A Wide Interval for Efficient Self-Scaling Quasi-Newton Algorithms

M. Al-Baali* and H. Khalfan†

July 2003

Abstract

This paper uses certain conditions for the global and superlinear convergence of the two-parameter self-scaling Broyden family of quasi-Newton algorithms for unconstraiend optimization to derive a wide interval for self-scaling updates. Numerical testing shows that such algorithms not only accelerate the convergence of the (unscaled) methods from the so-called convex class, but increase their chances of success as well. Self-scaling updates from the preconvex and the postconvex classes are shown to be effective in practice, and new algorithms which work well in practice, with or without scaling, are also obtained from the new interval. Unlike the behaviour of unscaled methods, numerical testing shows that varying the updating parameter in the proposed interval has little effect on the performance of the self-scaling algorithms.

Key Words. Unconstrained optimization, quasi-Newton methods, Broyden's family, self-scaling, global and superlinear convergence.

1 Introduction

This paper is concerned with self-scaling quasi-Newton algorithms for finding a local minimum of the unconstrained optimization problem

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

where $f : R^n \to R$ is twice continuously differentiable. These algorithms start with an initial approximation, x_1, of a solution, x_*, and generate new approximations by the basic iteration

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

*Department of Mathematics and Statistics, Sultan Qaboos University, P.O. Box 36, Al-Khodh 123, Muscat, Sultanate of Oman. (albaali@squ.edu.om)

†Department of Mathematics and Computer Science, UAE University, Al-Ain 17551, United Arab Emirates. (hfayez@emirates.net.ae)

where α_k is a steplength (calculated by line search or trust-region frameworks), and B_k approximates the Hessian matrix $\nabla^2 f(x_k)$. Given an initial symmetric and positive definite B_1, new Hessian approximations are generated by the two-parameters family of updates

$$B_{k+1} = \tau_k \left(B_k - \frac{B_k \delta_k \delta_k^T B_k}{\delta_k^T B_k \delta_k} + \theta_k w_k w_k^T \right) + \frac{\gamma_k \gamma_k^T}{\delta_k^T \gamma_k}, \tag{2}$$

where

$$w_k = (\delta_k^T B_k \delta_k)^{1/2} \left(\frac{\gamma_k}{\delta_k^T \gamma_k} - \frac{B_k \delta_k}{\delta_k^T B_k \delta_k} \right), \quad \delta_k = x_{k+1} - x_k, \quad \gamma_k = \nabla f(x_{k+1}) - \nabla f(x_k)$$

and τ_k and θ_k are scaling and updating parameters, respectively. (For background on these updates, see e.g. Oren and Spedicato [18].)

If $\tau_k = 1$, class (2) reduces to the classical (unscaled) Broyden family of Hessian approximation updates. In this case, although for $\theta_k \in [0, 1)$, which is called the convex class, iteration (1) converges globally and q-superlinearly for convex objective functions under appropriate conditions (see e.g. Byrd, Liu and Nocedal [6]), only updates with $\theta_k = 0$ (which corresponds to the BFGS update) have been shown to be effective in practice; and the performance worsens as θ_k increases from 0 to 1. See in particular, Al-Baali [3], Byrd, Liu and Nocedal [6], Powell [20], Zhang and Tewarson [22] and also Section 4.

Several attempts have been made to improve the performance of the above class of algorithms by choosing τ_k and θ_k in such a way to improve the conditioning of B_{k+1} (see e.g. Oren and Spedicato [18]). Numerical experiments, however, have not shown significant improvement. In fact, Nocedal and Yuan [16] showed that the best self-scaling BFGS algorithm of Oren and Luenberger [17] performs badly compared to the BFGS method when applied with inexact line searches to a simple quadratic function of two variables.

Al-Baali [2], however, used the theory of Byrd, Liu and Nocedal [6] for unscaled methods to determine conditions on τ_k and θ_k that ensure global and superlinear convergence for scaling algorithms with inexact line searches under the additional restriction that

$$\tau_k \leq 1. \tag{3}$$

Using these conditions, Al-Baali [2] and [3] showed that the performance of some members of the Broyden family, including the BFGS update, was improved substantially.

Condition (3) is motivated by the fact that the eigenvalues of the bracketed matrix in (2) can be reduced if $\tau_k < 1$ (even for $\theta_k = 0$) and, hence, smaller eigenvalues are introduced into B_{k+1} if the eigenvalues of B_k are large. On the other hand, since the BFGS update corrects small eigenvalues of B_k (see e.g. Byrd, Liu and Nocedal [6] and Powell [20]), it is sensible to use $\tau_k = 1$ if $\theta_k \leq 0$.

This paper extends the work of Al-Baali [2] and [3] and proposes a wide interval of globally and superlinearly convergent self-scaling updates, that not only accelerate the convergence of the unscaled methods, but also succeed in solving many problems that the unscaled methods failed to solve, especially when $\theta_k \geq 1$. Numerical experience presented also shows that, unlike the behaviour of unscaled methods, varying the updating parameter θ_k in the proposed interval (which contains $[0, 1]$) has little effect on the performance of the self-scaling algorithms.

