ISSN 2278-3091
Volume 8, No.6, November – December 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse74862019.pdf
https://doi.org/10.30534/ijatcse/2019/74862019

A New Quasi-Newton (SR1) With PCG Method for Unconstrained Nonlinear Optimization

Salah Gazi Shareef [1], Alaa Luqman Ibrahim [2]
[1, 2] Department of Mathematics, Faculty of Science, University of Zakho, Zakho, Kurdistan Region, Iraq.
[1]salah.shareef@uoz.edu.krd, [2] alaa.ibrahim@uoz.edu.krd

ABSTRACT

The quasi-newton equation (QN) plays a key role in contemporary nonlinear optimization. In this paper, we present a new symmetric rank-one (SR1) method by using preconditioning conjugate gradient (PCG) method for solving unconstrained optimization problems. The suggested method has an algorithm in which the usual SR1 Hessian is updated. We show that the new quasi-newton (SR1) method maintains the Quasi- Newton condition and the positive definite property. Numerical experiments are reported which produces by the new algorithm better numerical results than those of the normal (SR1) method by using PCG algorithm based on the number of iterations (NOI) and the number of functions evaluation (NOF).

Key words: Unconstrained optimization, Quasi-Newton equation, Hessian approximation, Symmetric rank-one update.

1. INTRODUCTION

In this paper, we are concerned to resolving unconstrained optimization problem by using quasi-Newton methods, given by

$$\min f(z) \qquad \forall\, z \in R^n \tag{1}$$

where $f : R^n \to R$ is twice continuously differentiable. Starting from a point z_1 and a symmetric positive-definite matrix B_1, a quasi-Newton method is an iterative scheme that generates sequences $\{z_i\}$ and $\{B_i\}$ by the iteration

$$z_{i+1} = z_i + \lambda_i d_i, \quad i = 0,1,2,\dots, \tag{2}$$

where $\lambda_i > 0$ is the step length and d_i is a search direction of f at z_i.The search direction d_i determines the line search method (see [9, 13]). The quasi-Newton method is defined by

$$B_i s_i = -g_i \tag{3}$$

where $g_i = \nabla f\,(z_i)$, B_i is the quasi-Newton update matrix, and the sequence $\{B_i\}$ satisfies the so-called quasi-Newton condition

$$B_{i+1} s_i = y_i \tag{4}$$

(for more details, see [7]).

1.1 The Symmetric Rank-One (SR1)

The symmetric rank-one (SR1) update comes from solving σ, ν from the general update form

$$B_{i+1} = B_i + \sigma \nu \nu^T \tag{5}$$

The solution of the problem gives the update the approximated Hessian matrix,

$$B_{i+1} = B_i + \frac{(y_i - B_i s_i)(y_i - B_i s_i)^T}{(y_i - B_i s_i)^T s_i} \tag{6}$$

By using the Sherman-Morrison-Woodbury formula [4], we obtain the following inversed Hessian update for SR1 [6]:

$$H_{i+1} = H_i + \frac{(s_i - H_i y_i)(s_i - H_i y_i)^T}{(s_i - H_i y_i)^T y_i} \tag{7}$$

Then the Quasi-Newton equation or Quasi-Newton condition [10] is defined as

$$H_{i+1} y_i = s_i \tag{8}$$

1.2 Preconditioning Conjugate Gradient (PCG)

The PCG method was presented in paper by Axelsson in 1972 [3]. It was advanced with object of accelerating the convergence of the conjugate gradient (CG) method by a transformation of variables while keeping the basic properties of the method. Many authors have extended this type of algorithms [1].
The search direction to the preconditioned (PCG) method is defined by

$$d_0 = -H_0 g_0 \tag{9}$$

$$d_{i+1} = -H_i g_{i+1} + \beta_i d_i \tag{10}$$

$$\beta_i = \frac{y_i^T H_i g_{i+1}}{d_i^T y_i}, \tag{11}$$

Where H_i defined in equation (7).

