International Journal of Mathematical Archive-4(5), 2013, 184-191
Available online through www.ijma.info ISSN 2229 – 5046

A NEW SELF-SCALING VARIABLE METRIC ALGORITHM FOR NONLINEAR OPTIMIZATION

Abbas Y. Al Bayati[1*] and Sabah A. Mohamed[2]

[1]Professor of Numerical Optimization, College of Basic Education-Telafer, Mosul University, Iraq
[2]Assistant Lecturer, Department of Computer, College of Education Scientific, Salahaddin University, Iraq

(Received on: 05-04-13; Revised & Accepted on: 05-05-13)

ABSTRACT

In this paper a new self-scaling Variable Metric (VM) method for solving a number of nonlinear unconstrained optimization problems is proposed. In this work we have suggested a new formula for the VM-update with a new Quasi-Newton (QN) like condition used for designing this update. Numerical experiments indicate that this new self-scaling method is effective and superior to both BFGS and Biggs VM-methods, with respect to the number of function evaluations (NOF) and the number of iterations (NOI).

Keywords: *Self-Scaling, Variable Metric, Quasi-Newton like Condition, Exact Line Searches.*

1. INTRODUCTION

The QN family of the VM formula which was introduced by (Broyden, 1970) is the most efficient technique for minimizing non–linear functions $f(x)$ by generating a sequence of points x_k and matrices H_k as follows:

$$d_k = -H_k g_k, \; k = 0,1,2,... \tag{1}$$

where g_k is the gradient of f at x_k.

$$x_{k+1} = x_k + \alpha_k d_k. \tag{2}$$

where the step-size α_k satisfies Wolfe's conditions:

$$f(x_k + \alpha_k d_k) \le f(x_k) + \delta_1 \alpha_k d_k^T g_k \tag{3a}$$

$$g(x_k + \alpha_k d_k)^T d_k \ge \delta_2 d_k^T g_k \tag{3b}$$

$$0 < \delta_1 < \frac{1}{2} \text{ and } \delta_1 < \delta_2 < 1.$$

Or at least it satisfies another equivalent condition:

$$f(x_{k+1}) < f(x_k) - \delta_3 \alpha_k d_k^T g_k \tag{3c}$$

for some predetermined δ_3, or more likely in theoretical analysis it is chosen to

$$\min \; f(x_k + \alpha_k d_k) \tag{4}$$

so that

$$g_{k+1}^T d_k = 0, \tag{5}$$

Corresponding author: *Abbas Y. Al Bayati[1*]*
[1]Professor of Numerical Optimization, College of Basic Education-Telafer, Mosul University, Iraq

this is an exact line search (ELS). However, based on a new nonlinear conjugate gradient method investigated by (Hagar and Zhang, 2005) and under some mild conditions, (Chen, 2012) proved it's global convergence property with some sort of Wolfe type line search procedure. Now, having determined the point x_{k+1} an improved inverse Hessian matrix H_{k+1} is obtained by incorporating the information generated in the last iteration. The matrix H_{k+1} is given by for the parameter $\phi \in [0,1]$.

$$H_{k+1} = H_k + \frac{v_k v_k^T}{v_k^T y_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \phi r_k r_k^T \tag{6a}$$

where

$$y_k = g_{k+1} - g_k \tag{6b}$$

$$v_k = x_{k+1} - x_k \tag{6c}$$

$$r_k = \frac{v_k}{v_k^T y_k} - \frac{H_k^T y_k}{y_k^T H_k y_k} \tag{6d}$$

$$H_0 = I \tag{6e}$$

Different values of the scalar ϕ in (6) correspond to different member of QN-family. It will be noted that $\phi = 0$ corresponds to the original DFP algorithm introduced by (Davidon, 1959) and (Fletcher and Powell, 1963). In studying the theoretical behavior of this technique it was shown by Fletcher and Powell that, on quadratic function with the accurate line search defined in (5), the original ($\phi = 0$) formula generates conjugate directions and hence minimizes a quadratic function in at most k iterations.

