SIAM J. OPTIMIZATION
Vol. 6, No. 4, pp. 1025–1039, November 1996
© 1996 Society for Industrial and Applied Mathematics
008

Downloaded 01/02/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

ANALYSIS OF A SYMMETRIC RANK-ONE TRUST REGION METHOD*

RICHARD H. BYRD†, HUMAID FAYEZ KHALFAN‡, AND ROBERT B. SCHNABEL†

Abstract. Several recent computational studies have shown that the symmetric rank-one (SR1) update is a very competitive quasi-Newton update in optimization algorithms. This paper gives a new analysis of a trust region SR1 method for unconstrained optimization and shows that the method has an $n+1$ step q-superlinear rate of convergence. The analysis makes neither of the assumptions of uniform linear independence of the iterates nor positive definiteness of the Hessian approximations that have been made in other recent analyses of SR1 methods. The trust region method that is analyzed is fairly standard, except that it includes the feature that the Hessian approximation is updated after all steps, including rejected steps. We also present computational results that show that this feature, safeguarded in a way that is consistent with the convergence analysis, does not harm the efficiency of the SR1 trust region method.

Key words. quasi-Newton method, symmetric rank-one update, superlinear convergence

AMS subject classifications. 65, 49

1. Introduction.

The symmetric rank-one (SR1) secant update has been shown to be very effective in optimization calculations, especially when used in conjunction with a trust region method. The theoretical understanding of its convergence behavior on nonquadratic problems is still rather incomplete, however. This paper improves that understanding by providing a local convergence analysis of a trust region, SR1 method for unconstrained optimization under fairly mild assumptions. In doing so it extends the work in Conn, Gould, and Toint [1991] and in Khalfan, Byrd, and Schnabel [1993]. The new results are particularly interesting, first because a trust region method can be argued to be the appropriate context for the SR1 and second because the trust region structure allows some of the less desirable assumptions made by Khalfan, Byrd, and Schnabel [1993] in analyzing the SR1 for a line search method to be relaxed. In particular, no assumption about the positive definiteness of the Hessian approximations is made.

In solving the nonlinear unconstrained minimization problem

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

the SR1 update can be used to approximate the Hessian matrix of the objective function $\nabla^2 f(x)$. Given an approximation B_k to the Hessian, the SR1 formula updates it by

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{s_k^T(y_k - B_k s_k)}, \tag{1.2}$$

where s_k is a step and $y_k = \nabla f(x_k + s_k) - \nabla f(x_k)$. Minimization algorithms using this update in both a line search and a trust region context have been shown in

* Received by the editors July 30, 1993; accepted for publication (in revised form) May 12, 1995. This research was supported by Air Force Office of Scientific Research grants AFOSR-90-0109 and F49620-94-1-0101, Army Research Office grants DAAL 03-91-G-0151 and DAAH04-94-0228, and National Science Foundation grant CCR-9101795.

† Department of Computer Science, Campus Box 430, University of Colorado, Boulder, CO 80309 (richard@hessian.cs.colorado.edu, bobby@hessian.cs.colorado.edu).

‡ Department of Mathematics and Computer Science, United Arab Emirates University, Al-Ain, United Arab Emirates.

Downloaded 01/02/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

computational experiments by Conn, Gould, and Toint [1988, 1992] and by Khalfan, Byrd, and Schnabel [1993] to be competitive with methods using the widely accepted Broyden–Fletcher–Goldfarb–Shanno (BFGS) update. The convergence of such algorithms is not as well understood, however, as convergence of the BFGS method. For example, no convergence results that are similar to the global results for the line search BFGS method given by Powell [1975] or the local superlinear results of Broyden, Dennis, and Moré [1973] exist, so far, for the SR1 method when applied to a nonquadratic function. A significant difference between the BFGS and SR1 updates that contributes to this situation is that although the BFGS algorithm is guaranteed to produce a positive definite B_{k+1} if B_k is positive definite and $s_k^T y_k > 0$, the SR1 update does not have this property. In practical implementations of an SR1 method, the Hessian approximations B_k may be indefinite at some iterations.

Conn, Gould, and Toint [1991] proved that the sequence of matrices generated by the SR1 formula converges to the actual Hessian at the solution $\nabla^2 f(x_*)$, provided that the sequence of steps taken, $\{s_k\}$, is uniformly linearly independent, that the denominator in (1.2) is sufficiently different from zero, and that the iterates converge to x_*. Using this result it is simple to prove that the rate of convergence under these assumptions is q-superlinear. (Interestingly, for the BFGS method Ge and Powell [1983] proved, under a different set of assumptions, that the sequence of generated matrices converges but not necessarily to $\nabla^2 f(x_*)$.) The assumption of uniform linear independence is unusual and fairly strong, but Conn, Gould, and Toint [1991] present experiments on minimization of randomly generated quartics in which it holds. On the other hand, experiments by Khalfan, Byrd, and Schnabel [1993] showed that nearly dependent steps occurred commonly when using the SR1 in both a line search and a trust region method on a standard set of unconstrained optimization test problems.

The study of the SR1 in a line search context by Khalfan, Byrd, and Schnabel [1993] did not make a uniform linear independence assumption and instead made the weaker assumptions that the matrices B_k are positive definite and uniformly bounded. This analysis showed the convergence rate of the iterates to be $n+1$-step q-superlinear and $2n$-step q-quadratic. Although the matrices B_k did appear in experiments to stay bounded, they were not always positive definite at every iterate. In fact, an example was constructed in that paper in which B_k is indefinite at every iterate and convergence is only linear.

In this paper we show that when used with a trust region method, indefiniteness does not cause such problems to the local convergence properties of the SR1 method. In §2 we describe in detail a class of SR1 trust region methods that is fairly standard with one exception and show that the superlinear convergence result proved in Khalfan, Byrd, and Schnabel [1993] holds for this method without any assumption of positive definiteness. The one unusual feature of the method that is analyzed is that it updates the Hessian approximation B_k at all steps, including rejected steps. In §3 we briefly present computational results that show that this feature, when safeguarded in a manner that is consistent with the convergence analysis, is reasonable in practice in comparison with the more common alternative of updating only at successful steps.

We will use the symbol $\|x\|$ to denote an arbitrary norm of a vector x. When the argument is a matrix the norm is the corresponding induced matrix norm.

2. Convergence results. In this section we establish convergence properties of the SR1 update in the following general trust region algorithm. In a trust region method, an attempt is made to search for an acceptable step in a neighborhood around x_k, instead of searching only in one direction as is done in line search algorithms.

