A new modified Leverrier's VM Update for solving Unconstrained Optimization

Abbas Y. Al-Bayati and Salah G. Shareef
College of Computer Sciences and Mathematics-University of Mosul

Abstract

In this paper, a new VM-updating formula for solving unconstrained non-linear optimization problems is proposed. The new method is derived by using Faddeev's modification in order to modify Leverrier's VM-update.The new algorithm is tested by (8) nonlinear test functions, with different dimensions and compared with the standard BFGS algorithm.

Introduction

Quasi-Newton (QN) methods are subclasses of the more general variable metric methods, but we content ourselves here with the quasi-Newton approach (Davidon,1959) . The basic form of these algorithms is given as:

$$g_k(x) = \nabla f(x) \qquad \ldots(1)$$

Where f is twice continually differentiable function $x \in R^n$

Here the notation g_k is used, as before, for the gradient vector of the objective function at x_k, and we also retain the notations:

$$v_k = x_{k+1} - x_k = \lambda_k d_k \qquad \ldots(2)$$

$$y_k = g_{k+1} - g_k \qquad \ldots(3)$$

Methods vary according to how the metric matrix H_k is updated from iteration to iteration. Once such an updating formula has been defined, according to QN-Condition, the algorithm is completed. The matrices are called metric and they are assumed to be positive definite and defined on R^n. The search direction is defined in the above algorithm as the steepest descent direction relative to this metric. The idea is to update H_k, in such a way that the search directions used for the first few iterations are close to the steepest descent direction until, the minimum value is approached. Thus, the matrix H_k can be viewed as an approximation to the inverse Hessian matrix of the objective function f at its minimum x^*. It is usual to take H_o=I so that the initial search direction is the scaled steepest descent direction.The development of these algorithms started with Davidon (1959), whose Argoune National Laboratory report was improved upon by

Fletcher and Powell (1963) to produce the famous Davidon-Flethcher-Powell (DFP), method. We begin our study for the quasi-Newton update formulas with this DFP algorithm. For the development of this theory, we will again restrict our attention to the positive definite quadratic objective function.

$$f(x)=c+b^T x+\frac{x^T Gx}{2} \qquad \ldots(4)$$

In this paper we are going to develop a much wider applicability algorithm than the DFP method.

We recall that for any step $\mathrm{v}_k=\mathrm{x}_{k+1}-\mathrm{x}_k$

We have $\mathrm{y}_k=G\mathrm{v}_k$ ...(5)

And so, if H_{k+1} is to be viewed as an approximation to G^{-1}, it is natural to require that;

$$\mathrm{H}_{k+1}\mathrm{y}_k=\mathrm{v}_k\,, \qquad \ldots(6)$$

which is called the quasi-Newton condition. Since G is positive definite, it is natural to demand that the H_k's should be also. This requirement can also be viewed as the requirement that the search directions are downhill, since the positive definiteness of H_k implies that:

$$\mathrm{d}_k^T\mathrm{g}_k=-\mathrm{g}_k^T\mathrm{H}_k\mathrm{g}_k<0 \qquad \ldots(7)$$

Provided that $g_k \neq 0$, that is, provided the minimum has not already been found.

The derivation of the particular updating formula is also based on the fact that generally conjugate search directions will guarantee the finite termination property; in ths case the final matrix generated would be H_k. Ideally, we would like this to be G^{-1}.

All the requirements are fulfilled by the DFP updating formula:

$$\mathrm{H}_{k+1}=\mathrm{H}_k-\frac{\mathrm{H}_k\mathrm{y}_k(\mathrm{H}_k\mathrm{y}_k)^T}{\mathrm{y}_k^T\mathrm{H}_k\mathrm{y}_k}+\frac{\mathrm{v}_k\mathrm{v}_k^T}{\mathrm{v}_k^T\mathrm{y}_k} \qquad \ldots(8)$$

