Indonesian Journal of Electrical Engineering and Computer Science
Vol. 13, No. 3, March 2019, pp. 945~953
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v13.i3.pp945-953

A new class of BFGS updating formula based on the new quasi-newton equation

Basim A. Hassan, Hussein K. Khalo
Department of Mathematics, College of Computers Sciences and Mathematics, University of Mosul, Iraq.

Article Info

Article history:

Received Sep 15, 2018
Revised Nov 28, 2018
Accepted Dec 17, 2018

Keywords:

Convergence properties
Quasi-Newton equations
Quasi-Newton methods

ABSTRACT

Quasi-Newton methods” are amongst the mainly useful and competent iterative process for solving unrestrained minimization functions. In this paper we derive a new quasi-Newton equation with on the Hessian estimate updates and alterations intended at developing their performance. The “Numerical results” illustrate that the proposed technique useful for the known test functions.

Corresponding Author:

Basim A. Hassan,
Department of Mathematics,
College of Computers Sciences and Mathematics,
University of Mosul, Iraq.
Email: basimabas39@gmail.com, husseinzomar54@gmail.com

1. INTRODUCTION

Uncovering a solution to a general broad degree nonlinear optimization Problem:

$$Min \ \ell(x) \ , x \in R^n \tag{1}$$

where where ℓ is a smooth function of n variables, by quasi-Newton methods is painstaking. Quasi-Newton methods are awfully useful utensils for solving unrestrained optimization problems [1]. At the k th iteration of the quasi-Newton method, a symmetric and non negative definite B_k is known, and a search direction is computed by:

$$d_k = - B_k^{-1} J_k \ , \tag{2}$$

where J_k is the gradient of ℓ evaluated at the current iterate x_k. One then computes the next iterate by:

$$x_{k+1} = x_k + \alpha_k d_k \tag{3}$$

It computed a step length α_k that make sure the approval of the Wolfe conditions:

$$\ell(x_k + \alpha_k d_k) \le \ell(x_k) + \delta \alpha_k J_k^T d_k \tag{4}$$

$$d_k^T J(x_k + \alpha_k d_k) \ge \sigma d_k^T J_k \tag{5}$$

where $0 < \delta < \sigma < 1$. For more details can be found in [2]. By tradition, $\{B_k\}$ satisfies the quasi-Newton equation:

$$B_{k+1}\partial_k = p_k , \tag{6}$$

where $\partial_k = x_{k+1} - x_k$ and $p_k = J_{k+1} - J_k$. Let H_k be the inverse of B_k. The famous inverse update H_k is the standard BFGS formula:

$$H_{k+1}^{BFGS} = H_k - \frac{H_k p_k \partial_k^T + \partial_k p_k^T H_k}{\partial_k^T p_k} + \left[1 + \frac{p_k^T H_k p_k}{\partial_k^T p_k}\right] \frac{\partial_k \partial_k^T}{\partial_k^T p_k} \tag{7}$$

Certainly, BFGS method is one of the most excellent methods and doing to now for solving (1). For more details can be found in [3]. A lot of approaches have been suggested to find better the quasi- Newton Hessian estimate updates. In this fragment we sketch a few latest suggested updates take by modifying the vector p_k, as shown in Table 1.

Table 1. Modifying The Vector

Name methods	Difference in gradients	References
P	$p_k^* = \varphi_k \ p_k + (1-\varphi_k) B_k \ \partial_k$	[4]
LF	$p_k^* = p_k + \tau_k \partial_k \ , \quad \tau_k \le 10^{-6}$	[5]
WLQ	$p_k^* = p_k + \frac{2(\ell_k - \ell_{k+1}) + (J_{k+1} + J_k)^T \partial_k}{\lVert \partial_k \rVert^2} \partial_k$	[6]
ZDC	$p_k^* = p_k + \frac{6(\ell_k - \ell_{k+1}) + 3(J_{k+1} + J_k)^T \partial_k}{\lVert \partial_k \rVert^2} \partial_k$	[7]

The idea of variant QN methods had been studied by many researchers for example, [8], [9].

Now we will derive new quasi-Newton equations and analyze its convergence.