Specifically the quadratic model problem

$$\text{(2.1)} \qquad \text{minimize } \ m(x_k + s) = f(x_k) + \nabla f(x_k)^T s + \frac{1}{2} s^T B_k s$$

$$\text{(2.2)} \qquad \text{subject to } \ \|s\| \le \Delta_k$$

is solved approximately, where B_k is an approximation to the Hessian matrix and Δ_k is a positive number representing the radius of the ball around the current point in which $m(x_k + s)$ is trusted to represent the objective function. The radius of the trust region, Δ_k, is adjusted from iteration to iteration by monitoring the actual reduction in functional values,

$$\text{ared}_k = f(x_k) - f(x_k + s_k),$$

and the reduction predicted by the quadratic model,

$$\text{pred}_k = f(x_k) - m(x_k + s_k),$$

where s_k is an approximate minimizer of $m(x_k + s)$ over $\|s\| \le \Delta_k$. If $\frac{\text{ared}_k}{\text{pred}_k}$ is close to 1, this indicates that the quadratic model represents $f(x)$ well enough and the trust region radius may be increased. If, on the other hand, this ratio is smaller than a prescribed small number, the size of the trust region is reduced. For a background on trust region algorithms and their global convergence, see, for example, Moré and Sorensen [1983] and Shultz, Schnabel, and Byrd [1985].

Below we give a formal description of the algorithm that we will analyze. It is meant to be a fairly typical trust region method, with the exception of Step 6. The step computed is a possibly approximate solution to the trust region problem, which is required to be accurate enough that for all k

$$\text{(2.3)} \qquad \text{pred}_k \ge \sigma_1 \|\nabla f(x_k)\| \min\{\Delta_k, \sigma_2 \|\nabla f(x_k)\| / \|B_k\|\}$$

for positive constants σ_1 and σ_2 and such that

$$\text{(2.4)} \qquad \text{whenever } \|s_k\| < 0.8\Delta_k, \text{ then } B_k s_k = -\nabla f(x_k).$$

These conditions are satisfied if the trust region problem (2.1, 2.2) is solved exactly and are satisfied by most approximate trust region strategies that are used in practice (see, e.g., Shultz, Schnabel, and Byrd [1985]). Note that in the notation used, it can happen that $x_{k+1} = x_k$ if the step s_k is rejected even though the update (1.2) is made.

As described in Khalfan, Byrd, and Schnabel [1993], we skip the update if the denominator in (1.2) is too small. Specifically, we update by (1.2) only if

$$\text{(2.5)} \qquad |s_k^T (y_k - B_k s_k)| \ge r \|s_k\| \|y_k - B_k s_k\|,$$

where r is a constant $\in (0, 1)$; otherwise, we set $B_{k+1} = B_k$.

ALGORITHM 2.1.

0) Choose an initial point x_0, an initial symmetric matrix B_0, an initial trust region radius Δ_0, $\eta \in (0, 0.1)$, $\tau_1 \in (0, 1)$, $\tau_2 > 1$, and set $k = 0$.
1) Compute $f(x_k)$ and $\nabla f(x_k)$. If $\nabla f(x_k) = 0$, then stop.

Downloaded 01/02/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 01/02/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

2) Compute a trust region step: find a feasible approximate solution s_k to

$$\min_{s \in R^n} \nabla f(x_k)^T s + \frac{1}{2} s^T B_k s \quad \text{subject to } \|s\| \leq \Delta_k$$

such that (2.3) and (2.4) are satisfied for some positive constants, σ_1 and σ_2.

3) Compute $\text{ared}_k = f(x_k) - f(x_k + s_k)$ and $\text{pred}_k = -(\nabla f(x_k)^T s_k + \frac{1}{2} s_k^T B_k s_k)$.

4) If $\dfrac{\text{ared}_k}{\text{pred}_k} > \eta$, then
   set $x_{k+1} = x_k + s_k$;
   else set $x_{k+1} = x_k$.

5) If $\dfrac{\text{ared}_k}{\text{pred}_k} > 0.75$, then
   if $\|s_k\| < 0.8\Delta_k$, then set $\Delta_{k+1} = \Delta_k$;
   else set $\Delta_{k+1} = \tau_2 \Delta_k$;
else
   if $0.1 \leq \dfrac{\text{ared}_k}{\text{pred}_k} \leq 0.75$, then set $\Delta_{k+1} = \Delta_k$;
   else set $\Delta_{k+1} = \tau_1 \Delta_k$.

6) Compute the next Hessian approximation B_{k+1} using the SR1 update along the direction s_k. (Observe here that the update is made even if $x_{k+1} = x_k$.)

7) Set $k = k + 1$ and go to Step 1.

Note that the trust region updating procedure in Step 5 requires increasing the trust region radius if there is good model-function agreement. Such a feature is usually present in implemented algorithms and is essential to our analysis. We specify the increase and decrease factors, τ_2 and τ_1, to be constants. Actually it is only essential to our results that the decrease factor be bounded away from zero, and the increase factor be bounded above, and bounded below by a number greater than one. However, considering such possibilities would complicate some proofs significantly. Also for simplicity we have specified reasonable numerical values, 0.1 and 0.75, for the algorithmic constants used in deciding when to decrease or increase the trust region radius, but in our analysis any values in (0,1) will work. We have not included the empirically efficient device of internal doubling (i.e., letting $x_{k+1} = x_k$ while increasing the radius when model-function agreement is very good) in order to keep the analysis simpler, even though such a device could be handled by similar analysis.

It is essential to the convergence analysis that even when $x_{k+1} = x_k$, the matrix B_k is updated along the failed direction s_k to get B_{k+1}. Analyses of Powell–Symmetric–Broyden (PSB) trust region methods by Powell [1970, 1975] make a similar provision, as does Khalfan [1989] in analyzing a related method. As far as we know these are the only other analyses proving superlinear convergence of a trust region secant method. Such updates along failed directions seem to be necessary to the convergence analysis because if the Hessian approximation is incorrect along such a direction and is not updated along that direction very similar directions could, in principle, be generated repeatedly at later iterations. Such steps would be rejected, resulting in the trust region being reduced at these iterations, potentially keeping it small enough that it would interfere with making a superlinear step even if the Hessian approximation is accurate enough to provide one. In §3 we present some numerical experiments that assess the effects and costs of this strategy of updating at failed points. It is possible to include safeguards on this strategy that do not make updates at "very bad" failed points which do not interfere with the theoretical results

Downloaded 01/02/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

in this paper. We have included such a safeguard in our implementation in §3 but for simplicity have not used it in the algorithm that we analyze.