The general formula, in a slight modification of Fletcher's (1970) parameterization, is given by

$$\mathrm{H}_{k+1}=\mathrm{H}_k-\frac{\mathrm{H}_k\mathrm{y}_k(\mathrm{H}_k\mathrm{y}_k)^T}{\mathrm{y}_k^T\mathrm{H}_k\mathrm{y}_k}+\frac{\mathrm{v}_k\mathrm{v}_k^T}{\mathrm{v}_k^T\mathrm{y}_k}+\varphi\mathrm{w}_k\mathrm{w}_k^T \qquad \ldots(9)$$

Where $\phi \geq 0$ is the free parameter and

$$\mathrm{w}=\frac{\mathrm{v}_k}{\mathrm{v}_k^T\mathrm{y}_k}-\frac{\mathrm{H}_k\mathrm{y}_k}{\mathrm{y}_k^T\mathrm{H}_k\mathrm{y}_k} \qquad \ldots(10)$$

Much effort, both analytically and computationally, has been devoted to identifying the best quasi-Newton formula or even the best from the much wider class of variable metric methods introduced by Huang (1970). The choice with the widest support is the BFGS algorithm which was derived independently in 1970 in four different ways by Broyden, Fletcher, Goldfarb, and Shanno. In the parameterization of (9), this BFGS update formula corresponds to choosing.

$$\phi = y_k^T H_k y_k \qquad \ldots(11)$$

Faddeev's Modification of Leverrier's Method

Consider that we have the system of equations:

$$Gv= \lambda v \qquad \ldots(12)$$

where the scalar λ is an eigenvalue (or a characteristic value) of matrix G_{nxn} and the vector v is the eigenvector (or a characteristic value) corresponding to λ.

The problem often requires the solution of the homogeneous set of equation:

$$(G-\lambda I)\ v= 0 \qquad ...(13)$$

Where I is the identity matrix.

To determine the values of λ and v which satisfy this set. However, before we proceed with developing method of solution. Let's examine equation (13) from a geometric perspective. The multiplication of a vector by a matrix is a linear transformation of the original vector to a new vector of different direction and length. For example, matrix G transforms the vector $\overline{v}$ to the vector z in the operation, i.e.,

$$G\overline{v} = z \qquad ...(14)$$

In contrast to this, if v is the eigenvector of G, then the multiplication of the eigenvector v by matrix G yields the same vector v multiplied by a scalar λ, that is, the same vector but of different length:

$$Gv= \lambda v \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (15)$$

It can be stated that for a nonsingular matrix G of order n there are n characteristic directions in which the multiplication by G does not change the direction of the vector, but only changes its length. More simple stated, matrix G has n eigenvectors and n eigenvalues.

The homogenous problem (13) possesses nontrivial solutions if the determinate of the matrix (G-λI), called the characteristic matrix of G, vanishes:

$$|G-\lambda I| = \begin{bmatrix} a_{11}-\lambda & a_{12} & a_{1n} \\ a_{21} & a_{22}-\lambda & a_{2n} \\ a_{n1} & a_{2n} & a_{nn}-\lambda \end{bmatrix} \quad ...(16)$$

The determinate can be expanded by minors to yield a polynomial of nth degree:

$$\lambda^n - \alpha_1\lambda^{n-1} - \alpha_2\lambda^{n-2}..........-\alpha_n = 0 \quad ...(17)$$

This polynomial, which is called the characteristic equation of matrix G, has n roots which are the eigenvalues of G. A nonsingular real symmetric matrix of order n has n real distinct eigenvalues and n linearly independent eigenvectors. The eigenvectors of a real symmetric matrix are orthogonal to each other. The coefficients α_i of the characteristic polynomial are functions of the matrix elements a_{ij}, and must be determined before the polynomial can be used.

