Mathematical Programming 26 (1983) 190–212
North-Holland Publishing Company

TRUNCATED-NEWTON ALGORITHMS FOR LARGE-SCALE UNCONSTRAINED OPTIMIZATION*

Ron S. DEMBO

School of Organization and Management, Yale University, New Haven, CT 06520, U.S.A.

Trond STEIHAUG

Department of Mathematical Sciences, Rice University, Houston, TX 77001, U.S.A.

Received 14 April 1982
Revised manuscript received 20 July 1982

We present an algorithm for large-scale unconstrained optimization based on Newton's method. In large-scale optimization, solving the Newton equations at each iteration can be expensive and may not be justified when far from a solution. Instead, an inaccurate solution to the Newton equations is computed using a conjugate gradient method. The resulting algorithm is shown to have strong convergence properties and has the unusual feature that the asymptotic convergence rate is a user specified parameter which can be set to anything between linear and quadratic convergence. Some numerical results on a 916 variable test problem are given. Finally, we contrast the computational behavior of our algorithm with Newton's method and that of a nonlinear conjugate gradient algorithm.

Key words: Unconstrained Optimization, Modified Newton Methods, Conjugate Gradient Algorithms.

1. Introduction

Consider the unconstrained optimization problem

$$\underset{x\in R^n}{\text{minimize}} \quad f(x) \tag{1.1}$$

where $f: R^n \to R$ is a nonlinear function with the following properties:

(1) f is twice continuously differentiable.

(2) For all $x_0 \in R^n$ the level sets

$$L(x_0) \triangleq \{x: f(x) \leq f(x_0)\} \tag{1.2}$$

are bounded.

We wish to find a local minimizer x^*, that is, a point x^* for which there exist

* This research was supported in part by DOT Grant CT-06-0011, NSF Grant ENG-78-21615 and grants from the Norwegian Research Council for Sciences and the Humanities and the Norway–America Association.

This paper was originally presented at the TIMS–ORSA Joint National Meeting, Washington, DC, May 1980.

$\delta > 0$ so that

$$f(x)^* \leq f(x) \quad \text{for all } x: \|x - x^*\| < \delta. \tag{1.3}$$

If the Hessian matrix at x^*, $H(x^*)$, is positive definite, then we refer to x^* as a strong local minimizer. It is the only local minimizer in a neighborhood of x^* and corresponds to a strict inequality in (1.3) for $x \neq x^*$.

A first-order necessary condition for a local minimum of $f(x)$ is that

$$g(x) = 0$$

where $g: R^n \to R^n$ is the gradient vector of f. An important method for solving this system of nonlinear equations is Newton's method which, given an initial guess x_0, computes a sequence of steps $\{p_k\}$ and iterates $\{x_k\}$ as follows:

FOR $k = 0$ STEP 1 UNTIL convergence DO

$$\text{Solve} \quad H(x_k)p_k = -g(x_k) \tag{1.4}$$

$$\text{Set} \quad x_{k+1} = x_k + p_k \tag{1.5}$$

We will refer to (1.4) as the Newton equation and to p_k as the Newton step or Newton direction.

Newton's method is important because it provides a standard with which to compare rapidly convergent methods for solving (1.1); one way of characterizing superlinear[1] convergence is that the computed step should approach the Newton step asymptotically in both magnitude and direction [8].

The positive and negative aspects of Newton's method have been well documented in the literature. Briefly, on the positive side, the algorithm is locally and quadratically convergent provided that f is sufficiently smooth. That is, for x_0 sufficiently close to a strong local minimizer x^* there exists a constant c so that

$$\|x_{k+1} - x^*\| \leq c\|x_k - x^*\|^2. \tag{1.7}$$

Its drawbacks, however, are significant in a practical setting. They are:

(i) The method is not globally convergent.[2]

(ii) It is not defined at points where $H(x_k)$ is singular.

(iii) For nonconvex problems it does not necessarily generate a sequence of descent directions (that is, directions satisfying $g(x_k)^T p_k < 0$).

(iv) An n-dimensional linear system (1.4) must be solved at each iteration.

With the exception of (iv) above, these difficulties may all be overcome by

[1] A sequence $\{x_k\}$ is said to converge superlinearly to x^* if

$$\lim_{k \to \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0. \tag{1.6}$$

[2] By this we mean that it does not converge to a local solution from an arbitrary starting point, x_0.

embedding the method in the following modified Newton framework (see [14] for example).

Given a starting guess x_0;

major iteration
- Compute $f(x_k)$, $g(x_k)$ and $H(x_k)$
- Test for convergence
- minor iteration
  - Caculate p_k so that
    $$\bar{H}_k p_k = -g(x_k) \tag{1.8}$$
    where $\bar{H}_k$ is some positive definite approximation to $H(x_k)$
- Calculate some λ_k satisfying
  $$f(x_k + \lambda_k p_k) \leq f(x_k) + \alpha \lambda_k g(x_k)^T p_k; \quad \alpha \in \left(0, \frac{1}{2}\right) \tag{1.9}$$
  $$g(x_k + \lambda_k p_k)^T p_k \geq \beta g(x_k)^T p_k; \quad \beta \in (\alpha, 1) \tag{1.10}$$
  set $x_{k+1} = x_k + \lambda_k p_k$
- $k \leftarrow k + 1$

We will refer to (1.9) and (1.10) as conditions guaranteeing 'sufficient descent' sometimes referred to as the Goldstein–Armijo conditions [8]. An alternative condition to (1.10) that does not involve gradient evalulations is

$$f(x_k + \lambda_k p_k) \geq f(x_k) + \beta \lambda_k g(x_k)^T p_k \tag{1.11}$$

where $\beta \in (\frac{1}{2}, 1)$. If the sequence of matrices $\{\bar{H}_k\}$ has a uniformly bounded condition number, then (1.9) and (1.10) or (1.9) and (1.11) guarantee that $\{g(x_k)\}$ will converge to zero [8]. Many modified Newton methods can also be shown to have stronger convergence properties, namely that the Hessian is positive semidefinite at all limit points of $\{x_k\}$ (see [10]).

The major iteration serves to modify the method so as to control its global behavior. It plays a significant role in the early stages of the computation. The minor iteration, on the other hand, assumes an important role when close to a solution x^* since, in the vicinity of a strong minimizer, $H(x_k)$ is positive definite and the computed search direction p_k is the Newton direction. Furthermore, a stepsize $\lambda_k = 1$ is acceptable in this region [8], and the algorithm possesses the same asymptotic rate as Newton's method.

Since the benefits of the Newton direction are mainly local (i.e., in the vicinity of a solution), there appears to be no justification for expending the effort

required to get an accurate solution to the modified Newton equations (1.8) when far from a local minimizer. This points to the main deficiency in modified Newton methods: the computational burden of solving the Newton or modified Newton equations when far from a solution.

In a large-scale setting one is forced to solve the Newton equations using an iterative method with low storage overhead. Therefore, in this environment, there is a direct trade-off between the amount of work required to compute a search direction and the accuracy with which the Newton equations are solved.

