A new self-scaling variable metric (DFP) method for unconstrained optimization problems

Salah Gazi Shareef a, Alaa Luqman Ibrahim b,*, Zinah Talal Yaseen c

a,b Department of Mathematics, Faculty of Science, University of Zakho, Zakho, Kurdistan Region, Iraq.
c Department of Mathematics, College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq.
E-mails: a salah.shareef@uoz.edu.krd, b alaa.ibrahim@uoz.edu.krd, c Zena-talal@uomosul.edu.iq

Abstract: In this study, a new self-scaling variable metric (VM)-updating method for solving nonlinear unconstrained optimization problems is presented. The general strategy of (New VM-updating) is to propose a new quasi-Newton condition used for update the usual DFP Hessian to a number of times in a way to be specified in some iteration with PCG method to improve the performance of the Hessian approximation. We show that it produces a positive definite matrix. Experimental results indicate that the new suggested method was more efficient than the standard DFP method, with respect to the number of functions evaluations (NOF) and number of iterations (NOI).

Keywords: Unconstrained optimization; self-scaling; Variable metric; Hessian approximation; DFP update.

2010 MSC No: 49J15, 74Pxx, 78M50.

1. Introduction

In 1970 Broyden [1] introduce the Quasi-Newton family of variable metric formula that is the most efficient technique for minimizing a non-linear function $h(x)$.

$$\min h(x) \quad \forall \ x \in R^n$$

(1)

Often needed to update the iterate matrix. Traditionally, satisfies the following quasi-Newton equation:

$$B_{j+1} s_j = y_j$$

(2)

If $H_j$ is to be viewed as an approximation to $B_j^{-1}$, it is natural to require that:

$$H_{j+1} y_j = s_j$$

(3)

Where $s_j = x_{j+1} - x_j = a_j d_j$ and $y_j = g_{j+1} - g_j$ [1,2,10]. The search direction is computed by

$$d_j = -B_j^{-1} g_j, \quad j = 0,1,2,...$$

(4)

where $g_j$ is the gradient of $h$ evaluated at the current iteration $x_j$. One then computes the next iteration by $x_{j+1}$

$$x_{j+1} = x_j + a_j d_j, \quad j = 0,1,2,...$$

(5)

where $a_j > 0$ is the step length satisfies the Wolfe’s conditions [8]:

*Corresponding author
Email address: alaa.ibrahim@uoz.edu.krd (Alaa Luqman Ibrahim)
doi: 10.31559/glm2020.9.1.6
Received: 21 Dec 2019; Accepted 18 Jun 2020
\[ h(x_j + \alpha_j d_j) - h(x_j) \leq \rho \alpha_j g_j^T d_j \] (6)
\[ g(x_j + \alpha_j d_j)^T d_j \geq \sigma g_j^T d_j \] (7)

Where \( \rho \in (0, \frac{1}{2}) \) and \( \sigma \in (0, 1) \).

Now, after determined the point \( x_{j+1} \) we obtain the improved inverse Hessian matrix \( H_{j+1} \) by merge the information generated in the last iteration. The matrix \( H_{j+1} \) is given by for the parameter

\[ H_{j+1} = H_j + \frac{s_j g_j}{s_j^T g_j} + \frac{R_j}{s_j^T g_j} \] (8)

where \( R_j = \frac{s_j g_j}{s_j^T g_j} + \frac{H_j}{\gamma_j H_j} - H_0 = I \) and \( \theta \in [0, 1] \)

Different values of the scalar \( \theta \) corresponding to different member of Broyden's Quasi-Newton -family, noted that if \( \theta = 0 \) then equation (8) corresponds to the standard DFP algorithm introduced by Davidon [13] and Fletcher and Powell [9]. In studying the theoretical behavior of this technique it was shown by Fletcher and Powell that, on quadratic function with the exact line search, the standard DFP formula generates conjugate directions and hence minimizes a quadratic function in at most the iterations. Many modulations have been applied on QN-methods in bid to improve its performance. In 1974 Oren [11] develops the self-scaling VM-algorithms, Oren's formula can be written as:

\[ H_{j+1} = [H_j - \frac{H_j g_j^T g_j}{g_j^T H_j g_j} + \theta R_j H_j] y_j + \frac{s_j g_j}{s_j^T g_j} \] (9)