1.3 Self-Scaling Quasi-Newton Methods

Several self-scaling have been propounded by some scholars like Yang, Xu and Gao [11] made a little modification for self-scaling symmetric rank one update with Davidon's optimal condition [5] (SHSR1) as follow

$$H_{i+1} = \mu_i H_i + \frac{(s_i-\mu_i H_i \hat{y}_i)(s_i-\mu_i H_i \hat{y}_i)^T}{\hat{y}_i{}^T(s_i-\mu_i H_i \hat{y}_i)} \qquad (12)$$

where μ_i is the scaling factor, $\hat{y}_i = \left(\frac{1+\theta_i}{s_i^T y_i}\right) y_i$

$$\theta_i = 6\big(f(z_i) - f(z_{i+1})\big) + 3(g_i - g_{i+1})^T s_i$$

To improve the performance of QN update, S. Shareef, and H. Jameel [8] proposed to choose H_{k+1} to satisfy the following modified equation $H_{i+1}\bar{y}_i = \omega s_i$, where $\omega > 0$, this class of updates can be written as

$$H_{i+1} = H_i + \frac{(s_i-\omega H_i \hat{y}_i)(s_i-\omega H_i \hat{y}_i)^T}{\omega \hat{y}_i{}^T(s_i-\omega H_i \hat{y}_i)} \qquad (13)$$

Where $\omega = t(1 + (1-\theta)(\frac{1}{\sigma} - 1)), \quad 0 < \theta < 1$

And $\sigma = \frac{2\sqrt{\epsilon_i}(1+\|z_{i+1}\|)}{\|s_i\|}$, ϵ_i is error machine.

2. DERIVATION OF A QUASI-NEWTON METHOD SR1

In this section, we will drive the SR1 methods named as H_{i+1}^{New}. The major idea of our method is to improve the performance of QN update, by choosing H_{i+1} to satisfy the following modified QN condition:

$$H_{i+1}\,\bar{y}_i = \tau s_i \qquad (14)$$

Where $\bar{y}_i = (1 + (1-\vartheta)\rho_i)y_i, \quad \vartheta \in (0,1)$ and $\rho_i = \frac{s_i^T y_i}{\|s_i\|^2}$

$$d_{i+1} = -H_{i+1} g_{i+1} \qquad (15)$$

Now by multiplying both side of above equation by $\bar{y}_i$

$$d_{i+1}^T \bar{y}_i = -g_{i+1}^T H_{i+1} \bar{y}_i \qquad (16)$$

By using equation (14) we obtained

$$(1 + (1-\vartheta)\rho_i) d_{i+1}^T y_i = -\tau g_{i+1}^T s_i \qquad (17)$$

By using the conjugacy condition [12],

$$d_{i+1}^T y_i = -t g_{i+1}^T s_i\,, t \geq 0 \qquad (18)$$

So,

$$\tau = t(1 + (1-\vartheta)\rho_i) \text{ where } t \geq 0. \qquad (19)$$

Then, the new SR1 becomes as follows:

$$H_{i+1}^{New} = H_i + \frac{(\tau s_i - H_i \bar{y}_i)(\tau s_i - H_i \bar{y}_i)^T}{(\tau s_i - H_i \bar{y}_i)^T \bar{y}_i} \qquad (20)$$

2.1 The Algorithm of the PCG-Method with New SR1

Step (1): Set $i = 0$, select z_0 and a real symmetric positive definite $H_0 = I$, $\varepsilon = 1 \times 10^{-5}$ and Compute g_0.
Step (2): Compute $g_i = \nabla f(z_i)$.
Step (3): Compute $d_i = -H_i g_i$.
Step (4): Find $\alpha_i > 0$, satisfying the strong Wolfe condition.
Step (5): Set $s_i = \alpha_i d_i$, $z_{i+1} = z_i + s_i$ and $y_i = g_{i+1} - g_i$.
Step (6): Compute g_{i+1} if $\|g_{i+1}\| < \varepsilon$, then stop.
Step (7): Calculate H_{i+1}^{New} using equation (20), and

$$d_{i+1} = -H_{i+1} g_{i+1} + \frac{\bar{y}_i{}^T H_{i+1} g_{i+1}}{d_i^T \bar{y}_i} d_i$$

Step (8): If $|g_i^T g_{i+1}| \geq 0.2\|g_{i+1}\|^2$ go to step (3) else continue.
Set $i = i + 1$ and repeat from Step (4).

Theorem 1: If the new SR1 method which is defined by (20) is applied to the quadratic with Hessian $G = G^T$, then $H_{i+1}^{New}\,\bar{y}_i = \tau s_i\,, i \geq 0$.

