Computational Methods for Differential Equations
http://cmde.tabrizu.ac.ir
Vol. 7, No. 1, 2019, pp. 28-41

The modified BFGS method with new secant relation for unconstrained optimization problems

Razieh Dehghani*
Faculty of Mathematics, Yazd University, Yazd, Iran.
E-mail: r.dehghani9080@yahoo.com

Mohammad Mehdi Hosseini
Faculty of Mathematics, Yazd University, Yazd, Iran.
E-mail: hosse_ m@yazd.ac.ir,

Narges Bidabadi
Faculty of Mathematics, Yazd University, Yazd, Iran.
E-mail: n_ bidabadi@yazd.ac.ir.

Abstract Using Taylor's series we propose a modified secant relation to get a more accurate approximation of the second curvature of the objective function. Then, based on this modified secant relation we present a new BFGS method for solving unconstrained optimization problems. The proposed method make use of both gradient and function values while the usual secant relation uses only gradient values. Under appropriate conditions, we show that the proposed method is globally convergent without needing convexity assumption on the objective function. Comparative results show computational efficiency of the proposed method in the sense of the Dolan-Moré performance profiles.

Keywords. Unconstrained optimization, Modified BFGS method, Global convergence.

2010 Mathematics Subject Classification. 90C30, 90C53, 49M37, 65K05.

1. Introduction

Consider the unconstrained nonlinear optimization problem

$$\min f(x), \quad x \in \mathbb{R}^n, \tag{1.1}$$

where, $f : \mathbb{R}^n \to \mathbb{R}$, is twice continuously differentiable.

Notation 1. Throughout our work, we consider the following notations:

$f_k = f(x_k), \quad s_k = x_{k+1} - x_k, \quad g_k = \nabla f(x_k), \quad y_k = g_{k+1} - g_k, \quad G_{k+1} = \nabla^2 f(x_{k+1}).$
Also, $\|.\|$ is the Euclidean norm.

As we all know, Newton method based on the second order Taylor's series approximation involves computation of the Hessian matrix of second order derivatives at

Received: 24 December 2017 ; Accepted: 27 October 2018.
$*$ Corresponding author.

each iteration. In practice it is often preferred to approximate the Hessian matrix (or sometimes its inverse) with a symmetric positive definite matrix through some effective procedure instead of its exact computation. This idea of approximating the Hessian with a symmetric positive definite matrix was first introduced by Davidon [7]. The class of methods that approximates Newton method by utilizing some symmetric positive definite approximation of the Hessian or the inverse Hessian instead of the corresponding exact value is termed as quasi-Newton methods.
The quasi-Newton methods possess a number of important theoretical properties (see [4], [5], [8], [9], [22]), for example, quadratic termination, invariance under non-singular affine transformations, heredity of positive-definite updates, and generating identical iterate points with exact line searches (see [11]), locally and superlinearly convergence under mild conditions (see [8], [9]).

These methods generate a sequence $\{x_k\}$ by the iterative scheme

$$x_{k+1} = x_k + \alpha_k d_k, \tag{1.2}$$

where, $\alpha_k > 0$ is a step length and d_k is the search direction obtained by solving $B_k d_k = -g_k$, where B_k is an approximation of the Hessian matrix of f at x_k.

A famous class of quasi-Newton methods is the Broyden family [2] in which the Hessian updates are defined by

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k} + \mu w_k w_k^T, \ w_k = (s_k^T B_k s_k)^{1/2} [\frac{y_k}{s_k^T y_k} - \frac{B_k s_k}{s_k^T B_k s_k}],$$

where μ is a scalar and B_{k+1} satisfies the following standard secant relation:

$$B_{k+1} s_k = y_k. \tag{1.3}$$

The popular BFGS, DFP and SR1 updates are obtained by setting $\mu = 0$, $\mu = 1$ and $\mu = 1/(1 - s_k^T B_k s_k / s_k^T y_k)$, respectively.

From the numerical experiment on the quasi-Newton methods, it is proved that the BFGS method is the most successful one among all the quasi-Newton methods. But the global convergence for general function f is still open even if it is convergent (global and superlinear) for convex minimization [1]-[3], [10]. Hence, it is very interesting to investigate whether there is any new quasi-Newton method that not only possess global convergence but also superior than the BFGS method from the computation point of view.
When the fuction f is convex, global convergence of the BFGS method's has been studied by some authors (see [3, 4, 17, 22, 24]). Dai [6] constructed an example to show that the standard BFGS method may fail for non-convex functions with inexact line search. Mascarenhas [20] showed standard BFGS methods may not be convergent even with precise line searching. To overcome this problem, Li and Fukushima [18, 19] made a modification on the standard BFGS method and developed a modified BFGS method that is globally convergent without a convexity assumption on the objective function f.

The usual secant relation employs only the gradients and the available function values are ignored. To overcome this problem, several researchers have modified the usual secant relation (1.3) to make full use of both the gradient and function values (see [25]-[29]).