In this section we will show that under certain assumptions the rate of convergence of Algorithm 2.1 is $n+1$-step q-superlinear. This is the same kind of superlinear convergence established for a line search SR1 method in Khalfan, Byrd, and Schnabel [1993]. As in that paper we will first show that for a large fraction of the iterates near the solution $B_k s_k$ is quite close to $\nabla^2 f(x_*) s_k$ and that if such an s_k satisfies $B_k s_k = -\nabla f(x_k)$ good progress is made toward the solution.

Throughout this section the following assumptions will frequently be made.

Assumptions.

(A1) The sequence of iterates does not terminate and remains in a closed, bounded, convex set D on which the function f is twice continuously differentiable and in which f has a unique stationary point x_*. The Hessian $\nabla^2 f(x_*)$ is positive definite, and $\nabla^2 f(x)$ is Lipschitz continuous in a neighborhood of x_*; that is, there exists a constant $\gamma > 0$ such that for all x, y in some neighborhood of x_*

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \le \gamma \|x - y\|.$$

(A2) The sequence of matrices $\{B_k\}$ is generated from each iterate x_k by the SR1 update, using s_k, and for each iteration

$$|s_k^T (y_k - B_k s_k)| \ge r \|s_k\| \|y_k - B_k s_k\|, \tag{2.6}$$

where r is a constant $\in (0,1)$.

(A3) The sequence of matrices $\{B_k\}$ is bounded by a constant M such that $\|B_k\| \le M$ for all k.

The assumptions that the iterates remain in a bounded set and that the denominator of the SR1 update obeys (2.6) appear reasonable in practice and have been made in the analyses of Conn, Gould, and Toint [1991] and Khalfan, Byrd, and Schnabel [1993]. Assumption (A2) implies that no updates are skipped due to violation of condition (2.6). As described in Khalfan, Byrd, and Schnabel [1993], this is usually the case in practice. The assumption that $\|B_k\|$ is bounded above is made in Khalfan, Byrd, and Schnabel [1993] (although not in Conn, Gould, and Toint [1991] due to their stronger assumption of uniform linear independence) and also appears to be reasonable in practice. Note that the latter two assumptions are not needed in analyses of the BFGS method.

Note that Assumption (A1) implies that if we choose positive constants

$$\beta_1 < \frac{1}{\|\nabla^2 f(x_*)^{-1}\|} \text{ and } \beta_2 > \|\nabla^2 f(x_*)\| \tag{2.7}$$

then if $\|x_k - x_*\|$ is sufficiently small

$$\beta_1 \|x_k - x_*\| \le \|\nabla f(x_k)\| \le \beta_2 \|x_k - x_*\| \tag{2.8}$$

and

$$\frac{2}{\beta_2}[f(x_k) - f(x_*)] \le \|x_k - x_*\|^2 \le \frac{2}{\beta_1}[f(x_k) - f(x_*)]. \tag{2.9}$$

First we point out that these assumptions are enough to imply convergence to x_* using standard global convergence theory.

Downloaded 01/02/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

THEOREM 2.1. *If the sequence $\{x_k\}$ is generated by Algorithm* 2.1 *and Assumptions* (A1) *and* (A3) *hold, then $x_k \to x_*$.*

Proof. Note that Algorithm 2.1 is a trust region method of the type considered by Theorem 2.2 in Shultz, Schnabel, and Byrd [1985]. Since the assumption that $\{B_k\}$ be bounded is satisfied, it follows from that theorem that all cluster points of $\{x_k\}$ are stationary points of f. By Assumption (A1) the sequence stays in a compact set and thus has at least one cluster point, which is a stationary point. Since there is only one stationary point, x_*, the entire sequence must converge to x_*. □

Now we establish a general relation between steplength and trust region which in fact holds for essentially any trust region method, provided the iterates converge.

LEMMA 2.2. *Suppose that the sequence of iterates $\{x_k\}$, generated by Algorithm* 2.1, *converges to x_*. Then either*

$$\Delta_k \to 0 \tag{2.10}$$

or there exist $K > 0$ and $\Delta > 0$ such that for all $k > K$

$$\Delta_k = \Delta. \tag{2.11}$$

In either case $s_k \to 0$.

Proof. Let $\Delta = \liminf \Delta_k$ and suppose first that $\Delta > 0$. From Step 5 of Algorithm 2.1, Δ_k is increased only if $\|x_{k+1} - x_k\| \geq 0.8\Delta_k$. Therefore since $\|x_{k+1} - x_k\| \to 0$, there exists $K \geq 0$ such that Δ_k is not increased for any $k \geq K$. Since $\Delta > 0$, this implies that $\Delta_k \geq \Delta$ for all $k \geq K$. Also we know that for some $K_1 > K$, $\Delta_{K_1} < \frac{1}{\tau_1}\Delta$. If the trust region radius were to be decreased we would have $\Delta_{K_1+1} < \Delta$, which we have ruled out. Since neither increase nor decrease can occur we must have $\Delta_k = \Delta$ for all $k \geq K_1$.

Suppose now that $\Delta = 0$. Since $x_k \to x_*$, for every $\epsilon > 0$ there exists $K_\epsilon \geq 0$ such that $\|x_{k+1} - x_k\| < \epsilon$ for all $k \geq K_\epsilon$. Since $\liminf \Delta_k = 0$, there exists $j \geq K_\epsilon$ such that $\Delta_j < \epsilon$. But since Δ_k is increased only if $\Delta_k \leq \frac{1}{0.8}\|x_{k+1} - x_k\| < \frac{\epsilon}{0.8}$, and the increase factor is τ_2, we have that $\Delta_k < \frac{\tau_2\epsilon}{0.8}$ for all $k \geq j$. Therefore (2.10) follows.

To show that $s_k \to 0$, note that if (2.10) is true, then clearly $s_k \to 0$. If (2.11) is true, then for all $k \geq K$, the step s_k is accepted and $s_k = \|x_{k+1} - x_k\|$. Thus $s_k \to 0$ by convergence. □

In analyzing the convergence rate of our algorithm we will make reference to the following result, which is a slight variation of Lemma 3.2 from Khalfan, Byrd, and Schnabel [1993]. It states that if the sequence of steps generated by an iterative process using the SR1 update satisfies Assumptions (A1) and (A2) and the sequence of matrices, $\{B_k\}$, is bounded, then out of any set of $n+1$ steps, B_k is accurate along at least one step.

