SIAM J. OPTIMIZATION
Vol. 1, No. 3, pp. 333–357, August 1991

A CURVILINEAR SEARCH USING TRIDIAGONAL SECANT UPDATES FOR UNCONSTRAINED OPTIMIZATION

J. E. DENNIS, JR.[†], N. ECHEBEST[‡], M. T. GUARDARUCCI[‡], J. M. MARTÍNEZ[§], H. D. SCOLNIK[¶], AND C. VACCHINO[‡]

Abstract. The idea of doing a curvilinear search along the Levenberg–Marquardt path $s(\mu) = -(H+\mu I)^{-1}g$ always has been appealing, but the cost of solving a linear system for each trial value of the parameter μ has discouraged its implementation. In this paper, an algorithm for searching along a path which includes $s(\mu)$ is studied. The algorithm uses a special inexpensive QT_cQ^T to QT_+Q^T Hessian update which trivializes the linear algebra required to compute $s(\mu)$. This update is based on earlier work of Dennis and Marwil and Martínez on least-change secant updates of matrix factors. The new algorithm is shown to be local and q-superlinearly convergent to stationary points, and to be globally q-superlinearly convergent for quasi-convex functions. Computational tests are given that show the new algorithm to be robust and efficient.

Key words. unconstrained optimization, trust regions, curvilinear search, Levenberg–Marquardt, factor updating, least change secant methods

AMS(MOS) subject classification. 65K

1. Introduction.

In this paper, we consider iterative methods for solving the smooth unconstrained minimization problem:

$$\min_x f(x); \qquad f:\Omega\subset \mathbb{R}^n \to \mathbb{R}; \qquad f\in C^1(\Omega)\ ,$$

for Ω open in $\mathbb{R}^n$. We denote $g(x) = \nabla f(x)$ for all $x \in \Omega$. We will use the ℓ_2 norm whenever another norm is not indicated.

Our methods are based on the common notion of choosing a trial step from the current iterate x_c to the next iterate x_+ based on a local quadratic model of $f(x_c+s) - f(x_c)$ of the form:

$$(1)\qquad q_c(s) \equiv g_c^T s + \frac{1}{2}s^T H_c s, \quad \text{where } g_c = \nabla f(x_c) \quad \text{and} \quad H_c = H_c^T.$$

Our methods belong to a class often called curvilinear search methods, and the curvilinear path we search along is the same one in $\mathbb{R}^n$ from which the trust-region method based on the same model would choose its step. The major difference from trust-region methods is that, even if we eventually choose the same trial step, we do our search based on the "Levenberg–Marquardt" parameter rather than on the length of the step. Methods based on other curvilinear paths have been published, but since none are in general use, we omit any comparative discussion. Most relevant is that Schramm and

[*] Received by the editors January 11, 1990; accepted for publication (in revised form) January 4, 1991.

[†] Mathematical Sciences Department, Rice University, Houston, Texas 77251-1892. This work was begun under a Fulbright Fellowship to Argentina. This research was partially supported by Air Force Office of Scientific Research grants AFOSR-89-0363, DOE/ER/25017-3, DAAL03-90-0093, and National Science Foundation grant DMS-8903751.

[‡] Departamento de Matemática, Universidad de La Plata, La Plata, Buenos Aires, Argentina.

[§] Universidade Estadual de Campinos, Campinos, Brasil. This work was done while the author visited the Mathematical Sciences Department, Rice University, and was supported by a fellowship from FAPESP, Brasil.

[¶] Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Argentina.

Zowe [11] in their B-T algorithm for nonsmooth optimization search the analogous curve.

The key to the practicality of the particular method we test is that we build the local model (1) in a form that trivializes the linear algebra needed to compute any trial step along the search path. For example, standard approaches would require a Cholesky factorization at each trial step, but we need only solve a tridiagonal system and do two matrix-vector products.

This paper is organized as follows: §2 contains a global convergence analysis in which we assume that the sequence of model Hessians is bounded, but we do not specify how the Hessians are to be chosen. We define the set from which a trial step must be chosen that satisfies an Armijo criterion. We show that there are steps in the set that satisfy the sufficient decrease criterion, but we do not specify how the step is to be found.

In §3, we assume that $\nabla^2 f$ is Lipschitz continuous on Ω, and we present a new least-change secant method for defining H_+ from H_c and apply the results of §2 to the resulting algorithm. This method is in the spirit of [2], [7], and [5] in that there is never any need to form H_+. Instead, H_c is held in the form $Q_0 T_c Q_0^T$, Q_0 orthogonal, T_c tridiagonal; and $H_+ = Q_0 T_+ Q_0^T$ is defined by doing a sparse symmetric secant update of T_c to get T_+.

In §4, we validate the new update by giving a local convergence analysis of the corresponding full step quasi-Newton method to stationary points of f. In §5, we add a convexity assumption on f and prove that the particular method from §3 that always tries the Newton step first when H_c is positive definite is globally q-superlinearly convergent. This order of convergence result is no better than we could prove if we did not do the updates, but the updates cost a low multiple of n, and they are certainly worthwhile computationally, as is shown in §7.3. Section 6 discusses an implementation and §7 gives some numerical results for a particular method from §3.

2. The general algorithm: Global convergence.

In this section we state a general algorithm of the type studied here. We make the algorithm only as specific as necessary to prove a global convergence result.

Given $x \in \Omega$, H a symmetric $n \times n$ matrix, $\lambda_1 = \lambda_1(H)$ the smallest eigenvalue of H, V_1 the corresponding eigenspace, we define a curve parameterized by *mu*:

$$\Gamma_1(x, H) = \{x - (H + \mu I)^{-1} g(x) : 0 \leq \mu > -\lambda_1\} .$$

If $g(x) \notin V_1^\perp$, or if $\lambda_1 > 0$, we define $\Gamma(x, H) = \Gamma_1(x, H)$. Otherwise, we choose $v \in V_1$, $v \neq 0$, and we define a curve parameterized by *mu*:

$$\Gamma(x, H) = \Gamma_1(x, H) \cup \Gamma_2(x, H) ,$$

where

$$\Gamma_2(x, H) = \{x - (H - \lambda_1 I)^+ g(x) + \mu v : \mu \in \mathbb{R}\} .$$

The following lemma, which follows from Gay [4] and Moré and Sorensen [8], gives a geometrical meaning to $\Gamma(x, H)$. It shows that if $\lambda_1 \leq 0$ and if $g(x) \in V_1^\perp$, then any $v \in V_1$ gives the same result for the quadratic. In our implementation, we always choose trial steps that stand in the same relation to the current iterate that z has to x in the hypotheses of the lemma. However, we have no need to be so specific in order to prove global convergence in the next section.

LEMMA 2.1. *Let $x \in \Omega$, $z \in \Gamma(x, H)$. Then z is a minimizer of*

$$q(w) = \tfrac{1}{2}(w-x)^T H(w-x) + g(x)^T(w-x) \quad \textit{subject to } \|w-x\| \le \|z-x\| ,$$

and the direction from x to z is a descent direction for q. Furthermore, assume $z \in \Gamma_1(x, H)$; then z is the unique minimizer. If $0 < \delta \le \|z-x\|$, then there is a unique $w \in \Gamma(x, H)$ such that $\|w-x\| = \delta$. Also, $w \in \Gamma_1(x, H)$.

Proof. This is just a slight restatement of a standard result of Gay[4] and Sorensen. For example, see Lemma 2.3 of Moré and Sorensen [8]. □

The following algorithm describes the way of obtaining a new approximation x_+ to the minimizer of f, starting from a current approximation $x_c \in \Omega$ such that $g_c \neq 0$ and using a current Hessian approximation H_c. A large positive number Δ is used to bound the steplength, and Δ_c and $\bar{\Delta}_c$ are constants needed in the convergence proof. The algorithm parameters $\alpha \in (0, \frac{1}{2}), \beta \in (0,1)$ are used to guarantee sufficient decrease. We use $\alpha = 10^{-4}$ and $\beta =$ macheps.

ALGORITHM 2.1.

```
Given $H_c, x_c$;
If $\lambda_1(H_c) \le 0$; Then $\Delta_c = \bar{\Delta}_c = \Delta$;
   Else $s_c^N = -H_c^{-1} g(x_c)$; $\Delta_c = \bar{\Delta}_c = \min\{\Delta, 1/\beta \|s_c^N\|\}$;
Set $\bar{x} = x_c$;
While ($\bar{x} = x_c$ or $f(\bar{x}) > f(x_c) + \alpha g(x_c)^T(\bar{x} - x_c)$) DO
        Choose $\bar{x} \in \Gamma(x_c, H_c)$ such that $\beta^2 \Delta_c \le \|\bar{x} - x_c\| \le \Delta_c$;
        $\Delta_c = \Delta_c/2$;
   ENDO;
   Set $x_+ = \bar{x}$;
```

Remark. Obviously, the efficiency of Algorithm 2.1 depends on the way $\bar{x}$ is selected. "Choose" is a very ambiguous word that we use deliberately to show that many strategies are possible.

Let us now prove that, given x_c, H_c, with $g_c = g(x_c) \neq 0$, Algorithm 2.1 is always able to finish by finding a point $\bar{x}$ which satisfies the sufficient decrease condition

$$f(\bar{x}) \le f_c + \alpha g_c^T(\bar{x} - x_c). \tag{2}$$

THEOREM 2.2. *After a finite number of DO loop executions, Algorithm 2.1 obtains a point $\bar{x} = x_+$ that satisfies (2).*

Proof. We only need to prove that, if $\|\bar{x} - x_c\|$ is small enough and $\bar{x} \in \Gamma(x_c, H_c)$, then (2) is satisfied. Using Lemma 2.1, it is easy to see that

$$\lim_{\substack{\bar{x} \to x_c \\ \bar{x} \in \Gamma(x_c, H_c)}} \frac{\bar{x} - x_c}{\|\bar{x} - x_c\|} = \lim_{\substack{\bar{x} \to x_c \\ \bar{x} \in \Gamma_1(x_c, H_c)}} \frac{\bar{x} - x_c}{\|\bar{x} - x_c\|} = \frac{-g(x_c)}{\|g(x_c)\|} , \tag{3}$$

since if $\|\bar{x} - x_c\|$ is small enough, then $\bar{x} \in \Gamma_1(x_c, H_c)$. Therefore, using (3) and the Mean Value Theorem, we have

$$\frac{f(\bar{x}) - f(x_c)}{\|\bar{x} - x_c\|} = \frac{g(x_c + \xi(\bar{x} - x_c))^T(\bar{x} - x_c)}{\|\bar{x} - x_c\|} \quad \text{with } \xi \in (0,1) .$$

Hence,

$$\lim_{\substack{\bar{x}\to x_c\\ \bar{x}\in\Gamma(x_c,H_c)}} \frac{f(\bar{x})-f(x_c)}{\|\bar{x}-x_c\|} = g(x_c)^T \lim_{\substack{\bar{x}\to x_c\\ \bar{x}\in\Gamma_1(x_c,H_c)}} \frac{\bar{x}-x_c}{\|\bar{x}-x_c\|} = -\|g(x_c)\|$$

$$\leq -\alpha\|g(x_c)\| \leq +\alpha\frac{g_c^T(\bar{x}-x_c)}{\|\bar{x}-x_c\|}$$

for any $\bar{x}\neq x_c$, and the required result follows from this inequality. □

We now give a result that we need to prove global convergence of Algorithm 2.1.