In the next section some classes of self-scaling algorithms which converge globally and superlinearly for convex functions are described. A wide interval for θ_k covering the convex class as well as updates from the preconvex and postconvex classes is developed in Section 3. In section 4, we present and discuss the results of numerical testing of the proposed self-scaling algorithms. Finally, we give a brief conclusion in Section 5.

2 Convergence Analysis

In this section we establish global and superlinear convergence for the classes of self-scaling algorithms we introduce in this paper. The analysis is based on the convergence results of Al-Baali [2] which are extensions of those of Byrd, Liu and Nocedal [6], Zhang and Tewarson [22], Byrd, Nocedal and Yuan [7] and Powell [19].

Throughout this section the following assumptions will frequently be made.

Assumptions 2.1

1. The function f is twice continuously differentiable.

2. The level set $N = \{x : f(x) \leq f(x_1)\}$ is convex, and there exist positive constants c_1 and c_2 such that
$$c_1\|z\|^2 \leq z^T\nabla^2 f(x)z \leq c_2\|z\|^2$$
for all $z \in R^n$ and all $x \in N$. (This implies that the Hessian matrix $\nabla^2 f(x)$ is positive definite on N and that f has a unique minimizer x_* in N.)

3. The Hessian matrix $\nabla^2 f(x)$ satisfies a Lipschitz condition for all x in a neighborhood of x_*.

The framework of the self-scaling algorithm we consider is given below.

Algorithm 2.1 Given an initial estimate x_1 and an initial symmetric and positive definite matrix B_1, at the kth iteration find the new iterate by

$$x_{k+1} = x_k + \alpha_k s_k, \qquad k \geq 1, \tag{4}$$

where α_k is a steplength calculated by a line search satisfying the Wolfe conditions

$$f(x_{k+1}) \leq f(x_k) + \zeta_1 \delta_k^T \nabla f(x_k), \qquad \delta_k^T \nabla f(x_{k+1}) \geq \zeta_2 \delta_k^T \nabla f(x_k), \tag{5}$$

where $\zeta_1 \in (0, 0.5)$ and $\zeta_2 \in (\zeta_1, 1)$,

$$s_k = -B_k^{-1}\nabla f(x_k)$$

is the search direction, and B_k is updated to a positive definite Hessian approximation by (2) for some values of θ_k and τ_k.

To maintain positive definiteness of Hessian approximations, values of the parameters τ_k and θ_k are chosen such that $\tau_k > 0$ and $\theta_k > \bar{\theta}_k$, where

$$\bar{\theta}_k = -\frac{1}{a_k}, \quad a_k = b_k h_k - 1, \quad b_k = \frac{\delta_k^T B_k \delta_k}{\delta_k^T \gamma_k}, \quad h_k = \frac{\gamma_k^T B_k^{-1} \gamma_k}{\delta_k^T \gamma_k}. \tag{6}$$

Note that $\delta_k^T \gamma_k > 0$ (by the Wolfe conditions (5)) and $a_k \geq 0$ (by Cauchy's inequality) for any positive definite B_k.

The following convergence result which is a slight variation of Theorem 3.2 of Al-Baali [2] gives conditions for the global convergence of Algorithm 2.1.

Theorem 2.1 Suppose that Assumptions 2.1 hold. Assume also that the parameters θ_k and τ_k are chosen such that

$$(1-\nu_1)\bar{\theta}_k \leq \theta_k \leq \nu_2, \tag{7}$$

$$\theta_k \tau_k \leq 1 - \nu_3, \tag{8}$$

$$\nu_4 \leq \tau_k \leq 1, \tag{9}$$

where $\nu_2 \geq 1$, and $\nu_1, \nu_3, \nu_4 > 0$ are small enough. Then the sequence $\{x_k\}$ generated by Algorithm 2.1 converges to x_* R-linearly and $\sum_{k=1}^{\infty} \|x_k - x_*\| < \infty$.

(Note that conditions (7)-(9) imply that $(1-\nu_1)\bar{\theta}_k \leq \tau_k\theta_k \leq 1 - \nu_3$ which is the condition required in Theorem 3.2 of Al-Baali [2].)

The following result gives conditions for superlinear convergence of Algorithm 2.1. It is essentially the same result as Theorem 5.2 of Al-Baali [2] except that it does not require information about the exact Hessian when $\theta_k \leq 0$.