The theory of the VM-methods is beautiful and we have a fairly good understanding of their properties. One of the best known VM-method is the BFGS method which was proposed independently by (Broyden, 1970); (Fletcher, 1970); (Goldfarb, 1970) and (Shanno, 1970). It is a line search method. At the k^{th} iteration, a symmetric and positive H_k is given and a search direction is computed by (1) and (2). It has been found that it is best to implement BFGS formula defined by formulae (6) where $\phi = 1$. Broyden had shown that if the search along all k conjugate directions is necessary with analysis based on the error matrix $R_k = G^{-\frac{1}{2}} H G^{-\frac{1}{2}}$ and determined a value of ϕ then the sequence H_k converges steadily to G^{-1}.

1.1. Self-Scaling VM-Updates.

Many modifications have been applied on QN-methods in attempt to improve its efficiency. In this section, the discussion will be on the self-scaling VM-algorithms developed by Oren (1974). Oren's formula can be written for the parameter ϕ:

$$H_{k+1} = [H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \phi r_k r_k^T]\eta_k + \frac{v_k v_k^T}{v_k^T y_k} \tag{7a}$$

where

$$\phi = 1 \text{ and } \eta_k = \frac{y_k^T v_k}{y_k^T H_k y_k} \tag{7b}$$

The formula (7) is known as self-scaling VM-method. Clearly when $\eta_k = 1$, formula (7) reduces to Broyden's class update defined in (6). Oren's update (7) processes the following properties for a quadratic function:

(a) If α_k minimizes $f(x_k - \alpha H_k g_k)$ for all k, then the vector d_k are mutually conjugate (with respect to G) and hence the solution is obtained in at most n iterations.

(b) The condition number of the matrix $R_k = G^{\frac{1}{2}} H_k G^{\frac{1}{2}}$ is strictly monotonically decreasing.

(c) If $\alpha_k = 1$ for all k, then the algorithm convergent" two- step super linearly", i.e.

$$\lim_{k\to\infty} \frac{\|x_{k+1} - x_{\min}\|}{\|x_{k-} x_{\min}\|} = 0 .$$

The proofs of these properties can be found in Oren (1974). Another self-scaling VM-update had been investigated by (Biggs, 1973):

$$H_{k+1} = H_k - \frac{H_k y_k y_k^{\mathrm{T}} H_k}{y_k^{\mathrm{T}} H_k y_k} + \phi r_k r_k^{\mathrm{T}} + t_k (\frac{v_k v_k^{\mathrm{T}}}{v_k^{\mathrm{T}} y_k}) \tag{8a}$$

$$\phi = 1 \text{ and } t_k = (v_k^T y_k)/(4v_k^T g_{k+1} + 2v_k^T g_k - 6(f(x_k) - f(x_{k+1})) \tag{8b}$$

Al-Bayati found another interesting family of VM-updates by further scaling of Biggs and Oren families of VM-updates with a scalar:

$$\phi = 1; \ \sigma_k = \frac{1}{\eta_k} \text{ and } t_k = 1 \quad \text{(Al-Bayati, 1991)} \tag{9a}$$

So that the updating formula becomes:

$$H_{k+1} = H_k + \frac{1}{v_k^T y_k}\left[\left(\sigma_k + \frac{y_k^T H_k y_k}{v_k^T y_k}\right) v_k v_k^T - v_k y_k^T H_k - H_k y_k v_k^T\right] \tag{9b}$$

1.2. Some other Scaling Factors

(Luksan and Vlcek, 1995) in their contribution, proposed an extremely simple scaling strategy which considerably decreases number of function evaluations if VM-methods from the (Broyden, 1970) class are used for unconstrained minimization of functions with number of variables not extremely small. They deal with the BFGS method, since scaling of other VM-methods has very similar properties. After describing their scaling strategy, they compare six scaling techniques, using an extensive collection of test problems, and they presented some useful conclusions.

