\lim_{\delta \to \delta_{\min}} s_k + \alpha(\delta) p_k = \lim_{\delta \to \delta_{\min}} -H_\delta^{-1} (s_k + \alpha(\delta) p_k) g_k \to \infty.$$

□

The first unexpected property states that in the context of modified CG, whenever indefiniteness is detected, given an $\epsilon > 0$ there always exists a sufficient large modification matrix E_k such that the next residual vector r_{k+1} satisfies

$$\|r_{k+1}\| \leq \epsilon.$$

An important result of this theorem is understanding that the size of the residual vector, while crucial to convergence in the positive-definite case, can be made arbitrarily small for any CG iteration where indefiniteness is detected. Moreover, as this modified residual vector goes to 0, the corresponding modified CG search direction s_{k+1} converges to s_k. Ultimately this means that a small residual in the indefinite case is not necessarily a good indicator that s_k will be a good search direction. This theorem thus helps highlight the crux of the problem with current line-search modification strategies and points towards a new goal: determine a modification matrix E_k so that $H_k + E_k$ is positive-semidefinite and

$$\beta_1 \|g\| \leq (H_k + E_k)^\dagger g \leq \beta_2 \|g\|,$$

for some appropriate choice of $\beta_1, \beta_2 \geq 0$.

This theorem also provides insight into why small Hessian modification can be detrimental in the context of line-search algorithms. Let us consider

$$s_{k+1} = s_k + \alpha(\delta) p_k$$

from Theorem 6.1. Suppose that we wish to use s_{k+1} as our search direction. Further suppose that we are sufficiently close to a global minimizer of $f(x)$ so that $f(x_k) < f(x)$ for all $\|x - x_k\| \geq 1$. This implies that for a line-search algorithm to achieve sufficient reduction in $f(x)$ at the current point x_k, we must have that the search direction s_{k+1} is scaled so that

$$\|\gamma s_{k+1}\| = \|\gamma s_k + \alpha(\delta) p_k\| \leq 1.$$

Since $\|\alpha(\delta) p_k\| \to \infty$ as $\delta \to \delta_{\min}$ by Theorem 6.1, we must have the $\gamma \to 0$ as $\delta \to \delta_{\min}$. This implies that in the limit, for arbitrarily small modifications,

$$s_{k+1} \approx \eta p_k,$$

for some η providing sufficient decrease. Thus we are numerically discarding all previous conjugate gradient vectors, and placing all our emphasis on a single dimension. Alternatively, as $\delta \to \infty$ $s_{k+1} = s_k$, which is arguably a much preferred search direction by Lemma 3.2 it denotes the unconstrained minimizer of the quadratic model with the span of conjugate-gradient vectors $\{p_0, \ldots, p_k\}$.

7. Acknowledgments. The authors profusely thank Manoj Chari, Tao Huang, Trevor Kearney and Aysegul Peker for many insightful discussions, and continual support for research and development of this work.

REFERENCES

[1] I. Akrotirianakis, J. Griffin, and W. Zhou, *Simultaneous iterative solutions for the trust-region and minimum eigenvalue subproblems*, SAS Technical Report 2009-02, 2009.

[2] M. Arioli, T. F. Chan, I. Duff, N. Gould, and J. Reid, *Computing a search direction for large-scale linearly-constrained nonlinear optimization calculations*, tech. report, 1993.

[3] C. Baker, P. Absil, and K. A. Gallivan, *An implicit turst-region method on riemannian manifolds*, Tech. Report FSU-SCS-2007-449, Florida State University, 2007.

[4] I. Bongartz, A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *CUTE: Constrained and unconstrained testing environment*, Report 93/10, Département de Mathématique, Facultés Universitaires de Namur, 1993.

[5] 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.

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

[7] J. B. Erway and P. E. Gill, *An interior method for computing a trust-region step*, Numerical Analysis Report 08-1, Department of Mathematics, University of California San Diego, La Jolla, CA, 2008.

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

[9] R. Fletcher, *Practical Methods of Optimization*, John Wiley and Sons, Chichester and New York, second ed., 1987.

[10] ———, *Practical methods of optimization*, Wiley-Interscience [John Wiley & Sons], New York, 2001.

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

[12] E. M. Gertz and P. E. Gill, *A primal-dual trust-region algorithm for nonlinear programming*, Math. Program., Ser. B, 100 (2004), pp. 49–94.
[13] P. E. Gill and J. Kroyan, *Trust-search algorithms for unconstrained optimization*, Numerical Analysis Report NA 04-1, University of California San Diego, 2004.
[14] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London and New York, 1981.
[15] 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.
[16] 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.
[17] J. D. Griffin, *Interior-point methods for large-scale nonconvex optimization*, PhD thesis, Department of Mathematics, University of California San Diego, March 2005.
[18] J. Kroyan, *Trust-Search Algorithms for Unconstrained Optimization*, PhD thesis, Department of Mathematics, University of California San Diego, February 2004.
[19] L. Luksan, C. Matonoha, and J. Vlcek, *A shifted steihaug-toint method for computing a trust-region step*, Tech. Report V914-04, Institute of Computer Science, Academy of Sciences of the Czech Republic, 2004.
[20] S. G. Nash, *Newton-type minimization via the Lanczos method*, SIAM J. Numer. Anal., 21 (1984), pp. 770–788.
[21] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.
[22] J. Nocedal and Y.-x. Yuan, *Combining trust region and line search techniques*, in Advances in Nonlinear Programming (Beijing, 1996), vol. 14 of Appl. Optim., Kluwer Acad. Publ., Dordrecht, 1998, pp. 153–175.
[23] D. P. O'Leary, *A discrete Newton algorithm for minimizing a function of many variables*, Math. Program., 23 (1982), pp. 20–33.
[24] C. C. Paige, *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*, PhD thesis, University of London, 1971.
[25] 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.
[26] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.