Theorem 2.2 Suppose that the assumptions of Theorem 2.1 are true and that the line search scheme is initiated with $\alpha_k = 1$. Assume in addition that

$$\sum_{\theta_k \leq 0} \ln \tau_k^{n-1} + \sum_{\theta_k > 0} \ln((1+\theta_k a_k)\tau_k^{n-1}) > -\infty \tag{10}$$

and

$$\tau_k \leq \min(1 + \theta_k a_k, 1). \tag{11}$$

Then the sequence $\{x_k\}$ generated by Algorithm 2.1 converges to x_* q-superlinearly, the sequences $\{\|B_k\|\}$ and $\{\|B_k^{-1}\|\}$ are bounded, and

$$\lim_{k\to\infty} b_k = \lim_{k\to\infty} h_k = 1. \tag{12}$$

Proof For any $\eta_k \in [0,1]$, conditions (7) and (11) imply that

$$\theta_k \geq \bar{\theta}_k \min(1-\nu_1, 1-\eta_k\tau_k) \tag{13}$$

(this is trivially true for $\tau_k = 1$, and if $\tau_k = 1 + \theta_k a_k$, then $\eta_k \tau_k \leq 1 + \theta_k a_k$). Inequality (13) with an appropriate η_k (replacing $(1 - \nu_8 \epsilon_k) p_k^{2(1-r)}$ in (5.3) of Al-Baali [2]), and (8) imply that

$$\tau_k \bar{\theta}_k \min(1 - \nu_1, 1 - \eta_k \tau_k) \leq \tau_k \theta_k \leq 1 - \nu_3.$$

This last inequality is equivalent to condition (5.2) of Al-Baali [2] and the result now follows from Theorem 5.2 of Al-Baali [2]. □

We note that (11) is satisfied for any $\theta_k \geq 0$ and $\tau_k \leq 1$, and it does not depend on the true Hessian if $\theta_k < 0$ (as required in Al-Baali [2]). We therefore use it together with (10) and (8) to modify the class of self-scaling methods in Al-Baali [3] to obtain the class

$$\tau_k = \begin{cases} \min(\sigma_k^{-1/(n-1)}, (1 - \nu_3)\theta_k^{-1}) & \text{if } \theta_k > 0 \\ \sigma_k & \text{if } \theta_k \leq 0, \end{cases} \tag{14}$$

where

$$\sigma_k = \min(1 + \theta_k a_k, \nu_5) \tag{15}$$

and $\nu_5 \geq 1 + \nu_2$. Note that $\sigma_k > 0$ for any $\theta_k > \bar{\theta}_k$ and that $\sigma_k < 1$ only if $\theta_k < 0$ and $a_k \neq 0$.

The following result gives conditions under which class (14) is globally and superlinearly convergent.

Theorem 2.3 Suppose that Assumptions 2.1 hold. If θ_k is chosen from interval (7) and τ_k is defined by (14), then the result of Theorem 2.1 is true. If in addition the line search scheme is initiated with $\alpha_k = 1$,

$$\sum_{\theta_k \leq 0} \ln \sigma_k > -\infty, \tag{16}$$

$$\theta_k^{n-1} - (1 - \nu_3)^{n-1}(1 + a_k \theta_k) \leq 0 \tag{17}$$

for $\theta_k \geq 0$, and ν_5 is sufficiently large so that $1 + a_k \theta_k \leq \nu_5$, then the result of Theorem 2.2 is true.

Proof The first part is straightforward, since condition (8) trivially holds and (9) holds with $\nu_4 = \min(\nu_1, (1-\nu_3)/\nu_2, 1/\nu_5^{n-1})$. For the second part notice that for $\theta_k \leq 1$, (14) and (16) imply that both conditions (10) and (11) hold, since $a_k \geq 0$ (see also Al-Baali [2] for the proof with $\theta_k \in [0, 1]$). For $\theta_k > 1$, condition (17) with sufficiently large ν_5 implies that $\theta_k \leq (1-\nu_3)\sigma_k^{1/(n-1)}$ and hence (14) defines τ_k such that condition (10) holds. □

For $\theta_k \geq 0$, choice (14) is the same as that of Al-Baali [3]. If $\theta_k = 0$, then both conditions (10) and (11) are satisfied and Algorithm 2.1 reduces to the (unscaled) BFGS method. For $\theta_k < 0$ we let τ_k be the largest value satisfying (11), in order to come as close as possible to satisfying condition (10).

Letting $\hat{\theta}_k$ denotes the largest value satisfying (17), then for ν_3 small enough (in practice, $\nu_3 = 0$), $\hat{\theta}_k > 1$ and therefore class (14) converges globally and superlinearly for $\theta_k \in (\xi_k, \hat{\theta}_k]$ with certain $\xi_k < 0$. In practice, we observed that this class outperforms the unscaled one. In particular for $\theta_k \geq 1$, this class solves all our test problems, while the corresponding unscaled methods failed to solve most of them (detail is given in Sections 3 and 4). On the convex interval, however, class (14) worsens a little as θ_k increases from 0 to 1, but it is still better than the unscaled methods which deteriorate substantially in this case. As done in Al-Baali [3], we handle this case by defining τ_k as given in the following convergence result.