Proof: Multiplying both sides of equation (20) by $\bar{y}_i$ from the right, we have:

$$H_{i+1}^{New}\,\bar{y}_i = H_i\,\bar{y}_i + \frac{(\tau s_i - H_i \bar{y}_i)(\tau s_i - H_i \bar{y}_i)^T \bar{y}_i}{(\tau s_i - H_i \bar{y}_i)^T \bar{y}_i} \qquad (21)$$

Since $(\tau s_i - H_i \bar{y}_i)^T \bar{y}_i$ is scalar. So, we have

$$H_{i+1}^{New}\,\bar{y}_i = H_i\,\bar{y}_i + +(\tau s_i - H_i\,\bar{y}_i).$$

Then, $H_{i+1}^{New}\,\bar{y}_i = \tau s_i$. (22)

The proof is then complete. ■

Theorem 2: If H_i^{New} is positive definite, then the matrix H_{i+1}^{New} which is generated by (20) is also positive definite.

Proof: Multiplying both sides of equation (20) by $\bar{y}_i$ from the right and by $\bar{y}_i{}^T$ from the left, we get

$$\bar{y}_i{}^T H_{i+1}^{New}\,\bar{y}_i = \bar{y}_i{}^T H_i\,\bar{y}_i + \frac{\bar{y}_i{}^T(\tau s_i - H_i \bar{y}_i)(\tau s_i - H_i \bar{y}_i)^T \bar{y}_i}{(\tau s_i - H_i \bar{y}_i)^T \bar{y}_i} \qquad (23)$$

After some algebraic operation, we get

$$\bar{y}_i{}^T H_{i+1}^{New}\,\bar{y}_i = \tau \bar{y}_i{}^T s_i \qquad (24)$$

Then

$$\bar{y}_i^T H_{i+1}^{New}\, \bar{y}_i = t(1 + (1 - \vartheta)\rho_i)^2\, y_i^T s_i$$

Obviously, t, $(1 + (1 - \vartheta)\rho_i)^2$ and $y_i^T s_i$ are positive. Hence, we obtain

$$\bar{y}_i^T H_{i+1}^{New}\, \bar{y}_i = t(1 + (1 - \vartheta)\rho_i)^2\, y_i^T s_i > 0\,.$$

The proof is then complete.

3. NUMERICAL RESULTS

This section is devoted to test the implementation of the new methods. We compare New SR1 method with standard SR1 method. The comparative tests involve well-known nonlinear problems (standard test function) [2] with different dimensions $10 \le n \le 5000$ see the appendix, all programs are written in FORTRAN95 language. Experimental results in Table (1) specifically quote the number of iteration NOI and the number of function NOF. Table (2) shows the rate of improvement the new algorithm and confirms that the new method is superior to standard method with respect to the NOI and NOF and we notate:

n: The variables number.
NOI: Number of iterations
NOF: Number of functions evaluation.

Table 1: Comparison between the performance of the new SR1 update and standard SR1 update.

Number of Problem	n	SR1 − HS		New − HS	
		NOI	NOF	NOI	NOF
1	10	22	54	22	53
	100	23	57	23	56
	500	23	57	23	56
	1000	23	57	23	56
	5000	23	57	23	56
2	10	36	253	32	185
	100	43	331	32	185
	500	60	496	32	185
	1000	66	554	32	185
	5000	72	616	32	185
3	10	14	42	13	37
	100	15	48	13	37
	500	16	49	13	37
	1000	16	50	13	37
	5000	16	180	13	37
4	10	35	90	27	74
	100	55	133	31	84
	500	66	158	31	83
	1000	70	162	32	84
	5000	47	114	50	131
5	10	31	91	30	83
	100	32	93	30	83
	500	32	93	30	83
	1000	34	98	30	83
	5000	37	101	30	83
6	10	34	329	38	146
	100	47	182999	42	175
	500	53	183098	43	178
	1000	53	183099	50	225
	5000	65	189124	50	228
7	10	30	80	32	88
	100	32	95	32	88
	500	33	97	32	88
	1000	33	97	32	88
	5000	33	97	32	88
8	10	6	34	6	34
	100	14	83	14	83
	500	21	119	20	103
	1000	23	123	22	107
	5000	38	176	38	180
9	10	25	51	25	51
	100	44	89	44	89
	500	47	95	47	95
	1000	50	101	49	99
	5000	106	294	105	212
Total		1694	744314	1413	4703