2. DERIVING NEW QUASI-NEWTON EQUATION AND AALGORITHM

In this fragment we derive the new quasi-Newton equations. Therefore we can apply it to functions more general than quadratic as from:

$$\ell_k = \ell_{k+1} - J_{k+1}^T \partial_k + \frac{1}{2} \partial_k^T U_{k+1} \partial_k + \frac{1}{6} \partial_k^T T_{k+1} \partial_k \tag{8}$$

where T_{k+1} is the tensor of ℓ at the point x_k. we attain, by revoking the conditions which comprise the tensor:

$$\partial_k^T U_{k+1} \partial_k = p_k^T \partial_k + 6(\ell_k - \ell_{k+1}) + 3(J_{k+1} + J_k)^T \partial_k \tag{9}$$

For more details can be found in [10]. So, would like derivative of manages to formulate general quadratic, we have:

$$\frac{1}{2}\partial_k^T U_{k+1}\partial_k = \frac{1}{2} p_k^T \partial_k + 3(\ell_k - \ell_{k+1}) + \frac{3}{2}(J_{k+1} + J_k)^T \partial_k \tag{10}$$

The step size scalar α_k, which minimizes $\ell(x_k)$, is approximated by:

$$\alpha_k = -\frac{J_k^T d_k}{d_k^T U d_k} \tag{11}$$

After some algebraic manipulations one obtains:

$$\partial_k^T U_{k+1}\partial_k = \frac{1}{2} p_k^T \partial_k + 3(\ell_k - \ell_{k+1}) + \frac{3}{2} J_{k+1}^T \partial_k + J_k^T \partial_k \tag{12}$$

Since $B_{k+1}\partial_k$ is need to estimated $U_{k+1}\partial_k$, it is reasonable to need:

$$\partial_k^T B_{k+1}\partial_k = \frac{1}{2} p_k^T \partial_k + 3(\ell_k - \ell_{k+1}) + \frac{3}{2} J_{k+1}^T \partial_k + J_k^T \partial_k \tag{13}$$

A good choice to estimate $B_{k+1}\partial_k$ is known by:

$$B_{k+1}\partial_k = p_k^*, \quad p_k^* = \frac{1}{2} p_k + \frac{3(\ell_k - \ell_{k+1}) + 3/2(J_{k+1}^T\partial_k) + J_k^T\partial_k}{\partial_k^T \omega_k}\omega_k \tag{14}$$

where u_k is any vector such that $\partial_k^T \omega_k \neq 0$.

Varieties of this quasi-Newton equations differ in the way of selecting the vector ω_k in (14) we have the forms:

1. First case $\omega_k = p_k$ gives:

$$B_{k+1}\partial_k = p_k^*, \quad p_k^* = \frac{1}{2} p_k + \frac{3(\ell_k - \ell_{k+1}) + 3/2(J_{k+1}^T\partial_k) + J_k^T\partial_k}{\partial_k^T p_k} p_k \tag{15}$$

2. Second case $\omega_k = J_{k+1}$ gives:

$$B_{k+1}s_k = p_k^*, \quad p_k^* = \frac{1}{2} p_k + \frac{3(\ell_k - \ell_{k+1}) + 3/2(J_{k+1}^T\partial_k) + J_k^T\partial_k}{\partial_k^T J_{k+1}} J_{k+1} \tag{16}$$

Different change gradient used in quasi-Newton equation for yield different quasi-Newton methods. The new CBFGS algorithm can be stated formally as follows.

Step 1: Data $x_0 \in R^n$ and $H_0 = I$. Set $k = 0$.

Step 2: Stop if $\|J_k\| = 0$.

Step 3: Calculate d_k by:

$$d_k = -H_k J_k$$

Step 4: Finds a α_k which satisfies the (4) and (5).

Step 5: Iterative process be as $x_{k+1} = x_k + \alpha_k d_k$.

Step 6: Update H_0 for times to find H_{k+1} by (7) and (14).
Step 7: Put $k = k+1$. Go to step 2.

A property positive definite of H_k is awfully important, can be verify in the next theorem.

Theorem 2.1.