Zhang and Xu [29], using Taylor's series, modified usual secant equation (1.3) as follows:

$$B_{k+1}s_k = \overline{y}_k,\ \overline{y}_k = y_k + \frac{\vartheta_k}{\|s_k\|^2}s_k, \tag{1.4}$$

with

$$\vartheta_k = 6(f_k - f_{k+1}) + 3(g_k + g_{k+1})^T s_k. \tag{1.5}$$

A merit of the new modified secant method can be seen from the following theorem [29].

Theorem 1.1. Assume that the functions $f(x)$ are smooth enough and $\overline{y}_k$ is defined by (1.4). If $\|s_k\|$ is sufficiently small, then we have

$$s_k^T(G_{k+1}s_k - \overline{y}_k) = O(\| s_k \|^4),$$
$$s_k^T(G_{k+1}s_k - y_k) = O(\| s_k \|^3).$$

Resently, Yabe and Takano [27] extended the modified secant relation (1.4) by multiplying a fixed parameter $\rho \geq 0$, as follows:

$$B_{k+1}s_k = z_k,\ z_k = y_k + \rho\frac{\vartheta_k}{\|s_k\|^2}s_k,$$

where ϑ_k is given by (1.5).

Theorem 1.1, demonstrate if $\|s_k\| > 1$, the standard secant relation is expected to be more accurate than the modified secant relation (1.4). In this case, the use of (1.4)) does not seem to be suitable. To overcome these problems, Peyghami et al. [21] modified (1.4) as follows:

$$B_{k+1}s_k = w_k,\ w_k = y_k + \rho_k\frac{\vartheta_k}{\|s_k\|^2}s_k, \tag{1.6}$$

with

$$\rho_k = \min(\rho_{\max}, \frac{a}{b + \|s_k\|^m}), \tag{1.7}$$

where a, b, $\rho_{\max}$ and m is a nonnegative integer.

Here, we construct alternative estimates of the secant relation (1.3), to get a more accurate approximation of the second curvature of the objective function. Then, we make use of the new secant relation in a BFGS updating formula. This work is organized as follows: In section 2, we employ Taylor's series to derive an alternative secant relation. In section 3, we investigate the global convergence of the proposed method. Finally, in section 4, we report some numerical results.

2. Proposed modified quasi-Newton method

Consider the following auxiliary function

$$f_k(x) = f(x) + \frac{r_k}{2}(x - x_k)^T(x - x_k). \tag{2.1}$$

Obviously the functions f_k and f have the same value in x_k.

By using (2.1), we have

$$B_{k+}s_k = y_k^*, \ y_k^* = y_k + r_k s_k. \tag{2.2}$$

Clearly, different choices of the r_k in (2.2) define a variety of secant relation. Here, we introduce a reasonable r_k, that leads a new secant relation.

Using the Taylor formula for the functions $f(x)$, we obtain

$$\begin{aligned} f_k \simeq\ & f_{k+1} - g_{k+1}^T s_k + \tfrac{1}{2} s_k^T G_{k+1} s_k - \tfrac{1}{6}\textstyle\sum_{i,j,l=1}^n \frac{\partial^3 f_{k+1}}{\partial x^i \partial x^j \partial x^l} s_k^i s_k^j s_k^l \\ & + \tfrac{1}{24}\textstyle\sum_{i,j,k,l=1}^n \frac{\partial^4 f_{k+1}}{\partial x^i \partial x^j \partial x^k \partial x^l} s_k^i s_k^j s_k^k s_k^l, \end{aligned} \tag{2.3}$$

$$\begin{aligned} s_k^T g_k \simeq\ & s_k^T g_{k+1} - s_k^T G_{k+1} s_k + \tfrac{1}{2}\textstyle\sum_{i,j,l=1}^n \frac{\partial^3 f_{k+1}}{\partial x^i \partial x^j \partial x^l} s^i s^j s^l \\ & - \tfrac{1}{6}\textstyle\sum_{i,j,d,l=1}^n \frac{\partial^4 f_{k+1}}{\partial x^i \partial x^j \partial x^d \partial x^l} s_k^i s_k^j s_k^d s_k^l, \end{aligned} \tag{2.4}$$

$$s_k^T G_k s_k \simeq s_k^T G_{k+1} s_k - \sum_{i,j,l=1}^n \frac{\partial^3 f_{k+1}}{\partial x^i \partial x^j \partial x^l} s_k^i s_k^j s_k^l + \frac{1}{2}\sum_{i,j,d,l=1}^n \frac{\partial^4 f_{k+1}}{\partial x^i \partial x^j \partial x^d \partial x^l} \partial s_k^i s_k^j s_k^d s_k^l. \tag{2.5}$$