LEMMA 2.3. *Let $\{x_k\}$ be a sequence of iterates which converges to the local minimizer x_* of f, and let s_k be a sequence of vectors such that $x_k + s_k \to x_*$. Suppose that Assumptions* (A1)–(A3) *hold. Then there exists $K \geq 0$ such that for any set of $n+1$ steps $\mathcal{S} = \{s_{k_j} : K \leq k_1 < \cdots < k_{n+1}\}$ there exists an index k_m with $m \in \{2, 3, \ldots, n+1\}$ such that*

$$\frac{\|(B_{k_m} - \nabla^2 f(x_*))s_{k_m}\|}{\|s_{k_m}\|} < \bar{c}\epsilon_{\mathcal{S}}^{\frac{1}{n}}, \tag{2.12}$$

where

$$\epsilon_{\mathcal{S}} = \max_{1\leq j\leq n+1}\{\|x_{k_j} - x_*\|, \|x_{k_j} + s_{k_j} - x_*\|\}$$

Downloaded 01/02/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

and

$$\bar{c} = 4\left[\gamma + \sqrt{n}\frac{\gamma}{2}\left(\frac{2}{r}+1\right)^{k_{n+1}-k_1-2} + M + \|\nabla^2 f(x_*)\|\right].$$

Actually Lemma 3.2 from Khalfan, Byrd, and Schnabel [1993] assumed in addition that $x_{k+1} = x_k + s_k$, but this fact is not used except in the definition of $\epsilon_{\mathcal{S}}$, and we have taken this difference into account in defining $\epsilon_{\mathcal{S}}$ in (2.12).

Applying Lemma 2.3 in a trust region context we can show that out of any set of p steps generated by Algorithm 2.1 where $p > n$ there are at least $p - n$ steps that are accepted by Algorithm 2.1, and for which the approximate Hessian is accurate.

LEMMA 2.4. *Suppose that Assumptions* (A1)–(A3) *are satisfied. Then there exists N such that for any set of $p > n$ consecutive steps $s_{k+1}, s_{k+2}, \dots, s_{k+p}$ with $k \geq N$, there exists a set, $\mathcal{G}_k$, of at least $p-n$ indices contained in the set $\{i : k+1 \leq i \leq k+p\}$ such that for all $j \in \mathcal{G}_k$,*

$$\frac{\|(B_j - \nabla^2 f(x_*))s_j\|}{\|s_j\|} < c\epsilon_k^{\frac{1}{n}}, \tag{2.13}$$

where

$$\epsilon_k = \max_{k+1\leq j\leq k+p}\{\|x_j - x_*\|, \|x_j + s_j - x_*\|\}$$

and

$$c = 4\left[\gamma + \sqrt{n}\frac{\gamma}{2}\left(\frac{2}{r}+1\right)^{p-2} + M + \|\nabla^2 f(x_*)\|\right].$$

Furthermore, for k sufficiently large, if $j \in \mathcal{G}_k$, then

$$\|s_j\| < \frac{2\beta_2}{\beta_1}\|x_j - x_*\| \tag{2.14}$$

and

$$\frac{\text{ared}_j}{\text{pred}_j} \geq 0.75. \tag{2.15}$$

Proof. By Theorem 2.1 $x_k \to x_*$ and by Lemma 2.2 $s_k \to 0$. Therefore, by Lemma 2.3, applied to the set

$$\{s_k, s_{k+1}, \dots, s_{k+p}\}, \tag{2.16}$$

there exists N such that for any $k \geq N$ there exists an index l_1, with $k+1 \leq l_1 \leq k+p$ satisfying

$$\|(B_{l_1} - \nabla^2 f(x_*))s_{l_1}\| < c\epsilon_k^{\frac{1}{n}}\|s_{l_1}\|, \tag{2.17}$$

where the fact that the set (2.16) consists of $p+1$ consecutive steps allows us to replace $\bar{c}$ by c. Now we can apply Lemma 2.3 to the set $\{s_k, s_{k+1}, \dots, s_{k+p}\} - s_{l_1}$ to

Downloaded 01/02/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

get l_2. Repeating this $p-n$ times we get a set of $p-n$ indices $\mathcal{G}_k = \{l_1, l_2, \ldots, l_{p-n}\}$ such that if $j \in \mathcal{G}_k$, then

$$\|(B_j - \nabla^2 f(x_*))s_j\| < c\epsilon_k^{\frac{1}{n}} \|s_j\|. \tag{2.18}$$

Now consider $j \in \mathcal{G}_k$. By (2.7) and (2.18) for k sufficiently large $s_j^T B_j s_j > \beta_1 \|s_j\|^2$. Therefore using (2.8) with j large enough we have

$$\begin{aligned} 0 \le \text{pred}_j &= -\nabla f(x_j)^T s_j - \frac{1}{2} s_j^T B_j s_j \\ &< \|\nabla f(x_j)\| \|s_j\| - \frac{1}{2}\beta_1 \|s_j\|^2 \\ &\le \beta_2 \|x_j - x_*\| \|s_j\| - \frac{1}{2}\beta_1 \|s_j\|^2. \end{aligned}$$

Therefore (2.14) holds for $j \in \mathcal{G}_k$, for k sufficiently large.

Now let $j \in \mathcal{G}_k$. Then using Taylor's theorem (see, e.g., Ortega and Rheinboldt [1970]), the Lipschitz continuity of $\nabla^2 f(x)$, (2.18) and (2.14) and assuming k is large enough we have that