A natural and scale independent measure of this accuracy is the relative residual, $\|r_k\|/\|g(x_k)\|$, where if p_k is the computed direction, then r_k is given by

$$r_k = H(x_k)p_k + g(x_k). \tag{1.12}$$

The norm $\|\cdot\|$ is any vector norm in R^n.

In this paper we describe a globally convergent algorithm that is based on truncating the conjugate gradient method applied to the Newton equations when $\|r_k\|/\|g_k(x_k)\|$ is 'small enough'. Hence the name Truncated-Newton method. In Section 2 we describe a Truncated-Newton algorithm, show that it is globally convergent and specify how small $\|r_k\|/\|g(x_k)\|$ has to be to achieve a prespecified convergence rate. However, by solving (1.8) by the conjugate gradient method we lose the property that the Hessian is positive semidefinite at all limit points of $\{x_k\}$. In Section 3 we discuss extensions and alternative methods for the minor iteration. Some computational results are presented in Section 4 where the method is compared with a nonlinear conjugate gradient method [24].

2. Truncated-Newton methods

Newton's method is based on approximating the function $f(x_k + p)$ by the quadratic model

$$\phi_k(p) = f(x_k) + g(x_k)^T p + \tfrac{1}{2} p^T H(x_k) p.$$

The Newton direction is obtained from an exact solution to $\min_p \phi_k(p)$. Since a conjugate direction algorithm has the property that ϕ is minimized over the subspace spanned by the directions that are generated, it is ideally suited to computing a Truncated-Newton direction.

Consider the Truncated-Newton method, defined by the minor iteration below, that uses a conjugate-gradient (CG) algorithm [1, 15, 16] to compute a search direction. We have named this the TNCG algorithm. For the sake of clarity we omit the major iteration, which is assumed to be identical to that in Section 1, and suppress the major iteration subscript k. The unsubscripted vectors g and p, matrix H and scalar η take on the major iteration subscript and should be read as $g(x_k)$, p_k, $H(x_k)$ and η_k.

Minor Iteration

Step 1: Set $p_0 = 0$, $r_0 = -g$, $d_0 = r_0$,
$\delta_0 = r_0^T r_0$, $i = 0$

Step 2: Set $q_i = Hd_i$

$$\textit{If } d_i^T q_i \leq \epsilon(\delta_i) \textit{ exit: } p = \begin{cases} d_0 & \text{if } i = 0 \\ p_i & \text{otherwise} \end{cases} \tag{2.1}$$

else continue with Step 3

Step 3: $\alpha_i = r_i^T r_i / d_i^T q_i$

$p_{i+1} = p_i + \alpha_i d_i$

$r_{i+1} = r_i - \alpha_i q_i$

$$\textit{If } \|r_{i+1}\| / \|g\| \leq \eta \textit{ exit: } p = p_{i+1} \tag{2.2}$$

else continue with Step 4

Step 4: $\beta_i = r_{i+1}^T r_{i+1} / r_i^T r_i$

$d_{i+1} = r_{i+1} + \beta_i d_i$

$\delta_{i+1} = r_{i+1}^T r_{i+1} + \beta_i^2 \delta_i$

$i \leftarrow i + 1$

Return to Step 2.

The sequence $\{\eta_k\}$ is called a *forcing sequence* and condition (2.2) is referred to as Truncated-Newton termination. A direction p_k satisfying (2.1) or (2.2) is called a Truncated-Newton direction.

There are three different ways in which the TNCG minor iteration can terminate.

Case i: The gradient vector, g, points in a direction of negative curvature i.e., $g^T H g < 0$. In this case the minor iteration returns with $p = d_0 = -g$, the steepest descent direction.

Case ii: A direction of negative curvature is encountered in the CG iteration (i.e., $d_i^T H d_i < 0$) prior to satisfying the Truncated-Newton termination criterion (2.2). For reasons mainly having to do with stability of the method, the CG iteration is terminated and the current estimate p_i is used. We will show that p_i is a direction of descent. Other choices could be the steepest descent direction $-g$, or the direction with negative curvature d_i. It can be shown that d_i is a descent direction [25].

Case iii: The algorithm terminates with the Truncated-Newton criterion (2.2). We will show that this case always occurs in the vicinity of a strong local minimizer provided ϵ is sufficiently small.

In practice, an additional safeguard (such as limiting the number of CG iterations) is necessary to account for roundoff. It is easily shown that $\delta_i = d_i^T d_i$ so (2.1) guarantees stability and sufficient positive curvature.

Each iteration of the CG algorithm requires one matrix vector product, and $5n$ multiplications for computing scalar vector and vector inner-products. In addition to H, storage is required for the four vectors p, r, d and q. We note that the TNCG algorithm never requires the Hessian matrix explicitly. What is required is the product $q = H(x)d$ where d is a direction vector generated within the conjugate gradient method. In many applications, such as finite element analysis, constrained optimization, or the numerical example in Section 4, the matrix-vector multiplication can be performed without 'assembling' the Hessian[1], thereby reducing storage requirements.

The conjugate gradient minor iteration possesses some important properties which link major and minor iterations and are important to the robustness and speed of Truncated-Newton algorithms. They also provide some clues as to what alternate iterative methods might be used.

Property 1. The sequence $\{\|g(x_k)\|\}$ should converge to zero.

Property 2. A stepsize $\lambda_k = 1$ should guarantee sufficient descent in the vicinity of a strong local minimizer.

This property is an assurance that sufficiently close to a solution the major iteration will no longer be needed to monitor convergence and the local properties of the minor iteration will take over.

Property 3. As $x_k \to x^*$ the algorithm should exhibit a rapid rate of convergence, usually superlinear or better.

Property 4. If $x_k \to x^*$, then $H(x^*)$ should be positive semidefinite.

The results below show that the TNCG algorithm possesses the Properties 1, 2 and 3. The proof of the results are in the Appendix.

Theorem 2.1. *The sequence of iterates $\{x_k\}$ is well defined and*

$$\lim_{k\to\infty} \|g(x_k)\| = 0.$$

If x^* is a limit point of $\{x_k\}$ and $H(x^*)$ is positive definite, then $x_k \to x^*$.

Since TNCG is a descent method and the level set, $L(x_0)$, is compact we know that the sequence $\{x_k\}$ has limit points where $g \equiv 0$. So, if one of these limit points is a strong local minimizer, then the sequence $\{x_k\}$ converges.

Theorem 2.2. *Let $x_k \to x^*$ where $H(x^*)$ is positive definite. Then, for ϵ sufficiently small there exists an index k_0 such that the termination criterion (2.2) is satisfied and $\lambda_k = 1$ is acceptable for all $k \geq k_0$.*

Thus, for $k \geq k_0$, the Truncated-Newton algorithm reduces to the following Inexact Newton method [5].