Recently, (Bin Embong, et al, 2011) proposed a new self-scaling VM-method in which the smallest eigenvalue of the Hessian approximation λ_0 was proposed as an alternative scaling factor of initial scaling on H_1. An improvement over unscaled BFGS is achieved, as for most of the cases, of their trails, the number of iterations are reduced. They suggested further investigation by using an alternative scaling factor, λ_0 on the other types of QN-methods. The relationship between the smallest eigenvalue of Hessian approximation and the optimal step-size was also of the interest in their future research, triggering the possibility of using eigenvalue as a new step-size in the QN-methods. Replacing step-size, α with the smallest eigenvalue of H_1, yields:

$$H_1 = \{H_0 - \frac{H_0 y_k y_k^T H_0}{y_k^T H_0 y_k} + \phi r_k r_k^T\}\lambda_0 + \frac{v_k v_k^T}{v_k^T y_k} \tag{10}$$

where r_k is a vector defined in (6d). As proposed by Shanno and Phua (1978) this update is also an initial scaling on the Hessian approximation. After the initial iteration, the Hessian approximation is never rescaled.

2. A NEW SELF-SCALING VM-METHOD

In this section a new formula for a self scaling VM-method is presented. If H_{k+1} is to be viewed as an approximation to G^{-1}, it is natural to require that:

$$H_{k+1} y_k = v_k \tag{11}$$

which is called the exact QN-condition. For the new method we have investigated a new expression for the QN-condition as follows : let for a weak QN-like condition:

$$H_k y_k = \xi_k v_k \tag{12a}$$

and let for the actual QN- like condition:

$$H_{k+1}y_k = \gamma_k v_k \tag{12b}$$

where

$$\xi_k, \gamma_k > 0 \tag{12c}$$

Then we get the following relationship from (11) and (12):

$$H_{k+1} = \frac{\gamma_k}{\xi_k} H_k \tag{13}$$

This implies that:

$$\frac{H_{k+1}}{H_k} = \frac{\gamma_k}{\xi_k}. \tag{14}$$

These implies that the condition number

$$R_k(H) = \gamma_k / \xi_k, \tag{15}$$

Therefore, to compute the new formula, γ_k, it may be chosen so that, from (12):

$$H_{k+1}y_k = v_k(\gamma_k + \xi_k) - H_k y_k \tag{16}$$

Now multiplying (16) by y_k^{-1} and multiplying and dividing the first and second terms of right hand side of equation (16) by v_k and y_k respectively, yields:

$$H_{k+1} = \frac{(\gamma_k + \xi_k)v_k v_k^T}{y_k^T v_k} - \frac{H_k y_k y_k^T}{y_k^T y_k} \tag{17}$$

Equation (17) can be further simplified using equation (11) as follows:

$$H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \gamma_k \frac{v_k v_k^T}{y_k^T v_k} \tag{18}$$

Equation (18) is a scaled DFP method. Now taking $\phi = 0$ in (7) yields:

$$H_{k+1} = \{H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k}\}\eta_k + \frac{v_k v_k^T}{y_k^T v_k} \tag{19}$$

Multiplying (19) by (1/ η_k) and comparing with (18) we get:

$$\gamma_k = \frac{y_k^T H_k y_k}{v_k^T y_k} \tag{20}$$

which is the new parameter for the self-scaling DFP-algorithm. Certainly, it satisfies the new QN-like condition.

2.1. Convergence Property of the New Proposed Method.

Theorem (2.1): assume that $f(x)$ be a quadratic function defined by:

$$f(x) = \frac{1}{2}x^T Gx + b^T x \tag{21}$$

and the line searches are exact. If H_k is any symmetric positive definite matrix and H_{k+1}^{new} is a new VM-formula defined by:

$$H_{k+1}^{new} = H_k + \frac{1}{v_k^T y_k}\left[\gamma_k(v_k v_k^T) - (v_k y_k^T H_k) - (H_k y_k v_k^T)\right] \tag{22}$$

where γ_k is a positive scalar defined by (20); denoting the new values by (*), then the search directions $d^*_{new} = -H^* g^*$ are identical to the (Hestenes and Stiefel, 1952) Conjugate Gradient (CG) direction d^*_{CG} defined by:

$$d^*_{CG} = -g^* + (y_k^T g^* / y_k^T d_k) d_k \text{ for } k>1 \tag{23}$$

Proof: the update (22) can be written as:

$$H^*_{new} = H_k - v_k y_k^T H_k / v_k^T y_k - H_k y_k v_k^T / v_k^T y_k + \gamma_k (v_k v_k^T / v_k^T y_k)$$