Cancellation of $\sum_{i,j,l=1}^n \frac{\partial^3 f_{k+1}}{\partial x^i \partial x^j \partial x^l} s_k^i s_k^j s_k^l$ *and* $\sum_{i,j,d,l=1}^n \frac{\partial^4 f_{k+1}}{\partial x^i \partial x^j \partial x^d \partial x^l} \partial s_k^i s_k^j s_k^d s_k^l$, from (2.3), (2.4) and (2.5) yields,

$$s_k^T G_{k+1} s_k \simeq 12(f_k - f_{k+1}) + 6(g_k + g_{k+1})^T s_k + s_k^T G_k s_k. \tag{2.6}$$

Since B_{k+1} approximate $G_{k+1} = \nabla^2 f(x_{k+1})$, we have

$$s_k^T B_{k+1} s_k = s_k^T y_k + 12(f_k - f_{k+1}) + 7 g_k^T s_k + 5 g_{k+1}^T s_k + s_k^T B_k s_k. \tag{2.7}$$

On the other hand, multiplying both sides of (2.2) by s_k, we get

$$s_k^T B_{k+1} s_k = s_k^T y_k + r_k s_k^T s_k. \tag{2.8}$$

The relation (2.7), together with (2.8), result in

$$r_k = \frac{\theta_k}{\|s_k\|^2}, \ \theta_k = 12(f_k - f_{k+1}) + 7 g_k^T s_k + 5 g_{k+1}^T s_k + s_k^T B_k s_k. \tag{2.9}$$

Now, Based on the above observation, we can modify the secant relation (1.3) as follows

$$B_{k+1} s_k = y_k^*, \tag{2.10}$$

with

$$y_k^* = y_k + \frac{\theta_k}{\|s_k\|^2} s_k, \;\; \theta_k = 12(f_k - f_{k+1}) + 7g_k^T s_k + 5g_{k+1}^T s_k + s_k^T B_k s_k. \tag{2.11}$$

A merit of the new modified secant method is revealed by the following result.

Theorem 2.1. Assume that the functions f is smooth enough and y_k^* is defined by (2.11). If $\|s_k\|$ is sufficiently small, then we have

$$s_k^T(G_{k+1}s_k - y_k^*) = O(\| s_k \|^5),$$
$$s_k^T(G_{k+1}s_k - \overline{y}_k) = O(\| s_k \|^4),$$

and

$$s_k^T(G_{k+1}s_k - y_k) = O(\| s_k \|^3).$$

Proof. The result follows immediately from (2.3), (2.4) and (2.5) (for details, see Theorem 2.1 of [25]. □ □

Theorem 2.1 implies that the quantity $s_k^T y_k^*$ generated by the modified secant method approximates the second-order curvature $s_k^T G_{k+1} s_k$ with a higher accuracy respect to quantity $s_k^T y_k$ and $s_k^T \overline{y}_k$.

Obviously, determine v_k by (2.11) require expensive computations at each iteration (because of the matrix-vector products), especially for large problems. We are going to propose an effective approach to to reduce the level of computation to determining v_k.

We know direction search computed by

$$d_k = -B_k^{-1} g_k. \tag{2.12}$$

On the other hand, using (1.2), we get

$$s_k = \alpha_k d_k. \tag{2.13}$$

From (2.12) and (2.13), we obtain

$$B_k s_k = -\alpha_k g_k. \tag{2.14}$$

Also, we know that B_k satisfies

$$B_k s_{k-1} = y_{k-1}. \tag{2.15}$$

Using (2.14) and (2.15), the relation (2.11) can be written as

$$y_k^* = y_k + \frac{\theta_k}{\|s_k\|^2} s_k, \;\; \theta_k = 12(f_k - f_{k+1}) + 7g_k^T s_k + 5g_{k+1}^T s_k - \alpha_k^2 d_k^T g_k. \tag{2.16}$$

Hence, we can be written as the secant relation (2.10) as follows:

$$B_{k+1} s_k = y_k^*, \;\; y_k^* = y_k + \rho_k \frac{\theta_k}{\|s_k\|^2} s_k, \tag{2.17}$$

where ρ_k and θ_k are given by (1.7) and (2.16) respectively.

Clearly, this relation is considerably less expensive than that given in (2.10), especially for large scale problems.

For general functions, if B_k is not positive definite d_k may not be a descent direction. To overcome this, we using the idea in [18] and update B_{k+1} by the following rule:

$$B_{k+1} = \begin{cases} B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k^* y_k^{*T}}{s_k^T y_k^*}, & \frac{s_k^T y_k^*}{\|s_k\|^2} \geq \delta, \\ B_k, & \text{otherwise.} \end{cases} \tag{2.18}$$

An attractive property of this update for B_k is that

$$y_k^{*T} s_k > 0, \tag{2.19}$$

which guarantees the positive definiteness of matrix B_k.

We can now give a new BFGS algorithm using our new secant relation as follows.

Algorithm 1: A modified BFGS method.
Step 1: Give ε as a tolerance for convergence, $\sigma_1 \in (0,1)$, $\sigma_2 \in (\sigma_1, 1)$, a starting point $x_0 \in \mathbb{R}^n$, and a positive definite matrix B_0. Set $k = 0$.