Theorem 2.4 Suppose that Assumptions 2.1 hold and let the scaling parameter be defined by

$$\tau_k = \begin{cases} \rho_k \min(\sigma_k^{-1/(n-1)}, (1-\nu_3)\theta_k^{-1}) & \text{if } \theta_k > 0 \\ \min(\rho_k \sigma_k^{-1/(n-1)}, \sigma_k) & \text{if } \theta_k \le 0, \end{cases} \tag{18}$$

where ρ_k is chosen such that

$$\nu_6 \le \rho_k \le 1, \tag{19}$$

$\nu_6 > 0$ and θ_k is chosen from interval (7). Then the result of Theorem 2.1 is true. If in addition the line search scheme is initiated with $\alpha_k = 1$, conditions (16), (17), $1 + \theta_k a_k \le \nu_5$ and

$$\sum_{k=1}^{\infty} \ln \rho_k > -\infty \tag{20}$$

hold, then the result of Theorem 2.2 is true.

Proof The first part follows from Theorem 2.1, since condition (8) trivially holds and (9) holds with ν_4 replaced by $\min(\nu_1, (1-\nu_3)/\nu_2, 1/\nu_5^{n-1}, \nu_6)$.

Now consider the second part. For $\theta_k > 0$, let $\hat{\tau}_k = \tau_k$, where τ_k is given by (18). Then $\hat{\tau}_k = \rho_k \tau_k$, where τ_k is given by (14). Substituting this last form of $\hat{\tau}_k$ for τ_k in (10), we get

$$\sum_{\theta_k \le 0} \ln \tau_k^{n-1} + \sum_{\theta_k > 0} \ln((1 + a_k\theta_k)\tau_k^{n-1}) + \sum_{k=1}^{\infty} \ln \rho_k^{n-1} > -\infty. \tag{21}$$

Using (20), the result follows from Theorem 2.3.

For $\theta_k \le 0$, we note that $\tau_k \ge \min(\rho_k, \sigma_k)$. Therefore using (16) and (20), it follows that condition (10) holds. Hence the result follows as in Theorem 2.3. □

In practice we observed that class (18) with

$$\rho_k = \min(1, \frac{1}{b_k}) \tag{22}$$

which is recommended by Al-Baali [3] works substantially better than (14).

3 Intervals for the Updating Parameter

We now propose an interval for the updating parameter based on conditions (11), (8) and

$$\rho_k^- \le \tau_k \le \rho_k^+, \tag{23}$$

where

$$\rho_k^{\pm} = \min(1, h_k(1 \pm c_k))$$

and

$$c_k = \left(\frac{a_k}{1+a_k}\right)^{1/2}.$$

We consider the bounds in (23), because they have been used by some authors (e.g. Al-Baali [1]) to obtain useful features for the scaling parameter τ_k.

Combining intervals (23), (11) and (8) with any $\nu_3 \geq 0$, it follows that

$$\theta_k^- \leq \theta_k \leq \theta_k^+, \tag{24}$$

where

$$\theta_k^- = \frac{\rho_k^- - 1}{a_k}, \quad \theta_k^+ = \frac{1}{\rho_k^-}. \tag{25}$$

Note that $1/\rho_k^- = \max(1, b_k(1+c_k))$ and choice (22) belongs to interval (23) and decreases as b_k increases. This is desirable since a large value of b_k indicates large eigenvalues of B_k (e.g. Al-Baali [3]). If both b_k and h_k tend to 1, intervals (23) and (24) tend to the single number 1 and [0,1], respectively, which is in line with the result of Al-Baali [3]. Although there exist other intervals containing (24) (e.g. Al-Baali [1]), we do not consider them here because they do not have this property (see Al-Baali and Khalfan [4] for detail).

Now interval (24) can be used to define θ_k and, using (22), the self-scaling classes (14) and (18) are well defined and converge globally for convex functions, since ρ_k^- is bounded away from zero (see e.g. Al-Baali [3] who also shows that ρ_k^- is smaller than choice (22)). In addition, these classes converge superlinearly if $\theta_k \in [0, \hat{\theta}_k]$ and for class (18) if condition (20) holds. We let the right endpoint of interval (24) to be $1/\rho_k^-$ rather than $\min(\hat{\theta}_k, 1/\rho_k^-)$, since the cost of computing it is negligible compared to that of computing $\hat{\theta}_k$.

It is clear that interval (24) is wide enough to include values of θ_k not only from the convex and preconvex parts of the Broyden family, but also from the postconvex one. For our numerical experiments, we considered the following updates.