$$\begin{aligned} |\text{ared}_j - \text{pred}_j| &= \left| f(x_j) - f(x_j + s_j) - \left(f(x_j) - f(x_j) - \nabla f(x_j)^T s_j - \frac{1}{2} s_j^T B_j s_j \right) \right| \\ &\le \left| \frac{1}{2} s_j^T B_j s_j - \int_0^1 s_j^T \nabla^2 f(x_j + \zeta s_j) s_j (1-\zeta) d\zeta \right| \\ &\le \|s_j\| \int_0^1 \|(B_j - \nabla^2 f(x_j + \zeta s_j)) s_j\| (1-\zeta) d\zeta \\ &\le \|s_j\| \|(B_j - \nabla^2 f(x_*)) s_j\| \\ &\quad + \|s_j\|^2 \int_0^1 \|\nabla^2 f(x_*) - \nabla^2 f(x_j + \zeta s_j)\| (1-\zeta) d\zeta \\ &\le \|s_j\| \|(B_j - \nabla^2 f(x_*)) s_j\| + \gamma \|s_j\|^2 (\|x_j - x_*\| + \|s_j\|) \\ &\le c\epsilon_k^{\frac{1}{n}} \|s_j\|^2 + \gamma (\|s_j\|^2 \|x_j - x_*\| + \|s_j\|^3) \\ &\le \left[c\epsilon_k^{\frac{1}{n}} + \gamma \left(1 + \frac{2\beta_2}{\beta_1} \right) \|x_j - x_*\| \right] \|s_j\|^2 \\ &\le 2c\epsilon_k^{\frac{1}{n}} \|s_j\|^2, \end{aligned} \tag{2.19}$$

where in the last step we note that for k, and thus j, large enough $\|x_j - x_*\|/c\epsilon_k^{\frac{1}{n}}$ becomes arbitrarily small.

Since $\|s_j\| < \left(\frac{2\beta_2}{\beta_1}\right) \|x_j - x_*\| \le \left(\frac{2\beta_2}{\beta_1^2}\right) \|\nabla f(x_j)\|$ and $\|s_j\| \le \Delta_j$ it follows that

$$\|s_j\|^2 \le \left(\frac{2\beta_2}{\beta_1^2}\right) \|\nabla f(x_j)\| \min \left\{ \Delta_j, \left(\frac{2\beta_2}{\beta_1^2}\right) \|\nabla f(x_j)\| \right\}. \tag{2.20}$$

From the condition (2.3) on Step 2 of the algorithm, we have that

$$\text{pred}_j \ge \sigma_1 \|\nabla f(x_j)\| \min \left\{ \Delta_j, \sigma_2 \frac{\|\nabla f(x_j)\|}{\|B_j\|} \right\}. \tag{2.21}$$

Therefore using (2.19), (2.20), and (2.21) we have that

$$|\text{ared}_j - \text{pred}_j| \le \alpha \epsilon_k^{\frac{1}{n}} |\text{pred}_j|,$$

Downloaded 01/02/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

where α is a constant, which implies (2.15). $\square$

The following lemma is a slight extension of Lemma 3.3 given in Khalfan, Byrd, and Schnabel [1993], which is a sort of single step version of the Dennis–Moré superlinear convergence condition.

LEMMA 2.5. *Suppose the function f satisfies Assumption* (A1). *If the quantities*

$$e_k = \|x_k - x_*\| \text{ and } \frac{\|(B_k - \nabla^2 f(x_*))s_k\|}{\|s_k\|}$$

are sufficiently small and if $B_k s_k = -\nabla f(x_k)$, then

$$\|x_k + s_k - x_*\| \leq \|\nabla^2 f(x_*)^{-1}\| \left[2\frac{\|(B_k - \nabla^2 f(x_*))s_k\|}{\|s_k\|}e_k + \frac{\gamma}{2}e_k^2\right] \tag{2.22}$$

and

$$h(x_k + s_k) \leq \sqrt{\frac{\beta_2}{\beta_1}}\|\nabla^2 f(x_*)^{-1}\| \left[2\frac{\|(B_k - \nabla^2 f(x_*))s_k\|}{\|s_k\|}h(x_k) + \frac{\gamma}{\sqrt{2\beta_1}}h(x_k)^2\right], \tag{2.23}$$

where $h(x) = [f(x) - f(x_*)]^{\frac{1}{2}}$.

Proof. Inequality (2.22) is proved in Lemma 3.3 of Khalfan, Byrd, and Schnabel [1993], and (2.23) follows from (2.22) using (2.9) three times. $\square$

Up to now the analysis has been quite similar to the SR1 line search analysis of Khalfan, Byrd, and Schnabel [1993]. We have shown that for many steps $B_k s_k$ is accurate and the step is accepted, and Lemma 2.5 says that when the trust region is inactive such a step results in good progress toward the solution. However, since B_k can be indefinite, we must worry about the possibility that the trust region interferes with progress toward the solution. In the following lemma we show that in fact the trust region radius is much larger than the distance to the solution (measured, for convenience, by $h(x_k)$) for all iterates with k sufficiently large, so that the trust region will not interfere with rapid progress when $B_k s_k$ is accurate. This is the case because, since at least $p - n$ out of p steps are accurate, trust radius decreases occur less often than trust radius increases and very good steps.

LEMMA 2.6. *Suppose the assumptions of Lemma* 2.4 *are satisfied and the computed step satisfies* (2.4). *Then, letting $h_k = h(x_k)$,*

$$\lim_{k\to\infty} \frac{h_k}{\Delta_k} = 0. \tag{2.24}$$

Proof. Let p be the smallest integer greater than $2n + n(-\log \tau_1/\log \tau_2)$. Then

$$\tau_1^n \tau_2^{p-2n} \geq 1, \tag{2.25}$$

and applying Lemma 2.4 with this value of p there exists N such that if $k \geq N$, then there exists a set of at least $p - n$ indices, $\mathcal{G}_k \subset \{j : k + 1 \leq j \leq k + p\}$, such that if $j \in \mathcal{G}_k$, then $\|(B_j - \nabla^2 f(x_*))s_j\|/\|s_j\| < c\epsilon_k^{\frac{1}{n}}$ and

$$\frac{\text{ared}_j}{\text{pred}_j} \geq 0.75. \tag{2.26}$$

Downloaded 01/02/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

We show that this implies that for such steps

$$\frac{h_{j+1}}{\Delta_{j+1}} \leq \frac{1}{\tau_2}\frac{h_j}{\Delta_j}. \tag{2.27}$$

If $\|s_j\| \geq 0.8\Delta_j$, then since from Step 5 of Algorithm 2.1 $\Delta_{j+1} = \tau_2\Delta_j$ and since $\{h_i\}$ is decreasing, (2.27) follows. If on the other hand $\|s_j\| < 0.8\Delta_j$, then from Step 5 of Algorithm 2.1 we have that

$$\Delta_{j+1} = \Delta_j.$$

Also since the trust region is inactive, by condition (2.4) $B_j s_j = -\nabla f(x_j)$. Therefore (2.13) and (2.23) imply that if N is large enough, we have that

$$h_{j+1} \leq \frac{1}{\tau_2} h_j.$$

This implies that (2.27) is true for $j \in \mathcal{G}_k$, where $k \geq N$.