Step 2: If $\|g_k\| < \varepsilon$ then stop.

Step 3: Compute a search direction d_k: Solve $B_k d_k = -g_k$.

Step 4: Compute α_k by using the following Wolfe conditions:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \sigma_1 \alpha_k g_k^T d_k, \tag{2.20}$$

and

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma_2 g(x_k)^T d_k. \tag{2.21}$$

Step 5: Set $x_{k+1} = x_k + \alpha_k d_k$. Compute y_k^* by (2.17). Update B_{k+1} by (2.18).

Step 6: Set $k = k + 1$ and go to Step 2.

Next, we will investigate the global and asymptotic superlinear convergence of the proposed algorithm.

3. Convergence analysis

In order to establish the global convergence of the Algorithm 1, we need some commonly used assumptions.

Assumption A. The level set $D = \{x \mid f(x) \leq f(x_0)\}$ is bounded, where x_0 is the starting point of Algorithm 1.

Assumption B. In an open set N containing D, there exists a constant $L > 0$ such that

$$\|g(x) - g(y)\| \leq L \parallel x - y \parallel, \forall x, y \in N.$$

It is clear that assumptions A and B imply that there exists a positive constant γ such that

$$\parallel g(x) \parallel \leq \gamma, \ \forall x \in D. \tag{3.1}$$

From Assumption A and the Wolfe conditions, $\{f(x_k)\}$ is a non-increasing sequence, which ensures $\{x_k\} \subset D$ and the existence of x^* such that

$$\lim_{k\to\infty} f(x_k) = f(x^*). \tag{3.2}$$

To establish convergence of Algorithm 1, we first provide some lemmas.

Lemma 3.1. Let f satisfy assumptions A and B, and $\{x_k\}$ be generated by Algorithm 1 and there exist constants a_1 and a_2 such that,

$$\|B_k s_k\| \le a_1\|s_k\| \text{ and } s_k^T B_k s_k \ge a_2\|s_k\|^2, \tag{3.3}$$

for infinitely k, then we have

$$\liminf_{k\to\infty} g(x_k) = 0. \tag{3.4}$$

Proof. From (3.3) and the relation $g_k = -B_k d_k$ we have

$$d_k^T B_k d_k \ge a_2\|d_k\|^2,\ \ a_2\|d_k\| \le \|g_k\| \le a_1\|d_k\|. \tag{3.5}$$

Let Λ be the set of indices k such that (3.5) holds. By using (2.21) and Assumption B, we have

$$L\alpha_k\|d_k\|^2 \ge (g_{k+1} - g_k)^T d_k \ge (1-\sigma_2) g_k^T d_k. \tag{3.6}$$

This implies that, for any $k \in \Lambda$,

$$\alpha_k \ge \frac{(1-\delta) g_k^T d_k}{L\|d_k\|^2} = \frac{(1-\delta) d_k^T B_k d_k}{L\|d_k\|^2} = \frac{(1-\delta)a_2}{L}. \tag{3.7}$$

On the other hand, from (3.2), we obtain

$$\sum_{k=1}^{\infty}(f_k - f_{k+1}) = \lim_{N\to\infty}\sum_{k=1}^{N}(f_k - f_{k+1}) = \lim_{N\to\infty}(f(x_1) - f_N) = f(x_1) - f^*,$$

which yields

$$\sum_{k=1}^{\infty}(f_k - f_{k+1}) < \infty,$$

Using (2.20), we get

$$\sum_{k=1}^{\infty}\alpha_k g_k^T d_k < \infty,$$

which ensures

$$\lim_{k\to\infty}\alpha_k g_k^T d_k = 0,$$

this together with (3.7) lead to

$$\lim_{k\in\Lambda, k\to\infty} d_k^T B_k d_k = \lim_{k\in\Lambda, k\to\infty} -g_k^T d_k = 0,$$

which along with (3.5), yields (3.4). □

Lemma 3.2. (Theorem 2.1 of [3]) Suppose that there are positive constants m and M such that for all $k \ge 0$

$$\frac{s_k^T y_k^*}{\|s_k\|^2} \ge m \text{ and } \frac{\|y_k^*\|^2}{s_k^T y_k^*} \le M. \tag{3.8}$$

Then, there exist constants a_1 and a_2 such that for any positive integer t (3.3) holds for at least $[\frac{t}{2}]$ values of $k \in \{1, 2, ..., t\}$.

Now, we prove the global convergence for Algorithm 1.

Theorem 3.1. Let f satisfies in assumption A and B, and $\{x_k\}$ be generated by Algorithm 1. Then, we have

$$\liminf_{k\to\infty} g(x_k) = 0. \tag{3.9}$$

Proof. In view of Lemma 3.1, sufficiently show that (3.3) holds for infinitely k.
Let $K = \{k| \frac{s_k^T y_k^*}{\|s_k\|^2} \geq \delta\}$. If K is a finite set, then from (2.18) B_k, is a constant matrix after some finite iterations, clearly (3.3) holds for all large k.