LEMMA 2.3. *Assume that* $\|H_k\|\leq B$ *for* $k=0,1,2,\cdots$ *and* $\lim_{k\to\infty} x_k = x_*$ *with* $g(x_*)\neq 0$. *Let* $\{\bar{x}_k\}$ *be any sequence such that* $\bar{x}_k\in\Gamma(x_k,H_k)$, $\lim_{k\to\infty}\|\bar{x}_k-x_k\|=0$. *Then there exists a subsequence* $\{\bar{x}_{k_j}-x_{k_j}\}$ *such that, for this subsequence,*

$$\lim_{j\to\infty}\frac{\bar{x}_{k_j}-x_{k_j}}{\|\bar{x}_{k_j}-x_{k_j}\|} = \frac{-g(x_*)}{\|g(x_*)\|}\ .$$

Proof. Let $\{H_k\}_{k\in K_1}$ be a convergent subsequence of $\{H_k\}$. Then for some H,

$$\lim_{k\in K_1} H_k = H\ , \qquad \|H\|\leq B\ .$$

For $k\in K_1$ let us write

$$H_k = Q_k D_k Q_k^T\ , \tag{4}$$

where $D_k=\operatorname{diag}(\lambda_1(H_k),\cdots,\lambda_n(H_k))$, $\lambda_1(H_k)\leq\cdots\leq\lambda_n(H_k)$. By the continuity property of eigenvalues (see Wilkinson [12, p. 63] or Ostrowski [9, p. 225]), we have:

$$\lim_{k\in K_1}\lambda_i(H_k)=\lambda_i(H)\ , \qquad i=1,\cdots,n,$$

where $\lambda_i(H)$, $i=1,\cdots,n$ are the eigenvalues of H in increasing order. Now, the matrices $\{Q_k\}_{k\in K_1}$ are contained in a compact set of $\mathbb{R}^{n\times n}$. Therefore, there exists a convergent subsequence $\{Q_k\}_{k\in K_2}$, $K_2\subset K_1$ such that

$$\lim_{k\in K_2} Q_k = Q\ ,$$

and Q is an orthogonal $n\times n$ matrix. Hence, taking limits in (4) for $k\in K_2$, we have:

$$H = QDQ^T\ ,$$

where $D=\operatorname{diag}(\lambda_1(H),\cdots,\lambda_n(H))$, $Q=(v_1,\cdots,v_n)$. Now, $g(x_*)\neq 0$, so there exists $m\in\{1,\cdots,n\}$ such that

$$g(x_*)^T v_m \neq 0\ . \tag{5}$$

Therefore, there exists $\mu > -\lambda_1$ such that

$$\frac{|g(x_*)^T v_m|}{\lambda_m+\mu}\geq\frac{\gamma}{2}\left(\equiv\frac{1}{2}\min\left\{\frac{|g(x_*)^T v_m|}{\lambda_m-\lambda_1},1\right\}\right)\ . \tag{6}$$

Hence, taking limits for $k\in K_2$, we have, for large enough $k\in K_2$,

$$\frac{|g(x_k)^T v_m^k|}{\lambda_m(H_k)+\mu}\geq\frac{\gamma}{4}\ . \tag{7}$$

But,

$$x_k - (H_k + \mu I)^{-1} g(x_k) \in \Gamma_1(x_k, H_k),$$

and

$$\| - (H_k + \mu I)^{-1} g(x_k)\| \geq \frac{|g(x_k)^T v_m^k|}{\lambda_m(H_k) + \mu}\,.$$

Therefore, for large enough $k \in K_2$, by Lemma 2.1 and (7) there exists $z_k \in \Gamma_1(x_k, H_k)$ such that

$$\|x_k - z_k\| = \frac{\gamma}{4}\,. \tag{8}$$

Hence, since $\lim_{k\to\infty} \|\bar{x}_k - x_k\| = 0$, Lemma 2.1 and (8) imply that $\bar{x}_k \in \Gamma_1(x_k, H_k)$ for large enough $k \in K_2$ (say, $k \in K_3$).

We now want to prove that $\lim_{k\to\infty} \mu_k = \infty$. We proceed by contradiction. Assume that $\mu_k \leq \mu_0 < \infty$ for $k \in K_4 \subset K_3$. Then, $\bar{x}_k \in \Gamma_1(x_k, H_k)$ for $k \in K_4$, so for $Q_k = (v_1^k, \cdots, v_n^k)$,

$$\begin{aligned}
\|\bar{x}_k - x_k\|^2 &= \| - (H_k + \mu_k I)^{-1} g(x_k)\|^2 = \| - Q_k(D_k + \mu_k I)^{-1} Q_k^T g(x_k)\|^2 \\
&= \|(D_k + \mu_k I)^{-1} Q_k^T g(x_k)\|^2 \\
&= \left(\frac{g(x_k)^T v_1^k}{\lambda_1(H_k) + \mu_k}\right)^2 + \cdots + \left(\frac{g(x_k)^T v_n^k}{\lambda_n(H_k) + \mu_k}\right)^2 \qquad (9)\\
&\geq \left(\frac{g(x_k)^T v_1^k}{\lambda_1(H_k) + \mu_0}\right)^2 + \cdots + \left(\frac{g(x_k)^T v_n^k}{\lambda_n(H_k) + \mu_0}\right)^2 \,.
\end{aligned}$$

But the limit of the right-hand side of (9) when $k \to \infty$ is clearly a nonzero positive number, therefore $\|\bar{x}_k - x_k\|^2$ is bounded away from zero if $k \in K_4$ is large enough, contradicting the hypothesis. Hence, $\lim_{k\in K_3} \mu_k = \infty$. Therefore, we may write

$$\begin{aligned}
\frac{\bar{x}_k - x_k}{\|\bar{x}_k - x_k\|} &= \frac{-(H_k + \mu_k I)^{-1} g(x_k)}{\| - (H_k + \mu_k I)^{-1} g(x_k)\|} \\
&= \frac{-(H_k/\mu_k + I)^{-1} g(x_k)}{\|(H_k/\mu_k + I)^{-1} g(x_k)\|}\,,
\end{aligned}$$

and the thesis follows for the subsequence indexed by K_3 using boundedness of $\{H_k\}$ and $\lim_{k\in K_3} \mu_k = \infty$. □

Now we are able to prove the following global convergence theorem. Note that we do not assume that $\nabla^2 f(x_k)$ exists, much less that H_k approximates it well.

THEOREM 2.4. *Assume that* $\|H_k\| \leq B$ *for* $k = 0, 1, 2, \cdots$, $x_0 \in \Omega$ *and* x_{k+1}, $k = 0, 1, 2, \cdots$ *is obtained from Algorithm* 2.1. *Let* $x_* \in \Omega$ *be a limit point of* $\{x_k\}$. *Then* $g(x_*) = 0$.

Proof. Assume that $x_* \in \Omega$, $x_* = \lim_{k\in K_1} x_k$, and $g(x_*) \neq 0$. We consider two possibilities:

(a) Some subsequence of $\{\|x_{k+1} - x_k\|\}_{k\in K_1}$ is bounded away from 0.

(b) $\lim_{k\in K_1} \|x_{k+1} - x_k\| = 0$.

Using Lemma 3.2 of Powell and Yuan [10], we see that

$$g(x_k)^T(x_{k+1} - x_k) \leq -\frac{\|g(x_k)\|^2 \|x_{k+1} - x_k\|}{2\|H_k\|\,\|x_{k+1} - x_k\| + \|g(x_k)\|} \leq -\frac{\|g(x_k)\|^2 \|x_{k+1} - x_k\|}{2B\Delta + \|g(x^k)\|}\,.$$

Hence, if (a) holds, using (2) and the continuity of ∇f at x_*, we see that $\lim_{k\to\infty} f(x_k) = -\infty$. This contradicts the assumption $x_* \in \Omega$.

Therefore, it remains to analyze (b). Since, in Algorithm 2.1, x_{k+1} is set to $\bar{x}$, which is chosen such that $\|\bar{x} - x_k\| \geq \beta^2\Delta_k/2$, it follows that $\lim_{k\in K_1} \Delta_k = 0$. We consider two possibilities:

(i) For some $K_2 \subset K_1$, $\lim_{k\in K_2} \bar{\Delta}_k = 0$.

(ii) For every $K_3 \subset K_1$, $\lim_{k\in K_3} \bar{\Delta}_k \neq 0$.

If (i) holds, then we can assume for $k \in K_2$ that $\lambda_1(H_k) > 0$ since otherwise $\bar{\Delta}_k = \Delta$. Thus $\bar{\Delta}_k$ is set in Algorithm 2.1 to be the minimum of Δ and $1/\beta\|H_k^{-1}g(x_k)\|$, and it follows that $\lim_{k\in K_2} \| - H_k^{+}g(x_k)\| = 0$. But

$$\|g(x_k)\| \leq \|H_k\|\;\|H_k^{-1}g(x_k)\| \leq B\|H_k^{-1}g(x_k)\|\;.$$

Hence $\lim_{k\in K_2} g(x_k) = 0$ and so, $g(x_*) = 0$, contradicting the initial assumption.

Now consider (ii). It means that the sequence $\{\bar{\Delta}_k\}_{k\in K_1}$ is bounded away from zero. Therefore the first trial point of the algorithm failed to satisfy (2). This is so because $\Delta_k = \bar{\Delta}_k$, the first pass through the DO loop at each iteration, and our working hypothesis at this point is $\lim_{k\in K_1} \Delta_k = 0$. Thus, for all iterations indexed by K_1, there is at least one failed trial point. Let us set the sequence of last failed trials to $\{\bar{x}_k\}_{k\in K_1}$. We have that each $\bar{x}_k$ satisfies

$$\beta^2 2\Delta_k \leq \|\bar{x}_k - x_k\| \leq 2\Delta_k\;.$$

It follows that

$$\lim_{k\in K_1} \|\bar{x}_k - x_k\| = 0$$

and

$$f(\bar{x}_k) - f(x_k) > -\alpha|g(x_k)^T(\bar{x}_k - x_k)| > -\alpha\|g(x_k)\|\;\|\bar{x}_k - x_k\|\;.$$

Hence, using the Mean Value Theorem,

$$g(x_k - \xi_k(\bar{x}_k - x_k))^T\frac{(\bar{x}_k - x_k)}{\|\bar{x}_k - x_k\|} > -\alpha\|g(x_k)\|\;. \tag{10}$$

Now we are under the hypotheses of Lemma 2.3. So, taking limits on both sides of (10) for a suitable subsequence, we obtain

$$g(x_*)^T\left(\frac{-g(x_*)}{\|g(x_*)\|}\right) \geq -\alpha\|g(x_*)\|\;.$$

But this inequality implies that $\alpha \geq 1$, contradicting the initial hypothesis. Therefore the theorem is proved. □

3. Updating H_k. In §2, we used a uniform bound on $\{\|H_k\|\}$ to obtain a global convergence result for Algorithm 2.1. Algorithm 3.1 proposes a way of updating H_k that under reasonable conditions preserves uniform boundedness of $\{\|H_k\|\}$ and, in addition, incorporates second-order information using secant approximations.

