Mathematical Programming 50 (1991) 177-195
North-Holland

Convergence of quasi-Newton matrices generated by the symmetric rank one update

A.R. Conn*
IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA

N.I.M. Gould
Rutherford Appleton Laboratory, Chilton, UK

Ph.L. Toint
Department of Mathematics, Facultés Universitaires ND de la Paix, B-5000, Namur, Belgium

Received 23 July 1987
Revised manuscript received 25 August 1989

Quasi-Newton algorithms for unconstrained nonlinear minimization generate a sequence of matrices that can be considered as approximations of the objective function second derivatives. This paper gives conditions under which these approximations can be proved to converge globally to the true Hessian matrix, in the case where the Symmetric Rank One update formula is used. The rate of convergence is also examined and proven to be improving with the rate of convergence of the underlying iterates. The theory is confirmed by some numerical experiments that also show the convergence of the Hessian approximations to be substantially slower for other known quasi-Newton formulae.

Key words: Quasi-Newton updates, convergence theory.

1. Introduction

Quasi-Newton methods are recognized today as one of the most efficient ways to solve nonlinear unconstrained or bound constrained optimization problems. These methods are mostly used when the second derivative matrix of the objective function is either unavailable or too costly to compute. They are very similar to Newton's method, but avoid the need of computing Hessian matrices by recurring, from iteration to iteration, a symmetric matrix which can be considered as an approximation of the Hessian. They allow, therefore, the curvature of the problem to be exploited in the numerical algorithm, despite the fact that only first derivatives (gradients) and function values are required. We refer the reader to [3, 4, 6, 8] for further motivation and analysis concerning these now classical algorithms.

* The work of this author was supported by the National Sciences and Engineering Research Council of Canada, and by the Information Technology Research Centre, which is funded by the Province of Ontario.

The problem we consider is that of finding a local solution x_* in $\mathbb{R}^n$ of

$$\min f(x), \tag{1}$$

where $f(x)$ is a smooth function from $\mathbb{R}^n$ into $\mathbb{R}$, using a quasi-Newton method. We denote the sequence of iterates generated by this method by $\{x_k\}$. It is important to note that most of the theory and practice in this field is based on a "line search" model algorithm whose iteration is of the form

$$x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla f(x_k), \tag{2}$$

where B_k is the symmetric approximation to the Hessian mentioned above, which satisfies the "secant equation"

$$B_{k+1}(x_{k+1} - x_k) = \nabla f(x_{k+1}) - \nabla f(x_k), \tag{3}$$

and where α_k is a suitable relaxation parameter that is computed by solving (sometimes quite inexactly) the one-dimensional problem

$$\min_{\alpha > 0} f(x_k + \alpha d_k), \tag{4}$$

where the search direction d_k is given by

$$d_k = -B_k^{-1} \nabla f(x_k). \tag{5}$$

This last procedure is often called the "line search", hence giving its name to the algorithmic model. In this context, maintaining B_k positive definite has the distinct advantage that it guarantees that d_k is a descent direction with respect to the objective function. This positive definiteness of the matrices B_k can be ensured by imposing suitable conditions on the value of α_k and using adequate recurrence relations to update the B_k themselves (see the above cited references again). In this context, updating the matrices B_k with the Broyden–Fletcher–Goldfarb–Shanno (BFGS) updating formula

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}, \tag{6}$$

where

$$s_k \overset{\text{def}}{=} x_{k+1} - x_k \tag{7}$$

and

$$y_k \overset{\text{def}}{=} \nabla f(x_{k+1}) - \nabla f(x_k), \tag{8}$$

has been quite unanimously recommended for the past 15 years.

Maintaining positive definite approximations to the Hessian matrix of the objective has, however, some important drawbacks. The first one is that, even though the true Hessian is usually positive definite at the solution of the problem, this may not be the case when the current iterates of the algorithm are far away, and the concept of an *approximation* of the Hessian at these iterates must therefore be revised. One also notes that if the solution of the minimization problem has to lie within a feasible region defined by bounds on the variables, or is subject to more general constraints, then the Hessian may be indefinite even at the solution. Finally, positive definiteness is known to be incompatible in practice with symmetry, the secant equation(3) and sparsity of B_k (see [14]). All these observations justify, in the authors' opinion, the continuing research on indefinite quasi-Newton methods, especially in the context of large-scale problems for which preserving structure and sparsity in B_k is of paramount importance.

On the other hand, the development of "trust region" methods has, in the recent past, allowed the design of efficient algorithms that are capable of handling indefinite Hessian approximations. Primarily used in conjunction with exact Hessian information (that is within Newton's method), they rapidly became an important research subject in a more general context (see [10] for an excellent survey of these techniques). Their success is probably due to their remarkable numerical reliability and their comprehensive theoretical basis. Amongst the major differences between trust region methods and the line search methods described above is the fact that the step s_k is no longer a multiple of the direction d_k, and that no line search is performed.

In two recent papers [1, 2], the authors describe and analyse a class of trust region methods for unconstrained and bound constrained minimization. The second of these papers, in particular, gives a numerical comparison of several quasi-Newton updating formulae used in a trust region framework and applied on small dimensional problems. The BFGS formula (6) is considered, together with the Davidon-Fletcher-Powell (DFP) formula

$$B_{k+1} = B_k + \frac{r_k y_k^{\mathrm{T}} + y_k r_k^{\mathrm{T}}}{y_k^{\mathrm{T}} s_k} - \frac{(r_k^{\mathrm{T}} s_k) y_k y_k^{\mathrm{T}}}{(y_k^{\mathrm{T}} s_k)^2}, \tag{9}$$

the Symmetric Rank One (SR1) formula

$$B_{k+1} = B_k + \frac{r_k r_k^{\mathrm{T}}}{r_k^{\mathrm{T}} s_k} \tag{10}$$

and the Powell-symmetric-Broyden (PSB) formula

$$B_{k+1} = B_k + \frac{r_k s_k^{\mathrm{T}} + s_k r_k^{\mathrm{T}}}{s_k^{\mathrm{T}} s_k} - \frac{(r_k^{\mathrm{T}} s_k) s_k s_k^{\mathrm{T}}}{(s_k^{\mathrm{T}} s_k)^2}, \tag{11}$$

where we have defined the residual r_k by

$$r_k \overset{\mathrm{def}}{=} y_k - B_k s_k. \tag{12}$$

Somewhat surprisingly, the traditional supremacy of the BFGS update is questioned by these numerical experiments, and the SR1 formula appears to be substantially more efficient in the trust region framework than any other quasi-Newton method tested in this context. It is suggested in [2] that this interesting behaviour may be linked to the fact that a better convergence of the matrices B_k to the true Hessian of the objective at the solution has been observed when using the SR1 update, in comparison with the other updating formulae.