Where \( y_j = \frac{s_j g_j}{s_j^T g_j} \) and \( \theta = 1 \)

Clearly when \( y_j = 1 \), formula (9) reduces to Broyden's class update defined in (7). Also, to improve the performance of VM updates Biggs [6] proposed to choose \( H_{j+1} \) to satisfy the following modified equation \( H_{j+1} y_j = \epsilon_j s_j \), where \( \epsilon_j > 0 \) is a scaling parameter. The modified BFGS may be written as:

\[ H_{j+1} = H_j + \frac{H_j g_j s_j}{s_j^T g_j} + \frac{\tau_j}{s_j^T g_j} + \frac{\gamma_j g_j}{s_j^T g_j} \] (10)

where \( \tau_j = \frac{1}{\epsilon_j} - \frac{6}{s_j^T g_j} (h(x_j) - h(x_{j+1}) + s_j^T g_{j+1}) - 2 \)

also S. Shareef, and A. Ibrahim [12] made a modification for self-scaling symmetric rank one update with QN condition \( H_{j+1} y_j = \tau s_j \) as follow

\[ H_{j+1} = H_j + \frac{(\tau s_j - h_j \hat{y}_j)(\tau s_j - h_j \hat{y}_j)^T}{(s_j - \hat{y}_j)^T \hat{y}_j} \] (11)

where \( \tau = \frac{1}{t(1 + (1 - \theta) \rho_j)}, t \geq 0, \theta \in (0, 1) \) and \( \rho_j = \frac{s_j^T y_j}{||s_j||^2} \).

We end this general introduction by content of this paper which is organized as follows: In section two, we present the new method. In section three, Numerical results, percentages and discussion are reported. In the final section, we present a conclusion. Throughout this paper, \( ||.|| \) denotes the Euclidean norm of a vector or matrix.

2. Derivation of New Self-scaling VM methods

In this section a new formula for a self-scaling VM-method with preconditioned conjugate gradient (PCG) method is presented. Further, Zhang et al. [4] and Zhang and Xu [5] expanded this condition and derived a class of modified secant condition with a vector parameter, in the form

\[ H_{j+1} y_j = s_j \hat{y}_j = y_j + \frac{\theta_j}{s_j^T u} u \] (12)

where \( u \) is any vector satisfying \( s_j^T u > 0 \), and \( \theta_j \) is defined by:

\[ \theta_j = 6 (f_j - f_{j+1}) + 3 (g_j + g_{j+1}) s_j \]

For the new method we have investigated a new expression for the QN-condition as follows:

\[ H_{j+1} y_j = s_j \hat{y}_j = y_j + \mu \frac{\theta_j}{s_j^T \gamma_j \gamma_j} \] (13)
Where $\mu \in (0,1)$ and $\theta_j = \alpha^{BB} = \frac{s_j^T \bar{y}_j}{s_j^T y_j}$ for more details see [3]

Then, the new self-scaling VM method becomes as follows

$$H_{j+1} = H_j - \frac{H_j y_j^T H_k}{y_j^T H_j y_j + \frac{\mu}{s_j^T y_j}} \frac{s_j^T y_j}{s_j^T y_j}$$

(14)

2.1 The Outlines of the New Self-Scaling VM Method with PCG Method

Step (1): Set $j = 0$, select $x_0$ and a real symmetric positive definite $H_0 = I$, $\epsilon = 10^{-5}$.
Step (2): If $g_j = 0$, stop; else $d_j = -H_j g_j$, where $g_i = \nabla h(x_j)$.
Step (3): Find $a_j = \text{arg} \min h(x_{j+1} + s_j)$.
Step (4): Set $x_{j+1} = x_j + s_j$ and $y_j = g_j + g_j$.
Step (5): Compute $g_{j+1}$ if $\|g_{j+1}\| < \epsilon$, then stop.
Step (6): Calculate $H_{j+1}$ from equation (14).
Step (7): Evaluate $d_{j+1} = -H_{j+1} g_{j+1} + \frac{s_j^T H_{j+1} \bar{y}_{j+1}}{s_j^T y_j} d_j$.
Step (8): If $|g_j^T g_{j+1}| \geq 0.2 \|g_{j+1}\|^2$ go to step (2) else continue.