ALGORITHM 3.1. Let $\mathcal{H} \subset \mathbb{R}^{n\times n}$ be a family of symmetric matrices uniformly bounded in norm by M. Let q be a positive integer, $\theta \in (0, \frac{1}{2})$ be a small number, and $\Upsilon \subset \mathbb{R}^{n\times n}$ be the set of tridiagonal symmetric matrices. We now particularize Algorithm 2.1 by specifying that if $k+1 \equiv 0 \pmod{q}$, then we choose $H_{k+1} \in \mathcal{H}$. Otherwise, we assume that $H_k = Q_kT_kQ_k^T$, $T_k \in \Upsilon$, Q_k orthogonal, and we obtain H_{k+1} by the following steps:

Step 1. Let $s = s_k = Q_k^T(x_{k+1} - x_k)$. If s does not satisfy

$$\sqrt{s_i^2 + s_{i+1}^2} \geq \theta\|s\| \leq \sqrt{s_n^2 + s_1^2}\,, \tag{11}$$

$i = 1, \cdots, n-1$, replace s by any vector satisfying (11) with $\|s\| = \|x_{k+1} - x_k\|$ and $x_k + Q_k s \in \Omega$. We used $s = ((\|x_{k+1} - x_k\|)/\sqrt{n})e$ in our implementation.

Step 2. Define $y = y_k = Q_k^T[g(x_k + Q_k s) - g(x_k)]$. (Observe that $x_k + Q_k s = x_{k+1}$ if s was not replaced at Step 1.)

Step 3. Obtain T_{k+1} as the solution of the problem

$$\min_{\substack{Ts=y \\ T\in\Upsilon}} \ \|T - T_k\|_F^2.$$

Step 4. $H_{k+1} = Q_{k+1}T_{k+1}Q_{k+1}^T$ with $Q_{k+1} = Q_k$. (Of course, neither H_k nor H_{k+1} need to be formed.)

The solution of (3) may be obtained using the least-change theory for updates by an algorithm which will be described in §5. See, for example, Dennis and Schnabel [3, Chap. 7]. The rest of this section is essentially to prove that the sequence of matrices obtained using Algorithms 2.1 and 3.1 is bounded, and so, that the global convergence Theorem 2.2 holds. Some auxiliary lemmas will be necessary.

LEMMA 3.1. *Let s be such that $s_i^2 + s_{i+1}^2 > 0$; $i = 1, \cdots, n-1$. Define $\tilde{A} \in \mathbb{R}^{n\times(2n-1)}$ as:*

(12)

$$\tilde{A} = \begin{bmatrix} s_1 & s_2/\sqrt{2} & & & & & & & & & \\ & s_1/\sqrt{2} & s_2 & s_3/\sqrt{2} & & & & & & & \\ & & & s_2/\sqrt{2} & s_3 & s_4/\sqrt{2} & & & & & \\ & & & & & & \ddots & & & & \\ & & & & & & & s_{n-2}/\sqrt{2} & s_{n-1} & s_n/\sqrt{2} & \\ & & & & & & & & & s_{n-1}/\sqrt{2} & s_n \end{bmatrix}.$$

Then rank $\tilde{A} = n$.

Proof. Form $\tilde{A}\tilde{A}^T$ and note that it is symmetric and strictly diagonally dominant. □

COROLLARY 3.1. *Under condition* (11), *if $s \neq 0$, rank $\tilde{A} = n$.*

Proof. The proof is trivial using Lemma 3.1. □

Under condition (11) and $s \neq 0$, either $|s_1| \geq (\theta/\sqrt{2})\|s\|$ or $|s_n| \geq (\theta/\sqrt{2})\|s\|$. Let us suppose, without loss of generality, that $|s_n| \geq (\theta/\sqrt{2})\|s\|$ (otherwise the following lemma may be reformulated in an obvious way).

LEMMA 3.2. *Let β_i be the angle between the row $i+1$ of $\tilde{A}$ and the subspace S_i spanned by the i first rows. Assume $s \neq 0$ and* (11). *Then $|\sin\beta_i| \geq (\theta/\sqrt{2})$, $i = 1, \cdots, n-1$.*

Proof. Consider S_i', the subspace of $\mathbb{R}^{(2n-1)}$ formed by the vectors of the form:

$$(z_1, z_2, \cdots, z_{2i}, 0, \cdots, 0)^T\ .$$

Obviously, $S_i \subset S_i'$, $i = 1, \cdots, n-1$.

Let β_i' be the angle between the row $i+1$ of $\tilde{A}$ and s_i'. Then, $|\sin\beta_i| \geq |\sin\beta_i'|$. Now if $i \leq n-2$, then

$$|\sin\beta_i'| = \frac{\sqrt{s_i^2 + s_{i+1}^2/2}}{\sqrt{(s_{i-1}^2/2) + s_i^2 + (s_i^2/2)}}$$

$$\geq \frac{(1/\sqrt{2})\sqrt{s_i^2 + s_{i+1}^2}}{\sqrt{s_{i-1}^2 + s_i^2 + s_{i+1}^2}} \geq \frac{(1/\sqrt{2})\sqrt{s_i^2 + s_{i+1}^2}}{\|s\|} \geq \frac{\theta}{\sqrt{2}}.$$

If $i = n-1$, then

$$|\sin\beta_{n-1}'| = \frac{|s_n|}{\sqrt{(s_{n-1}^2/2) + s_n^2}} \geq \frac{\theta\|s\|/\sqrt{2}}{\|s\|} = \frac{\theta}{\sqrt{2}},$$

so, $|\sin\beta_i| \geq |\sin\beta_i'| \geq (\theta/\sqrt{2})$, $i = 1, \cdots, n-1$. □

LEMMA 3.3. *The product* $\prod = \prod_{i=1}^{n-1} |\sin\beta_i|$ *is invariant under permutations of the rows of* $\tilde{A}$.

Proof. Set $\bar{A} = \binom{\tilde{A}}{H}$ such that $\bar{A}$ is nonsingular. Suppose further that the rows of H are orthogonal and span the orthogonal complement to the rows of $\tilde{A}$. Thus (see [6])

$$\Pi = \frac{|\det\bar{A}|}{W}, \tag{13}$$

where W is the product of the norms of the rows of $\bar{A}$. But the right-hand side of (13) is invariant under permutations of the rows of $\bar{A}$ (and hence, of $\tilde{A}$), so the same happens with Π. □

LEMMA 3.4. *Let* γ_i *be the angle between the row* i *of* $\tilde{A}$ *and the subspace spanned by the other rows of* $\tilde{A}$. *Then* $|\sin\gamma_i| \geq \prod \geq \theta^{n-1}2^{(1-n)/2}$.

Proof. Fix the row i and permute the rows of $\tilde{A}$ so that row i becomes the last one. So $|\sin\gamma_i| = |\sin\beta_{n-1}| \geq \prod = \prod|\sin\beta_i| \geq \left(\theta/\sqrt{2}\right)^{n-1}$. □

LEMMA 3.5. *Let* $s \neq 0$ *and* $\tilde{A}^+ = \tilde{A}^T\ (\tilde{A}\tilde{A}^T)^{-1}$. *Then* $\tilde{A}^+ \in \mathbb{R}^{(2n-1)\times n}$. *Let*

$$\tilde{A}^+ = (h_1, \cdots, h_n), \qquad \tilde{A} = \begin{pmatrix} r_1^T \\ \vdots \\ r_n^T \end{pmatrix}. \text{ Then } \|h_i\| \leq \frac{2^{(n-1)/2}}{\theta^n\|s\|}. \tag{14}$$

Proof. Each column h_i of $\tilde{A}^+$ is a linear combination of $r_1, \cdots, r_n$. Moreover $h_i^T r_i = 1$ and $h_i^T r_j = 0$ if $j \neq i$. Let S be the subspace spanned by $\{r_1, \cdots, r_n\}$ (and hence, by $\{h_1, \cdots, h_n\}$). Each r_i may be expressed as

$$r_i = v_i + w_i,$$

where v_i is the projection of r_i on the subspace spanned by $\{r_j, j \neq i\}$ and w_i is the projection of r_i into the line spanned by h_i. So

$$w_i = \frac{h_i^T}{\|h_i\|} r_i \frac{h_i}{\|h_i\|} = \frac{h_i}{\|h_i\|^2}. \tag{15}$$

But

$$\sin\gamma_i = \frac{\|w_i\|}{\|r_i\|}, \quad \text{so} \quad \frac{\|w_i\|}{\|r_i\|} \geq \left(\frac{\theta}{\sqrt{2}}\right)^{n-1}. \tag{16}$$

Thus, by (15) and (16), $1/\|h_i\|\,\|r_i\| \geq \left(\theta/\sqrt{2}\right)^{n-1}$ and hence,

$$\|h_i\| \leq \frac{\left(\theta/\sqrt{2}\right)^{1-n}}{\|r_i\|}.$$

But $\|r_i\| \geq \theta\|s\|$, so

$$\|h_i\| \leq \frac{2^{(n-1)/2}}{\theta^n\|s\|}. \tag{17}$$ □

LEMMA 3.6. *If $s \neq 0$, then for any norm $|\cdot|$ fixed in $\mathrm{IR}^{(2n-1)\times n}$, there exists a constant $K_1 = K_1(|\cdot|, \theta, n)$ such that $|\tilde{A}^+| \leq K_1/\|s\|$.*

Proof. The proof is a consequence of (17).