FOR $k = k_0$ STEP 1 UNTIL convergence DO

Find p_k satisfying

$$H(x_k)p_k = -g(x_k) + r_k, \quad \text{where } \frac{\|r_k\|}{\|g(x_k)\|} \leq \eta_k \tag{2.3}$$

Set $x_{k+1} = x_k + p_k$.

The importance of the next theorem lies in its constructive nature. That is, it indicates precisely how to construct a Truncated-Newton algorithm that will possess any *prescribed* order of convergence between 1 and 2.

Theorem 2.3. [5]. *Let $x_k \to x^*$ where $H(x^*)$ is positive definite and assume that H is Lipschitz continuous at x^*. Then*

(i) *$x_k \to x^*$ superlinearly if and only if $\|r_k\|/\|g(x_k)\| \to 0$ as $k \to \infty$;*

(ii) *$x_k \to x^*$ with order[3] $(1+t)$ if and only if*

$$\limsup_{k\to\infty} \frac{\|r_k\|}{\|g(x_k)\|^{1+t}} < \infty.$$

By choosing

$$\eta_k = \min\{1/k, \|g(x_k)\|^t\} \tag{2.4}$$

for some $0 < t \leq 1$, we have a Truncated-Newton method of order $1+t$. This forcing sequence will result in an adaptive algorithm for solving the unconstrained optimization problem (1.3). When far from a solution $\|g(x_k)\|$ is large and hence very little work is required to compute a direction satisfying the Truncated-Newton termination condition. As $g(x_k) \to 0$ which implies $\eta_k \to 0$ thereby forcing p_k closer to the Newton direction in both magnitude and direction. The forcing sequence (2.4) is used in the numerical experiments in Section 4. However, it should be pointed out that the efficiency depends on the specific choice of forcing sequence and in general, the best sequence depends on the problem.

[3] A sequence x_k is said to converge to x^* with order $(1+t)$ if

$$\limsup_{k\to\infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|^{1+t}} < \infty.$$

In the notation of Ortega and Rheinboldt [21] this would be a sequence of Q-order *at least* $1+t$.

The behavior and theoretical properties of the conjugate gradient minor iteration depend on the initial residual, $-g(x)$. If started at a stationary point the algorithm will terminate. However, the theorem below shows that it may be modified to move away from stationary points that are not local minimizers except in rare instances (that is, when $g(x)$ is orthogonal to the eigenspaces of H corresponding to nonpositive eigenvalues). The modification entails the use of the direction of negative curvature, d_i in case (ii) above.

Theorem 2.4. *Let* $\eta_k = \epsilon = 0$ *in* (2.1) *and* (2.2). *If* $H(x_k)$ *is not positive definite, then*

either $d_i^T H(x_k) d_i \leq 0$ *for some minor iteration* i

or $g(x_k)$ *is orthogonal to the eigenspaces corresponding to the nonpositive eigenvalues.*

The above result therefore states that if $x_k \to x^*$ and we terminate using (2.2), then, except in the rare case above, x^* will be a strong local minimizer.

3. Extensions

3.1. *Truncated-Newton with finite differencing*

In many practical situations, it may place too large a burden on the user to require a subroutine that computes the Hessian matrix. There are three different modifications to the TNCG algorithm that overcome this difficulty.

The first approach is to replace the Hessian by a sparse finite difference approximation that attempts to minimize the number of gradient evaluations [4, 22]. Let $\bar{H}_k$ be the approximate Hessian at x_k. Then Steihaug [25] shows that the method will exhibit a superlinear rate of convergence if both η_k and $\|\bar{H}_k - H(x_k)\|$ converge to zero and will have order $1 + t$ if

$$\|\bar{H}_k - H(x_k)\| \leq c\|g(x_k)\|^t$$

with the forcing sequence (2.4).

A second approach is based on the observation that the matrix vector product $H(x)d$, where d is a direction generated within the conjugate gradient method, can be approximated using the finite difference formula

$$H(x)d \simeq \frac{1}{\sigma}[g(x + \sigma d) - g(x)] = q \tag{3.1}$$

where σ is chosen appropriately. This obviates the need for storing $H(x)$ (and therefore requires less storage than TNCG) but requires an additional gradient evaluation for each minor iteration. Garg and Tapia [11] also have used this idea in a different context.

If we choose

$$\alpha_i = d_i^T r_0 / d_i^T q_i \quad i \geq 0 \tag{3.2}$$

and provided that $d_i^T r_0 > 0$, then we can prove the first part of Theorem 2.1 (see Appendix). This result is similar to Shanno's [23] for his nonlinear conjugate gradient method. Unfortunately, not much is known about the convergence characteristics of conjugate gradient methods with (3.1) replacing $H(x)d$. Although, in many cases (see Section 4) with a proper choice of σ we expect the method to be almost indistinguishable from TNCG.

The above approach places Truncated-Newton methods in direct competition with nonlinear conjugate gradient algorithms [19, 24], which are currently considered to be the only available methods for truly large-scale problems. This is because Truncated-Newton methods with finite differencing have very similar storage and the same informational (f and g) requirements as nonlinear conjugate gradient algorithms, and convergence can be shown under the same conditions.

Lastly, when the Hessian is not available but there is sufficient storage for a sparse finite-difference approximation to it, we advocate a combination of the above two approaches. At iterate x_k we will determine which approach to use dependent on the relative cost and accuracy needed.

(1) Preprocess to determine the number of gradient-evaluations required to approximate the Hessian. Denote this by NG.

(2) Approximate Hd using (3.1) until the expected number of minor iterations exceeds NG.

(3) At this stage switch to using a sparse finite-difference approximation of the Hessian.

Such a hybrid algorithm has the same properties as TNCG. Theorems 2.1 and 2.2 hold provided ϵ and σ_k are sufficiently small. Superlinear convergence follows if η_k and $\sigma_k \to 0$.

3.2. *Alternative minor iterations*

In problems where some a-priori information on the structure of the problem is available, the use of preconditioning in the conjugate gradient algorithm can often speed up convergence considerably [1, 2]. The modifications required to introduce preconditioning into the TNCG algorithm are minor [1, 2, 25]. The preconditioned algorithm will, in general, require more storage than TNCG but, with a positive-definite preconditioning matrix with a uniformly bounded condition number, it satisfies the theorems in Section 2.

In the *conjugate residual method* the 2-norm of the residual decreases monotonically [2]. Thus, it is an attractive alternative to CG in view of the Truncated-Newton termination criterion which depends on $\|r\|$. However, numerical experiments indicated that the direction p_k has to be scaled so that

$r_k^T p_k$ is sufficiently small to ensure that $\lambda_k = 1$ is acceptable in the limit. Also, a stability condition similar to (2.1) must be introduced [3, 9, 17]. An attractive conjugate residual method is the one presented in [3].

4. Computational results

The results of this section are preliminary findings based on a limited amount of testing and are in no way conclusive statements regarding the behavior of the algorithms tested. We prefer that the results be read as indications of the possible limitations and potential benefits of the various methods.