Set $j = j + 1$ and repeat from Step (3).

Theorem 1: If the new self-scaling VM method is applied to the quadratic with Hessian $B = B^T$, then $H_{j+1} \bar{y}_j = s_j$ if $j \geq 0$.

Proof: By multiplying equation (14) by $\bar{y}_j$ from the right, we obtained:

$$H_{j+1} \bar{y}_j = H_j \bar{y}_j - \frac{H_j y_j \bar{y}_j^T H_k}{y_j^T H_j y_j} \bar{y}_j + \frac{s_j^T \bar{y}_j}{s_j^T y_j} \bar{y}_j$$

(15)

Since $\bar{y}_j^T H_j \bar{y}_j$ and also $s_j^T \bar{y}_j$ are scalars. Then

$$H_{j+1} \bar{y}_j = H_j \bar{y}_j - H_j \bar{y}_j + s_j$$

(16)

Then, $H_{j+1} \bar{y}_j = s_j$

The proof is complete. ■

Theorem 2: In the new self-scaling VM method, if $H_j$ is positive definite, then so is the matrix $H_{j+1}$.

Proof: we can write the equation (14) in quadratic form:

$$X^T H_{j+1} X = X^T H_j X - \frac{X^T H_j \bar{y}_j^T H_k}{\bar{y}_j^T H_j \bar{y}_j} X^T \left( \frac{\mu \bar{y}_j^T}{s_j^T y_j} \right) X,$$

(17)

$$X^T H_{j+1} X = X^T H_j X - \frac{X^T \bar{y}_j^T H_k}{\bar{y}_j^T H_j \bar{y}_j} + \left( \frac{X^T \bar{y}_j^T}{s_j^T y_j} \right)^2.$$

(18)

Now, define $a = H_j^2 \bar{y}_j$, $b = H_j^2 \bar{y}_j$ where $H_j = H_j^2 \bar{y}_j$.
And by using the definitions of $a$ and $b$, we obtain

$$X^T H_j X = X^T H_j \bar{y}_j = a^2 a,$$

$$X^T H_j \bar{y}_j = X^T H_j^2 \bar{y}_j = a^2 b$$

and

$$\bar{y}_j^T H_j \bar{y}_j = \bar{y}_j^T H_j^2 \bar{y}_j = b^2 b.$$

Hence, $X^T H_{j+1} X = a^2 a - \frac{(a^2 b)^2}{b^2 b} + \left( \frac{X^T \bar{y}_j^T}{s_j^T y_j} \right)^2$,

$$X^T H_{j+1} X = \frac{\|a\|^2 \|b\|^2 - (a^2 b)^2}{\|b\|^2} + \left( \frac{X^T \bar{y}_j^T}{s_j^T y_j} \right)^2.$$

(19)

We know that $s_j^T \bar{y}_j$ and $s_j^T s_j$ are positive and $\mu \in (0,1)$, so we have
Therefore, to show that $X^TH_j^+X > 0$, for $X \neq 0$, we need only to demonstrate that these terms do not both vanish simultaneously. The first term vanishes only if $a$ and $b$ are proportional, that is, if $a = \beta b$ for some scalar $\beta$. To complete the proof it is enough to show that if $a = \beta b$ then

$$\frac{(x^Tv)^2}{v^Ty_j} > 0.$$ First observe that

$$H_j^TX = a = \beta b = \beta H_j^T\tilde{y}_j = H_j^T(\beta \tilde{y}_j).$$

Hence $X = \beta \tilde{y}_j$.

By using the above expression for $X$ and $v_j^T\tilde{y}_j$,

$$\frac{(x^Tv)^2}{v_j^Tv_j} = \frac{v_j^Tv_j}{v_j^Tv_j} = \beta^2(\frac{v_j^Tv_j}{v_j^Tv_j} + \mu \frac{v_j^Tv_j}{v_j^Tv_j} v_j^Tv_j) > 0.$$