It is the purpose of the present paper to clarify this suggestion by considering the convergence of the matrices B_k to the true Hessian at a limit point of the sequence of iterates produced by the minimization algorithm, provided the search directions s_k remain uniformly linearly independent, which is defined in the next section. The rate of convergence of the matrices is also examined, and proven to be strongly related to that of the sequence $\{x_k\}$.

The next section briefly reviews what is known about the convergence of the quasi-Newton matrices and discusses the assumptions made. Section 3 is devoted to the convergence proof for the SR1 update, while Section 4 presents some numerical results supporting this theory, together with a comparison with the PSB, BFGS and DFP formulae.

2. Convergence of the Hessian approximations in quasi-Newton methods

We consider solving the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^n} f(x), \tag{13}$$

for a local minimum, using the following algorithmic model.

Algorithm 1. Let x_0, an initial point, be given, as well as an initial $n \times n$ symmetric matrix B_0.

Step 1. Set $k = 0$ and compute $g_0 \overset{\text{def}}{=} \nabla f(x_0)$.

Step 2. Find a vector s_k.

Step 3. Set

$$x_{k+1} = x_k + s_k, \tag{14}$$

compute $g_{k+1} = \nabla f(x_{k+1})$ and y_k according to

$$y_k = g_{k+1} - g_k. \tag{15}$$

Step 4. Update the matrix B_k to obtain B_{k+1} by using one of the quasi-Newton formulae as described above.

Step 5. Increment k by one and go to Step 2.

This very broad outline naturally calls for some comments.

(i) We have not specified in which way the step s_k should be computed. We note, however, that in the classical line search framework we have that

$$s_k = -\alpha_k B_k^{-1} g_k, \tag{16}$$

from (2), where we have assumed that B_k is nonsingular.

This freedom in the choice of s_k is, nonetheless, quite useful. In particular, we have in mind defining the step by a trust region scheme (using a truncated conjugate gradient technique, as in [2], for example), which would not yield (16). Another case where (16) would not hold is in the important context of large scale optimization using partially separable functions and partitioned updating [9], where the step is computed by taking all element functions into account and hence is only indirectly related to any particular element Hessian approximation.

(ii) No stopping criterion has been given. We are, indeed, interested in the asymptotic behaviour of the algorithm, and will assume that it produces an infinite sequence of iterates $\{x_k\}$ and an infinite sequence of quasi-Newton matrices $\{B_k\}$.

(iii) In the case where the SR1 formula (10) is used for updating B_k, we have to make sure that it is well defined. Therefore, formula (10) will be used only if

$$|r_k^{\mathrm{T}} s_k| \geq c_1 \|r_k\| \, \|s_k\|, \tag{17}$$

where $c_1 \in (0, 1)$ is a constant. (Here and below, the norm considered is the usual l_2-norm on vectors and the corresponding induced norm on matrices.) If (17) is violated, we will simply set $B_{k+1} = B_k$.

(iv) It is assumed that the same quasi-Newton updating formula is used for all iterations: we do not consider using the BFGS update for certain iterations and the DFP update for others, for example.

We now present our assumptions on the problem and the sequence of iterates generated by our algorithmic model.

(AS.1) $f(x)$ is twice continuously differentiable everywhere.

(AS.2) $\nabla^2 f(x) \overset{\mathrm{def}}{=} H(x)$ is Lipschitz continuous, that is there exists a constant $c_2 > 0$ such that, for all $x, y \in \mathbb{R}^n$,

$$\|H(x) - H(y)\| \leq c_2 \|x - y\|. \tag{18}$$

(AS.3) The sequence $\{x_k\}$ converges to some finite limit point x_*, say.

(AS.4) The sequence $\{s_k\}$ is uniformly linearly independent, that is, there exist a $c_3 > 0$, a k_0 and an $m \geq n$ such that, for each $k \geq k_0$, one can choose n distinct indices

$$k \leq k_1 < \cdots < k_n \leq k + m$$

with

$$\sigma_{\min}(S_k) \geq c_3, \tag{19}$$

where $\sigma_{\min}(S_k)$ is the minimum singular value of the matrix

$$S_k \overset{\mathrm{def}}{=} \left(\frac{s_{k_1}}{\|s_{k_1}\|}, \ldots, \frac{s_{k_n}}{\|s_{k_n}\|} \right). \tag{20}$$

(We refer the reader to [11] for an equivalent definition of a uniformly independent sequence.)

The motivation for the last of these assumptions (AS.4) comes from a careful analysis of the numerical results reported in [2]: in these tests, the smallest singular value of the matrices S_k containing n successive normalized directions is always above 10^{-4}. The superiority of the SR1 update on the BFGS observed in this paper can therefore by analysed in the context of this assumption.

It is, however, clear that (AS.4) is relatively strong: there are, indeed, known circumstances where it may fail. Firstly, the denominator in (10) can become arbitrarily small and even vanish completely during the solution of well-defined and well-conditioned minimization problems. This happens, in particular, when (2) is used with $\alpha_k = 1$, in which case (10) becomes

$$B_{k+1} = B_k + \frac{g_{k+1} g_{k+1}^{\mathrm{T}}}{g_{k+1}^{\mathrm{T}} s_k}. \tag{21}$$

The update (10) is then unusable if $g_{k+1}^{\mathrm{T}} s_k = 0$, although in most quasi-Newton methods it is advantageous if a steplength of one minimizes the objective function along the search direction. Assuming that the denominator in this formula is not exactly zero, but merely very small, a very large correction is then made to B_k to obtain B_{k+1}, possibly obliterating the useful information already contained in B_k. Furthermore, the fact that now

$$B_{k+1} \approx \frac{g_{k+1} g_{k+1}^{\mathrm{T}}}{g_{k+1}^{\mathrm{T}} s_k}. \tag{22}$$

may cause subsequent directions s_{k+j} to be nearly orthogonal to g_{k+1}, with the effect that these directions become linearly dependent, hence contradicting (AS.4). Other cases where this assumption is unsuitable arise in the solution of separable and partially separable problems, when the minimum of $f(x)$ is found at different iterations in independent subspaces. The components of the search directions along the subspaces where the minimum is found first may then become very small and linearly dependent.