The results above are going to be used in a "vector formulation of the least-change update." Let us write

$$T_k = \begin{bmatrix} a_1^k & b_1^k & & \\ b_1^k & a_2^k & b_2^k & \\ & \ddots & \ddots & \\ & & b_{n-1}^k & a_n^k \end{bmatrix}, \tag{18}$$

$$T = \begin{bmatrix} a_1 & b_1 & & \\ b_1 & a_2 & b_2 & \\ & \ddots & \ddots & \\ & & b_{n-1} & a_n \end{bmatrix}. \tag{19}$$

The least-change update is the solution of

$$\min_{\substack{Ts=y\\ T\in\Upsilon}} \quad \|T - T_k\|_F^2. \tag{20}$$

By (18) and (19), (20) may be formulated as follows:

(21)

$$\min \quad (a_1-a_1^k)^2 + 2(b_1-b_1^k)^2 + (a_2-a_2^k)^2 + \cdots + 2(b_{n-1}-b_{n-1}^k)^2 + (a_n-a_n^k)^2$$

$$\text{s.t.} \quad \begin{cases} a_1s_1 + b_1s_2 & = y_1 \\ \quad b_1s_1 + a_2s_2 + b_2s_3 & = y_2 \\ \qquad \ddots & \\ \qquad\quad b_{n-1}s_{n-1} + b_ns_n & = y_n \end{cases}.$$

Let us now consider the isomorphism between Υ and IR^{2n-1}, which maps

$$T = \begin{bmatrix} a_1 & b_1 & & \\ b_1 & a_2 & b_2 & \\ & \ddots & \ddots & \\ & & b_{n-1} & a_n \end{bmatrix} \overset{\Phi}{\longleftrightarrow} \begin{bmatrix} a_1 \\ b_1 \\ a_2 \\ \vdots \\ b_{n-1} \\ a_n \end{bmatrix} = t. \tag{22}$$

We write $\Phi(T) = t$, $\Phi(T_k) = t_k$, and so on. Therefore, the problem (21) may be written in $\mathbb{R}^{2n-1}$ as:

$$\min_t \|t - t_k\|_G^2 \equiv (t - t_k)^T G (t - t_k) \quad \text{with } G = \begin{pmatrix} 1 &&&&&&& \\ & 2 &&&&&& \\ && 1 &&&&& \\ &&& 2 &&&& \\ &&&& 1 &&& \\ &&&&& \ddots && \\ &&&&&& 2 & \\ &&&&&&& 1 \end{pmatrix}$$

such that $A_k t = y$,

$$(23) \qquad \text{where} \quad A_k = \begin{pmatrix} s_1 & s_2 &&&& \\ & s_1 & s_2 & s_3 && \\ && \ddots & \ddots && \\ &&& s_{n-1} & s_n & \end{pmatrix} \quad \text{and} \quad s_i = (s_k)_i \,.$$

By Lemma 3.1, the matrix $A_k = \tilde{A}_k G^{1/2}$ has full rank, so by straightforward calculations, the solution of (23) is

$$(24) \qquad t_{k+1} = t_k - G^{-1/2} \tilde{A}_k^+ (A_k t_k - y),$$

where $\tilde{A}_k$ is defined in (12) and

$$(25) \qquad \tilde{A}_k^+ = \tilde{A}_k^T (\tilde{A}_k \tilde{A}_k^T)^{-1} = G^{-1/2} A_k^T (A_k G^{-1} A_k^T)^{-1}.$$

So

$$(26) \qquad t_{k+1} = t_k - G^{-1} A_k^T (A_k G^{-1} A_k^T)^{-1} (A_k t_k - y) \,.$$

Therefore

$$\|t_{k+1}\|_G \le \|(I - G^{-1} A_k^T (A_k G^{-1} A_k^T)^{-1} A_k) t_k\|_G + \|G^{-1/2} \tilde{A}_k^+ y\|_G \,.$$

But $(I - G^{-1} A_k^T (A_k G^{-1} A_k^T)^{-1} A_k) t_k$ is the solution of

$$\min \|t - t_k\|_G \quad \text{s.t. } A_k t = 0,$$

so $\|(I - G^{-1} A_k^T (A_k G^{-1} A_k^T)^{-1} A_k) t_k\|_G \le \|t_k\|_G$. Therefore

$$(27) \qquad \|t_{k+1}\|_G \le \|t_k\|_G + \|G^{-1/2} \tilde{A}_k^+ y\|_G \,.$$

Now,

$$\|G^{-1/2} \tilde{A}_k^+ y\|_G \le \|G^{-1/2}\|_G \|\tilde{A}_k^+ y\|_G$$

and

$$\begin{aligned} \|\tilde{A}_k^+ y\|_G = \|y_1 h_1 + \cdots + y_n h_n\|_G &\le |y_1| \, \|h_1\|_G + \cdots + |y_n| \, \|h_n\|_G \\ &\le \|y\| (\|h_1\|_G + \cdots + \|h_n\|_G) \,. \end{aligned}$$

But $\|h_1\|_G + \cdots + \|h_n\|_G$ defines a norm in $\mathbb{R}^{(2n-1)\times n}$, so by Lemma 3.6,

$$\|\tilde{A}_k^+ y\|_G \le \frac{K_1 \|y\|}{\|s\|}$$

and so

$$\|G^{-1/2}\tilde{A}_k^{+}y\|_G \leq K_2\frac{\|y\|}{\|s\|}\ . \tag{28}$$

Now we are able to prove the main result of this section. □

Let

$$L_0 = \{x : f(x) \leq f(x_0)\}.$$

THEOREM 3.7. *Assume that L_0 is bounded and contained in Ω, $f \in C^2(\Omega)$, Ω convex, and that for some $L \geq 0$,*

$$\|\nabla^2 f(x) - \nabla^2 f(w)\| \leq L\|x - w\| \tag{29}$$

for all $x, w \in \Omega$.

Assume that the sequences $\{x_k\}$ and $\{H_k\}$ are generated using Algorithms 2.1 *and* 3.1. *Then the sequence $\{H_k\}$ is bounded by some constant B.*

Proof. Since $\{x_k\}$ is generated by Algorithms 2.1 and 3.1, $\|s\| = \|x_{k+1} - x_k\|$, and Lemma 2.1 implies that $\|s\| = 0$ only if $\{x_k\}$ converges to a stationary point in finitely many steps. Using (29), we have

$$\|y - \nabla^2 f(x_k)s\| \leq \frac{L}{2}\|s\|^2\ .$$

Since $\|\nabla^2 f(x)\|$ is bounded uniformly on L_0 by continuity, and since $\{x_k\}$ contained in L_0 implies that $\|s\|$ is uniformly bounded,

$$\|y\| \leq \|\nabla^2 f(x_k)\|\ \|s\| + \frac{L}{2}\|s\|^2 \leq K_3\|s\|$$

for a suitably defined constant K_3. If $k + 1 \not\equiv 0 \pmod{q}$, then by (27) and (28),

$$\|t_{k+1}\|_G \leq \|t_k\|_G + K_2\frac{\|y\|}{\|s\|} \leq \|t_k\|_G + K_2K_3. \tag{30}$$

Hence, by (30),

$$\begin{aligned}\|H_{k+1}\| &= \|T_{k+1}\| \leq \|T_{k+1}\|_F = \|t_{k+1}\|_G \leq \|t_k\|_G + K_2K_3\\ &= \|T_k\|_F + K_2K_3 \leq \sqrt{n}\|T_k\| + K_2K_3 = \sqrt{n}\|H_k\| + K_2K_3\\ &= (\sqrt{n})^q M + q\sqrt{n}K_2K_3\ .\end{aligned}$$

□

COROLLARY 3.2. *Under the hypothesis of Theorem* 3.7, *the sequence $\{x_k\}$ is well defined by Algorithms* 2.1 *and* 3.1, *and there is at least one limit point of the sequence. Every limit point is a stationary point for f.*

Proof. The proof follows directly from Theorem 2.4, Theorem 3.7, and the compactness of L_0. □

4. Local superlinear convergence.

In §3, we proved that Algorithm 2.1, with the approximate Hessian matrices $\{H_k\}$ chosen by Algorithm 3.1, is globally convergent in the sense that every limit point of the sequence $\{x_k\}$ must satisfy the first-order stationary condition. In this section, we will do two things at once by doing a local analysis of the direct-prediction method associated with the tridiagonal factor update method. This means that we will take $x_{k+1} = x_k + s_k^N$. Unhappily, the good local behavior of this iteration imposes that $H_k \equiv \nabla^2 f(x_k)$ if $k \equiv 0 \pmod{q}$. First, we will prove some strong bounded deterioration results for $\{H_k\}$ which will be crucial to our global convergence result in §5. Then, almost as a sidelight to the main theme of this paper, we will prove that the direct-prediction method is locally q-superlinearly convergent to stationary points at which the Hessian is nonsingular. It will turn out that this result is also useful in the global analysis of §5.

Let us define the algorithm under consideration in this section as an independent algorithm.

ALGORITHM 4.1. Assume that $x_0 \in \mathbb{R}^n$, $H_0 = \nabla^2 f(x_0)$. Given $x_k \in \mathbb{R}^n$, $H_k \in \mathbb{R}^{n\times n}$, $H_k = Q_k T_k Q_k^T$, Q_k orthogonal, $T_k \in \Upsilon$, obtain x_{k+1}, H_{k+1} as follows:

Step 1. $x_{k+1} = x_k - H_k^+ \nabla f(x_k)$.
Step 2. If $k+1 \equiv 0 \pmod{q}$, set $H_{k+1} = \nabla^2 f(x_{k+1})$. Else, obtain H_{k+1} using Algorithm 3.1.

Let us state the assumptions on f which allow us to obtain a local superlinear convergence result.

ASSUMPTION 4.1. Let $f \in C^2(\Omega)$, Ω an open and convex set. We assume that $x_* \in \Omega$ is such that $\nabla f(x_*)$ is symmetric and nonsingular. Further, we assume that (29) holds for all $x, w \in \Omega$.

Let P_Υ denote the Frobenius norm projection operator onto the subspace of symmetric tridiagonal matrices Υ.

LEMMA 4.1. *Assume that* $k \equiv 0 \pmod{q}$ *and that* $x_k \in \Omega$ *is well defined. Then,*

$$\|P_\Upsilon(Q_k{}^T \nabla^2 f(x_*) Q_k) - Q_k{}^T \nabla^2 f(x_*) Q_k\|_F \le 2\sqrt{n}\, L \|x_k - x_*\|\ .$$

Proof.

$$\begin{aligned}
&\|P_\Upsilon(Q_k{}^T \nabla^2 f(x_*) Q_k) - Q_k{}^T \nabla^2 f(x_*) Q_k\ \|_F \\
&\quad\le \|P_\Upsilon(Q_k{}^T \nabla^2 f(x_*) Q_k) - Q_k{}^T \nabla^2 f(x_k) Q_k\|_F \\
&\qquad + \|Q_k{}^T \nabla^2 f(x_k) Q_k - Q_k{}^T \nabla^2 f(x_*) Q_k\|_F\ .
\end{aligned}$$

But $Q_k{}^T \nabla^2 f(x_k) Q_k \in \Upsilon$. Therefore,

$$\begin{aligned}
&\|P_\Upsilon(Q_k{}^T \nabla^2 f(x_*) Q_k) - Q_k{}^T \nabla^2 f(x_k) Q_k\|_F \\
&\qquad = \|P_\Upsilon(Q_k{}^T \nabla^2 f(x_*) Q_k) - P_\Upsilon(Q_k{}^T \nabla^2 f(x_k) Q_k)\|_F \\
&\qquad \le \|Q_k{}^T \nabla^2 f(x_*) Q_k - Q_k{}^T \nabla^2 f(x_k) Q_k\|_F.
\end{aligned}$$

Hence, by (29),

$$\begin{aligned}
\|P_\Upsilon(Q_k{}^T \nabla^2 f(x_*) Q_k) - Q_k{}^T \nabla^2 f(x_*) Q_k\|_F &\le 2\|Q_k^T(\nabla^2 f(x_k) - \nabla^2 f(x_*))Q_k\|_F \\
&\le 2\sqrt{n}\ \|Q_k{}^T(\nabla^2 f(x_k) - \nabla^2 f(x_*))Q_k\| \\
&= 2\sqrt{n}\ \|\nabla^2 f(x_k) - \nabla^2 f(x_*)\| \\
&= 2\sqrt{n}\, L\|x_k - x_*\|.
\end{aligned}$$

□

From now on, let us use the notation $e_\ell = \|x_\ell - x_*\|$, $\ell = 0, 1, 2, \cdots$.