Now using $d^*_{new} = -H^* g^*$, yields

$$d^*_{new} = -H_k g^* + (y_k^T H_k g^* / v_k^T y_k) v_k + (v_k^T g^*)(H_k y_k)/(v_k^T y_k) - \gamma_k (v_k^T g^*)/(y_k^T v_k) v_k$$

Using the property $v_k^T g^* = 0$, for exact line searches:

$$= -H_k g^* + (y_k^T H_k g^* / y_k^T v_k) v_k \tag{24}$$

This is the preconditioned Hestenes and Stiefel CG-method. Hence for a positive definite matrix H_k, the vector g^* can be substituted for $H_k g^*$ by using the following property: $H_{i+1} g^* = H_k g^*$, for all $0 \le i < k \le n$ (see Al-Bayati, 1991).

$$d^*_{new} = -g^* + (y_k^T g^* / y_k^T v_k) v_k \tag{25}$$

We also know that d^*_{DFP} and d^*_{CG} are identical; for the proof of this fact see (Nazareth , 1979). Hence d^*_{new} is identical to d^*_{DFP} with ELS and equation (23) becomes:

$$d^*_{new} = -g^* + (y_k^T g^* / y_k^T d_{CG}) d_{CG} = d^*_{CG}. \tag{26}$$

Hence, the proof is complete.

2.2. New Algorithm.

Step (1): Initialize x_0, $H_0 = I$, $\varepsilon = 10^{-5}$, k=1

Step (2): Set $d_k = -H_k g_k$

Step (3): Set $x_{k+1} = x_k + \alpha_k d_k$; α_k is optimal step-size determined by the cubic Interpolation Uni-model line search.

Step(4): If the stopping criterion is satisfied, stop; otherwise

Step (5): Set $v_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$

Step (6): Compute $H^{new}_{k+1} = H_k + \frac{1}{v_k^T y_k}\left[\gamma_k v_k v_k^T - v_k y_k^T H_k - H_k y_k v_k^T\right]$; γ_k is defined by (20)

Step (7): Set $k = k+1$ and go to Step (2).

3. NUMERICAL RESULTS

Ten well-known test functions (given in the Appendix) with two different dimensions**,** are tested in the range ($100 \le n \le 1000$). The Programs are written in Fortran 90 language and for all cases the stopping criterion is taken to be:

$$\{\|g_{k+1}\| < 10^{-5} \text{ or (either NOI exceeds 500 or NOF exceeds 1000)}\} \tag{24}$$

The line search routine used is the cubic interpolation which uses function and gradient values and it is an adaptation of the routine published by (Bunday, 1984). Our numerical results are given in the Tables (3.1); (3.2) and (3.3) and specifically quoting the number of function calls (NOF) and the number of iterations (NOI). Experimental results in

Table (3.1) confirms that the new algorithm is superior to both standard BFGS and Biggs algorithms. Table (3.2) confirms that the improvement of the new algorithm against the others; namely there are (75-77)% (NOI-NOF) improvements comparing with BFGS. Also, there are (77-81)% (NOI-NOF) improvements comparing with Biggs algorithm. However, Oren and Al-Bayati algorithms are beat the new algorithm in about (4-6)% (NOI-NOF) only. Finally, we have found from (20) best results (indicated in Table (3.1)) that the New algorithm has (40)% NOI best results and (20)% NOF best results. This indicates that self-scaling plays a good rule in the theory of VM-updates.