Now, suppose K is a infinite set, then we have

$$s_k^T y_k^* \geq \delta \|s_k\|^2, \qquad \forall k \in K. \tag{3.10}$$

By the definitions of y_k^* and Assumption B, it is easy to see that

$$\frac{\|y_k^*\|^2}{s_k^T y_k^*} \leq M.$$

where $M > 0$ is a constant. Applying Lemma 3.2 to the subsequence $\{B_k\}_{k\in K}$, there exist, constants $a_1 > 0$ and $a_2 > 0$ such that (3.3) holds for infinitely many k. Then, Lemma 3.1 completes the proof. □

FIGURE 1. Total number of iterations performance profiles for the Algorithms.

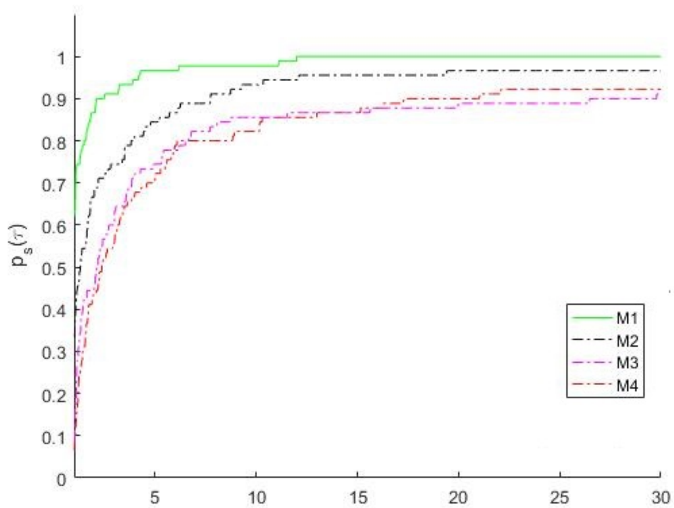
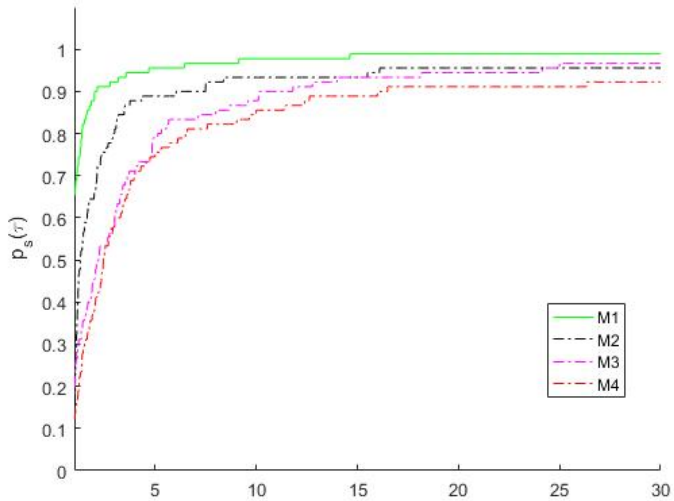


FIGURE 2. Total number of function evaluations performance profiles for the Algorithms.



4. Numerical results

We compare the performance of the following four methods on some unconstrained optimization problems:

M1: proposed method (Algorithm 1) with $a = b = \rho_{max} = 1,\ m = 10$.
M2: the modified BFGS of Peyghami et al. using (1.6) [21] with $a = b = \rho_{max} = 1,\ m = 10$.
M3: the modified BFGS method of Zhang and Xu using (1.4) [29].
M4:the usual BFGS method using (1.3) [3].

We have tested all the considered algorithms on 120 test problems from CUTEr library [16]. A summary of these problems are given in Table 1 of [8]. All codes were written in Matlab 2012 and run on a PC with CPU Intel(R) Core(TM) i5-4200 3.6 GHz, 4 GB of RAM memory and Centos 6.2 server Linux operating system.

In the four algorithms, the initial matrix is set to be the identity matrix, and the step length α_k was computed satisfying the Wolfe conditions, with $\sigma_1 = 0.01$, $\sigma_2 = 0.9$ and $\varepsilon = 10^{-6}$. In Algorithm 1 we set $\delta = 10^{-6}$.

Tables 1-3 in the appendix list estimation errors of these algorithms, where f^* stand for optimal value objective function.
We used the performance profiles of Dolan and Mor̈e (see [12]) to evaluate performance of these two algorithms with respect to the number of iterations and the total number of function and gradient evaluations computed as

$$N_f + nN_g,$$

where N_f and N_g, respectively denote the number of function and gradient evaluations (note that to account for the higher cost of N_g, as compared to N_f, the former is multiplied by n).

Figures 1 and 2 demonstrate the results of the comparisons. From these figures, it is clearly observed that the proposed method (M1) is the most efficient for solving these 120 test problems.