Of course, a convergence analysis for the matrices generated by the SR1 update without (AS.4) is desirable, but the authors believe that such an analysis is likely to be very difficult.

Maybe the oldest result about the convergence of the quasi-Newton matrices B_k, without assuming exact linesearches, is that given a quadratic objective function and a sequence of n linearly independent steps $\{s_k\}_{k=1}^{n}$, the SR1 formula determines the exact Hessian $H = \nabla^2 f$ (that is $B_{n+1} = H$) provided formula (10) is well defined for these steps. This result seems to have been proved first by Fiacco and McCormick [5], and is rather well-known (see [6], for instance). It has the very remarkable feature that it does not depend in any way on the method used to compute the steps s_k, and thus holds for our general algorithmic mode.

The convergence of the matrices generated by the PSB update has also been studied. In [12], Powell proved that, assuming (AS.1) and (AS.2), the convergence of sequence $\{x_k\}$ to a point x_* and the uniform linear independence of the steps $\{s_k\}$ (see [11]), the sequence $\{B_k\}$ generated by the PSB update converges to $H(x_*)$. The proof is based on the fact that the PSB update defines B_{k+1} as the symmetric matrix closest to B_k in the Frobenius norm that satisfies (3). (It is interesting to note that [12] has also been a seminal paper in the study of trust region methods.) A similar theorem has been proved for the sparse PSB update in [15].

The DFP and BFGS updates have been considered by Ge and Powell in a more recent paper [7]. They consider the case where the step s_k is obtained by (16) with the choice $\alpha_k = 1$, and proved that the sequence $\{B_k\}$ is convergent when the objective function is a strictly convex quadratic. However, the limit of the sequence $\{B_k\}$ need not be, in general, equal to the Hessian matrix of the quadratic. They, indeed, provide an example where this unfortunate behaviour is observed (other examples may be found in [4]). Ge and Powell also show that their result can be extended to nonquadratic objective functions provided that (AS.1) and (AS.2) hold, and that the sequence $\{x_k\}$ converges to the limit x_* where $H(x_*)$ is positive definite. The proofs in their paper are quite involved and depend rather crucially on their choice of steps s_k. Quite remarkably, they do not depend on an assumption of the type of (AS.4).

The convergence of the quasi-Newton matrices generated by the BFGS formula has also been analysed by Schuller in [13]. He shows the interesting relation

$$\|B_{k+n+1} - H(x_*)\| \leq c_4 \|x_k - x_*\| \tag{23}$$

for some constant $c_4 > 0$, but his assumptions are quite strong. Indeed, besides (AS.3) and (AS.4), he also assumes that the line search problem (4) is solved asymptotically exactly, that B_0 and $H(x_*)$ are positive definite, that the norms of B_k and its inverse stay bounded, that x_0 is sufficiently close to x_*, and that the sequence $\{x_k\}$ converges to x_* Q-linearly. The first of these additional assumptions is, in particular, quite unrealistic in our context, because we wish to cover the trust region case, where there is no line search at all.

3. Symmetric Rank One update for nonquadratic functions

We now turn our attention to the SR1 update formula again, but at variance with the results quoted above, we consider a general nonquadratic objective function that satisfies (AS.1) and (AS.2).

We first prove the following important lemma, a surprisingly simple variation of the quadratic case.

Lemma 1. *Assume that* (AS.1) *and* (AS.2) *hold, and also that* $\{x_k\}$ *is a sequence of iterates generated by the algorithm described above, using the* SR1 *updating formula* (10). *Assume, furthermore, that* (17) *holds at every iteration. Then*

$$\|y_j - B_{j+1}s_j\| = 0 \tag{24}$$

for all j and

$$\|y_j - B_i s_j\| \leq \frac{c_2}{c_1}\left(\frac{2}{c_1}+1\right)^{i-j-2} \eta_{i,j}\|s_j\| \tag{25}$$

for all j and $i \geq j+1$, where

$$\eta_{i,j} \overset{\text{def}}{=} \max[\|x_p - x_s\| \mid j \leq s \leq p \leq i]. \tag{26}$$

Proof. We first observe that (24) and (25) with $i = j+1$ immediately results from (3). The proof of (25) is by induction. We choose $k \geq j+1$ and assume that (25) holds for all $i = j+1, \ldots, k$. We now consider

$$|r_k^T s_j| = |y_k^T s_j - s_k^T B_k s_j| \leq |y_k^T s_j - s_k^T y_j| + \frac{c_2}{c_1}\left(\frac{2}{c_1}+1\right)^{k-j-2} \eta_{k,j} \|s_j\| \, \|s_k\|, \tag{27}$$

where we used (12), our inductive assumption and the Cauchy-Schwarz inequality. Now, using the mean value theorem (see [4], for instance), we obtain that, for all l,

$$y_l = H_l s_l, \tag{28}$$

where

$$H_l = \int_0^1 H(x_l + t s_l)\, \mathrm{d}t. \tag{29}$$

Hence (27) yields that

$$\begin{aligned} |r_k^T s_j| &\leq |s_k^T(H_k - H_j)s_j| + \frac{c_2}{c_1}\left(\frac{2}{c_1}+1\right)^{k-j-2} \eta_{k,j} \|s_j\| \, \|s_k\| \\ &\leq c_2 \eta_{k+1j} \|s_j\| \, \|s_k\| + \frac{c_2}{c_1}\left(\frac{2}{c_1}+1\right)^{k-j-2} \eta_{k,j} \|s_j\| \, \|s_k\|, \end{aligned} \tag{30}$$

where we used (29), (AS.2) and the definition (26). But, from (10), (17) and the triangle inequality,

$$\|y_j - B_{k+1}s_j\| = \left\| y_j - B_k s_j - \frac{r_k r_k^T s_j}{r_k^T s_k} \right\| \leq \|y_j - B_k s_j\| + \frac{|r_k^T s_j|}{c_1 \|s_k\|} \tag{31}$$