In addition, note that for any j, $h_{j+1} \leq h_j$ and $\Delta_{j+1} = \tau_1\Delta_j$ and so

$$\frac{h_{j+1}}{\Delta_{j+1}} \leq \frac{1}{\tau_1}\frac{h_j}{\Delta_j}. \tag{2.28}$$

Therefore (2.27) is true for $p-n$ values of $j \in \mathcal{G}_k$, and (2.28) holds for all j. Thus for all $k \geq N$, we have using (2.25) that

$$\frac{h_{k+p}}{\Delta_{k+p}} \leq \left(\frac{1}{\tau_1}\right)^n \left(\frac{1}{\tau_2}\right)^{p-n} \frac{h_k}{\Delta_k} \leq \left(\frac{1}{\tau_2}\right)^n \frac{h_k}{\Delta_k}. \tag{2.29}$$

Therefore, starting at $k = N$, it follows that

$$\frac{h_{N+lp}}{\Delta_{N+lp}} \to 0$$

as $l \to \infty$. Using (2.28) again, (2.24) is implied. $\square$

Now that we know the trust region is eventually inactive at most steps, it is fairly easy to show that the rate of convergence of the sequence $\{x_k\}$ generated by Algorithm 2.1 is $n+1$-step q-superlinear.

THEOREM 2.7. *Consider Algorithm* 2.1 *satisfying* (2.3) *and* (2.4) *and suppose that Assumptions* (A1)–(A3) *hold. Then the sequence* $\{x_k\}$ *generated by Algorithm* 2.1 *is* $n+1$*-step q-superlinear*; *i.e.*,

$$\frac{e_{k+n+1}}{e_k} \to 0.$$

Proof. By Lemma 2.4 there exists N such that if $k \geq N$, then the set of steps $\{s_{k+1}, \ldots, s_{k+n+1}\}$ contains at least one step s_{k+j}, $1 \leq j \leq n+1$, for which

$$\|(B_{k+j} - \nabla^2 f(x_*))s_{k+j}\|/\|s_{k+j}\| < c\epsilon_k^{\frac{1}{n}}.$$

By (2.14) and (2.9)

$$\|s_{k+j}\| < \frac{2\beta_2}{\beta_1} e_{k+j} \leq \left(\frac{2}{\beta_1}\right)^{\frac{3}{2}} \beta_2 h_{k+j}.$$

Therefore by Lemma 2.6 if N is large enough and $k \geq N$, then $\|s_{k+j}\| < 0.8\Delta_{k+j}$. This implies $B_{k+j}s_{k+j} = -\nabla f(x_{k+j})$. Therefore by Theorem 2.1 $e_k \to 0$, and thus by inequality (2.23) of Lemma 2.5 if N is large enough and $k \geq N$, then

$$h_{k+j+1} = h(x_{k+j} + s_{k+j}) \leq \sqrt{\frac{\beta_2}{\beta_1}}\|\nabla^2 f(x_*)^{-1}\| \left(2c\epsilon_k^{\frac{1}{n}} + \frac{\gamma}{\sqrt{2\beta_1}}h_{k+j}\right) h_{k+j}. \tag{2.30}$$

The first equality holds because (2.15) implies that the step is accepted.

Since the sequence $\{h_i\}$ is decreasing this implies that

$$h_{k+n+1} \leq \sqrt{\frac{\beta_2}{\beta_1}}\|\nabla^2 f(x_*)^{-1}\| \left(2c\epsilon_k^{\frac{1}{n}} + \frac{\gamma}{\sqrt{2\beta_1}}h_k\right) h_k.$$

Therefore by (2.9) we have that

$$\begin{aligned} e_{k+n+1} &\leq \sqrt{\frac{2}{\beta_1}}h_{k+n+1} \\ &\leq \frac{\sqrt{2\beta_2}}{\beta_1}\|\nabla^2 f(x_*)^{-1}\| \left(2c\epsilon_k^{\frac{1}{n}} + \frac{\gamma}{\sqrt{2\beta_1}}h_k\right) h_k \\ &\leq \frac{\sqrt{2\beta_2}}{\beta_1}\|\nabla^2 f(x_*)^{-1}\| \left(2c\epsilon_k^{\frac{1}{n}} + \frac{\gamma}{2}\sqrt{\frac{\beta_2}{\beta_1}}e_k\right) \sqrt{\frac{\beta_2}{2}}e_k. \end{aligned}$$

Thus $n+1$-step q-superlinear convergence follows. □

As we noted before, this $n+1$ step q-superlinear convergence rate is the same as that proved for the SR1 with a line search by Khalfan, Byrd, and Schnabel [1993] but with a trust region method there is no assumption of positive definiteness of B_k required. We could not establish a result corresponding to the $2n$-step quadratic convergence in that paper; however, it would be possible to show that in the limit the subsequence of iterations on which superlinear convergence occurs includes almost all iterates.

It is natural to ask to what extent the matrices B_k are in fact asymptotically positive definite if Algorithm 2.1 is used. It turns out that slightly stronger conditions on the step computations are required in order to guarantee any positive semidefiniteness.

Example. Let $f(x) = x_{(1)}^2 + x_{(1)}^3 + x_{(2)}^2$ with gradient

$$\nabla f(x) = \left(2x_{(1)} + 3x_{(1)}^2,\ 2x_{(2)}\right)^T$$

so that there is a local minimizer at $x = 0$. Consider an iterate and a Hessian approximation of the form

$$x_k = \begin{bmatrix} \xi \\ 0 \end{bmatrix}, \text{ and } B_k = \begin{bmatrix} \beta & 0 \\ 0 & -1 \end{bmatrix},$$

where ξ and β are constants and $\beta > 0$. Assume Δ_k is relatively large so that the step $-B_k^{-1}\nabla f(x_k)$ satisfies (2.4) as well as (2.3) and is selected. Thus the step is

$$s = \left(-(2\xi + 3\xi^2)/\beta,\ 0\right)^T,$$

and the next iterate is

$$x_{k+1} = \left(\xi - (2\xi + 3\xi^2)/\beta,\ 0\right)^T.$$

Downloaded 01/02/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 01/02/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

The difference in gradients has the form

$$y = \left(y_{(1)},\ 0\right)^T.$$

By (1.2) the SR1 update gives a new Hessian approximation of the form

$$B_{k+1} = \begin{bmatrix} \beta' & 0 \\ 0 & -1 \end{bmatrix},$$

where β' is a one-dimensional secant approximation to $\frac{\partial^2 f}{\partial x_{(1)}^2}$, using $(x_k)_{(1)}$ and $(x_{k+1})_{(1)}$. Thus the method behaves like the one-dimensional secant method on $x_{(1)}$, while $x_{(2)} = 0$ and $(B_k)_{2,2} = -1$ for all iterates.