5. Conclusion

We propose a modified secant relation to get a more accurate approximation of the second curvature of the objective function. Then, based on the proposed secant relation, we presented a modified BFGS for solving unconstrained optimization problem. The global convergence of all the proposed methods was established, under suitable assumptions. Numerical results on the collection of problems from the CUTEr library showed the proposed method to be more efficient as compared to several proposed BFGS methods in the literature.

Acknowledgements

We would like to thank the referees and the associate editor, whose very helpful suggestions led to much improvement of this paper.

References

[1] C. G. Broyden, *A class of methods for solving nonlinear simultaneous equations*, Math. Comp., *19* (1965), 577–593.

[2] C. G. Broyden, *The convergence of a class of double rank minimization algorithms-Part 2: The new algorithm*, J. Inst. Math. Appl., *6* (1970), 222–231.

[3] R. Byrd and J. Nocedal, *A tool for the analysis of quasi-Newton methods with application to unconstrained minimization*, SIAM J. Numerical Anal., *26* (1989), 727–739.

[4] R. Byrd, J. Nocedal, and Y. Yuan, *Global convergence of a class of quasi-Newton methods on convex problems*, SIAM J. Numer. Analysis, *24* (1987), 1171–1189.

[5] A. R. Conn, N. Gould, and Ph. L. Toint, *Convergence of quasi-Newton matrices generated by the symmetric rank one update*, Math. Programming, *50* (1991), 177–195.

[6] Y. Dai, *Convergence properties of the BFGS algorithm*, SIAM J. Optim., *13* (2003), 693–701.

[7] W. C. Davidon, *Variable metric methods for minimization*, Argonne National Labs Report, ANL-5990.

[8] R. Dehghani, N. Bidabadi, and M. M. Hosseini, *A new modied BFGS method for unconstrained optimization problems*, Computational and Applied Mathematics, *37* (2018), 5113-5125.

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

[10] J. E. Dennis and R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[11] L. C. W. Dixon, *Quasi-Newton algorithms generate identical points*, Math. Programming, *2* (1972), 383–387.

[12] E. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program., *91* (2002), 201–213.

[13] R. Fletcher, *A new approach to variable metric algorithms*, Comput., *13* (1970), 317–322.

[14] J. A. Ford and A. F. Saadallah, *On the construction of minimisation methods of quasi-Newton type*, J. Comput. Appl. Math., *20* (1987), 239–246.

[15] D. Goldfarb, *A family of variable metric methods derived by variational means*, Math. Comp., *24* (1970), 23–26.

[16] N. I. M. Gould, D. Orban, and Ph. L. Toint, *A constrained and unconstrained testing environment*, revisited, ACM Trans. Math. Softw., *29*(4) (2003), 373–394.

[17] A. Griewank, *The global convergence of partioned BFGS on problems with convex decompositons and Lipschitzian gradients*, Math. Program., *50* (1991), 141–175.

[18] D. H. Li and M. Fukushima, *On the global convergence of BFGS method for nonconvex unconstrained optimization problems*, SIAM Journal on Optimization, *11* (2001), 1054–1064.

[19] D. H. Li and M. Fukushima, *A modified BFGS method and its global convergence in nonconvex minimization*, J. Comput. Appl. Math., *129* (2001), 15–35.

[20] W. F. Mascarenhas, *The BFGS method with exact line searches fails for non-convex objective functions*, Math. Program., *99* (2004), 49–61.

[21] M. R. Peyghami, H. Ahmadzadeh, and A. Fazli, *A new class of efficient and globally convergent conjugate gradient methods in the Dai-Liao family*, Optimization Methods and Software, *30*(4) (2015), 843–863.

[22] M. J. D. Powell, *Some global convergence properties of a variable metric algorithm for minimization without exact line searches*, In: Cottle, R.W., Lemke, C.E. (eds.), Nonlinear Programming, SIAMAMS Proceedings, vol. IX. SIAM, Philadelphia, 1976.

[23] D. F. Shanno, *Conditioning of quasi-Newton methods for function minimization*, Math. Comp., *24* (1970), 647–656.

[24] Ph. L. Toint, *Global convergence of the partioned BFGS algorithm for convex partially separable opertimization*, Math. Program., *36* (1986), 290–306.

[25] Z. Wei, G. Li, and L. Qi, *New quasi-Newton methods for unconstrained optimization problems*, Appl. Math. Comput., *175* (2006), 1156–1188.

[26] G. Yuan and Z. Wei, *Convergence analysis of a modified BFGS method on convex minimizations*, Computational Optimization and Applications, *47* (2010), 237–255.

[27] H. Yabe and M. Takano, *Global convergence properties of nonlinear conjugate gradient methods with modified secant condition*, Comput. Optim. Appl., *28* (2004), 203–225.

[28] J. Z. Zhang, N. Y. Deng, and L. H. Chen, *New quasi-Newton equation and related methods for unconstrained optimization*, J. Optim. Theory Appl., *102* (1999), 147–167.