and therefore

$$\begin{aligned} \|y_j - B_{k+1}s_j\| \leq &\left(\frac{c_2}{c_1}\left(\frac{2}{c_1}+1\right)^{k-j-2} + \frac{c_2}{c_1^2}\left(\frac{2}{c_1}+1\right)^{k-j-2}\right)\eta_{k,j}\|s_j\| \\ &+ \frac{c_2}{c_1}\eta_{k+1,j}\|s_j\|, \end{aligned} \tag{32}$$

by using (30), the inductive assumption and (31). This last inequality then gives (25) for $i = k+1$, when one takes into account the fact that $c_1 \in (0, 1)$ and uses the simple inequality

$$\eta_{k,j} \leq \eta_{k+1,j}. \quad \square \tag{33}$$

We are now in position to prove the desired convergence theorem.

Theorem 2. *Assume that* (AS.1)-(AS.4) *hold, where* $\{x_k\}$ *is a sequence of iterates generated by the algorithm described above using the* SR1 *updating formula* (10). *Assume, furthermore, that* (17) *holds for every iteration. Then there exists a constant* $c_5 > 0$ *such that, for* $k \geq k_0$,

$$\|B_{k+m+1} - H(x_*)\| \leq c_5 \varepsilon_k, \tag{34}$$

where

$$\varepsilon_k \overset{\text{def}}{=} \max[\|x_s - x_*\| \mid k \leq s \leq k+m+1], \tag{35}$$

and

$$\lim_{k\to\infty} \|B_k - H(x_*)\| = 0. \tag{36}$$

Proof. In order to prove this theorem we first observe that, for any s and p,

$$\|x_s - x_p\| \leq \|x_s - x_*\| + \|x_p - x_*\|, \tag{37}$$

which implies that

$$\eta_{k+m+1,k} \leq 2\varepsilon_k, \tag{38}$$

because of the definitions (26) and (35). We also note that, because of (AS.2),

$$\|y_j - A(x_*)s_j\| = \|(H_j - H(x_*))s_j\| \leq c_2 \varepsilon_k \|s_j\| \tag{39}$$

for any $k \leq j \leq k+m$, where H_j is defined by (29). Moreover, for any such j we can deduce from Lemma 1 and (38) that

$$\|y_j - B_{k+m+1} s_j\| \leq \frac{2c_2}{c_1}\left(\frac{2}{c_1}+1\right)^m \varepsilon_k \|s_j\|. \tag{40}$$

Gathering (39) and (40) and using the triangle inequality, we obtain, for any $k \leq j \leq k+m$, that

$$\left\|(B_{k+m+1} - H(x_*)) \frac{s_j}{\|s_j\|}\right\| \leq \left(\frac{2c_2}{c_1}\left(\frac{2}{c_1}+1\right)^m + c_2\right) \varepsilon_k. \tag{41}$$

This inequality holds in particular for $j = k_1, \ldots, k_n$. We have also that

$$\|B_{k+m+1} - H(x_*)\| \leq \frac{1}{c_3} \|(B_{k+m+1} - H(x_*))S_k\|, \tag{42}$$

using (AS.4). This last inequality, together with (41), implies (34) with

$$c_5 \overset{\text{def}}{=} \frac{c_2}{c_3}\left(\frac{2}{c_1}\left(\frac{2}{c_1}+1\right)^m + 1\right)\sqrt{n}. \tag{43}$$

Now (AS.3) implies in turn that ε_k tends to zero, and hence (36) follows from (34). □

We note that this theorem is indeed quite powerful. Not only does it guarantee the global convergence of the sequence $\{B_k\}$ to the true Hessian $H(x_*)$, but it also provides some indication on its rate of convergence, because of the estimate (34). Indeed, one can obtain the following easy corollary.

Corollary 3. *Assume that* (AS.1)-(AS.4) *hold, where* $\{x_k\}$ *is a sequence of iterates generated by the algorithm described above using the* SR1 *updating formula* (10). *Assume, furthermore, that* (17) *holds for every iteration and that there exists a constant* $c_6 \geq 0$ *such that, for all large enough k,*

$$\|x_{k+1}-x_*\| \leq c_6\|x_k - x_*\|. \tag{44}$$

Then there exists a constant $c_7 \geq 0$ *such that*

$$\|B_{k+m+1}-H(x_*)\| \leq c_7\|x_k - x_*\| \tag{45}$$

for k sufficiently large.

Proof. The proof is obvious if one observes that (44) and the definition (35) imply that

$$\varepsilon_k \leq \max[1, c_6^m]\ \|x_k - x_*\| \tag{46}$$

for k sufficiently large. Hence (45) follows from (34) with

$$c_7 \overset{\text{def}}{=} \max[1, c_6^m]c_5. \quad \square \tag{47}$$

Thus we obtain the quite interesting result that, the fast the convergence of $\{x_k\}$ to x_*, the faster the convergence of $\{B_k\}$ to $H(x_*)$!

This result is similar to that of Schuller mentioned above, but if our proof only holds for the SR1 update, the assumptions made are substantially weaker. In particular, no line search is required, nor positive definiteness of any of the involved matrices.

We can also concentrate our attention on the last n iterations, and replace (AS.4) by the following statement.

(AS.4b) There exists a k_0 such that, for all $k \geq k_0$, the matrix

$$S_k^* \overset{\text{def}}{=} \left(\frac{s_k}{\|s_k\|}, \ldots, \frac{s_{k+n-1}}{\|s_{k+n-1}\|}\right) \tag{48}$$

is nonsingular.

We now obtain the following error estimate.