LEMMA 4.2. *Assume that* $k \equiv 0 \pmod{q}$, $0 \le j \le q-1$, *and that* x_{k+j}, x_{k+j+1}, $x_{k+j} + s_{k+j}$ *are well defined and belong to* Ω. *Then,*

$$\|y_{k+j} - [P_\Upsilon(Q_k{}^T \nabla^2 f(x_*) Q_k)] s_{k+j}\| \le L\|s_{k+j}\|(e_{k+j} + \tfrac{1}{2} e_{k+j+1} + 2\sqrt{n}\, e_k)\ .$$

Proof.

$$\begin{aligned}\|y_{k+j} - [P_\Upsilon({Q_k}^T\nabla^2 f(x_*)Q_k)]s_{k+j}\| &\le \|y_{k+j} - {Q_k}^T\nabla^2 f(x_*)Q_k s_{k+j}\| \\ &\quad + \|P_\Upsilon({Q_k}^T\nabla^2 f(x_*)Q_k) - {Q_k}^T\nabla^2 f(x_*)Q_k\|\,\|s_{k+j}\| \,.\end{aligned} \tag{31}$$

But, by (29), and the definition of y_{k+j},

$$\begin{aligned}&\|y_{k+j} - {Q_k}^T\nabla^2 f(x_*)Q_k s_{k+j}\| \\ &\quad = \|g(x_{k+j} + Q_k s_{k+j}) - g(x_{k+j}) - \nabla^2 f(x_*)Q_k s_{k+j}\| \\ &\quad \le \frac{L}{2}\|s_{k+j}\| \max\{e_{k+j}, \|x_{k+j} + Q_k s_{k+j} - x_*\|\} \,.\end{aligned} \tag{32}$$

Therefore, by (31), (32), and Lemma 4.1,

$$\begin{aligned}&\|y_{k+j} - [P_\Upsilon({Q_k}^T\nabla^2 f(x_*)Q_k)]s_{k+j}\| \\ &\quad \le \frac{L}{2}\|s_{k+j}\| \max\{e_{k+j}, \|x_{k+j} + Q_k s_{k+j} - x_*\|\} + 2\sqrt{n}\,L\|s_{k+j}\|e_k \,.\end{aligned}$$

Now, even if $s_{k+j} \neq x_{k+j+1} - x_{k+j}$, they are equal in norm, so

$$\begin{aligned}\|x_{k+j} - Q_k s_{k+j} - x_*\| \le e_{k+j} + \|s_{k+j}\| &= e_{k+j} + \|x_{k+j+1} - x_{k+j}\| \\ &\le 2e_{k+j} + e_{k+j+1} \,.\end{aligned}$$

Therefore,

$$\begin{aligned}&\|y_{k+j} - [P_\Upsilon({Q_k}^T\nabla^2 f(x_*)Q_k)]s_{k+j}\| \\ &\quad \le \frac{L}{2}\|s_{k+j}\|(2e_{k+j} + e_{k+j+1}) + 2\sqrt{n}\,L\|s_{k+j}\|e_k \\ &\quad \le L\|s_{k+j}\|(e_{k+j} + \tfrac{1}{2}e_{k+j+1} + 2\sqrt{n}\,e_k) \,,\end{aligned}$$

as we wanted to prove. □

The following lemma states a Bounded Deterioration Principle (see [1]) for the matrices T_k.

LEMMA 4.3. *Assume that* $k \equiv 0 \pmod q$, $0 \le j \le q-2$, *and that* x_{k+j}, x_{k+j+1}, $x_{k+j} + Q_k s_{k+j}$ *are well defined and belong to* Ω. *Then,*

$$\begin{aligned}&\|T_{k+j+1} - P_\Upsilon({Q_k}^T\nabla^2 f(x_*)Q_k)\|_F \\ &\quad \le \|T_{k+j} - P_\Upsilon({Q_k}^T\nabla^2 f(x_*)Q_k)\|_F + K_2 L(e_{k+j} + \tfrac{1}{2}e_{k+j+1} + 2\sqrt{n}\,e_k) \,.\end{aligned}$$

Proof. For matrices $T \in \Upsilon$, remember that $\|T\|_F = \|\Phi(T)\|_G$, where Φ is the isomorphism which maps Υ into $\mathbb{R}^{2n-1}$. The matrices

$$T_{k+j+1} - P_\Upsilon({Q_k}^T\nabla^2 f(x_*)Q_k)$$

and

$$T_{k+j} - P_\Upsilon({Q_k}^T\nabla^2 f(x_*)Q_k)$$

belong to Υ. So, using the convention $t = \Phi(T)$, we are going to prove the thesis in $\mathbb{R}^{2n-1}$ using $\|\cdot\|_G$.

By (26) we have, writing $y = y_{k+j}$,

$$\begin{aligned} t_* &= \Phi(P_\Upsilon({Q_k}^T \nabla^2 f(x_*) Q_k)), \\ t_{k+j+1} &= t_{k+j} - G^{-1} A_{k+j}^T (A_{k+j} G^{-1} A_{k+j}^T)^{-1} (A_{k+j} t_{k+j} - y). \end{aligned}$$

So,

$$\begin{aligned} t_{k+j+1} - t_* &= t_{k+j} - t_* - G^{-1} A_{k+j}^T (A_{k+j} G^{-1} A_{k+j}^T)^{-1} (A_{k+j} t_{k+j} - y) \\ &= t_{k+j} - t_* - G^{-1} A_{k+j}^T (A_{k+j} G^{-1} A_{k+j}^T)^{-1} \\ &\quad \times (A_{k+j} t_{k+j} - A_{k+j} t_* + A_{k+j} t_* - y) \\ &= [I - G^{-1} A_{k+j}^T (A_{k+j} G^{-1} A_{k+j}^T)^{-1} A_{k+j}](t_{k+j} - t_*) \\ &\quad + G^{-1} A_{k+j}^T (A_{k+j} G^{-1} A_{k+j}^T)^{-1} (y - A_{k+j} t_*). \end{aligned}$$

Hence,

$$\begin{aligned} \|t_{k+j+1} - t_* \|_G &\le \|[I - G^{-1} A_{k+j}^T (A_{k+j} G^{-1} A_{k+j}^T)^{-1} A_{k+j}](t_{k+j} - t_*)\|_G \\ &\quad + \|G^{-1} A_{k+j}^T (A_{k+j} G^{-1} A_{k+j}^T)^{-1} (y - A_{k+j} t_*)\|_G \\ &\le \|t_{k+j} - t_*\|_G + \|G^{-1} A_{k+j}^T (A_{k+j} G^{-1} A_{k+j}^T)^{-1} (y - A_{k+j} t_*)\|_G \ . \end{aligned}$$

Therefore, using the arguments which lead to (28), we have:

$$\|t_{k+j+1} - t_*\|_G \le \|t_{k+j} - t_*\|_G + \frac{K_2 \|y - A_{k+j} t_*\|}{\|s_{k+j}\|} \ .$$

But $A_{k+j} t_* = P_\Upsilon({Q_k}^T \nabla^2 f(x_*) Q_k) s_{k+j}$. Thus, the desired result follows using Lemma 4.2. □

LEMMA 4.4. *Assume that* $k \equiv 0 \pmod q$, $0 \le j \le q-2$, *and that* x_{k+j}, x_{k+j+1}, $x_{k+j} + Q_k s_{k+j}$ *are well defined and belong to* Ω. *Then,*

$$\begin{aligned} &\|T_{k+j+1} - {Q_k}^T \nabla^2 f(x_*) Q_k\|_F \\ &\quad \le \|T_{k+j} - {Q_k}^T \nabla^2 f(x_*) Q_k\|_F + L \left(K_2 e_{k+j} + \frac{K_2}{2} e_{k+j+1} + 2\sqrt{n}\,(K_2 + 1) e_k \right) . \end{aligned}$$

Proof. By Lemmas 4.1 and 4.3, we have:

$$\begin{aligned} &\|T_{k+j+1} - {Q_k}^T \nabla^2 f(x_*) Q_k \ \|_F \\ &\quad \le \|T_{k+j+1} - P_\Upsilon({Q_k}^T \nabla^2 f(x_*) Q_k)\|_F \\ &\qquad + \|P_\Upsilon({Q_k}^T \nabla^2 f(x_*) Q_k) - {Q_k}^T \nabla^2 f(x_*) Q_k\|_F \\ &\quad \le \|T_{k+j} - P_\Upsilon({Q_k}^T \nabla^2 f(x_*) Q_k)\|_F \\ &\qquad + K_2 L(e_{k+j} + \tfrac{1}{2} e_{k+j+1} + 2\sqrt{n}\, e_k) + 2L e_k \sqrt{n} \end{aligned}$$

and the desired result follows trivially from this inequality. □

LEMMA 4.5. *Assume the hypotheses of the previous lemmas. Then,*

$$\|T_k - {Q_k}^T \nabla^2 f(x_*) Q_k\|_F \le \sqrt{n} L e_k \ , \tag{33}$$

and for $0 \le j \le q - 2$,

$$\begin{aligned} &\|T_{k+j+1} - {Q_k}^T \nabla^2 f(x_*) Q_k\|_F \\ &\quad \le \sqrt{n}\, L e_k + \sum_{\nu=0}^{j} L \left(K_2 e_{k+\nu} + \frac{K_2}{2} e_{k+\nu+1} + 2\sqrt{n}\,(K_2 + 1) e_k \right) \ . \end{aligned}$$

Proof.

$$\begin{aligned}\|T_k - {Q_k}^T\nabla^2 f(x_*)Q_k\|_F &= \|{Q_k}^T\nabla^2 f(x_{k+j})Q_k - {Q_k}^T\nabla^2 f(x_*)Q_k\|_F \\ &\le \sqrt{n}\,\|\nabla^2 f(x_{k+j}) - \nabla^2 f(x_*)\| \le \sqrt{n}\,Le_{k+j} = \sqrt{n}\,Le_k\,.\end{aligned}$$