From the preconvex class, we considered the update defined by

$$\theta_k = \begin{cases} 0 & \text{if } b_k \leq 1 \text{ or } a_k \leq \epsilon_1 \\ \min\left(\theta_k^n, \theta_k^{SR1}\right) & \text{if } h_k \leq 1 - \epsilon_2 \\ 1 - 1/c_k & \text{otherwise,} \end{cases} \tag{26}$$

where

$$\theta_k^n = \frac{n}{2(n-1)}\left[1 - \left(1 + \frac{4(n-1)}{n^2 a_k}\right)^{1/2}\right], \tag{27}$$

$$\theta_k^{SR1} = \frac{1}{1-b_k} \tag{28}$$

and $\epsilon_1, \epsilon_2 \in (0, 1)$. Choice (26) is used by Al-Baali [3], (S520), which defines a globally convergent algorithm that works well in practice and includes the BFGS and SR1 updates. Although this choice may not belong to interval (24), we consider it here to test its behaviour when scaling is employed only for $\theta_k \geq \theta_k^-$ (and $\tau_k = 1$ otherwise).

Since the SR1 method, (28), has been shown to be efficient in practice (see e.g. Conn, Gould and Toint [9], Khalfan, Byrd and Schnabel [14] and Byrd, Khalfan and Schnabel [5]), but it is undefined when $b_k = 1$ and it does not preserve positive definiteness, we considered a modified SR1 update

$$\theta_k = \begin{cases} \theta_k^- & \text{if } b_k = 1 \\ \max(\theta_k^-, 1/(1-b_k)) & \text{if } b_k > 1 \\ \min(\theta_k^+, 1/(1-b_k)) & \text{if } b_k < 1. \end{cases} \tag{29}$$

which enforces the endpoints of interval (24) if θ_k^{SR1} does not belong to this interval.

We also considered the update

$$\theta_k = \begin{cases} \theta_k^- & \text{if } b_k \geq 1 \\ \theta_k^+ & \text{if } b_k < 1 \end{cases} \tag{30}$$

which switches between the endpoints of interval (24).

To consider other possibilities of switching algorithms that include the convex, preconvex and postconvex parts, we chose the following updates.

$$\theta_k = \max(\theta_k^-, 1 - b_k), \tag{31}$$

$$\theta_k = \min\left(\theta_k^+, \frac{1}{b_k}\right), \tag{32}$$

$$\theta_k = \max\left(\theta_k^-, \min\left(\theta_k^+, \frac{1-b_k}{b_k}\right)\right). \tag{33}$$

For more testing of scaling, we also included the following updates from the convex class

$$\theta_k = \frac{1}{1 + \sqrt{b_k h_k}} \tag{34}$$

and

$$\theta_k = \frac{1}{1 + b_k} \tag{35}$$

which correspond to the self-dual updates of Oren and Spedicato [18] and Hoshino [13], respectively.

Note that for most of the updates presented in this section, the updating parameter θ_k decreases as b_k increases.

4 Numerical Results

In this section we present the results of numerical experiments of testing the performance of the self-scaling algorithms developed in the previous sections against that of the unscaled BFGS method. All experiments were conducted on a SUN blade 2000 workstation in double precision arithmetic with machine $\epsilon = 2^{-52} \approx 2.22 \times 10^{-16}$.

For the self-scaling algorithms we selected the ones in which the updates are defined by (26), $\theta_k = \theta_k^-$, (29), (30), (31), (33), $\theta_k = 0$, (34), (35), (32), $\theta_k = 1$, and $\theta_k = \theta_k^+$ which we denote respectively by U_{-6}, U_{-5}, $\mathrm{U}_{\pm 4}$, $\mathrm{U}_{\pm 3}$, $\mathrm{U}_{\pm 2}$, $\mathrm{U}_{\pm 1}$, U_0, U_1, U_2, U_3, U_4, and U_5 as shown in Table 1 (the sign of the subscript in the notation U_i is used to denote the sign of the associated value of θ_k; the sign $\pm$ refers to unfixed sign).

We tested these updates on a set of 89 standard test problems, with dimensions ranging from 2 to 100. These problems are extensions of 31 problems from Moré, Garbow and Hillstrom [15], Conn, Gould and Toint [8], Fletcher and Powell [11] and Grandinetti [12] as outlined in Table A.1 in the Appendix.

We let the initial Hessian approximation $B_1 = I$, where I is the identity matrix, and the initial scaling parameter be defined by

$$\tau_1 = \frac{h_1}{1 + a_1\theta_1}. \tag{36}$$

Choice (36) gives the least value of the condition number of the matrix $\tau_1^{-1}B_2$ (Oren and Spedicato [18]). Thus in particular for $\theta_1 = 0$, choice (36) yields

$$\tau_1 = \frac{\gamma_1^T\gamma_1}{\delta_1^T\gamma_1} \tag{37}$$

which is recommended by Shanno and Phua [21] for scaling the initial matrix I before updating by U_0.

In all experiments, we used $\nu_1 = 0.05$ and $\nu_2 = \nu_5 = 10^{16}$ which were never exceeded, $\nu_3 = 0$ and $\nu_4 = \nu_6 = 10^{-4}$; and if $\tau_k < \nu_4$ which happened in some cases, we set $\tau_k = \nu_4$. We also let $\tau_k = 1$ if $\theta_k < \theta_k^-$. Since the value $a_k = 0$ makes the quasi-Newton family of updates (2) independent of θ_k, we use (as in Al-Baali [3]) the unscaled U_0 update if $a_k \leq \sqrt{\epsilon}$.