Corollary 4. *Assume that* (AS.1)-(AS.3) *and* (AS.4b) *hold, where* $\{x_k\}$ *is a sequence of iterates generated by the algorithm described above using the* SR1 *updating formula* (10). *Assume, furthermore, that* (17) *holds for every iteration. Then there exists a constant* $c_8 > 0$ *such that, for all* $k \geq k_0$,

$$\|B_{k+n+1}-H(x_*)\| \leq c_8\kappa(S_k^*)\varepsilon_k, \tag{49}$$

where S_k^* and ε_k are defined by (48) *and* (35) *respectively, and where $\kappa(X)$ is the condition number of the matrix X.*

Proof. This result can directly be deduced from the inequality (41), the definition (48), the bound

$$\kappa(S_k^*) \geqslant \frac{\|S_k^*\|}{\sigma_{\min}(S_k^*)} \geqslant \frac{1}{\sigma_{\min}(S_k^*)}, \tag{50}$$

and the definition

$$c_8 \overset{\text{def}}{=} c_2\sqrt{n}\left(\frac{2}{c_1}\left(\frac{2}{c_1}+1\right)^n+1\right). \quad \square \tag{51}$$

This result is probably best related to the behaviour one can notice in practice, as shown in the next section.

Needless to say, the bounds provided by Theorem 2 and Corollaries 3 and 4 can be very gross, mainly because they depend quite heavily on c_1 in (17) and also on the Lipschitz constant c_2.

It is also clear that, if condition (17) does not hold for some iterations, then the whole convergence process and the corresponding estimates are merely delayed. It would require (17) to be violated for an infinite number of iterations to prevent convergence of the quasi-Newton matrices. However, we may expect the number of iterations where the condition (17) does not hold to be rather small in practice.

Finally, it is interesting to note that the convergence of the matrices $\{B_k\}$ to $H(x_*)$ does not require x_* to be a stationary point. This is useful in the context of trust region methods for minimization, because it guarantees that any negative eigenvalue present in $H(x_*)$ will eventually be present in B_k, and steps of negative curvature will then prevent the convergence of the iterates to x_*.

4. Some numerical experiments

In this section, we report some numerical experiments that were performed with the double purpose of verifying the theoretical error estimates of the quasi-Newton matrices generated by the SR1 update, and also of comparing these convergence properties with that of other famous quasi-Newton updating formulae, such as the BFGS, DFP and PSB. As mentioned above, all these updates were already considered in [2].

All experiments were run in FORTRAN 77 on the CRAY X-MP/24 of the Harwell Laboratory, mainly because this machine has a large wordlength (64 bits) and therefore allows a more detailed analysis of the asymptotical behaviour of the quasi-Newon matrices close to an objective function minimizer. All tests were EXTENDED PRECISION under the CFT77 compiler. The corresponding machine precision ε_M is of the order of 10^{-29}.

As the main use of quasi-Newton formulae is within nonlinear optimization algorithms, our experiments were designed to analyse the convergence of the quasi-Newton matrices for a sequence $\{x_k\}$ converging to a minimizer of some objective function. We focus our attention on the unconstrained minimization of quartic test functions of the form

$$f(x)=\tfrac{1}{2}x^{\mathrm{T}}Hx+\tfrac{1}{3}\sum_{i=1}^{n}t_ix_i^3+\tfrac{1}{4}\sum_{i=1}^{n}q_ix_i^4. \tag{52}$$

We now detail the choices made for H and the coefficients t_i and q_i. These use the routine FA01 of the Harwell Subroutine Library for generating uniformly distributed pseudo-random numbers in a given range from a "seed" θ. To allow reproducibility of our results, the recurrence used by this routine is given in an appendix.

• Given an integer $\nu\geq 1$, coefficients u_i, t_i and q_i were first generated for $i=1,\ldots,n$ in the order $u_1, t_1, q_1, u_2, t_2, q_2, \ldots,$ using FA01 with initial seed $\theta=\nu+\nu\times 16^4$ and ranges $[0, 1]$, $[0, 1]$ and $[0, 10\times 2^{\nu}]$, respectively.

• The $n\times n$ matrix H is symmetric positive definite and was then generated by the following procedure.

Using the vector $u=(u_1,\ldots,u_n)^{\mathrm{T}}$, a Householder reflector

$$R=I-2uu^{\mathrm{T}}/\|u\|^2 \tag{53}$$

was defined, and the matrix H was then built as

$$H=RDR^{\mathrm{T}}, \tag{54}$$

where D is a diagonal matrix whose elements are spread at equal intervals between 1 and $2^{-\nu}$.

As the diagonal elements of D are the eigenvalues of H, the parameter ν allows the conditioning of the problem to be varied. A minimum for these problems (for all values of ν) clearly lies at the origin.

Three different types of sequences converging to this minimum were generated

• artificial sequences having a well determined convergence rate,

• sequences of iterates as produced by the trust region algorithm described in [2],

• sequences of iterates generated by a minimization algorithm using line searches, as described in Section 1, where the method of determining the stepsize α_k is specified below.

All these sequences start at

$$x_0^{\mathrm{T}}=(1,\ldots,1)$$

and the initial quasi-Newton approximation was always chosen to be the identity matrix.

The SR1 formula was skipped, as suggested by the above theory, when the test (17) was violated, where c_1 was chosen as 10^{12}. The BFGS and DFP updates were skipped whenever

$$y_k^{\mathrm{T}}s_k<10^{-8}\|y_k\|\,\|s_k\|. \tag{55}$$

It is worth noticing that these conditions *never* prevented updating the second derivative approximations in any of the tests reported here.

4.1. Artificially generated sequences

The sequence $\{x_k\}$ converging to the origin was, in these tests, generated by the following procedure.

Step 1. Determine the norm of the next iterate x_{k+1}. This is done according to the relations

$$\|x_{k+1}\| = \begin{cases} \beta_k \|x_k\| & \text{(linear convergence)}, \\ \beta_k \|x_k\|/k & \text{(superlinear convergence)}, \\ \beta_k \|x_k\|^2 & \text{(quadratic convergence)}, \end{cases} \tag{56}$$

where β_k is a random number uniformly distributed in one of the ranges $[0.7, 0.9]$, $[0.4, 0.5]$ and $[0.1, 0.3]$.

Step 2. Generate x_{k+1} by computing a random vector z_{k+1} on the unit sphere and setting

$$x'_{k+1} = \|x_{k+1}\| z_{k+1}. \tag{57}$$

The vector z_{k+1} is computed by randomly determining its Euler angles in the appropriate ranges.

Step 3. Verify that the new point gives a strict descent on the objective function. If $f(x'_{k+1}) < f(x_k)$, then accept $x_{k+1} = x'_{k+1}$ as the next iterate. Else return to Step 2.