Thus for all $X \neq 0$,

$$X^TH_j^+X > 0.$$ Then the proof is completed. ■

3. Numerical Results

This section is devoted to test the implementation of the new VM method. The comparative test involve well-known nonlinear problems (standard test function) [7] given in the Appendix with different dimensions, are tested in the range ($4 \leq n \leq 5000$), all programs are written in FORTRAN95 language. The numerical results in Table (1) illustrate that, the VM method is more efficient than standard DFP method with respect to NOI and NOF. Table (2) confirms that the new VM method is superior to standard DFP VM method. Namely there are an improvement of the new suggested about 25.77801 % in NOI and 33.66278 % in NOF by using the new suggested VM algorithm. Generally, the New VM method was improved by 29.72039 % compared with DFP method.

| Test Function | n  | DFP | NOI | NOF | NOI | NOF |
|---------------|----|-----|-----|-----|-----|-----|
| Powell        | 4  | 35  | 96  | 32  | 87  |
|               | 10 | 36  | 107 | 33  | 96  |
|               | 50 | 37  | 110 | 37  | 119 |
|               | 100| 38  | 112 | 35  | 105 |
|               | 500| 38  | 111 | 35  | 105 |
|               | 1000| 38 | 111 | 35  | 105 |
|               | 5000| 39 | 113 | 36  | 105 |
| Mile          | 4  | 66  | 300 | 32  | 153 |
|               | 10 | 67  | 302 | 32  | 145 |
|               | 50 | 80  | 374 | 36  | 160 |
|               | 100| 86  | 411 | 31  | 127 |
|               | 500| 94  | 454 | 38  | 177 |
|               | 1000| 99 | 496 | 47  | 231 |
|               | 5000| 112| 575 | 104| 533 |
| Central       | 4  | 34  | 237 | 19  | 91  |
|               | 10 | 49  | 423 | 22  | 132 |
|               | 50 | 39  | 295 | 21  | 122 |
|               | 100| 39  | 295 | 34  | 269 |
|               | 500| 49  | 423 | 31  | 221 |
|               | 1000| 54 | 479 | 30  | 216 |
|               | 5000| 71 | 682 | 55  | 546 |
| Rosen         | 4  | 34  | 94  | 33  | 92  |
|               | 10 | 34  | 94  | 34  | 95  |
|               | 50 | 34  | 94  | 34  | 92  |
|               | 100| 35  | 97  | 34  | 92  |
|               | 500| 35  | 97  | 34  | 95  |
|               | 1000| 34 | 95  | 34  | 93  |
|               | 5000| 38| 104 | 36  | 98  |
| Cubic         | 4  | 14  | 40  | 14  | 40  |
|               | 10 | 14  | 40  | 14  | 40  |
|               | 50 | 16  | 47  | 15  | 43  |
|               | 100| 17  | 49  | 16  | 46  |
|               | 500| 16  | 46  | 16  | 46  |
|               | 1000| 17 | 49  | 16  | 46  |
|               | 5000| 16| 46  | 16  | 46  |
| Non-Diagonal  | 4  | 27  | 74  | 27  | 73  |
|               | 10 | 36  | 92  | 35  | 90  |
Table (2): Percentage performance of the (new VM and DFP) methods

| Tools      | DFP     | New-VM  |
|------------|---------|---------|
| NOI        | 100%    | 74.22199% |
| NOF        | 100%    | 66.33722% |

Figure (1): Shows the comparison between New-VM method and the standard algorithm (DFP) according to the total number of iterations (NOI) and the total number of functions (NOF).

4. Conclusion

In this paper, we have offered a VM-type for unconstrained optimization problems based on a modified quasi-Newton condition. We showed that the new method satisfy the modified quasi-newton condition and positive definite property. It is clear that from the numerical results the new modified VM-updating formula has an improvement on the standard DFP method in about 29% in both NOI and NOF.