One of the difficulties we encountered while testing was the lack of test problems for large-scale optimization. For this reason we created a family of large test problems derived from a water distribution system simulator. The test problems are generated in a manner similar to those in [26] and are fully documented in [7]. For the purposes of this presentation we consider two 916 variable problems both of which are strictly convex. Problem I is well-conditioned, $\text{cond}(H(x^*)) = 20$ and Problem II is mildly ill-conditioned, $\text{cond}(H(x^*)) \simeq 10^4$. The norm $\|\cdot\|$ is the standard Euclidean vector norm.

The following four algorithms were compared on the above problems.

(i) Newton's method, which was implemented using a CG algorithm to solve the Newton equations. In all cases, the minor iteration was terminated when

$$\|r_k\|/\|g_k\| \leq 10^{-10}.$$

(ii) A Truncated-Newton method (TNCG) with the minor iteration termination condition (2.4) with $t = 1$ (i.e., quadratically convergent).

(iii) A Truncated-Newton method (DTNCG) as in (ii) above but using the finite difference formula (3.1) to approximate $H(x)d$. The finite difference stepsize, σ, was chosen as:

$$\sigma = \frac{\sqrt{\text{machine precision}}}{\|d\|} = 10^{-8}/\|d\|. \tag{4.1}$$

(iv) CONMIN, a state-of-the-art nonlinear conjugate gradient code [24].

The codes for NEWTON and TNCG differed essentially by a single IF statement which is used to terminate the minor iteration. The linesearch algorithm used by CONMIN differed from that used by NEWTON, TNCG and DTNCG which was based on [12]. In order to make the various codes comparable the same 'sufficient descent' criteria were used to terminate linesearch. In all cases a step λ_k was accepted when

$$f(x_k + \lambda_k p_k) \leq f(x_k) + 0.0001\, \lambda_k g_k^T p_k,$$
$$|g(x_k + \lambda_k p_k)^T p_k| \leq 0.9\, |g(x_k)^T p_k|.$$

Note that the second condition gives a slightly more restrictive steplength than is

needed for the global convergence of these algorithms. Since the test problems are all strictly convex, termination (2.1) was never used.

All codes were written in FORTRAN IV and all runs were made in double precision on a DEC 20/60 computer using the TOPS20 operating system (machine precision $= 1E-16$).

4.1. *Newton vs. Truncated-Newton* (TNCG)

Computational results of Newton vs. Truncated-Newton are presented in Tables 1 and 2 and Figs. 1 and 2. This experiment is one of those rare instances where there is an excellent measure for comparing the two algorithms, that is, the total number of minor iterations required to achieve a specified optimality tolerance. This is because the number of major iterations for the two algorithms is similar, the number of function and gradient values is approximately the same and their storage and informational requirements are identical.

For both problems the asymptotic rate of convergence is approximately quadratic yet TNCG solves the problem in a number of minor iterations comparable to obtaining the first Newton direction. For the ill-conditioned Problem II, TNCG is roughly eight times faster than NEWTON.

Fig. 1 and 2 indicate the number of minor iterations that were required in order to reduce $\|g\|$ by an order of magnitude over the course of the computation. The TNCG algorithm required roughly the same amount of work to reduce $\|g\|$ by an order of magnitude at all stages of the computation. Newton's method on the other hand was significantly worse than Truncated-Newton when $\|g\|$ was large and, as expected, the asymptotic behavior of the two methods was very similar.

Table 1
Newton vs. TNCG on Problem I. Major iteration termination $\|g\| \leq 1 \cdot E-7$, $\text{cond}(H(x^*)) \simeq 20$

	Newton		Truncated-Newton	
Major Iteration Number	Cumulative minor iterations	$\|g\|$	Cumulative minor iterations	$\|g\|$
0	0	$0.469E+2$	0	$0.469E+2$
1	35	$0.806E+1$	1	$0.128E+2$
2	74	0.639	3	$0.586E+1$
3	113	$0.573E-2$	6	$0.136E+1$
4	150	$0.809E-6$	11	0.210
5	183	$0.497E-13$	17	$0.204E-1$
6			26	$0.227E-3$
7			43	$0.404E-7$

Key: For each algorithm, there is one Hessian evaluation per major iteration. Newton's method required a total of 5 function and 5 gradient evaluations. Truncated-Newton required a total of 7 function and 7 gradient evaluations.

Table 2
Newton vs. TNCG on Problem II. Major iteration termination $\|g\| \leq 1 \cdot E-7$, $\text{cond}(H(x^*)) \simeq 10^4$

	Newton		Truncated-Newton	
Major Iteration Number	Cululative minor iterations	$\|g\|$	Cumulative minor iterations	$\|g\|$
0	—	$0.402E+4$	—	$0.402E+4$
1	770	$0.360E+4$	1	$0.216E+4$
2	1512	$0.327E+4$	33	$0.202E+4$
3	2270	$0.278E+4$	71	$0.220E+4$
4	2999	$0.291E+4$	106	$0.304E+4$
5	3588	$0.185E+3$	138	$0.255E+4$
6	4248	$0.418E+3$	149	$0.745E+3$
7	4862	$0.210E+3$	205	$0.508E+3$
8	5486	$0.109E+3$	262	$0.454E+3$
9	6125	$0.232E+2$	309	$0.229E+3$
10	6722	$0.601E+1$	358	$0.324E+2$
11	7290	0.736	437	$0.668E+1$
12	7838	$0.146E-1$	497	$0.116E+1$
13	8379	$0.595E-5$	569	$0.984E-1$
14	8916	$0.994E-12$	651	$0.697E-2$
15	—	—	806	$0.486E-4$
16	—	—	1114	$0.207E-8$

Key: Each algorithm required one Hessian evaluation per major iteration. Newton's method required a total of 20 function and 20 gradient evaluations; TNCG required 24 function and 24 gradient evaluations.

4.2. *Truncated-Newton with and without finite differencing*

By comparing Tables 1 and 2 with the results in the second column in Tables 3 and 4 it is clear that on these test problems Truncated-Newton with differencing (DTNCG) was almost indistinguishable from TNCG, in which the actual Hessian was used. The total number of major and minor iterations and function evaluation is very close for the two methods. The trade-off is simply one Hessian evaluation for each outer iteration versus one extra gradient evaluation for each inner iteration.

A number of runs were made in which the differencing parameter was varied from $\sigma = 10^{-8}/\|d\|$ to $\sigma = 10^{-2}/\|d\|$ in order to test the effect on DTNCG. As expected, the rate of convergence deteriorated progressively as σ got larger, however, the algorithm converged.