We observe that Step 2 can be repeated a number of times before a satisfactory point is found. As implied in (56), three different rates of convergence were considered: linear, superlinear and quadratic.

We note that using random directions can be quite beneficial for the convergence of the matrices generated by the SR1 update: indeed, the behaviour considered in the discussion of (AS.4) above is unlikely to happen in this context, because the steps s_{k+j} are no longer related to the matrix B_{k+1} and need not be nearly orthogonal to g_{k+1} anymore. As a matter of fact, assuming s_{k+1} has a sizeable component along g_{k+1}, the residual r_{k+1} itself is likely to be dominated by the term $-g_{k+1}(g_{k+1}^{\mathrm{T}} s_{k+1})/(g_{k+1}^{\mathrm{T}} s_k)$ because of (22), and we observe that

$$B_{k+2} - B_{k+1} = \frac{r_{k+1} r_{k+1}^{\mathrm{T}}}{r_{k+1}^{\mathrm{T}} s_{k+1}} \approx -\frac{g_{k+1} g_{k+1}^{\mathrm{T}}}{g_{k+1}^{\mathrm{T}} s_k}, \tag{58}$$

correcting for the large rank-one term introduced by the update in B_{k+1}. Random-like directions do, however, appear in conjunction with the SR1 update in the solution of a large number of well conditioned partially separable problems, for instance, or when the update is used in the context of an inexact trust region scheme.

We first report in Table 1 four sets of experiments involving linearly convergent sequences in 10 and 3 variables, respectively. In order to produce the results of these tables, linearly convergent sequences were generated as described above, and the process was stopped as soon as $\|x_{k+n+1}\| \leq 10^{-8}$. Then the error

$$E \overset{\text{def}}{=} \max_{i,j=1,\dots,n} |[B_{k+n+1} - H(x_*)]_{ij}| \tag{59}$$

was computed. These errors appear in the tables under the headings E_{SR1}, E_{PSB}, E_{BFGS} and E_{DFP} for the four considered updates, respectively. The values of ε_k and $\kappa(S_k^*)$ are also provided to allow a comparison of E_{SR1} with their product, as suggested by Corollary 4.

A few conclusions can already be drawn from these results. Firstly, the prediction of Corollary 4 seems to be verified quite nicely, with a value of the constant c_8 not much larger than 0.1 for our test functions. One observes indeed a quite smooth convergence of the quasi-Newton matrices generated by the SR1 update to the true Hessian matrix at the solution, even for the not so well conditioned examples (ν large). As far as the other updating formulae are concerned, they seem to produce much less accurate approximations. The quality of the approximations produced by BFGS and DFP substantially deteriorate when the dimension is increased from 3 to 10, especially for DFP. The theoretical convergence of the PSB matrices barely shows up for the best conditioned problems with $n = 3$, but again its performance decreases markedly with the quality of conditioning, and when the dimension of

Table 1
Linearly convergent sequences

n	β_k range	ν	E_{SR1}	ε_k	$\kappa(S_k^*)$	E_{PSB}	E_{BFGS}	E_{DFP}
3	[0.1, 0.3]	2	1.35×10^{-7}	5.51×10^{-6}	3.06	1.33	2.09×10^{-2}	9.75×10^{-1}
		4	6.53×10^{-8}	1.84×10^{-6}	1.47	3.70	1.38×10^{-1}	1.11×10^{2}
		6	1.04×10^{-6}	7.28×10^{-6}	3.65	2.68	4.34×10^{-1}	4.32×10^{1}
		8	9.52×10^{-7}	1.14×10^{-5}	3.05	2.46	1.50×10^{-1}	1.24×10^{3}
		10	1.70×10^{-8}	4.80×10^{-6}	8.89	1.36×10^{1}	2.02×10^{-1}	1.40×10^{1}
3	[0.4, 0.5]	2	1.72×10^{-7}	1.70×10^{-7}	9.49×10^{1}	6.09×10^{-2}	7.86×10^{-4}	1.31×10^{-2}
		4	8.82×10^{-9}	2.06×10^{-7}	2.16	2.89×10^{-4}	3.78×10^{-2}	3.68×10^{1}
		6	4.85×10^{-8}	2.08×10^{-7}	8.71	1.26×10^{-3}	3.65×10^{-3}	4.66
		8	1.42×10^{-7}	1.55×10^{-7}	6.96	2.87×10^{-3}	1.60×10^{-3}	6.62×10^{1}
		10	4.87×10^{-8}	1.80×10^{-7}	4.93	4.28×10^{-2}	3.84×10^{-2}	1.55×10^{2}
3	[0.7, 0.9]	2	9.06×10^{-9}	1.94×10^{-8}	4.57	7.13×10^{-10}	8.28×10^{-10}	8.28×10^{-10}
		4	7.59×10^{-9}	2.37×10^{-8}	5.59	4.90×10^{-9}	4.16×10^{-9}	4.16×10^{-9}
		6	1.23×10^{-8}	2.46×10^{-8}	6.03	4.12×10^{-7}	7.20×10^{-9}	7.20×10^{-9}
		8	1.56×10^{-8}	2.52×10^{-8}	8.88×10^{1}	1.89×10^{-4}	6.19×10^{-7}	4.08×10^{-7}
		10	1.38×10^{-8}	2.34×10^{-8}	7.61	1.24×10^{-1}	3.06×10^{-8}	2.61×10^{-8}
10	[0.7, 0.9]	2	1.06×10^{-6}	2.00×10^{-7}	1.72×10^{2}	1.21	3.82×10^{-1}	2.70
		4	6.85×10^{-7}	1.05×10^{-7}	4.05×10^{2}	6.67	4.49×10^{-1}	5.77
		6	6.77×10^{-6}	1.02×10^{-7}	2.38×10^{4}	1.23×10^{1}	8.22×10^{-1}	1.78×10^{1}
		8	6.57×10^{-5}	1.06×10^{-7}	4.60×10^{3}	5.90×10^{2}	8.82×10^{-1}	9.24×10^{2}
		10	1.63×10^{-5}	1.23×10^{-7}	4.75×10^{3}	8.36×10^{2}	6.09×10^{-1}	4.55×10^{2}

the problem increases. The deterioration associated with conditioning is somewhat to be expected, as PSB, in contrast with the other three updates, is not invariant with respect to linear scaling of the variables.