Table 3.1: Comparisons of Different VM algorithms

Test Functions	N	BFGS	Biggs	Oren	Al-Bayati	New
		NOI (NOF)	NOI (NOF)	NOI (NOF)	NOI (NOF)	NOI (NOF)
POWELL (3,-1,0,1,…)	100	42(117)	85(225)	35(121)	45(93)	55(120)
	1000	44(113)	108(283)	37(122)	51(106)	60(153)
WOOD (-3,-1,-3,-1,.)	100	347(1001)	336(898)	20(68)	21(46)	28(61)
	1000	259(1003)	247(1003)	20(68)	21(46)	30(65)
ROSEN (-1.2,...…....)	100	236(679)	295(752)	26(101)	23(57)	30(87)
	1000	448(1002)	316(801)	26(101)	23(57)	35(98)
NON-D. (-1,……..…)	100	107(268)	48(116)	23(90)	24(54)	31(101)
	1000	172(483)	113(283)	23(100)	24(54)	38(155)
Miele (1,2,2,2,….)	100	31(102)	31(99)	30(101)	32(88)	31(127)
	1000	45(147)	40(124)	44(151)	41(114)	32(129)
Cantrell (1,2,2,2,….)	100	14(69)	26(144)	14(62)	16(105)	12(82)
	1000	14(69)	28(179)	14(62)	16(105)	13(99)
CUBIC (-1.2,1.,….)	100	18(68)	24(192)	25(92)	22(57)	23(65)
	1000	18(68)	23(192)	25(92)	22(57)	23(65)
SUM (1,…………)	100	247(1001)	213(804)	11(57)	11(66)	10(65)
	1000	161(1003)	178(1003)	17(81)	15(91)	16(81)
SHALLOW (-2.,……….)	100	6(17)	6(17)	6 (26)	6 (16)	5 (15)
	1000	6(17)	6(17)	6 (26)	6 (16)	5(15)
WOLFE (-1,………)	100	72(145)	74(149)	43(128)	42(85)	41(83)
	1000	94(189)	96(193)	48(141)	47(95)	46(93)
TOTAL		2297(7561	2294(7474)	493(1790)	507(1316)	564(1759)

4. DISCUSSION.

From Table (3.1), the percentage performance of the new algorithm against other algorithms for 100% NOI and NOF are given in Table (3.2).

Table 3.2: Percentage Performance of the New Algorithm

Tools	**BFGS**	**Biggs**	**Oren**	**Al-Bayati**	**New**
NOI	100%	99%	21%	22%	25%
NOF	100%	98%	24%	17%	23%

Furthermore, counting the best results in Table (3.1), we have found that from (20) best results for both NOI and NOF and for each test problem, we have (2) best results for BFGS; (7) best results for Oren; (3) best results for Al-Bayati and (8) best results for the New algorithm. While for the NOF, we have (4) best results for Oren; (12) best results for Al-Bayati and (4) best results for the New algorithm. Arranging these results in a percentage Table (3.3), we have:

Table 3.3: Percentage Performance (Best Results)

Tools	**BFGS**	**Biggs**	**Oren**	**Al-Bayati**	**New**
NOI	10%	----	35%	15%	40%
NOF	----	----	20%	60%	20%

5. APPENDIX

All the test functions used in this paper are from general literature:

1. **Powell function**:

$$f = \sum_{i=1}^{n/4}[(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4] \ , x_0 = (3,-1,0,1;\ldots)^T$$

2. **Rosen function:**

$$f = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 ,\ x_0 = (-1.2,1.0)^T$$

3. **Cubic function:**

$$f = 100(x_2 - x_1^3)^2 + (1 - x_1)^2 ,\ x_0 = (-1.2,1.0)^T$$

4. **Shallow function:**

$$f = \sum_{i=1}^{n/2}(x_{2i-1}^2 - x_{2i})^2 + (1 - x_{2i-1})^2 \ ,\ x_0 = (-2;\ldots)^T$$