4.3. CONMIN *vs. Truncated-Newton with differencing* (DTNCG)

On both problems DTNCG significantly outperforms CONMIN on two counts; the final accuracy that it is able to achieve and the total number of

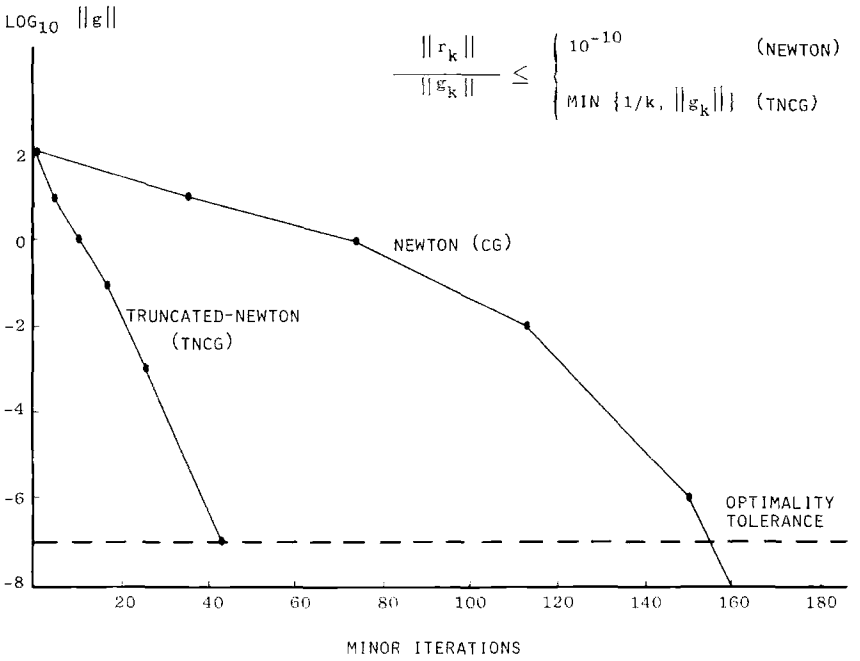

Fig. 1. Truncated-Newton vs. Newton (Problem I).

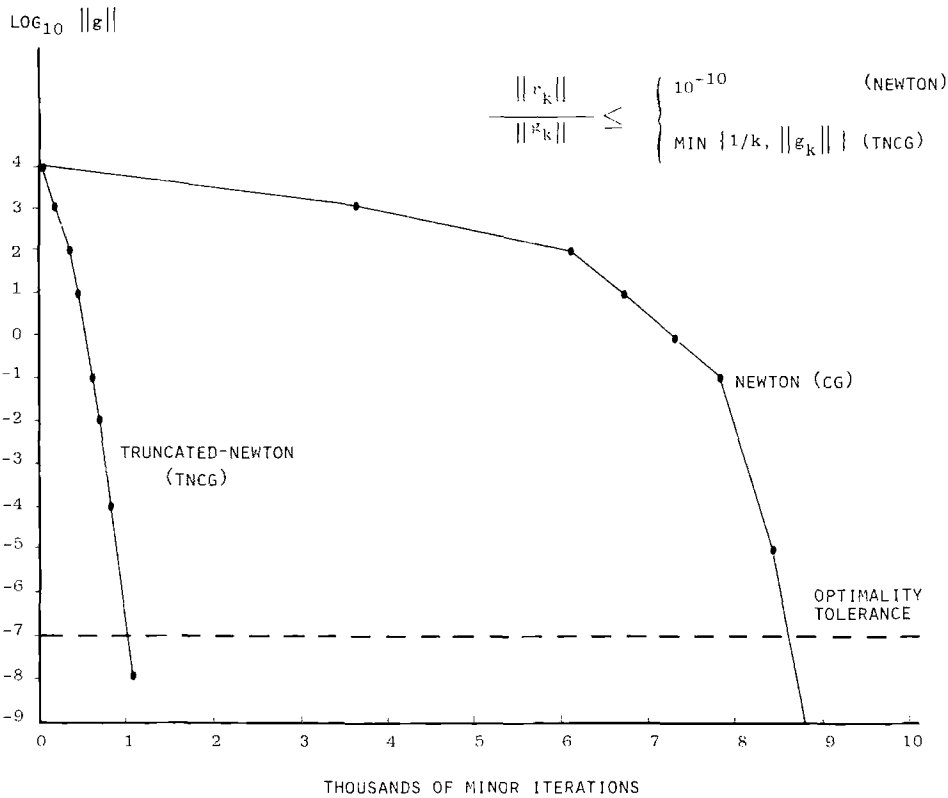

Fig. 2. Truncated-Newton vs. Newton (Problem II).

Table 3
CONMIN vs. DTNCG on Problem I. Major iteration termination $\|g\| \leq 1 \cdot E-7$, cond $(H(x^*)) \simeq 20$

Cumulative minor iterations for DTNCG; Cumulative iterations for CONMIN	CONMIN			Truncated-Newton (DTNCG) with finite difference (3.1)		
	Cumulative function evaluations	Cumulative gradient evaluations	$\|g\|$	Cumulative function evaluations	Cumulative gradient evaluations	$\|g\|$
0	1	1	$0.469E+2$	1	1	$0.469E+2$
1	3	3	$0.139E+2$	2	3	$0.128E+2$
3	7	7	$0.532E+1$	3	6	$0.586E+1$
6	13	13	$0.122E+1$	4	10	$0.136E+1$
11	23	23	0.130	5	16	0.210
17	35	35	$0.714E-2$	6	23	$0.204E-1$
26	53	53	$0.621E-4$	7	33	$0.227E-3$
43	87	87	$0.874E-7$	8	51	$0.404E-7$

Key: The entries in the table were chosen to correspond the major iterations of the DTNCG algorithm.

Table 4
CONMIN vs. DTNCG on Problem II.* Major iteration termination $\|g\| \leq 1 \cdot E-7$, $\text{cond}(H(x^*)) \simeq 10^4$

Cumulative minor iterations for DTNCG; Cumulative iterations for CONMIN	CONMIN			Truncated-Newton (DTNCG) with finite difference (3.1)		
	Cumulative function evaluations	Cumulative gradient evaluations	$\|g\|$	Cumulative function evaluations	Cumulative gradient evaluations	$\|g\|$
0	1	1	$0.402E+4$	1	1	$0.402E+4$
1	3	3	$0.228E+4$	2	3	$0.216E+4$
33	68	68	$0.198E+4$	4	37	$0.202E+4$
71	144	144	$0.339E+3$	6	77	$0.220E+4$
106	215	215	$0.158E+3$	8	114	$0.304E+4$
138	280	280	$0.613E+2$	10	148	$0.255E+4$
149	302	302	$0.458E+2$	11	160	$0.745E+3$
206	416	416	$0.801E+1$	13	219	$0.508E+3$
265	535	535	$0.185E+1$	15	280	$0.454E+3$
312	631	631	0.365	17	329	$0.229E+3$
363	733	733	0.124	18	381	$0.321E+2$
447	903	903	$0.171E-1$	19	466	$0.650E+1$
512	1033	1033	$0.248E-2$	20	532	$0.112E+1$
587	1183	1183	$0.209E-3$	21	608	$0.900E-1$
673(662)**	1333	1333	$0.241E-4$	22	695	$0.577E-2$
840	—	—	—	23	863	$0.313E-4$
1180	—	—	—	24	1210	$0.961E-9$

Key: *The entries in the table were chosen to correspond to major iterations of the DTNCG algorithm.
** The number in parenthesis refers to the cumulative number of iterations at which CONMIN terminated.