Table 2: The percentages of improving the New SR1 method

Tools	SR1 − HS	NSR1 − HS
NOI	100%	83.4104%
NOF	100%	0.6319%

Clearly there is an improvement of 16.5896% in NOI and 99.3681% in NOF for our new proposed algorithms. In general, the New SR1 method has been improved by 57.97885% as compared to standard SR1 method.

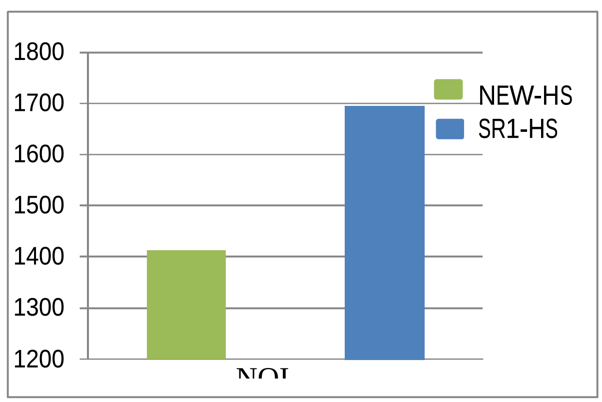

Figure 1: The comparison between SR1 method and the New SR1method according to the total number of iterations (NOI).

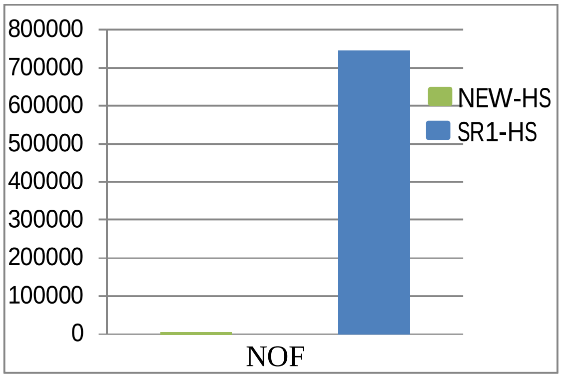

Figure 2: The comparison between SR1 method and the New SR1method according to the total number of functions (NOF).

4. CONCLUSION

In this paper, a modified of quasi-Newton matrices based on SR1 for solving nonlinear unconstrained optimization is presented. The quasi-newton condition and positive definite property of the new method have been proved. Our numerical results indicate that there are improvements of proposed new method techniques over standard SR1 method.

REFERENCES

1. Al-Bayati, **A New PCG Method for Unconstrained Non- Linear Optimization**, Journal of Abhath Al-yarmouk, Pure sci. and eng. Vol. (5), No. (1) Pp. 71-92 (1996).
2. Andrei, N., **An Unconstrained Optimization Test Functions Collection**. Advanced Modeling and Optimization, 10, 147- 161, (2008).
3. Axelsson, O., **On Preconditioning and Convergence acceleration in sparse matrix problems**, CERN Data Handling Division Report, 74-1 (1974).
4. Jorge Nocedal. Stephen J. Wright, **Numerical optimization**. New York : Springer, (1999). https://doi.org/10.1007/b98874
5. L. P. Sun, **Updating of the self-scaling symmetric rank one algorithm with limited memory** for large-scale unconstrained optimization, Computational Optimization and Applications, 27, (2004). https://doi.org/10.1023/B:COAP.0000004977.04103.fb
6. Nielsen, H. Frandsen; P. E. Jonasson; K. and Tingleff, O., **Unconstrained Optimization, Informatics and Mathematical Modeling** , Technical University of Denmark, 3rd Edition, March, (2004).
7. R. Fletcher, **Practical Methods of Optimization**, John Wiley & Sons, New York, (1987).
8. S. Shareef, and H. Jameel, **On Modification of Preconditioning Conjugate Gradient Method with Self-Scaling Quasi-Newton**. International Journal of Advanced Trends in Computer Science and Engineering, 8(5), 2395-2398, (2019). https://doi.org/10.30534/ijatcse/2019/80852019
9. Schropp, J, **A note on minimization problems and multistep methods**. Numer. Math. 78(1), 87-101 (1997). https://doi.org/10.1007/s002110050305
10. Sun W. and Yuan Y., **Optimization Theory Methods Nonlinear Programming**, Springer Science Business Media, LLC. New York, USA, (2006).
11. Y. T. Yang, C. X. Xu and Y. L. Gao, 34, **An optimal self-scaling strategy to the modified symmetric rank one updating**, Journal of Xian Jiaotong University, 100-103, (2005).
12. Y.H. Dai, L.Z. Liao, **New conjugacy conditions and related nonlinear conjugate gradient methods**, Appl. Math. Optim. (43), 87_101 (2001). https://doi.org/10.1007/s002450010019
13. Yuan, G, Wei, Z, Zhao, Q, **A modified Polak-Ribière-Polyak conjugate gradient algorithm for large-scale optimization problems**. IIE Trans. 46(4), 397-413 (2014). https://doi.org/10.1080/0740817X.2012.726757