[29] J. Z. Zhang and C. X. Xu, *Properties and numerical performance of quasi-Newton methods with modified quasi-Newton equation*, J. Comput. Appl. Math., *137* (2001), 269–278.

[30] D. T. Zhu, *Nonmonotone backtracking inexact quasi-Newton algorithms for solving smooth nonlinear equations*, Appl. Math. Comput., *161* (2005), 875–895.

Appendix

TABLE 1. Test results for the two algorithms.

No	$\|f_k - f^*\|_2$			
	M1	M2	M3	M4
1	7.89e-011	4.86e-009	3.48e-009	5.23e-005
2	9.90e-009	1.79e-006	2.09e-004	8.23e-005
3	3.46e-011	4.97e-011	7.09e-011	8.23e-005
4	2.35e-013	4.40e-011	4.48e-011	5.21e-006
5	3.17e-011	3.27e-011	3.27e-011	5.23e-009
6	4.11e-009	4.11e-009	4.11e-009	4.11e-007
7	5.64e-009	5.64e-009	5.64e-009	5.64e-009
8	1.26e-006	1.96e-0036	2.10e-006	68.23e-005
9	4.09e-009	6.95e-09	6.95e-009	7.23e-007
10	3.99e-005	1.09e-006	9.02e-006	9.02e-006
11	1.54e-004	1.54e-004	1.54e-004	1.54e-004
12	4.90e-009	9.74e-006	9.02e-006	9.21e-005
13	2.01e-002	2.01e-002	2.01e-002	2.01e-002
14	5.93e-008	2.18e-007	2.18e-007	3.25e-003
15	5.64e-009	5.64e-009	5.64e-009	5.64e-009
16	1.66e-011	2.96e-009	2.95e-009	3.93e-002
17	9.61e-011	6.35e-011	5.29e-009	5.23e-005
18	2.05e-008	6.68e-008	6.68e-008	8.23e-006
19	2.18e-010	2.15e-007	2.15e-007	8.23e-005
20	2.40e-011	2.32e-006	2.84e-009	3.23e-003
21	2.42e-011	3.45e-011	3.62e-010	3.57e-010
22	1.65e-008	6.34e-008	6.34e-008	5.21e-002
23	2.43e-011	4.42e-010	4.42e-010	4.64e-007
24	8.54e-011	4.16e-010	1.28e-010	68.23e-006
25	6.57e-011	4.63e-011	8.32e-011	8.22e-009
26	3.55e-008	1.39e-007	1.39e-007	6.02e-006
27	2.81e-011	1.93e-011	1.93e-011	3.54e-009
28	3.96e-010	1.62e-009	1.62e-009	5.09e-007
29	2.01e-010	2.01e-010	2.01e-010	2.01e-010
30	5.33e-009	2.18e-006	2.18e-006	3.25e-005
31	2.01e-006	2.01e-006	2.01e-006	3.23e-003
32	1.45e-011	1.45e-011	1.45e-011	2.57e-010
33	3.57e-008	3.75e-008	3.81e-008	3.89e-008
34	6.56e-011	6.49e-010	6.50e-010	6.94e-007
35	8.54e-011	4.16e-010	1.28e-009	8.23e-003
36	6.57e-011	4.63e-011	8.32e-011	8.22e-003
37	3.55e-008	1.39e-007	1.39e-007	6.02e-007
38	2.71e-011	1.33e-011	1.93e-011	3.24e-006
39	3.66e-010	1.65e-009	1.65e-009	5.08e-009
40	8.01e-011	8.01e-011	8.01e-011	2.01e-010

TABLE 2. Test results for the two algorithms.