However, we can guarantee that the Hessian approximations are eventually positive definite most of the time if we further qualify the step selection strategy. Requiring s_k to solve (2.1), (2.2) exactly would do this, but it turns out that all we need to require, in addition to (2.3) and (2.4), is that a Newton step is taken only if B_k is positive semidefinite.

THEOREM 2.8. *Suppose that Algorithm* 2.1 *is run, that Assumptions* (A1)–(A3) *hold, and that the steps s_k are computed such that $\|s_k\| < 0.8\Delta_k$ only when B_k is positive semidefinite and $B_k s_k = -\nabla f(x_k)$. Then*

$$\lim_{k\to\infty} \frac{1}{k} |\{j \in [1, k] | B_j \text{ is positive semidefinite }\}| = 1. \tag{2.31}$$

Proof. Under these assumptions Lemma 2.6 holds and so

$$\frac{h_k}{\Delta_k} \to 0.$$

Consider any $\delta > 0$. Choose p such that $\frac{n}{p} < \delta$. By Lemma 2.4 we can find $K_1 > 0$ such that for any $k > K_1$ there exists a set $\mathcal{G}_k \subset \{k+1, \ldots, k+p\}$ such that $|\mathcal{G}_k| \geq p - n$ and such that (2.14) holds for all $j \in \mathcal{G}_k$. In addition, if K_1 is sufficiently large, we have that

$$\frac{h_j}{\Delta_j} < 0.8 \left[\beta_2 \left(\frac{2}{\beta_1}\right)^{\frac{3}{2}}\right]^{-1} \quad \text{for } j > K_1$$

by Lemma 2.6. Therefore, for $j \in \mathcal{G}_k$ with $k > K_1$ by (2.14) and (2.9)

$$\|s_j\| \leq \frac{2\beta_2}{\beta_1} e_j < \beta_2 \left(\frac{2}{\beta_1}\right)^{\frac{3}{2}} h_j < 0.8\Delta_j.$$

By hypothesis this only occurs if B_j is positive semidefinite. Therefore B_j is positive semidefinite when

$$j \in \bigcup_{i=0}^{\infty} \mathcal{G}_{K_1+ip}.$$

However, for any $k > K_1$ we can bound the cardinality of this set by

$$\left| \left[\bigcup_{i=0}^{\infty} \mathcal{G}_{K_1+ip}\right] \bigcap [1, k] \right| \geq \left\lfloor \frac{k - K_1}{p} \right\rfloor (p - n)$$

$$
\begin{aligned}
&> \frac{k-(K_1+p)}{p}(p-n) \\
&> \frac{p-n}{p}k-(K_1+p) \\
&= \left[\frac{p-n}{p}-\frac{K_1+p}{k}\right]k \\
&> \left[1-\delta-\frac{K_1+p}{k}\right]k.
\end{aligned}
$$

Thus for k sufficiently large,

$$
\frac{1}{k}|\{j \in [1,k] | B_j \text{ is positive semidefinite }\}| > 1-2\delta.
$$

Since δ is chosen arbitrarily, (2.31) follows. □

The restriction used in Theorem 2.8, that the full secant step can only be taken when the Hessian approximation is at least positive semidefinite, is imposed in virtually all implementations of trust region methods that we are aware of. Thus Theorem 2.8 shows the interesting fact that a practical SR1 algorithm that uses the framework of Algorithm 2.1 will generate positive definite Hessian approximations asymptotically at most iterations for a broad range of problems. The fact that the algorithm makes updates to the Hessian approximations at failed points is needed to establish this property.

3. Computational results. Algorithm 2.1 differs from a standard trust region algorithm in that the Hessian approximating matrix, B_k, is updated at all generated steps including the rejected steps. This was necessary for the convergence analysis, but it imposes the cost of an extra gradient evaluation at the rejected points upon the algorithm. However, it is possible that updating the Hessian approximation more often could make it more accurate and thereby reduce the total number of iterations to offset, or partially offset, this extra evaluation cost. In this section we present results of a numerical experiment conducted to investigate this possibility.

The trust region algorithm we used is from the UNCMIN unconstrained optimization software package (Schnabel, Koontz, and Weiss [1985]), modified to use the SR1 update and so that the exact trust region step is taken in all iterations. We compared two variants, one where updates are made only at successful points and one where updates are made at successful and failed points. We found that the option to update at failed points worked better if we skipped the update at failed points where the increase in f was too large. Specifically, the update in Step 6 of Algorithm 2.1 was not done if $f(x_k+s_k)-f(x_k) > 0.5[f(x_0)-f(x_k)]$. This change does not affect the convergence theory because it is easy to show from Lemma 2.2 that as x_k converges to x_*, $f(x_k+s_k)-f(x_k)$ converges to zero. For these experiments we used analytic gradients and the gradient stopping tolerance was 10^{-5}. The test problems were selected from Moré, Garbow, and Hillstrom [1981].

The results of this experiment are given in Table A. In this table limited updating refers to the trust region–SR1 algorithm when updates are made only along accepted steps, and updating at all points refers to the same algorithm when updates are made at all steps. For each test function the table contains the number of the function as given in the original source, the dimension of the problem (n), the number of *successful* iterations required to solve the problem (itrn.), the number of function evaluations required to solve the problem (f-eval), the number of gradient evaluations required to

Downloaded 01/02/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

Downloaded 01/02/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

TABLE A
Comparisons of limited updating and updating at all points.