The well-known Cayley-Hamilton theorem states that a square matrix satisfies its own characteristic equation, i.e.

$$G^n - \alpha_1 G^{n-1} - \alpha_2 G^{n-2}.............\alpha_n I = 0 \quad ...(18)$$

The problem of evaluating the eigenvalues and eigenvector of matrices is a complex multipstip procedur. Several methods have been developed for this purpose some of these apply to symmetric matrices,others to tridiagonal matrices, and a few can be used for general matrices. The method in this section work the original matric G and its characteristic polynomial (17) to evaluate the coefficients α_i of the polynomial(Alkis ,1987). One such method is the Faddeev-Leverrier procedure. Once the coefficients of the polynomial are known, the methods use root-finding techniques, such as the Newton-Raphson or Graeffe's method. To determine the eigenvalues. Finally, the algorithms employ a reduction method, such a Gauss elimination, to calculate the eigenvectors (Fletcher,1970).

The Faddeev-Leverrier method calculates the coefficients α_i to α_n of the characteristic polynomial (17) by generating a series of matrices G_k whose traces are equal to the coefficients of the polynomial.

Let us start in matrix G and first coefficient are:

$$G_1=G, \qquad \alpha_1= \text{tr } G_1, \quad ...(19)$$

and the subsequence matrices are evaluated from the recursive equations:

$$\left.\begin{aligned} G_k &= G(G_{k-1}-\alpha_{k-1}I) \\ \alpha_k &= \frac{1}{k}\text{tr}G_k \end{aligned}\right\} k = 2,3,..........n\,. \quad ...(20)$$

In addition to this, the Faddeev-Leverrier method yields the inverse of the matrix G by: $H = \frac{1}{\alpha_n}\left(G_{n-1} - \alpha_{n-1} I\right)$ ...(21)

For more details, see Alkis (1987)

New VM-Updating Formula for Leverrier Method

In this section we present a new updating formula to minimize the function f(x), $x \in R^n$ where f(x) is assumed to be at least twice continuously differentiable and where the function and the first derivatives can be evaluated at any point x.

Now from equation (22) and (23) :

$$g_{k+1} = g_k + \lambda_k G d_k \qquad ...(22)$$

then $y_k = \lambda_k G d_k$ where ...(23)

$y_k = g_{k+1} - g_k$, and let G_k satisfy the equation

$$G\left(x_{k+1} - x_k\right) = y_k$$

let $v_k = x_{k+1} - x_k$ then we have

$$G_k = \frac{y_k}{v_k} \qquad ...(24)$$

Now multiplying and dividing equation (24) by y_k^T , then eq.(24) becomes:

$$G_k = \frac{y_k y_k^T}{y_k^T v_k} \qquad ...(25)$$

or from (23)

$$G_k = \frac{y_k}{\lambda_k d_k}$$

By using the follows equation (26)

$$x_{k+1} = x_{k+1} + \lambda_k d_k \qquad ...(26)$$

when $v_k = \lambda_k d_k$ then we have

$$G_k = \frac{y_k}{v_k} \qquad ...(27)$$

Also multiplying and dividing equation (26)by y_k^T ,we obtain the equation (25).

Now we let

$S_k = G_k y_k$, and $\overline{S}_k = I.y_k$ then the formula (20) becomes:

$$G_k = \frac{y_k S_k^T}{v_k^T y_k} - \alpha_k \frac{\overline{S}_k \overline{S}_k^T}{v_k^T y_k} \qquad ...(28)$$

Now we compute $\alpha_k = \frac{1}{k} trG_{k+1}$ and finally we find the inverse of the matrix G by:

$$H = \frac{1}{\alpha_k}\left(G_{k-1} - \alpha_{k-1} I\right) \qquad ...(29)$$

Outlines of New proposed Algorithm

Step (1): set x_1, ε, $H_1 = \overline{H}_1 = \overline{\overline{H}}_1 = I$

Step (2): for k=1 to n set $d_1 = - H_1 g_1$

Step (3): set $\alpha_k = \mathrm{tr}\overline{H}$

Step (4): compute $x_{k+1} = x_k + \lambda_k d_k$, where λ_k is obtained from line search procedure.