No	$\|f_k - f^*\|_2$			
	M1	M2	M3	M4
41	3.21e-002	3.21e-002	3.21e-002	2.01e-002
42	1.93e-008	1.18e-007	2.28e-007	3.15e-003
43	5.64e-009	5.64e-009	5.64e-009	5.64e-009
44	6.16e-011	5.96e-009	5.35e-009	2.13e-009
45	2.31e-011	3.15e-011	3.15e-009	7.51e-009
46	3.21e-008	6.18e-008	6.18e-008	6.26e-005
47	5.16e-010	7.15e-011	7.15e-011	8.29e-005
48	2.10e-010	2.12e-009	2.14e-009	2.13e-008
49	2.12e-011	8.41e-011	8.61e-010	9.97e-010
50	1.23e-008	2.33e-008	2.33e-008	3.21e-006
51	3.93e-011	9.12e-010	8.42e-010	5.24e-007
52	8.54e-011	4.16e-011	1.28e-011	8.23e-009
53	9.19e-011	8.86e-011	8.48e-010	6.23e-010
54	4.34e-011	5.45e-011	5.45e-011	6.87e-010
55	2.57e-008	1.75e-008	5.31e-008	2.29e-008
56	3.16e-011	6.89e-010	8.90e-010	5.34e-007
57	2.34e-011	3.16e-010	6.28e-006	8.23e-010
58	9.57e-011	9.63e-011	9.32e-011	8.12e-009
59	1.58e-008	2.32e-008	2.32e-008	3.52e-007
60	2.71e-010	1.33e-010	1.93e-010	2.14e-006
61	7.66e-010	3.65e-009	6.65e-009	5.08e-009
62	5.01e-011	3.01e-011	3.01e-011	6.01e-011
63	2.90e-009	3.79e-009	5.09e-009	7.23e-010
64	1.46e-011	2.97e-011	8.09e-011	9.23e-005
65	2.35e-010	4.40e-010	4.48e-010	5.13e-006
66	3.37e-009	3.97e-010	3.97e-010	6.83e-009
67	2.11e-003	1.11e-003	1.11e-003	1.11e-003
68	5.64e-009	5.64e-009	5.64e-009	5.64e-009
69	1.26e-008	1.96e-003	2.10e-006	68.23e-005
70	4.09e-009	6.95e-09	6.95e-009	7.23e-006
71	2.29e-011	3.09e-011	2.02e-006	7.06e-006
72	3.14e-004	3.52e-004	3.52e-004	3.52e-004
73	4.90e-006	9.74e-004	9.02e-006	9.21e-002
74	2.87e-011	6.68e-011	7.36e-011	6.29e-003
75	1.55e-008	1.39e-007	1.39e-007	6.02e-006
76	2.81e-010	1.93e-011	1.93e-011	3.54e-009
77	3.96e-010	1.62e-009	1.62e-009	5.09e-011
78	2.01e-011	2.01e-011	2.01e-011	2.01e-011
79	3.33e-009	2.18e-006	2.18e-006	1.25e-006
80	8.01e-006	8.01e-006	8.01e-006	7.23e-005

TABLE 3. Test results for the two algorithms.

No	$\|f_k - f^*\|_2$			
	M1	M2	M3	M4
81	6.23e-006	8.33e-006	8.33e-006	1.21e-002
82	2.13e-011	1.12e-010	1.42e-010	6.24e-007
83	2.54e-010	5.16e-010	5.28e-006	6.23e-003
84	9.19e-011	8.86e-011	8.48e-010	6.23e-005
85	4.34e-010	5.45e-010	5.45e-010	36.87e-010
86	6.17e-009	2.15e-009	3.31e-009	6.29e-009
87	1.16e-011	1.89e-010	1.90e-010	3.34e-007
88	3.34e-010	3.16e-010	3.28e-006	5.23e-003
89	1.57e-011	2.63e-011	2.32e-011	1.12e-003
90	3.58e-008	3.32e-007	3.32e-007	2.52e-007
91	3.71e-011	5.33e-011	5.93e-011	7.14e-006
92	1.26e-010	2.65e-009	2.65e-009	3.08e-009
93	9.01e-011	8.01e-011	8.01e-011	5.01e-011
94	4.90e-009	5.79e-006	5.09e-004	3.23e-005
95	2.46e-011	4.97e-011	4.09e-011	6.23e-006
96	1.35e-009	2.40e-010	2.48e-010	3.23e-009
97	4.37e-009	5.97e-010	5.97e-010	6.83e-009
98	6.11e-003	3.11e-003	3.11e-003	3.11e-003
99	164e-009	1.64e-009	1.64e-009	1.64e-009
100	6.09e-009	5.95e-09	5.95e-009	5.23e-011
101	2.42e-010	3.45e-011	3.62e-010	3.57e-010
102	1.65e-008	6.34e-008	6.34e-008	5.21e-002
103	2.43e-011	4.42e-010	4.42e-010	4.64e-007
104	1.54e-010	2.16e-010	2.28e-006	6.23e-009
105	8.57e-011	3.63e-011	3.32e-011	9.22e-003
106	3.15e-008	2.39e-007	2.39e-007	5.02e-006
107	2.81e-010	1.93e-009	1.93e-009	3.54e-008
108	3.96e-010	1.62e-009	1.62e-009	5.09e-010
109	2.01e-011	2.01e-011	2.01e-011	2.01e-011
110	3.53e-009	1.58e-006	1.58e-006	6.25e-005
111	1.03e-006	1.03e-006	1.03e-006	1.03e-006
112	2.45e-009	2.45e-009	2.45e-009	3.57e-010
113	8.57e-008	6.75e-008	6.81e-008	7.89e-008
114	5.16e-011	3.49e-010	3.80e-010	7.44e-007
115	1.14e-011	2.26e-010	3.38e-006	6.13e-008
116	3.27e-009	4.13e-010	5.31e-010	6.12e-003
117	1.22e-008	2.32e-007	2.32e-007	5.06e-007
118	5.21e-010	6.23e-011	6.93e-011	1.11e-006
119	6.16e-010	5.15e-009	5.15e-009	4.31e-009
120	2.21e-008	3.11e-008	3.11e-008	5.21e-005