Let $(\alpha_k, x_{k+1}, J_{k+1}, d_{k+1})$ be generated by the new algorithm. Then B_{k+1} is positive definite for all k provided that $\partial_k^T p_k^* > 0$.

Proof.

Now, we evaluate the quantity $\partial_k^T y_k^*$. If the step length α_k satisfies the Wolfe conditions (4) and (5), then we have:

$$\begin{aligned}\partial_k^T p_k^* &= \frac{1}{2} p_k^T \partial_k + 3(\ell_k - \ell_{k+1}) + \frac{3}{2} J_{k+1}^T \partial_k + J_k^T \partial_k \\ &= 3(\ell_k - \ell_{k+1}) + \frac{1}{2} J_k^T \partial_k + J_{k+1}^T \partial_k \\ &\geq \left[\frac{1}{2} - 3\delta + \sigma\right] J_k^T \partial_k\end{aligned} \tag{17}$$

To attain this intention, prefer the values of σ and δ with $1/2 > \delta > 1/3$ and $1/2 - 3\delta + \sigma < 0$. Noting the $\partial_k^T J_k = \alpha_k d_k^T J_k < 0$, we know that there exists a constant $m < 0$ such that:

$$\partial_k^T p_k^* \geq m\, d_k^T J_k > 0 \tag{18}$$

The proof is complete.

3. GLOBAL PROPERTY

We ready the local convergence property of the modified BFGS method. The following assumption is required.

Assumption

(i) The level set $S = \{x \in R^n \,|\, \ell(x) \leq \ell(x_0)\}$ is bounded.

(ii) The function f is twice continuously differentiable on S and there exists a constant $L > 0$ such that:

$$\|J(x) - J(y)\| \leq L\, \|x - y\| \tag{19}$$

Since $\{\ell_k\}$ is a diminishing series, it is obvious that the series $\{x_k\}$ generated by new Algorithm is enclosed in S, and there exists a constant ℓ^* such that:

$$\lim_{k\to\infty} \ell_k = \ell^* \tag{20}$$

(iii) The function ℓ is uniformly convex, i.e., there exist positive constants M and m such that:

$$m\,\|d\|^2 \leq d^T U(x) d \leq M\,\|d\|^2 \tag{21}$$

holds for all $x \in S$ and $d \in R^n$, where $U(x) = \nabla^2 \ell(x)$. These assumptions are the same as those in [11].

Theorem 2.2.

Let $\{x_k\}$ be generated by the new algorithm. Then we have:

$$\begin{aligned} m\,\|\partial_k\|^2 &\le \partial_k^T p_k^* \le M\,\|\partial_k\|^2, \\ \|p_k^*\| &\le (L+M)\,\|\partial_k\|, \end{aligned} \tag{22}$$

and

$$\sum_{k=1}^{\infty} -\alpha_k J_k^T d_k \le +\infty \tag{23}$$

Proof.

Following the definition of p_k^* and the Taylor's series, we get:

$$\begin{aligned} \partial_k^T p_k^* &= \frac{1}{2} p_k^T \partial_k + 3(\ell_k - \ell_{k+1}) + \frac{3}{2} J_{k+1}^T \partial_k + J_k^T \partial_k \\ &= p_k^T \partial_k + 6(\ell_k - \ell_{k+1}) + 3(J_{k+1} + J_k)^T \partial_k \\ &= 4J_{k+1}^T \partial_k + 2J_k^T \partial_k + 6(\ell_k - \ell_{k+1}) \end{aligned} \tag{24}$$

By using Taylor's series, and mean value theorem, we get:

$$\ell_k = \ell_{k+1} - J_{k+1}^T \partial_k + \frac{1}{2} \partial_k^T U(\eta_k) \partial_k \tag{25}$$