Thus the desired result follows straightforwardly from the previous inequality and Lemma 4.4. □

LEMMA 4.6. *Assume the hypotheses of the previous lemmas, and remember that*

$$H_\ell = Q_\ell T_\ell Q_\ell^T \qquad \textit{for } \ell = 1, 2, \cdots.$$

Then, for some $\eta \ge 0$

$$\|H_{k+j+1} - \nabla^2 f(x_*)\| \le \eta\left(\sum_{\nu=0}^{j+1} e_{k+\nu}\right).$$

Proof. By Lemma 4.5,

$$\begin{aligned}\|H_{k+j+1} - \nabla^2 f(x_*)\| &= \|Q_k(T_{k+j+1} - {Q_k}^T\nabla^2 f(x_*)Q_k){Q_k}^T\| \\ &\le \|T_{k+j+1} - {Q_k}^T\nabla^2 f(x_*)Q_k\| \\ &\le \|T_{k+j+1} - {Q_k}^T\nabla^2 f(x_*)Q_k\|_F \\ &\le \sqrt{n}\,Le_k + \sum_{\nu=0}^{j} L\left(K_2 e_{k+\nu} + \frac{K_2}{2}e_{k+\nu+1} + 2\sqrt{n}\,(K_2+1)e_k\right)\end{aligned}$$

and the result follows directly. □

THEOREM 4.7. *There exists* $\epsilon > 0$ *such that for any* x_0 *with* $\|x_0 - x_*\| \le \epsilon$, *the sequence* $\{x_\ell\}$ *generated by Algorithm* 4.1 *converges q-superlinearly to* x_*. *Furthermore, if* $\epsilon q\eta\|\nabla^2 f_*^{-1}\| \le \gamma < 1$, *then the sequence* $\{\|H_\ell^{-1}\|\}$ *is uniformly bounded by the constant*

$$B_N \equiv \frac{\|\nabla^2 f(x_*)^{-1}\|}{(1-\gamma)}$$

independent of the particular choice of x_0.

Proof. Algorithm 4.1 is locally linearly convergent and $\{\|H_\ell^{-1}\|\}$ is uniformly bounded if the matrices H_k remain in a suitable neighborhood of $\nabla^2 f(x_*)$. (See [3, Chap. 7].) This condition is easily verified using Lemma 4.6 if x_0 is close enough to x_*. The reason this condition and the bound on the inverses can be independent of the particular x_0 is that Algorithm 4.1 always takes $H_0 = \nabla^2 f(x_0)$. In particular,

$$\|H_\ell - \nabla^2 f(x_*)\| \le \epsilon q\eta$$

and so the bound B_N follows from the Banach lemma (see [3]). Now, using linear convergence and Lemma 4.6, we see that $\lim_{k\to\infty} H_k = \nabla^2 f(x_*)$. This implies that convergence is q-superlinear (see [1]). □

5. Global superlinear convergence.

In §3, we proved that Algorithm 2.1, with the approximate Hessian matrices $\{H_k\}$ chosen by Algorithm 3.1, is globally convergent in the sense that every limit point of the sequence $\{x_k\}$ is a first-order stationary point. In §4, we proved that if we require the Hessian update method to always choose $H_k = \nabla^2 f(x_k)$ every q iterations, then the direct-prediction method is locally q-superlinearly convergent to stationary points at which the Hessian is nonsingular. In this section, we put all this together. We update the Hessian approximations as in §4, and we modify Algorithm 2.1 to always try the full quasi-Newton step first when H_k is positive definite. We then prove that if f is quasi-convex on L_0 and $\nabla^2 f(x_*) = \nabla^2 f(x_*)$ is positive definite for some stationary point x_*, then from some point on, the Newton steps satisfy the sufficient decrease condition (2).

ALGORITHM 5.1 Assume that $x_0 \in \mathbb{R}^n$, $H_0 = \nabla^2 f(x_0)$. Given $x_k \in \mathbb{R}^n$, $H_k \in \mathbb{R}^{n\times n}$, $H_k = Q_k T_k Q_k^T$, Q_k orthogonal, $T_k \in \Upsilon$, obtain $\{x_{k+1}\}$, $\{H_{k+1}\}$ as follows:

Step 1. If H_k is positive definite, then in Algorithm 2.1, first try $x_{k+1} = x_k - H_k^{-1}\nabla f(x_k)$.

Step 2. If $k+1 \equiv 0 \pmod q$, set $H_{k+1} = \nabla^2 f(x_{k+1})$. Else, obtain H_{k+1} using Algorithm 3.1. Return to Step 1.

Now we give our main result. We assume that f is quasi-convex, i.e., that all level sets of f are convex.

THEOREM 5.1. *Let $f \in C^2(\Omega)$, Ω an open and convex set containing L_0, be a quasi-convex function on L_0. Assume that L_0 is bounded, and that some stationary point $x_* \in \Omega$ is such that $\nabla^2 f(x_*)$ is positive definite. Further, assume that the Lipschitz condition on the Hessian given by* (29) *holds for all $x, w \in \Omega$. Then, there exists some integer k_N such that Algorithm* 5.1 *takes $\mu_k = 0$ for $k \geq k_N$, and $\{x_k\}$ converges q-superlinearly to x_*, which is the global minimizer of f.*

Proof. Since f is quasi-convex and has a stationary point x_* at which $\nabla^2 f(x_*)$ is positive definite, x_* must be the unique stationary point for f on L_0, and the global minimizer of f.

Since L_0 is bounded and $\nabla^2 f$ is continuous, we can take

$$\mathcal{H} = \{\nabla^2 f(x) : x \in L_0\}.$$

Thus from Corollary 3.2, we have that $\{x_k\}$ is well defined and that some subsequence converges to a stationary point, which must then be x_*. Furthermore, there is some $B \geq \|H_k\|$ uniformly in k. Since x_* is the only possible limit point of $\{x_k\}$, the compactness of L_0 ensures that $\lim_k x_k = x_*$. In particular, the subsequence of the iterates indexed by $k \equiv 0 \pmod q$ converges to x_*.

The key to the proof will be to show below that eventually, starting at one of the $k \equiv 0 (\mathrm{mod}\ q)$ iterates, Algorithm 5.1 reduces to Algorithm 4.1, i.e., the step $s_k^N = -H_k^{-1} g_k$ eventually satisfies (2).

Let ϵ be small enough that Algorithm 4.1 is locally q-linearly convergent to x_* from any x_0^N with $\|x_0^N - x_*\| < \epsilon$. Now, let B_N be as in Theorem 4.7. Choose ϵ even smaller, if necessary, to make $1 - 2\alpha > (q\eta + L)B_N\epsilon$. The standard approach to proving Theorem 4.7 makes ϵ be chosen so that if $\nabla^2 f(x_*)$ is positive definite, then so are all H_k^N for $\|x_0^N - x_*\| < \epsilon$. Choose $k_N \equiv 0 \pmod q$ so that if $k \geq k_N$, then $\|x_k - x_*\| < \epsilon$.

There are still a couple of small points to deal with before we start to chain inequalities. First, since H_k is positive definite, we have $g_k^T s_k^N < 0$, and

$$\|s_k^N\|^2 = (H_k^{-1/2} g_k)^T H_k^{-1} H_k^{-1/2} g_k \leq \|(H_k)^{-1}\| (H_k^{-1/2} g_k)^T H_k^{-1/2} g_k$$

$$\le -\|(H_k)^{-1}\| g_k^T s_k^N .$$

Furthermore, $x_k^N, x_k^N + s_k^N$ are both within ϵ of x_*. Thus, any convex combination is also, and so for any $\xi \in (0,1)$, $\|x_k^N + \xi s_k^N - x_*\| < \epsilon$.

Now the proof that $\{x_k^N\} = \{x_k\}$ for $k > k_N$ is by Taylor's theorem and all these partial results. It can be done by induction, but we give only the main step here. Assume that the sequences are identical from the k_Nth to the ℓth iterate. Then $H_\ell = H_{\ell - k_N}^N$, and

$$\begin{aligned} f(x_\ell + s_\ell^N) - f_\ell &= g_\ell^T s_\ell^N + \tfrac{1}{2}(s_\ell^N)^T[\nabla^2 f(x_\ell + \xi s_\ell^N - x_*) \pm \nabla^2 f(x_*)] s_\ell^N \\ &= \tfrac{1}{2} g_\ell^T s_\ell^N + \tfrac{1}{2}(s_\ell^N)^T[\nabla^2 f(x_\ell + \xi s_\ell^N - x_*) \pm \nabla^2 f(x_*) - H_\ell] s_\ell^N \\ &\le \tfrac{1}{2} g_\ell^T s_\ell^N + \tfrac{1}{2}[L + q\eta]\epsilon \|s_\ell^N\|^2 \\ &\le \tfrac{1}{2} g_\ell^T s_\ell^N - \tfrac{1}{2}[L + q\eta]\epsilon B_N g_\ell^T s_\ell^N \\ &\le \tfrac{1}{2} g_\ell^T s_\ell^N - \tfrac{1}{2}(1 - 2\alpha) g_\ell^T s_\ell^N = \alpha g_\ell^T s_\ell^N \end{aligned}$$

since H_ℓ is positive definite and so $g_\ell^T s_\ell^N < 0$. □

6. Implementation.

6.1. Implementation of steps 4 and 5 of Algorithm 2.1.

Considering $s_k(\mu) = -(H_k + \mu I)^{-1} g(x_k)$ with

$$\mu \ge \bar{\mu} = \max(0, -\lambda_1 + \epsilon),$$

where λ_1 is the least eigenvalue of H_k and $\epsilon = 10^{-5}$ in the computer implementation, we choose

$$x_{k+1} = x_k + s_k(\mu_*),$$

where μ_* is an approximate solution to the problem

$$\text{(I)} \qquad \arg\min f(x_k + s_k(\mu)), \qquad \mu \ge \bar{\mu} .$$

In order to solve this problem it is necessary to follow the curvilinear path $s_k(\mu)$, $\mu \ge \bar{\mu}$, and therefore to find the solution of the linear system of equations

$$(H_k + \mu I) s_k(\mu) = -g(x_k) , \qquad \mu \ge \bar{\mu}$$

for several trial values of μ. These computations are carried out in $O(n)$ operations because the decomposition $H_k = Q_k T_k {Q_k}^T$ is available. This is because we can write the equivalent system

$$(T_k + \mu I)\hat{s}_k(\mu) = -\hat{g}(x_k),$$

where $\hat{s}_k(\mu) = {Q_k}^T s_k(\mu)$, $\hat{g}(x_k) = {Q_k}^T g(x_k)$.

The least eigenvalue of T_k is obtained by means of the IMSL routine EQRT1S, and the solution of the tridiagonal systems by the LINPACK routine SGTSL.

For solving (I) we modified the routine GSRCH originally written by Powell for MINPACK [10].