5. **Wolfe function:**

$$f = (-x_1(3 - x_{1/2}) + 2x_2 - 1)^2 \sum_{i=1}^{n-1}(x_{i-1} - x_i(3 - x_{i/2}) + 2x_{i+1} - 1)^2 + (x_{n-1} - x_n(3 - x_{n/2}) - 1)^2$$

$$x_0 = (-1;\ldots)^T$$

6. **Non-diagonal function:**

$$f = \sum_{i=1}^{n}[100(x_1 - x_i^2)^2 + (1 - x_i)^2],\ \ x_0 = (-1;\ldots)^T$$

7. Wood function:

$$f = \sum_{i=1}^{n/4}[100(x_{4i-2} - x_{4i-3}^2)^2 + (1 - x_{4i-3})^2 + 90(x_{4i} - x_{4i-1}^2)^2 + (1 - x_{4i-1})^2 + 10.1(x_{4i-2} - 1)^2$$
$$+ (x_{4i} - 1)^2 + 19.8(x_{4i-2} - 1)(x_{4i} - 1) \quad , \; x_0 = (-3,-1;-3,-1;...)^T$$

8. Miele function:

$$f = (e^{x_1} - 1)^2 \tan^4(x_3 - x_4) + 100(x_2 - x_3)^6 + x_1^{\ 8} + (x_4 - 1)^2 , \; x_0 = (1,2,2,2)^T$$

9. Cantrell function:

$$f = (e^{4i-3} - x_{4i-2})^4 + 100(x_{4i-2} - x_{4i-1})^6 + A\tan((x_{4i-1} - x_{4i})^4 + x_{4i-3}^8 \; ,$$
$$x_0 = (1,2,2,2)^T$$

10. SUM function:

$$f(x) = \sum_{i=1}^{n}(x_i - i)^4 \; , \;\; x_0 = (1.,1.,...,1.,1.)^T \; .$$

6. REFERENCES

1. Al-Bayati, A. Y., (A new family of self-scaling variable metric algorithms for unconstrained optimization), Journal of Education and Science, Mosul University, 12 (1991) 25-54.
2. Biggs, M. C., (A note on minimization algorithms making use of non-quadratic properties of the objective function) Journal of the Institute of Mathematics and Its Applications, 12 (1973) 337-338.
3. Bin Embong, M. F.; Rivaie, M.; Bin Mamat, M. and Bin Mohd, I. (An alternative scaling factor in BFGS method for unconstrained optimization) Australian Journal of Basic and Applied Sciences, 5 (2011) 29-34.
4. Broyden, C. G., (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.
5. Bunday, B., (A Basic Optimization Methods), Edward Arnold, Bedford Square, London, (1984).
6. Chen, Y., (Global Convergence of a New Conjugate Gradient Method with Wolfe Type Line Search) , Journal of Information and Computing Science, 7 (2012), 067-071.
7. Davidon, W. C., (Variable metric methods for minimization), Argon, National Laboratory, Pept-ANL-5990 (Rev)., (1959).
8. Fletcher, R., and M. J. D. Powell, “A Rapidly convergent descent method for minimization”, Computer Journal 6 (1963) 163-168.
9. Fletcher, R., (A new approach to variable metric algorithms) Computer Journal, 13 (1970) 317-322.
10. Goldfarb, D., (A family of Variable metric methods derived by variational means), Mathematics of Computation, 24 (1970) 23-26.
11. Hager, W.W. and Zhang, H. (A new conjugate gradient method with guaranteed descent and an efficient line search), SIAM J. Optim., 16 (2005) 170-192.
12. Hestenes, H. R. and Stiefel, E., (Methods of conjugate gradient for solving linear systems), Journal of Research of the National Bureau of Standards 40 (1952) 409-436.
13. Luksan, L. and Vlcek, J., (Simple Scaling For Variable Metric Updates) Technical Report No. V-611 (1995), Institute of Computer Science, Academy of Sciences of the Czech.
14. Nazareth, L., (A relationship between BFGS and conjugate gradient algorithm and it’s implementations for new algorithm), SIAM Journal Numerical Analysis 16 (1979) 794-800.
15. Oren, S. S., (On the selection of parameters in self-scaling variable metric algorithm), Mathematical Programming 3 (1974) 351-367.
16. Shanno, D. F., (Conditioning of quasi-Newton methods for function minimization), Mathematics of Computation, 24 (1970) 647-657.
17. Shanno, D. F. and K.H. Phua, 1978. (Matrix conditioning and nonlinear optimization), Mathematical Programming, 14 (1978)149-160.

Source of support: Nil, Conflict of interest: None Declared