From (24) and (25) we get:

$$\begin{aligned} s_k^T p_k^* &= \frac{1}{2} p_k^T \partial_k + 3(\ell_k - \ell_{k+1}) + \frac{3}{2} J_{k+1}^T \partial_k + J_k^T \partial_k \\ &= p_k^T \partial_k + 6(\ell_k - \ell_{k+1}) + 3(J_{k+1} + J_k)^T \partial_k \\ &= 4[\frac{1}{2} \partial_k^T U(\eta_k) \partial_k - (\ell_k - \ell_{k+1})] + 2J_k^T \partial_k + 6(\ell_k - \ell_{k+1}) \\ &= 2\partial_k^T U(\eta_k) \partial_k + 2J_k^T \partial_k + 2(\ell_k - \ell_{k+1}) \end{aligned} \tag{26}$$

By using quadratic function, and mean value theorem, we have:

$$\ell_{k+1} = \ell_k + \partial_k^T J_k + \frac{1}{2} \partial_k^T U(\eta_k) \partial_k \tag{27}$$

Therefore, it follows from (26) and (27) that:

$$\begin{aligned} \partial_k^T p_k^* &= 2\partial_k^T U(\eta_k) \partial_k + 2J_k^T \partial_k + 2(\ell_k - \ell_{k+1}) \\ &= 2\partial_k^T U(\eta_k) \partial_k - \partial_k^T U(\eta_k) \partial_k \\ &= \partial_k^T U(\eta_k) \partial_k \end{aligned} \tag{28}$$

where

$$\eta_k = x_k + \xi (x_{k+1} - x_k) \text{ and } \xi \in (0\,,1) \tag{29}$$

Mingling with Assumption (iii), it is simple to catch:

$$m\left\|\partial_k\right\|^2 \le \partial_k^T p_k^* \ \le M\left\|\partial_k\right\|^2 \tag{30}$$

Using definition of p_k^* and the Taylor's series once more, we get:

$$\begin{aligned}\left\|p_k^*\right\| &= \left\|\frac{1}{2}p_k + \frac{3(\ell_k-\ell_{k+1})+3/2(J_{k+1}^T\partial_k)+J_k^T\partial_k}{\partial_k^T\omega_k}\omega_k\right\| \\ &\le \frac{1}{2}\left\|p_k\right\| + \frac{\left|3(\ell_k-\ell_{k+1})+3/2(J_{k+1}^T\partial_k)+J_k^T\partial_k\right|}{\left\|\partial_k\right\|\left\|\omega_k\right\|}\left\|\omega_k\right\| \\ &\le \frac{1}{2}\left\|p_k\right\| + \frac{\left|\ [\partial_k^T U(\eta_k)\partial_k\ -1/2(\ \partial_k^T p_k)]\ \right|}{\left\|\partial_k\right\|\left\|\omega_k\right\|}\left\|\omega_k\right\| \\ &\le \left\|p_k\right\| + \frac{\left|\ [\partial_k^T U(\eta_k)\partial_k\,]\right|}{\left\|\partial_k\right\|} \ \le\ L\left\|\partial_k\right\| + M\left\|\partial_k\right\| \\ &\le \left(L+M\right)\left\|\partial_k\right\|\end{aligned} \tag{31}$$

Now we turn to the proof of (23). By the WWP rule (4) and Assumption (ii) we obtain:

$$-(1-\sigma)J_k^T d_k \le (J_{k+1}-J_k)^T d_k \le L\alpha_k\left\|d_k\right\|^2 \tag{32}$$

This yields that:

$$\alpha_k\ \ge\ \frac{-(1-\sigma)J_k^T d_k}{L\left\|d_k\right\|^2} = \frac{(1-\sigma)d_k^T B_k d_k}{L\left\|d_k\right\|^2} \ge \frac{(1-\sigma)\ m}{L} = \tilde{\alpha}_k \tag{33}$$

Conversely, from (1), we obtain:

$$\sum_{k=1}^{\infty}(\ell_k-\ell_{k+1}) = \lim_{N\to\infty}\sum_{k=1}^{\infty}(\ell_k-\ell_{k+1}) = \lim_{N\to\infty}(\ell_1-\ell_{k+1}) = \ell_1-\ell^* \tag{34}$$

Thus,

$$\sum_{k=1}^{\infty}(\ell_k-\ell_{k+1}) \le +\infty, \tag{35}$$

which mixed with the WWP law (4) that is:

$$\sum_{k=1}^{\infty}-\alpha_k J_k^T d_k \le +\infty \tag{36}$$

Currently we established the worldwide convergence of new Algorithm.