References

[1] C. G. Broyden, The convergence of a class of double rank minimization algorithms II. The new algorithm, Journal of The Institute of Mathematics And it's Applications, 6 (1970), 221-231, https://doi.org/10.1093/imamat/6.3.222.
[2] C. Storey and HU. Y. F., Preconditioned Low-Order Newton methods, J. of optimization theory and application, 79(2) (1993), 311-331, https://doi.org/10.1007/bf00940583.
[3] J. Barzilai, and J.M. Borwein, Tow point step size gradient methods, IMA J. Numer. Anal., 8(1) (1988), 141-148, https://doi.org/10.1093/imanum/8.1.141.
[4] J. Zhang, N. Y. Deng, and L. H. Chen, New quasi-Newton equation and related methods for unconstrained optimization, Journal of Optimization Theory and Applications, 1(1999), 147-167.
[5] J. Zhang and C. Xu, Properties and numerical performance of quasi-Newton methods with modified quasi-Newton equations, Journal of Computational and Applied Mathematics, 137(2)(2001), 269-278, https://doi.org/10.1016/s0377-0427(00)00713-5.
[6] M. C. Biggs, A note on minimization algorithms which make use of non-quadratic properties of objective function, J. Inst. Maths Applics, 12(3) (1973), 337-338, https://doi.org/10.1093/imi/12.3.337.
[7] N. Andrei, An Unconstrained Optimization Test Functions Collection, Advanced Modeling and Optimization, 10 (2008), 147-161.
[8] P. Wolfe, Convergence conditions for ascent methods, SIAM. Rev. 11(2) (1969), 226-235, https://doi.org/10.1137/1011036.
[9] R. Fletcher, and M. J. D. Powell, A Rapidly convergent descent method for minimization, Computer Journal, 6(2) (1963), 163-168, https://doi.org/10.1093/comjnl/6.2.163.
[10] R. Fletcher, Practical Methods of Optimization, Wiley., Chi Chester, (1987).
[11] S. S. Oren, On the selection of parameters in self-scaling variable metric algorithm, Mathematical Programming, 7(1) (1974), 351-367, https://doi.org/10.1007/bf01585530.

[12] S. Shareef and A. Ibrahim, A New Quasi-Newton (SR1) With PCG Method for Unconstrained Nonlinear Optimization, International Journal of Advanced Trends in Computer Science and Engineering, 8(6) (2019), 3124 – 3128, https://doi.org/10.30534/ijatcse/2019/74862019.

[13] W. C. Davidon, Variable metric methods for minimization, Argon, National Laboratory, Pept-ANL-5990 (Rev), (1959).
Appendix

Standard Test Functions for Nonlinear Unconstrained Optimization Problems

1. Generalized Central Function:
   \[ h(x) = \sum_{j=1}^{n/4} \left( \exp(x_{4j-3} + x_{4j-2})^4 + 100 \left( (x_{4j-2} - x_{4j-1})^6 + \arctan(x_{4j-1} - x_{4j})^4 + x_{4j-3} \right) \right), \]
   \[ x_0 = (1,2,2, \ldots, 1,2,2)^T. \]

2. Generalized Cubic Function:
   \[ h(x) = \sum_{j=1}^{n/2} \left( 100(x_{2j} - x_{2j-1})^2 + (1 - x_{2j})^2 \right), \quad x_0 = (-1.2,1, -1.2,1)^T. \]

3. Generalized Non-Diagonal Function:
   \[ h(x) = \sum_{j=2}^n \left( 100(x_1 - x_j)^2 + (1 - x_j)^2 \right), \quad x_0=(-1, \ldots, -1)^T. \]

4. Generalized Rosen Brock Banana Function:
   \[ h(x) = \sum_{j=1}^{n/2} \left( 100(x_{2j} - x_{2j-1})^2 + (1 - x_{2j-1})^2 \right), \quad x_0 = (-1.2,1, \ldots, -1.2,1)^T. \]

5. Mile Function:
   \[ h(x) = \sum_{j=1}^{n/4} \left( (e^{x_{4j-3}} + 10x_{4j-2})^2 + 100(x_{4j-2} + x_{4j-1})^6 + (\tan(x_{4j-1} - x_{4j}))^4 + (x_{4j-3})^8 + (x_{4j} - 1)^2 \right), \]
   \[ x_0 = (1,2,2, \ldots, 1,2,2)^T. \]

6. Powell Function:
   \[ h(x) = \sum_{j=1}^{n/4} \left( (x_{4j-3} - 10x_{4j-2})^2 + 5(x_{4j-1} - x_{4j})^2 + (x_{4j-2} - 2x_{4j-1})^4 + 10(x_{4j-3} - x_{4j})^4 \right), x_0 \]
   \[ = (3, -1,0,1, \ldots, 3, -1,0,1)^T. \]