The results for superlinear and quadratic sequences are presented in Table 2. The first of these sequences was stoped at the first k, such that $\|x_k\| \le 10^{-30}$, while the second one was stopped when $\|x_k\| \le 10^{-50}$. These values allow the asymptotic behaviour to take place before the machine precision level is reached, despite the relatively high speed of convergence. Both of these tests were run for $n = 3$ only, because the delay of n iterations involved in the bound given by Corollary 4 has to be small enough with respect to the speed of convergence to allow a meaningful asymptotical behaviour to be reached.

Again, one can observe the excellent convergence of the matrices generated by the SR1 update. The best of the other updates is still BFGS on average, with PSB doing relatively well for the superlinear case. DFP is clearly the worst.

It is also interesting to show numerically how much the convergence of the SR1 matrices depends on the rate of convergence of the underlying sequence of iterates to the minimizer. To illustrate this behaviour, we ran the tests with $n = 3$ and $\nu = 5$ for the four updates, and then counted the number of iterations that were required for the error (see (59)) to be strictly smaller than 10^{-6}. These counts are presented

Table 2

Superlinearly and quadratically convergent sequences ($n = 3$, $\beta_k \in [0.7, 0.9]$)

	ν	E_{SR1}	ε_k	$\kappa(S_k^*)$	E_{PSB}	E_{BFGS}	E_{DFP}
Superlinear convergence	2	2.52×10^{-29}	3.12×10^{-26}	2.31	4.41×10^{-3}	1.03×10^{-3}	4.10×10^{-2}
	4	2.75×10^{-27}	5.21×10^{-25}	3.73	1.73×10^{-2}	3.40×10^{-2}	4.12×10^{1}
	6	3.84×10^{-27}	5.33×10^{-25}	1.37	5.85×10^{-8}	2.50×10^{-2}	2.34×10^{1}
	8	4.66×10^{-27}	5.36×10^{-25}	2.54	5.36×10^{-5}	1.20×10^{-3}	3.38×10^{1}
	10	5.25×10^{-25}	5.68×10^{-25}	5.34×10^{1}	4.00×10^{-2}	4.83×10^{-2}	328×10^{2}
Quadratic convergence	2	1.45×10^{-11}	4.00×10^{-6}	7.90	4.70	3.47×10^{-1}	6.95
	4	1.18×10^{-4}	8.17×10^{-4}	1.02×10^{2}	4.02×10^{1}	1.56	1.06×10^{2}
	6	3.41×10^{-8}	2.95×10^{-4}	3.74	9.96×10^{-1}	8.33×10^{-1}	8.77×10^{1}
	8	1.71×10^{-9}	1.01×10^{-4}	2.61	1.45×10^{2}	1.49×10^{-1}	1.19×10^{2}
	10	2.50×10^{-8}	7.27×10^{-5}	2.20×10^{1}	1.72×10^{2}	5.92×10^{-1}	4.07×10^{2}

Table 3

Iteration counts to reach an accuracy of 10^{-6}

	β_k range	SR1	PSB	BFGS	DFP
	[0.7, 0.9]	70	79	72	73
Linear	[0.4, 0.5]	23	40	56	82
	[0.1, 0.3]	14	29	>88 (10^{-5})	>88 (10^{-2})
Superlinear	[0.7, 0.9]	12	29	>46 (10^{-3})	>46 (10^{0})
Quadratic	[0.7, 0.9]	10	>11 (10^{0})	>11 (10^{-1})	>11 (10^{-1})

in Table 3. When the desired precision was not reached before this new stopping criterio was achieved, this is denoted in the table by a ">" sign followed by the number of iteration performed and, between parenthesis, the order of magnitude of the obtained accuracy on the quasi-Newton matrix.

The decrease of the number of iterations required to reach this accuracy with the improving speed of convergence of the sequence $\{x_k\}$ is quite apparent for the SR1 update.

4.2. Sequences generated by a trust region algorithm

The next experiment uses sequences $\{x_k\}$ that were generated by a trust region algorithm. The algorithm used is described in full detail in [2], and was applied to the quartic test examples (52) with $n=3$. It was stopped as soon as

$$\|\nabla f(x_k)\| \leq \varepsilon_M^{2/3}. \tag{60}$$

The final error in the quasi-Newton matrices compared to the true Hessian at the minimum was then measured as above. The results are reported in Table 4, while Table 5 gives more detail for the case where the SR1 update is used. In these tables, the heading "ng" stands for the number of gradient evaluations that were necessary to achieve the required accuracy.

Table 4
Sequence generated by a trust region method

ν	SR1		PSB		BFGS		DFP	
	ng	E_{SR1}	ng	E_{PSB}	ng	E_{BFGS}	ng	E_{DFP}
2	15	9.74×10^{-10}	63	1.11×10^{-3}	24	1.39×10^{-3}	45	3.53×10^{-3}
4	25	3.67×10^{-13}	59	1.74×10^{-4}	34	2.02×10^{-3}	83	4.16×10^{-3}
6	24	4.96×10^{-9}	116	3.81×10^{-4}	30	1.37×10^{-3}	42	3.22×10^{-3}
8	35	8.55×10^{-10}	169	3.86×10^{-4}	43	2.38×10^{-3}	50	2.12×10^{-3}
10	50	8.63×10^{-13}	111	3.75×10^{-3}	49	1.89×10^{-3}	93	2.17×10^{-3}

Table 5
Details for the SR1 case within a trust region method

ν	ng	E_{SR1}	ε_k	$\kappa(S_k^*)$
2	15	9.74×10^{-10}	7.83×10^{-7}	5.40×10^{1}
4	25	3.67×10^{-13}	3.48×10^{-9}	9.23
6	24	4.96×10^{-9}	5.16×10^{-7}	3.63×10^{1}
8	35	8.55×10^{-10}	1.21×10^{-8}	3.57×10^{1}
10	50	8.63×10^{-13}	3.30×10^{-10}	7.30×10^{1}

Again, the behaviour predicted for SR1 by Corollary 4 is observed, while the other three formulae are significantly less efficient.

4.3. *Sequences generated by a line search algorithm*

Finally, we consider sequences $\{x_k\}$ generated by a simple line search algorithm. This framework is the traditional one in which the BFGS formula is used, and it was hoped that the latter formula might show improved efficiency in this context. We therefore restricted our test to the SR1 and BFGS updates only.