The steplength α_k was calculated by a line search routine satisfying the strong Wolfe conditions

$$f(x_{k+1}) \leq f(x_k) + \zeta_1\delta_k^T\nabla f(x_k), \qquad |\delta_k^T\nabla f(x_{k+1})| \leq -\zeta_2\delta_k^T\nabla f(x_k), \tag{38}$$

where $\zeta_1 = 10^{-4}$ and $\zeta_2 = 0.9$.

The stopping condition used was

$$\nabla f(x_k)^T\nabla f(x_k) \leq \epsilon \max(1, |f(x_k)|), \tag{39}$$

and the maximum number of line searches allowed was 5000 which, except for the (unscaled) U_4 method, was never reached in all experiments.

We made four main experiments using Algorithm 2.1: (i) no scaling is used at all, (ii) only initial scaling by (36) is used, (iii) initial scaling by (36) and subsequently by (14) is used, and, (iv) initial scaling by (36) and subsequently by (18) is used. The results of these experiments are given in Tables 2, 3, 4, and 5, respectively.

In each experiment, the performance of all the updates were compared against that of the unscaled U_0 update, using three statistics A_l, A_f and A_g, where A_l denotes the "average" of certain ratios of the number of line searches, denoted by nls, required to solve the test problems by a method to the corresponding nls required by the unscaled U_0 method; and A_f and A_g denote similar ratios with respect to the number of function evaluations, nfe, and the number of gradient evaluations nge.

The averages here are defined as, for example, in Al-Baali [3], where for a given algorithm, a value $A_l < 1$, for instance, indicates that the performance of the algorithm compared to that of the unscaled U_0 method improved by $100(1-A_l)$ as far as reducing the number of line searches required.

As Table 2 shows, the unscaled $\mathrm{U}_{-6}, \ldots, \mathrm{U}_{\pm 1}$ algorithms are better than the unscaled U_0 method (U_{-6} gives the most efficient algorithm). While this is expected for algorithms U_{-6} and U_{-5}, which define $\theta_k \leq 0$ for all k, it is rather surprising for $\mathrm{U}_{\pm 4}, \ldots, \mathrm{U}_{\pm 1}$ which use values of $\theta_k > 0$ as well as of $\theta_k \geq 1$, at some iterations. Comparing the results of these algorithms with those of $\mathrm{U}_1, \mathrm{U}_2, \ldots, \mathrm{U}_5$, indicates that values of $\theta_k \leq 0$ improve performance. A comparison of U_4 and U_5 with U_3 which switches between $\theta_k \in (0,1)$ and $\theta_k \geq 1$, also indicates that using $\theta_k < 1$ improves performance.

The results in Table 3 indicates that using scaling only at the first iteration improves slightly the performance of all methods, except U_4 and U_5 where the initial scaling worsens the performance.

Comparing Table 4 with Table 3 (or Table 2) shows that scaling with (14) makes further improvement especially for $\theta_k > 0$. For example, unlike the unscaled U_4 method, which failed to solve 36 problems in the first experiment and 47 problems in the second experiment, the scaled U_4 algorithm solved all the test problems. Similarly algorithm U_5 with this scaling solved all the problems, while the unscaled U_5 algorithm failed to solve 75 problems in the first experiment and 79 problems in the second experiment.

We also observe from Table 4 that the algorithms from the preconvex class, still outperform self-scaling methods of type (14) and that the performance of U_0 in both Tables 3 and 4 is identical, because the choice $\theta_k = 0$ reduces the scaling factor (14) to the unscaled case $\tau_k = 1$.

Table 5 shows that scaling by (18) and (36) yield the best performance. The algorithms in this table solved all the tests and perform substantially better than the algorithms of the previous tables. Moreover we observe that varying the value of θ_k in interval (24) makes slight effect on the performance of the algorithms.

Finally, we note that Tables 2–5 indicate that unscaled algorithms taken from the preconvex part work well with or without scaling. It is also worth mentioning that the modified SR1 algorithm, $\mathrm{U}_{\pm 4}$, works substantially better than a standard line-search SR1 method, like Algorithm 2.1, in which the SR1 update is employed only when the inequality $\min(b_k, h_k) < 1 - 10^{-7}$ is satisfied, (and skipped otherwise), in order to guarantee positive definiteness at each step.