Table 5

	Problem I		Problem II	
	DTNCG	CONMIN	DTNCG	CONMIN
Function evaluations	7	87	22	1333
Gradient evaluations	50	87	863	1333
Accuracy $\|g\|$	$0.4E-7$	$0.8E-7$	$0.3E-4$	$0.2E-4$

function and gradient evaluations to achieve a given accuracy. For both problems the total number of function and gradient evaluations using DTNCG was less than half the number required by CONMIN to reach the same accuracy. On the ill-conditioned Problem II, CONMIN could not reduce $\|g\|$ below $0.2E-4$ whereas DTNCG has no difficulty meeting the optimality criterion $\|g\| \leq 1 \cdot E-7$. The comparison is summarized in Table 5.

5. Conclusion

In recent years a large body of literature has developed around large scale nonlinear unconstrained optimization. Most of the existing methods are based on extending a (preconditioned) conjugate gradient method to non-quadratic problems. The approach here, however, is based on Newton's method and applying a (linear) conjugate gradient method to the Newton equation. The special case with finite difference approximation of the matrix vector multiplication was introduced by Garg and Tapia [11], however, their CG constants α_i and β_i and termination rule are different. This approach is also used in [15] and in conjunction with Lanczos' method [13, 20] to 'solve' the Newton equations.

In summary, the numerical results indicate that, if the Hessian is available, and is relatively inexpensive to compute. Truncated-Newton algorithms may significantly outperform nonlinear conjugate gradient methods. When it is not, they indicate that Truncated-Newton with differencing may be a viable alternative to nonlinear conjugate gradient algorithms. When sufficient storage is available for the Hessian, a hybrid algorithm such as the one discussed in Section 3.1 will probably be far more efficient than one that uses a difference approximation for $H(x)d$. Certainly, more experimentation with a wider variety of problems is needed before any firmer conclusions may be drawn.

The results of this paper extend in a natural way to large-scale linearly-constrained nonlinear programs (see for example [6, 18]).

Acknowledgements

We are indebted to Stan Eisenstat for his many useful comments.

An earlier draft of this report was written while the first author was visiting the Naval Postgraduate School in the summer of 1980. Their support is gratefully acknowledged.

Appendix

Let $\epsilon > 0$ and define

$$\phi(p) = g^T p + \tfrac{1}{2} p^T H p. \tag{A.1}$$

The space spanned by the vectors x and y is denoted by $[x, y]$.

Theorem A.1. *If* $d_i^T H d_i > \epsilon \delta_i$, $i = 0, 1, \ldots, k$, *then*

$$d_i^T H d_j = 0, \qquad i \neq j, \quad i, j = 0, 1, \ldots, k, \tag{A.2}$$

$$d_i^T r_j = 0, \qquad i < j, \quad i, j = 0, 1, \ldots, k+1 \tag{A.3}$$

$$r_i^T d_j = r_j^T r_j = r_0^T d_j, \quad i \leq j, \quad i, j = 0, 1, \ldots, k, \tag{A.4}$$

$$[d_0, d_1, \ldots, d_k] = [g, Hg, \ldots, H^{k-1} g], \tag{A.5}$$

$$\phi(p_{k+1}) = \min\{\phi(p): p \in [d_0, \ldots, d_k]\}, \tag{A.6}$$

$$\delta_i = d_i^T d_i, \quad i = 0, 1, \ldots, k. \tag{A.7}$$

For a proof of these conditions, see, for instance Hestenes and Stiefel [16] or Steihaug [25]. The property (A.6) is the basic motivation for choosing a conjugate gradient method for the minor iterations. It follows from (A.5) that the number of iterations is bounded by the number of distinct eigenvalues.

Let p be the direction computed in the minor iteration.

Lemma A.2. *There exist positive constants* γ_0 *and* γ_1 *that only depend on* $\|H\|$ *and* ϵ *so that*

$$g^T p \leq -\gamma_0 \|g\|^2 \tag{A.8}$$

and

$$\|p\| \leq \gamma_1 \|g\|. \tag{A.9}$$

Proof. Assume that $p = p_i$, $i \geq 1$. From Step 3 in the Conjugate Gradient method

we have

$$p = p_i = \sum_{j=0}^{i-1} \alpha_j d_j. \tag{A.10}$$

From (A.4) we have that

$$\alpha_j = \frac{d_j^T r_0}{d_j^T q_j} \tag{A.11}$$

hence using (A.10) and (A.11)

$$p^T g = -p_i^T r_0 = -\sum_{j=0}^{i-1} \frac{(d_j^T r_0)^2}{d_j^T q_j} \leq -\frac{d_0^T r_0}{d_0^T q_0} d_0^T r_0.$$

But $d_0 = r_0$ and

$$\frac{d_0^T r_0}{d_0^T q_0} = \frac{d_0^T d_0}{d_0^T H d_0} \geq \frac{1}{\|H\|}.$$

So for

$$\gamma_0 \equiv \min\{1, \frac{1}{\|H\|}\}$$

we have the desired result (A.8). Also, note from (A.10) and (A.11) that

$$p_i = \sum_{j=0}^{i-1} \frac{d_j^T r_0}{d_j^T q_j} d_j = \left[\sum_{j=0}^{i-1} \frac{1}{d_j^T q_j} d_j d_j^T\right] r_0.$$

Hence

$$\|p\| = \|p_i\| \leq \left[\sum_{j=0}^{i-1} \frac{d_j^T d_j}{d_j^T q_j}\right] \|r_0\| < i \frac{1}{\epsilon} \|r_0\| = i \frac{1}{\epsilon} \|g\|$$

using (2.1). Hence for

$$\gamma_1 = \max\{n\epsilon^{-1}, 1\}$$

we have the desired result (A.9) and (A.7) follows directly from the definition of d_i and (A.3).

We note that this lemma remains true when q_j is determined by finite differencing and α_j given in (3.2).

We can now prove the first part of Theorem 2.1.

Theorem A.3. *The sequence of iterates* $\{x_k\}$ *is well defined and*

$$\lim_{k \to +\infty} \|g(x_k)\| = 0.$$

Proof. From (A.8) in Lemma A.2, we have that p is a descent direction. Hence we

know [8] that there exists λ_k that satifies (1.9) and (1.10) and

$$\lim_{k\to\infty} \frac{g(x_k)^T p_k}{\|p_k\|} = 0.$$

From Lemma A.1 we have

$$\frac{g(x_k)^T p_k}{\|p_k\|} \le -\gamma_0 \frac{\|g(x_k)\|^2}{\|p_k\|} \le -\frac{\gamma_0}{\gamma_1}\|g(x_k)\| \le 0$$

using (A.8) and (A.9), and we have the desired result.

We will now prove a generalization of Theorem 6.4 of Dennis and Moré [8].