The new iterate x_{k+1} is accepted (step 5 of Algorithm 2.1) only if the condition

$$f(x_{k+1}) \le f(x_k) + \alpha g(x_k)^T (x_{k+1} - x_k)$$

is satisfied with $\alpha = 10^{-4}$. However, we may continue searching even if the Newton step satisfies this criterion.

We decide that $\Gamma_2(x_k, H_k)$ is not empty if the angle between g_k and $v_1^{(k)}$ is between 85 degrees and 95 degrees.

6.2. Choosing the sequence B_k. For those iterations in which $H_k = \nabla^2 f(x_k)$, the decomposition is computed with the IMSL routines EHOUSS and EHOBS, except when the Hessian itself is tridiagonal.

The stopping condition is (7.2.5) of Dennis and Schnabel [3, p. 160]

$$\max_{1\le i\le n}\left\{\frac{|\nabla f(x_k)_i|\max(|x_i^k|,1)}{\max(|f(x_k)|,1)}\right\}\le \text{eps}$$

(eps $= 10^{-15}$ in the computer implementation).

6.3. Efficiency. The computer program allows the user to compute the full decomposition every q iterations (we use $q = 3$) or to decide when to do so in between automatically, depending upon the following notion of efficiency of an iteration. We define efficiency of the kth iteration as

$$E_k = \frac{-\log r_k}{t_k},$$

where

$$r_k = \frac{f_{k+1} - f_*}{f_k - f_*},$$

f_* is an estimation of $f(x_*)$, $f_{k+1} = f(x_{k+1})$, and t_k is the CPU time required by the kth iteration.

Assuming r_k remains constant until convergence (denoted by r hereafter), the required number of iterations NITER is approximately given by

$$r^{NITER} = \text{eps}\,.$$

Therefore, the total CPU time T will be

$$T = \frac{\log \text{eps}}{\log r} t_k = -\frac{\log \text{eps}}{E_k}\,.$$

In order to decide what H_{k+1} will be (that is, $H_{k+1} = \nabla^2 f(x_{k+1})$ or $H_{k+1} = Q_k T_{k+1} {Q_k}^T$), we use E_k as follows. Let k_0 be the last iteration such that $B_{k_0} = \nabla^2 f(x_{k_0})$. If $k_0 \equiv k \pmod q$ or if $E_{k_0} > E_k$, then $H_{k+1} = \nabla^2 f(x_{k+1})$. Otherwise $H_{k+1} = Q_k T_{k+1} {Q_k}^T$.

7. Numerical experience. The class of algorithms described in the previous sections form the theoretical basis of subroutine TRIDI.

The decision about when Γ_2 is not empty is taken according to a user-supplied parameter defining a maximum deviation in degrees with respect to orthogonality. This parameter was defined as five degrees for the numerical experiments.

7.1. Test problems. In order to demonstrate the effectiveness of the new method, numerical results were obtained not only for well-known test examples appearing in the literature but also for some new functions. For brevity, the full details of the test problems are not given here except for the following new ones:

TEST FUNCTION PRUEBA.

$$f(x) = a(1)/x(1) + a(2)/x(2) + a(3)/x(3) + 0.5\langle x, Cx\rangle + \langle b, x\rangle$$

where $b(i) = 1. \times 10^{-6} * a(i) - (i+4) * 1. \times 10^3$ for $i = 1,\cdots,3$, a is as defined in Table 1, and

$$C = \begin{pmatrix} 1/3 & 1/10 & 1/10 \\ 1/10 & 1/4 & 1/10 \\ 1/10 & 1/10 & 1/5 \end{pmatrix}.$$

The underlying idea is that if a starting point is close to the origin, the "wavy behavior" of the function leads to a very small trust region, a phenomenon which leads to a rather inefficient performance of the classical method. This shortcoming does not exist for the new algorithm because of the curvilinear search, which can be considered as a way of computing an optimal radius in each iteration.

TEST FUNCTION SNLLSQ I. Generate data $(j, y(j))$ for $j = 1, \cdots, 15$ from

$$y(j) = a(1) * j{*}{*}xopt(1) + a(2) * j{*}{*}xopt(2) + a(3) * j{*}{*}xopt(3)$$

with $a(1) = 3$, $a(2) = 3.1$, $a(3) = 0.7$, $xopt(1) = 1.5$, $xopt(2) = 2.5$, $xopt(3) = -2.5$. Now with the given a, recover x by a least-squares fit to this data.

TEST FUNCTION SNLLSQ II. Generate data $(j, y(j))$ for $j = 1, \cdots, 15$ from

$$y(j) = a(1) * \sin(j * xopt(1)) + a(2) * \sin(j * xopt(2)) + a(3) * \sin(j * xopt(3))$$

with $a(i), xopt(i)$, $i = 1, \cdots, 3$ as in SNLLSQ I. Again, recover x by least squares.

TEST FUNCTION SNLLSQ III. Generate data $(j, y(j))$ for $j = 1, \cdots, 30$ from

$$y(j) = a(1) * \cos(j * xopt(1)) + a(2) * \cos(j * xopt(2)) + a(3) * \cos(j * xopt(3))$$

with $a(1) = 10$, $a(2) = 20$, $a(3) = 30$, $xopt(1) = 0.1$, $xopt(2) = 0.2$, $xopt(3) = 0.3$. Recover x by least squares.

TEST FUNCTION SNLLSQ IV. Generate data $(j, y(j))$ for $j = 1, \cdots, 45$ from

$$y(j) = a(1) * \exp(j * xopt(1)) + a(2) * \exp(j * xopt(2)) + a(3) * \exp(j * xopt(3))$$

with $a(1) = 1$, $a(2) = 2$, $a(3) = 3$, $xopt(1) = -0.1$, $xopt(2) = -0.2$, $xopt(3) = -0.3$. Now recover x by least squares.

From here on we use the notation tfn.n.cn.sp, where tfn is the test function number, n the number of variables, cn the case number, and sp the identification of the starting point.

Table 1 defines the problems.

TABLE 1

tfn	Name	n	cn	sp
1	Prueba	3	1: $a(i) = 1.d-1$	1: $(1.d-3,\ 1.d-3,\ 1.d-3)$
1		3	2: $a(1) = 1.d3$ $a(2) = a(3) = 1.d0$	2: (0.25, 0.25, 0.25)
1		3	3: $a(1) = a(2) = a(3) = 1.d1$	
2	Penalty I	4	1	1: $x(j) = j$
2	[3]	8	1	
3	Variable Dimensioned	4	1	1: $x(j) = 1 - j/n$
3	[3]	5		
3		8		
3		12		

TABLE 1
(*continued*)

tfn	Name	n	cn	sp
4	Rosenbrock	4	1	1: $x(2j-1) = -1.2,\ x(2j) = 1$
4	[3]	8		
4		10		
4		12		
5	Chained Rosenbrock [3]	25	1	1: $x(j) = -1$
6	Powell Extended	4	1	1: $x(4j-3) = 3,\ x(4j-2) = -1$
6	[3]	8	1	$x(4j-1) = 0,\ x(4j) = 1$
6		240	1	
6		400	1	
7	Brown–Dennis	4	1	1: $(25, 5, -5, 1)$
8	Gaussian [3]	3	1	1: $(0.4, 1, 0)$
9	Trigonometric	25	1	1: $x(j) = 1$
9	[5]	50		
9		100		
9		200		
10	Watson [3]	12	1	1: $x(j) = 0$
11	Wood [3]	4	1	1: $(-3, -1, -3, -1)$
12	Box [3]	3	1	1: $(0, 10, 20)$
13	Biggs Exp 6 [3]	6	1	1: $(1.2, 1, 1, 1, 1, 1)$
14	Dennis–Marwil I [2]	10	1: $r1 = 1;\ r2 = n$ $k1 = k3 = 1;\ k2 = 5$ 2: $r1 = 1;\ r2 = n$ $k1 = 4;\ k2 = k3 = 1$	1: $x(j) = -1$
		100	2	
15	Dennis–Marwil II [2]	5	1	1: $x(j) = -1$
16	Pseudo Penalty [3]	50	1	1: $x(j) = 0$
17	SNLLSQ I	3	1	1: $x(j) = 3.50 * xopt(j)$
18	SNLLSQ II	3	1	1: $x(j) = 1.15 * xopt(j)$
19	SNLLSQ III	3	1	1: $x(j) = 1.50 * xopt(j)$
20	SNLLSQ IV	3	1	1: $x(j) = 3.00 * xopt(j)$

TABLE 2

Problem	NIT	FE	GE	HE	T	FMIN
1.3.1.1	18	33	19	18	1.00	−.13e + 09
	11	12	12	5	0.29	−.13e + 09
	13	14	14	4	0.33	−.13e + 09
1.3.1.2	12	14	13	12	1.00	−.13e + 09
	5	6	6	2	0.24	−.13e + 09
	5	6	6	2	0.22	−.13e + 09
1.3.2.1			error 6			
	23	24	24	9	0.09	−.13e + 09
	22	23	23	8	0.06	−.13e + 09
1.3.2.2	13	26	14	13	1.00	−.13e + 09
	8	9	9	3	0.26	−.13e + 09
	8	9	9	2	0.24	−.13e + 09
1.3.3.1	23	33	24	23	1.00	−.13e + 09
	17	18	18	8	0.45	−.13e + 09
	18	19	19	5	0.45	−.13e + 09
1.3.3.2	12	20	13	12	1.00	−.13e + 09
	5	6	6	2	0.20	−.13e + 09
	5	6	6	2	0.19	−.13e + 09
2.4.1.1	34	48	35	34	1.00	0.23e − 04
	11	12	12	5	0.50	0.24e − 04
	12	13	13	4	0.33	0.24e − 04
2.8.1.1	34	43	35	34	1.00	0.54e − 04
	15	16	16	5	0.88	0.57e − 04
	17	21	21	6	1.09	0.57e − 04
3.4.1.1	10	11	11	10	1.00	0.24e − 27
	12	13	13	5	1.10	0.21e − 30
	12	13	13	4	1.88	0.78e − 12
3.5.1.1	11	12	12	11	1.00	0.13e − 28
	14	15	15	6	3.79	0.27e − 19
	14	34	34	4	3.74	0.61e − 17

7.2. Numerical results. Table 2 gives the obtained numerical results using the notation:

NIT = number of iterations
FE = number of function evaluations
GE = number of gradient evaluations
HE = number of Hessian evaluations
T = relative CPU time with respect to the IMSL routines
FMIN = Computed minimum

For each problem three sets of results are given; the first row corresponds to the routine DUMIAH (trust region algorithm), the second and third to the new method with efficiency and without efficiency, respectively. For the last four test problems the first row corresponds to the results obtained with the routine DUMIDH. Error 6 in DUMIAH means that five consecutive steps have been taken with the maximum step length.