Table 1: Selected choices for the updating parameter

Update	U_{-6}	U_{-5}	$U_{\pm 4}$	$U_{\pm 3}$	$U_{\pm 2}$	$U_{\pm 1}$	U_0	U_1	U_2	U_3	U_4	U_5
θ_k	(26)	θ_k^-	(29)	(30)	(31)	(33)	0	(34)	(35)	(32)	1	θ_k^+

Table 2: Unscaled Algorithms

Update	A_l	A_f	A_g
U_{-6}	0.838	0.886	0.869
U_{-5}	0.882	0.934	0.910
$U_{\pm 4}$	0.861	0.902	0.892
$U_{\pm 3}$	0.895	0.941	0.922
$U_{\pm 2}$	0.876	0.917	0.907
$U_{\pm 1}$	0.886	0.922	0.916
U_0	1.000	1.000	1.000
U_1	1.167	1.137	1.154
U_2	1.120	1.096	1.107
U_3	1.260	1.217	1.240
U_4	3.148	3.061	3.116
U_5	6.594	6.659	6.593

Table 3: Initially Scaled by (36)

Update	A_l	A_f	A_g
U_{-6}	0.738	0.744	0.765
U_{-5}	0.753	0.760	0.780
$U_{\pm 4}$	0.736	0.740	0.763
$U_{\pm 3}$	0.759	0.771	0.783
$U_{\pm 2}$	0.753	0.757	0.781
$U_{\pm 1}$	0.763	0.761	0.788
U_0	0.895	0.854	0.887
U_1	1.012	0.956	0.993
U_2	0.970	0.917	0.954
U_3	1.046	0.984	1.025
U_4	4.203	3.798	4.051
U_5	9.144	9.101	9.094

Table 4: Scaled by (14) and (36)

Update	A_l	A_f	A_g
U_{-6}	0.747	0.752	0.768
U_{-5}	0.753	0.760	0.780
$U_{\pm 4}$	0.768	0.790	0.785
$U_{\pm 3}$	0.777	0.807	0.801
$U_{\pm 2}$	0.733	0.741	0.759
$U_{\pm 1}$	0.746	0.747	0.767
U_0	0.895	0.854	0.887
U_1	0.949	0.897	0.933
U_2	0.932	0.884	0.917
U_3	0.970	0.934	0.957
U_4	1.349	1.263	1.317
U_5	0.935	0.976	0.944

Table 5: Scaled by (18) and (36)

Update	A_l	A_f	A_g
U_{-6}	0.700	0.708	0.716
U_{-5}	0.745	0.754	0.771
$U_{\pm 4}$	0.758	0.781	0.766
$U_{\pm 3}$	0.775	0.802	0.796
$U_{\pm 2}$	0.718	0.725	0.730
$U_{\pm 1}$	0.739	0.749	0.752
U_0	0.757	0.780	0.769
U_1	0.791	0.802	0.795
U_2	0.801	0.816	0.805
U_3	0.835	0.862	0.843
U_4	0.833	0.854	0.840
U_5	0.886	0.956	0.907

5 Conclusion

In this paper we have attempted to explore further the potential of self-scaling for improving the performance of quasi-Newton methods from the Broyden family of Hessian approximation updates. We derived scaling schemes and a wide interval of Hessian approximations. Numerical testing of these algorithms on a fairly large number of standard test problems has shown that the scaled algorithms outperform the unscaled ones significantly as far as the number of failures is concerned, and as far as varying the updating parameter in this interval.

Numerical experience also indicate that the postconvex class of updates contains useful choices for θ_k. Thus the modified SR1 update ($U_{\pm 4}$ in Section 4) and other new updates (such as $U_{\pm 1}$ and $U_{\pm 2}$) are proposed. They work better, with or without scaling, than the BFGS method. It would be interesting to investigate further convergence properties of such methods.

Acknowledgement We would like to thank the anonymous referees for carefully reading a draft of this paper and making a number of valuable comments.

References

[1] M. Al-Baali, (1995). "On measure functions for the self-scaling updating formulae for quasi-Newton methods," Optimization, 32, pp. 59–69.

[2] M. Al-Baali, (1998a). "Analysis of a family of self-scaling quasi-Newton methods," Computational Optimization and Applications, 9, pp. 191–203.

[3] M. Al-Baali, (1998b). "Numerical experience with a class of self-scaling quasi-Newton algorithms," JOTA, 96, pp. 533–553.

[4] M. Al-Baali and H. Khalfan, (2003). "Globally convergent self-scaling quasi-Newton methods for unconstrained optimization," Tech. Report DMCS 1/03, Dept. of Mathematics and Computer Science, UAE University, United Arab Emirates.

[5] R.H. Byrd, H. Khalfan and R.B. Schnabel, (1996). "Analysis of a symmetric rank-one trust tegion method," SIAM J. Optimization, 6, pp. 1025–1039.

[6] R. Byrd, D. Liu and J. Nocedal, (1992). "On the behaviour of Broyden's class of quasi-Newton methods," SIAM J. Optimization, 2, pp. 533–557.

[7] R. Byrd, J. Nocedal and Y. Yuan, (1987). "Global convergence of a class of quasi-Newton methods on convex problems," SIAM J. Numer. Anal., 24, pp. 1171–1190.

[8] A.R. Conn and N.I.M. Gould and Ph.L. Toint, (1988). "Testing a class of methods for solving minimization problems with simple bounds on the variables," Math. Comp., 50, pp. 399–430.