Theorem A.4. *Let $\{x_k\}$ be a sequence converging to x^* at which $H(x^*)$ is positive definite. If p_k is a descent direction, λ_k is chosen to satisfy* (1.9) *and* (1.10) *or* (1.11) *and*

$$\lim_{k\to\infty} \frac{p_k^T(g(x_k) + H(x_k)p_k)}{p_k^T p_k} = 0, \tag{A.12}$$

then there exists an index k_0 such that $\lambda_k = 1$ is admissible for $k \ge k_0$.

Proof. Let 2γ be the smallest eigenvalue of $H(x^*)$ and

$$\delta \equiv \frac{1}{3}\gamma \min\{1-2\alpha, 2\beta - 1\}. \tag{A.13}$$

Since $H(x^*)$ is positive definite, $\alpha < 1/2$ and $\beta > 1/2$, then $\gamma > 0$ and $\delta > 0$. Choose $\epsilon > 0$ so that

$$\|H(z) - H(x)\| \le \delta, \tag{A.14}$$

$$\|g(z) - g(x) - H(x)(z-x)\| \le \delta\|x - z\| \tag{A.15}$$

and

$$p^T H(x) p \ge \gamma\|p\|^2 \quad \text{for all } p \in R^n \tag{A.16}$$

for all x, z so that $\|x - x^*\| \le \epsilon$ and $\|z - x^*\| \le \epsilon$. This can be done since H is continuous and positive definite at x^*. Let k_0 be an index so that

$$|p_k^T(g(x_k) + H(x_k)p_k)| \le \delta\|p_k\|^2 \tag{A.17}$$

for all $k \ge k_0$. This can be done in view of (A.12).

$$\begin{aligned} p_k^T g(x_k) &= p_k^T(g(x_k) + H(x_k)p_k) - p_k^T H(x_k) p_k \\ &\le -(\gamma - \delta)\|p_k\|^2 \end{aligned} \tag{A.18}$$

using (A.17) and (A.16). Hence $\|p_k\| \le (\gamma - \delta)^{-1}\|g(x_k)\|$ so for k sufficiently large

$$\|x_k - x^*\| \le \epsilon \quad \text{and} \quad \|x_k + p_k - x^*\| \le \epsilon. \tag{A.19}$$

From the mean value theorem we have for some $u_k = x_k + t_k p_k$, $0 \le t_k \le 1$,

$$\begin{aligned} f(x_k + p_k) - f(x_k) &= g(x_k)^T p_k + \frac{1}{2} p_k^T H(u_k) p_k \\ &= \frac{1}{2} p_k^T g(x_k) + \frac{1}{2} p_k^T (g(x_k) + H(x_k) p_k) + \frac{1}{2} p_k^T (H(u_k) - H(x_k)) p_k \\ &\le \frac{1}{2} p_k^T g(x_k) + \frac{1}{2} \delta \|p_k\|^2 + \frac{1}{2} \delta \|p_k\|^2 \\ &= \frac{1}{2} p_k^T g(x_k) + \delta \|p_k\|^2 \end{aligned}$$

using (A.17), (A.19) and (A.14). Hence using (A.18) and (A.13) and from the choice of δ

$$\begin{aligned} f(x_k + p_k) - f(x_k) - \alpha g(x_k)^T p_k &\le \left(\frac{1}{2} - \alpha\right) g(x_k)^T p_k + \delta \|p_k\|^2 \\ &\le \left[-\left(\frac{1}{2} - \alpha\right)(\gamma - \delta) + \delta \right] \|p_k\|^2 \le 0 \end{aligned}$$

which is the desired result (1.9). By the same arguments, it follows that for $\beta \in (\frac{1}{2}, 1)$

$$\begin{aligned} f(x_k + p_k) - f(x_k) - \beta g(x_k)^T p_k &\ge \frac{1}{2} p_k^T g(x_k) - \delta \|p_k\|^2 - \beta g(x_k)^T p_k \\ &= -(\beta - 1/2) g(x_k)^T p_k - \delta \|p_k\|^2 \\ &\ge [(\beta - 1/2)(\gamma - \delta) - \delta] \|p_k\|^2 > 0 \end{aligned}$$

from the choice of δ, which is the desired result (1.11). Further, using (A.15) and (A.17)

$$\begin{aligned} g(x_k + p_k)^T p_k &= (g(x_k + p_k) - g(x_k) - H(x_k) p_k)^T p_k + p_k^T (g(x_k) + H(x_k) p_k) \\ &\ge -\delta \|p_k\|^2 - \delta \|p_k\|^2. \end{aligned} \tag{A.20}$$

Finally, by the choice of $\delta \le 1/3\beta\gamma$ and $\beta \le 1$ and using (A.20) we have

$$g(x_k + p_k)^T p_k - \beta g(x_k)^T p_k \ge [-2\delta + \beta(\gamma - \delta)] \|p_k\|^2 \ge 0$$

which is the desired result (1.10).

Theorem 2.2 follows now from (A.3) and Theorem A.3, since for k sufficiently large $H(x_k)$ is positive definite, hence the conjugate gradient methods terminate with (2.2) and

$$p^T (g(x) + H(x) p) = p_i^T r_i = \sum_{j=0}^{i-1} \alpha_j r_i^T d_j = 0.$$

To show the second part of Theorem 2.1, let $\{x_{k_i}\}$ be the subsequence converging to x^*. From Theorem A.4 we know that $\lambda_{k_i} = 1$ is admissible. But

from Lemma A.1, we have

$$\|p_j\| \leq \gamma_1 \|g(x_j)\|,$$

hence for k_i sufficiently large $x_{k_i+1} + p_{k_i+1}$ is sufficiently close to x^* so we may apply Theorem A.4 and we have that $x_k \to x^*$ [21, NR 14.1–3].

The second part of Theorem 2.1 remains valid in the finite difference approximation of the Hessian. However, it only remains valid in the finite difference approximation of $H(x)d$ when we can establish an upper bound on λ.

Let $(\cdot, \cdot)$ be the standard inner product on R^n, i.e.,

$$(x, y) = \sum_{i=1}^{n} \epsilon_i \eta_i \quad \text{where } x = (\epsilon_1, \ldots, \epsilon_n)^T, \quad \text{and } y = (\eta_1, \ldots, \eta_n)^T$$

and let $E(\lambda, H)$ be the eigenspace corresponding to eigenvalue λ of H, i.e.,

$$E(\lambda, H) = \{v: Hv = \lambda v\} \tag{A.21}$$

where H is a symmetric matrix.

Theorem A.5. *Let g have a nonzero projection on each eigenspace and let k be the number of distinct eigenvalues of H. If*

$$(d_i, Hd_j) = 0 \quad \textit{for } i \neq j,\ i = 0, 1, \ldots, k-1. \tag{A.22}$$

$$(d_i, Hd_i) > 0 \quad \textit{for } i = 0, 1, \ldots, k-1,$$

$$[d_0, \ldots, d_{k-1}] = [g, Hg, \ldots, H^{k-1}g], \tag{A.23}$$

then H is positive definite.