Function	n	SR1–Limited updating				SR1–Updating at all points					sp
		itrn.	f-eval	g-eval	rgx	itrn.	f-eval	g-eval	updf	rgx	
MGH05	2	17	24	17	0.5E-05	16	24	20	4	0.8E-05	1
MGH07	3	31	41	31	0.6E-06	31	41	36	5	0.8E-06	1
MGH09	3	3	5	3	0.1E-07	3	5	3	0	0.1E-07	1
MGH12	3	35	45	35	0.1E-05	29	38	36	7	0.2E-05	1
MGH14	4	38	46	38	0.2E-05	48	62	57	9	0.1E-05	1
MGH16	4	21	29	21	0.2E-06	22	30	24	2	0.3E-08	1
MGH18	6	58	70	58	0.6E-06	50	61	57	7	0.5E-06	1
MGH20	9	73	92	73	0.7E-05	56	69	64	8	0.8E-05	1
MGH21	10	97	125	97	0.3E-05	53	69	63	10	0.3E-05	1
MGH22	8	29	33	29	0.6E-05	29	33	29	0	0.6E-05	1
MGH23	10	8	16	8	0.3E-06	9	17	13	4	0.1E-05	1
MGH24	10	27	36	27	0.5E-05	26	33	29	3	0.3E-05	1
MGH25	10	14	18	14	0.7E-05	14	18	14	0	0.7E-05	1
MGH26	10	25	31	25	0.2E-05	25	31	30	5	0.6E-06	1
MGH35	9	20	25	20	0.5E-05	20	25	21	1	0.5E-05	1
MGH05	2	42	53	42	0.1E-05	42	54	50	8	0.2E-06	10
MGH07	3	32	40	32	0.5E-06	34	48	43	9	0.2E-05	10
MGH09	3	21	28	21	0.8E-05	22	29	25	3	0.2E-07	10
MGH14	4	99	124	99	0.6E-06	86	107	102	16	0.9E-06	10
MGH16	4	36	47	36	0.5E-05	36	44	39	3	0.6E-06	10
MGH18	6	56	77	56	0.4E-05	51	65	62	11	0.1E-05	10
MGH20	9	73	92	73	0.7E-05	56	69	64	8	0.8E-05	10
MGH21	10	71	92	71	0.8E-07	71	91	87	16	0.6E-06	10
MGH22	8	46	52	46	0.9E-05	46	52	47	1	0.9E-05	10
MGH24	10	475	602	475	0.1E-04	344	434	430	86	0.1E-04	10
MGH25	10	25	30	25	0.4E-08	25	30	25	0	0.4E-08	10
MGH26	10	46	60	46	0.2E-05	46	61	53	7	0.2E-05	10
MGH07	3	24	34	24	0.6E-06	20	34	31	11	0.3E-07	100
MGH09	3	16	22	16	0.1E-06	16	22	18	2	0.9E-05	100
MGH14	4	103	130	103	0.7E-09	86	106	101	15	0.2E-06	100
MGH16	4	52	61	52	0.6E-07	49	58	53	4	0.9E-06	100
MGH18	6	94	108	94	0.1E-04	112	130	122	10	0.1E-04	100
MGH20	9	73	92	73	0.7E-05	56	69	64	8	0.8E-05	100
MGH21	10	411	531	411	0.5E-06	264	338	334	70	0.3E-05	100
MGH22	8	93	105	93	0.1E-05	59	68	63	4	0.5E-05	100
MGH26	10	39	55	39	0.5E-06	56	70	69	13	0.1E-06	100

TABLE B
Ratios of costs of SR1 with all-point updating to costs of SR1 with limited updating.

	Iterations	Function evaluations	Gradient evaluations
Arithmetic mean	0.83	0.83	0.98
Geometric mean	0.93	0.93	1.07

Downloaded 01/02/13 to 128.148.252.35. Redistribution subject to SIAM license or copyright; see http://www.siam.org/journals/ojsa.php

solve the problem (g-eval), and the relative gradient at the solution (rgx). The tenth column (updf) contains the number of times updates were made along failed steps, and the last column (sp) indicates whether the starting point used is x_0, $10x_0$, or $100x_0$, where x_0 is the standard starting point.

Table B summarizes the data in Table A and contains the ratios of the mean number of iterations, the mean number of function evaluations, and the mean number of gradient evaluations required to solve these problems by the algorithm when updates are made at all steps to those required by the algorithm when updates are made only at successful steps. Both arithmetic and geometric means are presented. These numbers indicate that on the set of test problems we used, updating at failed points improved the performance of the algorithm with respect to iterations and function evaluations and may have slightly increased the cost with respect to gradient evaluations depending on the statistic used. Combined with the theoretical advantages that are derived from updating at failed points, these computational results indicate to us that an implementation of a trust region–SR1 method for unconstrained optimization probably should include this feature.

Acknowledgments. The authors would like to thank the referees for their very careful reading and many helpful corrections.

REFERENCES

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

A. R. CONN, N. I. M. GOULD, AND PH. TOINT [1988], *Testing a class of methods for solving minimization problems with simple bounds on the variables,* Math. Comp., 50, pp. 399–430.

——— [1991], *Convergence of quasi-Newton matrices generated by the symmetric rank one update,* Math. Programming, 50, pp. 177–195.

——— [1992], *Numerical Experiments with the LANCELOT Package (Release A) for Large-scale Nonlinear Optimization,* Report 92/16, Department of Mathematics, University of Namur, Belgium.

H. FAYEZ KHALFAN [1989], *Topics in Unconstrained Optimization,* Ph.D. thesis, Department of Mathematics, University of Colorado at Boulder, Boulder, CO.

H. FAYEZ KHALFAN, R. H. BYRD, AND R. B. SCHNABEL [1993], *A theoretical and experimental study of the symmetric rank one update,* SIAM J. Optim., 3, pp. 1–24.

R.-P. GE AND M. J. D. POWELL [1983], *The convergence of variable metric matrices in unconstrained optimization,* Math. Programming, 27, pp. 123–143.

J. J. MORÉ, B. S. GARBOW, AND K. E. HILLSTROM [1981], *Testing unconstrained optimization software,* TOMS, 7, pp. 17–41.

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

J. M. ORTEGA AND W. C. RHEINBOLDT [1970], *Iterative Solution of Nonlinear Equations in Several Variables,* Academic Press, New York.

M. J. D. POWELL [1970], *A new algorithm for unconstrained optimization,* in Nonlinear Programming, J. Rosen, O. Mangasarian, and K. Ritter, eds., Academic Press, New York, pp. 31–65.

——— [1975], *Convergence properties of a class of minimization algorithms,* in Nonlinear Programming 2, O. Mangasarian, R. Meyer, and S. Robinson, eds., Academic Press, New York, pp. 1–27.

R. B. SCHNABEL, J. E. KOONTZ, AND B. E. WEISS [1985], *A modular system of algorithms for unconstrained minimization,* TOMS, 11, pp. 419–440.

G. A. SHULTZ, R. B. SCHNABEL, AND R. H. BYRD [1985], *A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties,* SIAM J. Numer. Anal., 22, pp. 47–67.