APPENDIX

Test Functions for Unconstrained Optimization

1. Extended Wood Function:

$$f(z) = \sum_{i=1}^{n/4}\Big(100(z_{4i-3}^2 - z_{4i-2})^2 + (z_{4i-3} - 1)^2 +$$
$$90z4i{-}12{-}z4i2{+}1{-}z4i{-}12{+}10.1z4i{-}2{-}12{+}z4i{-}12$$
$$+19.8z4i{-}2{-}1z4i{-}1$$
$$z_0 = (-3, -1, \dots, -3, -1)^T.$$

2. Generalized Central Function:

$$f(x) = \sum_{i=1}^{n/4}(exp(z_{4i-3} + z_{4i-2})^4 + 100((z_{4i-2} - z_{4i-1})^6 +$$
$$arctan(z4i{-}1{-}z4i4{+}z4i{-}3, \;\; z0{=}(1,2,2,2,\dots,1,2,2,2)T.$$

3. Generalized Cubic Function:

$$f(z) = \sum_{i=1}^{n/2}(100(z_{2i} - z_{2i-1}^3)^2 + (1 - z_{2i})^2)\,,$$

$$z_0 = (-1.2,1, \dots, -1.2,1)^T.$$

4. Generalized Non-Diagonal Function:

$$f(z) =$$
$$\sum_{i=2}^{n}(100(z_1 - z_i^2)^2 + (1 - z_i)^2), \quad z_0(-1, \dots, -1)^T.$$

5. Generalized Rosen Brock Banana Function:

$$f(z) = \sum_{i=1}^{n/2}(100(z_{2i} - z_{2i-1}^2)^2 + (1 - z_{2i-1})^2),$$

$$z_0 = (-1.2,1, \dots, -1.2,1)^T.$$

6. Mile Function:

$$f(z) = \sum_{i=1}^{n/4}((e^{z_{4i-3}} + 10z_{4i-2})^2 + 100(z_{4i-2} + z_{4i-1})^6 +$$

$\tan z4i-1-z4i4+z4i-38+z4i-12, \; z0=(1,2,2, \dots,1,2,2)T.$

7. Powell Function:

$$f(z) = \sum_{i=1}^{n/4}((z_{4i-3} - 10z_{4i-2})^2 + 5(z_{4i-1} - x_{4i})^2 +$$

$z4i-2-2z4i-14+10z4i-3-z4i4,$

$$z_0 = (3,-1,0,1, \dots, 3,-1,0,1)^T.$$

8. Sum of Quadrics (SUM) Function:

$$f(z) = \sum_{i=1}^{n}(z_i - i)^4, \quad z_0 = (1,1, \dots, 1)^T.$$

9. Wolfe Function:

$$f(z) = \left(-z_1\left(3 - \frac{z_1}{2}\right) + 2z_2 - 1\right)^2 + \sum_{i=1}^{n-1}\left(z_{i-1} - z_i\left(3 - \frac{z_i}{2} + 2z_{i+1} - 1\right)\right)^2 + \left(z_{n-1} - z_n\left(3 - \frac{z_n}{2}\right) - 1\right)^2,$$

$$z_0 = (-1, \dots, -1)^T.$$