[9] A.R. Conn and N.I.M. Gould and Ph.L. Toint, (2000). "Trust-Region Methods," MPS-SIAM series on Optimization, Philadelphia, USA.

[10] R. Fletcher, (1987). "Practical Methods of Optimization," Second Edition, Wiley, Chichester, England.

[11] R. Fletcher and M.J.D. Powell, (1963). "A rapidly convergent descent method for minimization," Computer Journal, 6, pp. 163–168.

[12] L. Grandinetti, (1984). "Some investigations in a new algorithm for nonlinear optimization based on conic models of the objective function," JOTA, 43, pp. 1–21.

[13] S. Hoshino, (1972). "A formulation of variable metric methods," J. Inst. Maths. Applns., 10, pp. 394–403.

[14] H. Khalfan, R.H. Byrd and R.B. Schnabel, (1993). "A theoretical and experimental study of the symmetric rank one update," SIAM J. Optimization, 3, pp. 1–24.

[15] J.J. Moré, B.S. Garbow and K.E. Hillstrom, (1981). "Testing unconstrained optimization software," ACM transactions on mathematical software, 7, pp. 17–41.

[16] J. Nocedal and Y. Yuan, (1993). "Analysis of a self-scaling quasi-Newton method," Math. Programming, 61, pp. 19–37.

[17] S.S. Oren and D.G. Luenberger, (1974). "Self-scaling variable metric (SSVM) algorithms I: Criteria and sufficient conditions for scaling a class of algorithms," Management Science, 20, pp. 845–862.

[18] S.S. Oren and E. Spedicato, (1976). "Optimal conditioning of self-scaling variable metric algorithms," Math. Programming, 10, pp. 70–90.

[19] M.J.D. Powell, (1976). "Some global convergence properties of a variable metric algorithm for minimization without exact line searches," in Nonlinear Programming, SIAM-AMS Proceedings vol. IX, eds. R.W. Cottle and C.E. Lemke, SIAM Publications.

[20] M.J.D. Powell, (1986). "How bad are the BFGS and DFP methods when the objective function is quadratic?," Math. Programming, 34, pp. 34–47.

[21] D.F. Shanno and K.H. Phua, (1978). "Matrix conditioning and nonlinear optimization," Math. Programming, 14, pp. 149–160.

[22] Y. Zhang and R.P. Tewarson, (1988). "Quasi-Newton algorithms with updates from the preconvex part of Broyden's family," IMA J. Numer. Anal., 8, pp. 487–509.

Appendix

Table A.1 contains detail of the set of test problems we used in this paper. The number in the first column refers to the number of the function given in the original source and the symbol "†" indicates that the same test function is used again, but with the standard initial point multiplied by 100. The dimension of 59 problems ranges from 2 to 30 and the other 30 problems have dimension either 40 or 100 (indicated in the table by "‡").

Table A.1: List of test functions

Test Code	n	Function's name
MGH3	2	Powell badly scaled
MGH4	2	Brown badly scaled
MGH5	2	Beale
MGH7	3†	Helical valley
MGH9	3	Gaussian
MGH11	3	Gulf research and development
MGH12	3	Box three-dimensional
MGH14	4†	Wood
MGH16	4†	Brown and Dennis
MGH18	6	Biggs Exp 6
MGH20	6,9,12,20	Watson
MGH21	2†,10†,20†, ‡	Extended Rosenbrock
MGH22	4†,12†,20†, ‡	Extended Powell singular
MGH23	10,20, ‡	Penalty I
MGH25	10†,20†, ‡	Variably dimensioned
MGH26	10,20, ‡	Trigonometric of Spedicato
MGH35	8,9,10,20, ‡	Chebyquad
TRIGFP	10,20, ‡	Trigonometric of Fletcher and Powell
CH-ROS	10†,20†, ‡	Chained Rosenbrock
CGT1	8	Generalized Rosenbrock
CGT2	25	Another chained Rosenbrock
CGT4	20	Generalized Powell singular
CGT5	20	Another generalized Powell singular
CGT10	30, ‡	Toint's seven-diagonal generalization of Broyden tridiagonal
CGT11	30, ‡	Generalized Broyden tridiagonal
CGT12	30, ‡	Generalized Broyden banded
CGT13	30, ‡	Another generalized Broyden banded
CGT14	30, ‡	Another Toint's seven-diagonal generalization of Broyden tridiagonal
CGT15	10	Nazareth
CGT16	30, ‡	Trigonometric
CGT17	8, ‡	Generalized Cragg and Levy

†: Two initial points are used; one of them is the standard, $\bar{x}$, and the other is $100\bar{x}$.
‡: $n = 40$ and 100 are used to define large dimensional tests.
MGH: Problems from Moré, Garbow and Hillstrom [15].
CGT: Problems from Conn, Gould and Toint [8].
TRIGFP: Problem from Fletcher and Powell [11].
CH-ROS: Problem from Grandinetti [12].