The computational tests were carried out in double precision on a Hewlett-Packard 9000 825S computer using software written in Fortran 77 under the HP-UX operating system and on an IBM 4361. The reason for using two different computers

TABLE 2
(*continued*)

Problem	NIT	FE	GE	HE	T	FMIN
3.8.1.1	13	14	14	13	1.00	$0.53e-26$
	17	18	18	5	4.75	$0.22e-24$
	16	18	18	6	4.75	$0.19e-16$
3.10.1.1	14	15	15	14	1.00	$0.18e-25$
	18	21	21	5	7.07	$0.15e-14$
	18	19	19	6	6.13	$0.46e-19$
4.4.1.1	23	34	24	23	1.00	$0.55e-20$
	31	50	49	14	1.16	$0.39e-31$
	39	72	71	10	1.45	$0.77e-21$
4.8.1.1	23	34	24	23	1.00	$0.11e-19$
	35	63	61	16	1.85	$0.29e-27$
	42	91	88	11	2.21	$0.34e-23$
4.10.1.1	23	34	24	23	1.00	$0.14e-19$
	36	75	73	12	1.68	$0.28e-11$
	36	75	73	12	1.46	$0.23e-11$
4.12.1.1	23	34	24	23	1.00	$0.16e-19$
	38	87	84	13	1.80	$0.18e-15$
	38	87	84	13	1.78	$0.18e-15$
5.25.1.1	15	19	16	15	1.00	$0.14e-13$
	19	51	49	7	0.62	$0.13e-15$
	19	51	49	7	0.56	$0.13e-15$
6.4.1.1	15	17	16	15	1.00	$0.46e-08$
	19	20	20	7	1.10	$0.46e-08$
	19	20	20	7	1.00	$0.47e-08$
6.8.1.1	15	17	16	15	1.00	$0.92e-08$
	22	27	27	8	1.58	$0.63e-08$
	22	27	27	8	1.68	$0.63e-08$
6.240.1.1	15	17	16	15	1.00	$0.27e-06$
	23	38	39	6	0.39	$0.93e-06$
	20	39	40	7	0.47	$0.19e-05$
6.400.1.1	15	17	16	15	1.00	$0.45e-06$
	23	36	37	6	0.33	$0.16e-05$

was mainly that the efficiency idea is quite sensitive to the precision with which the CPU time is measured. Due to the fact that timing routines like the one provided in the IMSL Library or others available for UNIX systems do not fulfill the accuracy requirements in the sense that different runs of the same problem may give unacceptable differences for our purposes, some of the small-size problems were run on an IBM computer for which the staff of the University of LaPlata Computer Center wrote a very precise assembler routine for measuring CPU time. For several reasons, it was not feasible to run all examples on that computer, so most of the results are from the HP machine. In order to normalize comparisons, all results are given relative to the CPU time required by the IMSL optimization routines except in the examples in which they failed to converge properly. All comparisons of the new method have been made against the trust regions algorithm as implemented in subroutine DUMIAH of the IMSL Library (version 1.0, April 1987), with the only exception being the separable nonlinear least squares problems for which subroutine DUMIDH was

TABLE 2
(*continued*)

Problem	NIT	FE	GE	HE	T	FMIN
7.4.1.1	8	10	9	8	1.00	0.86e + 05
	9	16	16	5	1.26	0.86e + 05
	13	19	19	4	1.39	0.86e + 05
8.3.1.1	1	4	2	1	1.00	0.11e − 07
	2	3	3	1	0.41	0.11e − 07
	2	3	3	1	0.47	0.11e − 07
9.25.1.1	6	20	7	6	1.00	−0.75e + 04
	9	22	22	3	0.94	−0.75e + 04
	9	22	22	3	0.94	−0.75e + 04
9.50.1.1	8	26	9	8	1.00	−0.31e + 05
	13	16	15	6	0.90	−0.31e + 05
	17	28	27	5	0.91	−0.31e + 05
9.100.1.1	17	39	18	17	1.00	−0.12e + 06
	20	45	45	7	0.68	−0.12e + 06
	20	45	45	7	0.58	−0.12e + 06
9.200.1.1	23	43	64	35	1.00	−0.50e + 06
	22	43	43	8	0.70	−0.50e + 06
	22	43	43	8	0.72	−0.50e + 06
10.12.1.1	12	26	13	12	1.00	0.22e − 07
	22	52	48	8	0.97	0.23e − 07
	22	52	48	8	0.85	0.22e − 07
11.4.1.1	12	26	13	12	1.00	0.47e − 09
	12	59	56	7	0.76	0.49e − 07
	12	61	57	8	1.03	0.15e − 07
12.3.1.1	7	14	8	7	1.00	0.54e − 16
	10	14	14	4	1.00	0.14e − 11
	10	14	14	4	0.94	0.14e − 11
13.6.1.1	29	60	30	29	1.00	0.11e − 11
	33	52	46	13	0.77	0.13e − 12
	53	85	77	14	1.19	0.36e − 12
14.10.1.1	12	23	13	12	1.00	0.29e − 15
	1	7	6	1	0.76	0.23e − 21
	1	7	6	1	0.76	0.23e − 21
14.100.2.1	17	37	18	17	1.00	0.81e − 15
	1	6	6	1	0.16	0.71e − 25
	1	6	6	1	0.16	0.71e − 25
15.10.2.1	12	23	13	12	0.52	0.17e − 15
	1	10	10	1	0.19	0.38e − 22
	1	10	10	1	0.15	0.38e − 22
15.5.1.1	4	6	5	4	1.00	0.24e − 13
	5	6	6	2	1.00	0.67e − 12
	5	6	6	2	1.01	0.67e − 12
16.50.1.1	100	111	101	100	1.00	0.23e − 03
	27	73	70	8	0.20	0.23e − 03
	35	87	86	9	0.20	0.23e − 03

TABLE 2
(*continued*)

Problem	NIT	FE	GE	HE	T	FMIN
17.3.1.1	7	78	29	0	1.73	0.70e + 02
	64	182	237	0	5.91	0.33e − 18
	56	140	194	0	4.45	0.35e − 12
18.3.1.1			divergence			
	13	35	46	0	0.72	0.15e − 21
	16	36	54	0	0.87	0.92e − 25
19.3.1.1	26	84	105	0	2.97	0.42e − 18
	31	37	64	0	1.61	0.33e − 18
	31	39	72	0	1.54	0.42e − 22
20.3.1.1	4	19	17	0	0.92	0.17e − 01
	31	59	90	0	3.08	0.33e − 09
	29	59	88	0	2.93	0.93e − 07

used because a finite-difference Hessian was required.

In the following nonlinear least squares problems the absolute CPU time is given because of the poor performance of the trust-region algorithm, which led to divergence in one example, a large number of function evaluations in another, and to a very high functional value in the third.

The test examples show the new algorithm to be more robust (in fact, no example of divergence has been found) than the trust-region method, and that its efficiency tends to increase with the number of variables. This is so because of the savings in Hessian evaluations, and in spite of the CPU time spent on the computation of the least eigenvalue of the tridiagonal factor, which is relatively more important in small size problems.

7.3. Comparisons with not updating. In Table 3 are some examples to show that our update is better than if we kept the Hessian constant for q iterations. In particular, we compare not updating (we will call this method HC) against the method obtained updating the Hessian but without the test of §6.3 (WE = without efficiency).

The results of these tests convince us that our updating scheme is worthwhile. This is true despite the fact that no stronger convergence result holds for our updating scheme than for not updating.

Acknowledgments. The authors wish to thank Ms. Laura Carcione for programming help and for generating the test results.

TABLE 3

Problem	NIT	FE	GE	HE	T	FMIN	q	Method
1.3.2.1	22	23	23	8	1.00	$-0.13e+9$	4	WE
	40	41	41	14	1.39	$-0.13e+9$	4	HC
	21	22	22	4	1.00	$-0.13e+9$	6	WE
	63	64	64	11	1.73	$-0.13e+9$	6	HC
	31	32	32	4	1.00	$-0.13e+9$	10	WE
	91	92	92	10	1.62	$-0.13e+9$	10	HC
2.8.1.1	17	21	21	6	1.00	$+0.57e-4$	4	WE
	21	22	21	7	1.20	$+0.57e-4$	4	HC
	15	33	32	3	1.00	$+0.57e-4$	6	WE
	31	35	34	6	1.40	$+0.57e-4$	6	HC
	16	33	32	2	1.00	$+0.57e-4$	10	WE
	41	43	42	5	1.51	$+0.57e-4$	10	HC
10.12.1.1	22	52	48	8	1.00	$+0.22e-7$	4	WE
	51	61	58	17	2.31	$+0.43e-7$	4	HC
	37	98	92	7	1.00	$+0.24e-7$	6	WE
	72	177	159	12	1.66	$+0.42e-7$	6	HC
	51	108	102	6	1.00	$+0.43e-7$	10	WE
	96	260	232	10	1.69	$+0.43e-7$	10	HC
16.50.1.1	35	87	86	9	1.00	$+0.23e+3$	4	WE
	30	127	124	8	0.88	$+0.23e+3$	4	HC
	31	66	63	6	1.00	$+0.23e+3$	6	WE
	51	184	183	9	1.47	$+0.23e+3$	6	HC
	25	52	49	3	1.00	$+0.23e+3$	10	HC
	49	162	161	5	1.47	$+0.23e+3$	10	HC

REFERENCES

[1] C. G. BROYDEN, J. E. DENNIS, JR., AND J. J. MORÉ, *On the local and superlinear convergence of quasi-Newton methods*, IMA J. Appl. Math., 12 (1973), pp. 223–246.

[2] J. E. DENNIS, JR. AND E. S. MARWIL, *Direct secant updates of matrix factorizations*, Math. Comp., 38 (1982), pp. 459–474.

[3] J. E. DENNIS, JR. AND R. B. SCHNABEL, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, NJ, 1983. Russian edition: Mir Publishing Office, Moscow, 1988, O. Burdakov, trans.

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

[5] J. M. MARTÍNEZ, *A new family of quasi-Newton methods for nonlinear equations with direct secant updates of matrix factorizations*, SIAM J. Numer. Anal., 27 (1990) pp. 1034–1049.

[6] ———, *On the order of convergence of the Broyden–Gay–Schnabel method*, Comm. Math. Univ. Carol, 19 (1978), pp. 107–118.

[7] ———, *A quasi-Newton method with a new updating for the* LDU *factorization of the approximate Jacobian*, Mat. Apl. Comput., 2 (1983), pp. 131–142.

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

[9] A.M. OSTROWSKI, *Solution of Equations in Euclidean and Banach Spaces*, Academic Press, New York, 1973.

[10] M. J. D. POWELL AND Y. YUAN, *A trust region algorithm for equality constrained optimization*, DAMTP Report 1986/NA2, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, U.K., 1986.

[11] H. SCHRAMM AND J. ZOWE, *A combination of the bundle approach and the trust region concept*, Tech. Report Math. Inst. 1987/20, Mathematisches Institut, University of Bayreuth, Bayreuth, Germany, 1987.

[12] J.H. WILKINSON, *The Algebraic Eigenvalue Problem*, Oxford University Press, London, 1965.