Theorem 2.3.

Let $\{x_k\}$ be created by new Algorithm and let ℓ satisfies Assumptions i and ii. Suppose to facilitate there exists constants κ_1 and κ_2 such the subsequent relation:

$$\|B_k\partial_k\| \;\le\; \kappa_1\|\partial_k\| \text{ and } \partial_k^T B_k \partial_2 \ge \kappa_2\|\partial_k\|^2 \tag{37}$$

holds, Then we have:

$$\lim_{k\to\infty} \inf \;\|J_k\| = 0 \tag{38}$$

Proof.

We proceed by contradiction, we assume $\|J_k\| \ge \varepsilon$ for all k with some positive constant ε. Taking into account $B_k\partial_k = \alpha_k B_k d_k = -\alpha_k J_k$, then $\|J_k\|^2 = (1/\alpha_k)^2\|B_k\partial_k\|^2$, we get from (36) that:

$$\begin{aligned} \infty \;>\; & \sum_{k=1}^{\infty}(-J_k^T d_k) \;\ge\; \sum_{\forall k}(-J_k^T d_k) \\ & = \sum_{\forall k}\frac{1}{\alpha_k}\partial_k^T B_k\partial_k = \sum_{\forall k}\alpha_k \frac{\|J_k\|^2}{\|B_k\partial_k\|^2}\partial_k^T B_k \partial_k \\ & \ge \sum_{\forall k}\overset{\sim}{\alpha}_k\, \varepsilon^2\kappa_2\kappa_1^{-2} = +\,\infty \end{aligned} \tag{39}$$

where the inequality follows from $\alpha_k \ge \overset{\sim}{\alpha}_k$ inequalities $\|B_k\partial_k\| \;\le\; \kappa_1\|\partial_k\|$ and $\partial_k^T B_k\partial_2 \ge \kappa_2\|\partial_k\|^2$, which concludes the proof.

4. NUMERICAL RESULTS AND DISCUSSION

Now, we details the numerical results for Algorithms CBFGS and BFGS. The problems that we tested are from [12,13]. Himmeblau used the next stop law see in [14] “If $|\ell(x_k)| < 10^{-5}$, let $stop\,1 = |\ell(x_k) - \ell(x_{k+1})| \;/\; |\ell(x_k)|$; Otherwise, let $stop\,1 = |\ell(x_k) - \ell(x_{k+1})|$. For each problem, if $\|J_k\| < \varepsilon$ or $stop\,1 < 10^{-5}$ was satisfied, the program will be stopped. All codes were written in MATLAB 4.4 and Windows XP operation system. The parameters are chosen as: $\delta = 0.1$, $\sigma = 0.9$, $\varepsilon = 10^{-5}$ and the initial matrix $B_0 = I$ is the unit matrix”. Table 1 explains the results, where the columns contain the following implying:

Problem : the name of the test problem in MATLAB;
Dim: the dimension of the problem;
NI : the number of iterations;
NF : the number of function evaluations.

Their numerical experience signify that numerous updates from this idea labored well in applied, specially the modified update $\omega_k = J_{k+1}$ but the modified update $\omega_k = p_k$ give a slight improvement over the original BFGS method. Comparison of several amounts between different quasi-Newton methods as shown in Table 1.

Generally, we can compute the percentage performance of the new proposed algorithms compared against the standard BFGS algorithm for the general tools NI and NF as follows. Relative efficiency of the new algorithms as shown in Table 2.