The search direction was determined as in (5) and the stepsize α_k was computed by successive bisection (starting from 1) in order to satisfy the condition

$$f(x_k)-f(x_{k+1})\geqslant 0.1\alpha_k \nabla f(x_k)^{\mathrm{T}} d_k. \tag{61}$$

When the BFGS updating formula was used, the additional condition

$$y_k^{\mathrm{T}} s_k \geqslant 10^{-8}\|y_k\|\,\|s_k\| \tag{62}$$

was also enforced, in order to guarantee positive definiteness of the Hessian approximation.

We note that this strategy is not really recommended for the SR1 update, since these formulae may generate indefinite or even singular matrices B_k. In order to resolve this potential difficulty, the search direction was reversed whenever d_k was not a descent direction, which happended a few times. The singular case never occurred in our tests.

The test functions used were again the quartics (52) with $n=3$. The minimization algorithm was stopped as soon as

$$\|\nabla f(x_k)\|\leqslant 10\varepsilon_M \tag{63}$$

(which is asking slightly more than (60)). The relevant quantities were then computed at the final point, and are reported in Table 6. More details for the SR1 case are provided in Table 7.

Even in this inappropriate framework, SR1 stays remarkably efficient, and very coherent with the error estimates given in Corollary 4.

Table 6
Sequence generated by a line search method

ν	SR1		BFGS	
	ng	E_{SR1}	ng	E_{BFGS}
2	21	2.97×10^{-14}	33	2.16×10^{-3}
4	24	5.99×10^{-13}	39	5.57×10^{-4}
6	35	4.01×10^{-10}	47	1.36×10^{-3}
8	34	1.98×10^{-17}	56	1.04×10^{-3}
10	43	5.76×10^{-11}	61	3.37×10^{-4}

Table 7

Details for the SR1 case within a line search method

ν	ng	E_{SR1}	ε_k	$\kappa(S_k^*)$
2	21	2.97×10^{-14}	2.06×10^{-9}	1.40×10^{1}
4	24	5.99×10^{-13}	1.97×10^{-9}	4.55
6	35	4.01×10^{-10}	6.37×10^{-8}	2.38×10^{1}
8	34	1.98×10^{-17}	5.38×10^{-12}	1.79
10	43	5.76×10^{-11}	7.60×10^{-10}	4.90×10^{1}

5. Conclusion

This paper compares the behaviour of quasi-Newton updating formulae as a means to generate symmetric approximations to Hessian matrices. A known convergence result for the SR1 matrices on quadratics has been extended to the general class of sufficiently smooth nonlinear functions. This convergence is global, and is rate is shown to be improving with that of the sequence of iterates produced by the underlying minimization algorithm.

Numerical experiments are also presented that support the theory quite well. Furthermore, these computations show that, in comparison with other formulae such as BFGS, DFP or PSB, the SR1 formula generates more accurate Hessian approximations in a number of circumstances.

Although it may not completely explain why a trust region method based on the SR1 update outperformed the more classical line search based BFGS algorithm in the study [2], it certainly throws some light on their relative merits and differences.

Appendix: the recurrence for FA01

As mentioned above, the routine FA01 of the Harwell Library of Subroutines was used to generate the coefficients of the quartic test example (52), and other uniformly distributed pseudo-random numbers in a given value range. In order to allow reproduction of our numerical tests, or use of (52) for other purposes, we now detail the recurrence used by FA01 to generate those numbers.

FA01 is given a seed θ and the routine first resets

$$\theta = (9228907\times\theta) \bmod 16^8. \tag{A.1}$$

This new value of the seed is then used on the next call to the routine. A pseudo random number in the range [0, 1] is then calculated by the formula

$$\text{pseudo random number} = \theta\times16^{-8}. \tag{A.2}$$

This number is then finally scaled by the linear transformation that maps [0, 1] onto the desired value range.

Acknowledgement

The authors thank M.J.D. Powell for his thoughtful and constructive comments on a first version of this paper.

References

[1] A.R. Conn, N.I.M. Gould and Ph.L. Toint, "Global convergence of a class of trust region algorithms for optimization with simple bounds," *SIAM Journal on Numerical Analysis* 25 (1988) 433-460 (with a correction given in *SIAM Journal on Numerical Analysis* 26 (1989) 764-767).

[2] A.R. Conn, N.I.M. Gould and Ph.L. Toint, "Testing a class of methods for solving minimization problems with simple bounds on the variables," *Mathematics of Computation* 50 (1988) 399-430.

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

[4] J.E. Dennis and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (Prentice-Hall, Englewood Cliffs, NJ, 1983).

[5] A.V. Fiacco and G.P. McCormick, *Nonlinear Programming* (Wiley, New York, 1968).

[6] R. Fletcher, *Practical Methods of Optimization: Unconstrained Optimization* (Wiley, Chichester, 1980).

[7] R.P. Ge and M.J.D. Powell, "The convergence of variable metric matrices in unconstrained optimization," *Mathematical Programming* 27 (1983) 123-143.

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

[9] A. Griewank and Ph.L. Toint, "Partitioned variable metric updates for large structured optimization problems," *Numerische Mathematik* 39 (1982) 119-137.

[10] J.J. Moré, "Recent developments in algorithms and software for trust region methods," in: A. Bachem, M. Grötschel and B. Korte, eds., *Mathematical Programming: The State of the Art* (Springer, Berlin, 1983).

[11] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables* (Academic Press, New York, 1970).

[12] M.J.D. Powell, "A new algorithm for unconstrained optimization," in: J.B. Rosen, O.L. Mangasarian and K. Ritter, eds., *Nonlinear Programming* (Academic Press, New York, 1970).

[13] G. Schuller, "On the order of convergence of certain quasi-Newton methods," *Numerische Mathematik* 23 (1974) 181-192.

[14] D.C. Sorensen, "An example concerning quasi-Newton estimates of a sparse Hessian," *SIGNUM Newsletter* 16 (1981) 8-10.

[15] Ph.L. Toint, "On the superlinear convergence of an algorithm for solving a sparse minimization problem," *SIAM Journal on Numerical Analysis* 16 (1979) 1036-1045.