Proof. Since g has a nonzero projection on each eigenspace, and H is symmetric there exist, for $1 \leq i \leq k$, nonzero $v_i \in E(\lambda_i, H)$ so that

$$g = \sum_{i=1}^{k} v_i. \tag{A.24}$$

Let P_j be the polynomial

$$P_j(x) = (x - \lambda_1) \cdots (x - \lambda_{j-1})(x - \lambda_{j+1}) \cdots (x - \lambda_k). \tag{A.25}$$

Then, using (A.24), (A.25) and that $v_j \neq 0$

$$P_j(H)g = \sum_{i=1}^{k} P_j(H)v_i = \sum_{i=1}^{k} P_j(\lambda_i)v_i = P_j(\lambda_j)v_j \neq 0. \tag{A.26}$$

From (A.23), there exist $\alpha_i(j)$ so that

$$P_j(H)g = \sum_{i=0}^{k-1} \alpha_i(j) d_i. \tag{A.27}$$

Since $P_j(H)g \neq 0$, not all $\alpha_i(j)$, $i = 0, \ldots, k-1$, can be zero. Hence we have using (A.26), (A.27) and (A.22)

$$\begin{aligned}\lambda_j(P_j(H)g, P_j(H)g) &= (P_j(H)g, HP_j(H)g)\\ &= \left(\sum_{i=0}^{k-1} \alpha_i(j)d_i, H\sum_{i=0}^{k-1} \alpha_i(j)d_i\right)\\ &= \sum_{i=0}^{k-1} \{\alpha_i(j)\}^2(d_i, Hd_i).\end{aligned}$$

But $(d_i, Hd_i) > 0$ and $(P_j(H)g, P_j(H)g) > 0$ hence $\lambda_j > 0$.

From Theorem A.5, (A.2) and (A.5) we have that if $g(x)$ is not orthogonal to any eigenspace of $H(x)$ and $r = 0$, then $H(x)$ is positive definite. Hence, if $H(x)$ is not positive definite, then either $d_i^T Hd_i \leq 0$ for some i or $r = 0$ and $g(x)$ is orthogonal to some eigenspaces. To show that these are the eigenspaces for the nonpositive eigenvalues, let k in Theorem A.5 be the number of eigenspaces $g(x)$ meets. The result now follows directly.

References

[1] O. Axelsson, "Solution of linear systems of equations: Iterative methods", in: V.A. Barker, ed., *Sparse matrix techniques*. (Springer-Verlag, New York, 1976) pp. 1–51.

[2] R. Chandra, "Conjugate gradient methods for partial differential equations", Ph.D. dissertation, Yale University (New Haven, CT, 1978). Also available as Department of Computer Science Research Report No. 129.

[3] R. Chandra, S.C. Eisenstat and M.H. Schultz, "The modified conjugate residual method for partial differential equations", in: R. Vichnevetsky, ed., *Advance in computer methods for partial differential equations II*, Proceedings of the Second International Symposium on Computer Methods for Partial Differential Equations, Lehigh University, Bethlehem, PA (International Association for Mathematics and Computers in Simulation, June 1977) pp. 13–19.

[4] A.R. Curtis, M.J.D. Powell and J.K. Reid, "On the estimation of sparse Jacobian matrices", *Journal of the Institute of Mathematics and its Applications* **13** (1974) 117–119.

[5] R.S. Dembo, S.C. Eisenstat and T. Steihaug, "Inexact Newton methods", *SIAM Journal of Numerical Analysis* **19** (1982) 400–408.

[6] R.S. Dembo and J.G. Klincewicz, "A scaled reduced gradient algorithm for network flow problems with convex separable costs", *Mathematical Programming Studies* **15** (1981) 125–147.

[7] R. S. Dembo and T. Steihaug, "A test problem for large-scale unconstrained minimization", School of Organization and Management, Yale University (New Haven, CT) Working Paper Series B (in preparation).

[8] J.E. Dennis Jr. and J.J. Moré, "Quasi-Newton methods, motivation and theory", *SIAM Review* **19** (1977) 46–89.

[9] R. Fletcher, "Conjugate gradient methods for indefinite systems", in: G.A. Watson, ed., *Numerical Analysis*, Proceedings of Biennial Conference, Dundee, Scotland, 1975 (Springer-Verlag, New York, 1976) pp. 73–89.

[10] R. Fletcher, *Unconstrained optimization* (John Wiley and Sons, New York, 1980).

[11] N.K. Garg and R.A. Tapia, "QDN: A variable storage algorithm for unconstrained optimization", Technical Report, Department of Mathematical Sciences, Rice University (Houston, TX, 1977).

[12] P.E. Gill and W. Murray, "Safeguarded steplength algorithm for optimization using descent methods", Technical Report NPL NA 37, National Physical Laboratory (1974).

[13] P.E. Gill, W. Murray and S.G. Nash, "A conjugate-gradient approach to Newton-type methods", presented at ORSA/TIMS Joint National Meeting (Colorado Springs, November 1980).

[14] P.E. Gill, W. Murray and M.H. Wright, *Practical optimization* (Academic Press, New York, 1981).

[15] M.R. Hestenes, *Conjugate direction methods in optimization* (Springer-Verlag, New York, 1980).

[16] M.R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems", *Journal of Research of the National Bureau of Standards* 49 (1952) 409–436.

[17] D.G. Luenberger, "Hyperbolic pairs in the method of conjugate gradients", *SIAM Journal on Applied Mathematics* 17 (1969) 1263–1267.

[18] B. Murtagh and M. Saunders, "Large-scale linearly constrained optimization", *Mathematical Programming* 14 (1978) 41–72.

[19] J. Nocedal, "Updating Quasi-Newton matrices with limited storage", *Mathematics of Computation* 35 (1980) 773–782.

[20] D.P. O'Leary, "A discrete Newton algorithm for minimizing a function of many variables", *Mathematical Programming* 23 (1982) 20–33.

[21] J.M. Ortega and W.C. Rheinboldt, *Iterative solution of nonlinear equations in several variables* (Academic Press, New York, 1970).

[22] M.J.D. Powell and Ph.L. Toint, "On the estimation of sparse Hessian matrices", *SIAM Journal on Numerical Analysis* 16 (1979) 1060–1074.

[23] D.F. Shanno, "On the convergence of a new conjugate gradient method", *SIAM Journal on Numerical Analysis* 15 (1978) 1247–1257.

[24] D.F. Shanno and K.H. Phua, "Algorithm 500: Minimization of unconstrained multivariate functions", *Transactions on Mathematical Software* 2 (1976) 87–94.

[25] T. Steihaug, "Quasi-Newton methods for large scale nonlinear problems", Ph.D. dissertation, Yale University (New Haven, CT, 1980). Also available as Working Paper, Series B No. 49.

[26] Ph.L. Toint, "Some numerical results using a sparse matrix updating formula in unconstrained optimization", *Mathematics of Computation* 32 (1978) 839–851.