Table 1. Comparison of Several Amounts Between Different Quasi-Newton Methods

		BFGS algorithm		CBFGS with $\omega_k = p_k$		CBFGS with $\omega_k = J_{k+1}$	
P. No.	n	NI	NF	NI	NF	NI	NF
Rose	2	35	140	37	146	8	48
Froth	2	9	26	8	23	5	16
Badscp	2	43	166	39	142	3	11
Badsc	2	3	30	3	30	3	30
Beale	2	15	50	13	43	6	21
Jensam	2	2	27	2	27	2	27
Helix	3	34	113	28	90	7	20
Bard	3	16	54	17	55	6	19
Gauss	3	2	4	2	4	2	4
Gulf	3	2	27	2	27	2	27
Box	3	2	27	2	27	2	27
Sing	4	20	60	11	35	5	17
Wood	4	19	61	19	60	4	13
Kowosb	4	21	65	21	117	5	13
Bd	4	17	54	15	46	5	17
Osb1	5	2	27	2	27	2	27
Biggs	6	25	72	8	48	4	12
Osb2	11	3	31	3	13	3	31
Watson	20	31	102	31	97	4	13
Singx	400	64	209	35	109	5	17
Pen1	400	2	27	2	27	2	27
Pen2	200	2	5	2	5	2	5
Vardim	100	2	27	2	27	2	27
Trig	500	9	33	9	32	8	28
Bv	500	2	4	2	4	2	4
Ie	500	6	61	7	19	6	16
Band	500	57	281	15	82	5	16
Lin	500	2	4	2	4	2	4
Lin1	500	3	7	3	7	3	7
Lino	500	3	7	3	7	3	7
Total		453	1756	345	1380	118	551

Table 2. Relative Efficiency of the New Algorithms

	BFGS algorithm	CBFGS with $\omega_k = p_k$	CBFGS with $\omega_k = J_{k+1}$
NI	100%	76.15%	26.04%
NF	100%	78.58%	43.37%

5. CONCLUSIONS

In this paper, supported the new QN-equation, we've got projected some new quasi-Newton strategies. The convergence results and also the understanding experimentation make sure that the approach explicit during this paper is fruitful.

REFERENCES

[1] Nocedal J, Wright S. Numerical Optimization. New York: *Springer-Verlag*. 1999.

[2] Nocedal J, Yuan Y. Analysis of a self-scaling quasi-Newton method. *J. Mathematical programming*. 1993; 61: 19-37.

[3] Zhiwei Q. The relationships between CG, BFGS and two limited-memory algorithms. *Electronic Journal of Undergraduate Mathematics*. 2007; 12: 5-20.

[4] Powell D. Algorithms for Nonlinear Constraints that Use Lagrange Functions. *Math. Programming*. 1978; 14: 224-248.

[5] Li D, Fukushima M. A modified BFGS method and its global convergence in non convex minimization. *J. Comput. Appl. Math*. 2001; 129: 15-35.

[6] Wei Z, Li G, Qi L. New quasi-Newton methods for unconstrained optimization problems. Math. Program. *Applied Mathematics and Computation*. 2006; 175: 1156-1188.

[7] Zhang Z, Deng Y, Chen H. *Quasi-Newton equation and related methods for unconstrained optimization. JOTA.* 1999; 102: 147-167.

[8] Ghani N., Kamaruddin S., Ramli M., Musirin I. And Hashim H. Enhanced BFGS quasi-Newton backpropagation models on MCCI data. *Indonesian Journal of Electrical Engineering and Computer Science. 2017;8:101-106.*

[9] Yong L., Tuo S. Quasi-Newton method for absolute value equation based on upper uniform smoothing approximation function. *TELKOMNIKA Telecommunication, Computing, Electronics and Control* . 2016:14:1134-1141.

[10] JJianzhong Z, Xu C. Properties and numerical performance of quasi-newton methods with modified quasi-newton equations. *Journal of Computational and Applied Mathematics*. 2001; 137: 269–278.

[11] Wei Z, Yu G, Yuan G, Lian Z. The superlinear convergence of a modified BFGS-type method for unconstrained optimization. *Comput. Optim. Appl*. 2004; 29: 315-332.

[12] More J, Garbow B, Hillstrome K. Testing unconstrained optimization software. *ACM Trans. Math. Software*. 1981; 7: 17-41.

[13] LLi C.. A modified conjugate gradent method for unconstrained optimization. *TELKOMNIKA Telecommunication, Computing, Electronics and Control*. 2013; 11: 6373-6380.

[14] YYuan Y, Sun W. *Theory and Methods of Optimization. Science Press of China.* 1999.