Step (5): cheek if $\|g_{k+1}\| < \varepsilon$ then stop

Otherwise go to step (6)

Step (6): $y_k = g_{k+1} - g_k$

$v_k = x_{k+1} - x_k$

Step (7): set $S_k = \overline{\overline{H}} y_k$

$\overline{S}_k = I.y_k$

Step (8): compute $\overline{H}_k = \frac{y_k S_k^T}{v_k^T y_k} - \alpha_k \frac{\overline{S}_k \overline{S}_k^T}{v_k^T y_k}$

Step (9): compute $\alpha_k = \frac{1}{k}\mathrm{tr}\overline{H}_k$

Step(10):if k=1 then $\overline{\overline{H}} = \frac{y_k y_k^T}{v_k^T y_k}$

Step (11): set $H_{k+1} = H_k + \frac{1}{\alpha_k}\left(\overline{\overline{H}} - \alpha_{k-1} I\right)$

Step (12): set $\overline{\overline{H}}_k = \overline{H}_k$ and

$\alpha_{k+1} = \alpha_k$

Step (13): set $d_{k+1} = -H_{k+1} g_{k+1} + \beta_k d_k$ where

Where $\beta_k = \frac{g_{k+1} H_{k+1} y_k}{d_k^T y_k}$

If k= n+1 or $d_{k+1}\, g_{k+1} \geq 0$

go to step (1) else

k= k+1

go to step (4)

Numerical Results

In order to investigate the performance of the new proposed algorithm eight test functions were tested with different dimensions ($4 \le n \le 500$) and all programs are written in FORTRAN 90 language; and for all cases the stopping criterion is taken to be $||g_{k+1}|| < 1\times10^{-5}$. The numerical results are given in the table (1) is specifically quoted the number of functions NOF and the number of iterations NOI.

Table (1): Comparative performance of the two algorithms
(Classical BFGS and new Update method)

Test function	BFGS Classical NOI (NOF)	New Method NOI (NOF)
Powell (4)	21 (86)	15 (43)
Powell (20)	38 (123)	25 (75)
Powell (100)	71 (197)	31 (108)
Powell (500)	50 (148)	40 (142)
Wood (4)	37 (110)	25 (57)
Wood (20)	84 (244)	50 (108)
Wood (100)	251 (775)	60 (128)
Wood (500)	283 (791)	60 (128)
Cubic (4)	19 (58)	15 (42)
Cubic (20)	35 (99)	13 (37)
Cubic (100)	70 (167)	13 (37)
Cubic (500)	53 (124)	14 (41)
Rosen (4)	34 (87)	17 (49)
Rosen (20)	34 (87)	19 (58)
Rosen (100)	34 (87)	19 (58)
Rosen (500)	34 (87)	19 (58)
Shallow (4)	8 (26)	8 (24)
Shallow (20)	8 (26)	8 (24)
Shallow (100)	8 (26)	8 (24)
Shallow (500)	8 (26)	8 (24)
Non-diagonal (4)	24 (72)	24 (61)
Non-diagonal (20)	48 (115)	19 (52)
Non-diagonal (100)	74 (177)	22 (60)
Non-diagonal (500)	78 (188)	22 (59)
Gedger (4)	6 (17)	5 (14)
Gedger (20)	6 (17)	5 (14)
Gedger (100)	6 (17)	6 (17)
Gedger (500)	6 (17)	6 (17)
Beal (4)	8 (20)	9 (21)
Beal (20)	10 (25)	9 (21)
Beal (100)	10 (25)	9 (21)
Beal (500)	10 (25)	9 (21)
Total	1466 (4089)	612 (1643)

Tools	BFGS	New update
NOI	100	41.7
NOF	100	40

Conclusion

It is clear that from the above table the new modified VM-updating formula has an improvement on the standard BFGS algorithm in about 60% in both NOI and NOF according to our selected set of numerical problems.

References

- Alkis,C.,(1987): Applied Numerical Methods With Personal Computer, Mc Graw-Hill international Editions.
- Davidon, W.C.,(1959): Variable Metric Methods for Minimization, Argonn, National Laboratory, Rept-ANL-5990 (Rev).
- Fletcher, R., (1987): Practical Methods of Optimization, John Wiley and Son, Chichester.
- Fletcher,R., (1970): A New Approach to Variable Metric Algorithm, Computer Journal Vol.13, pp. 317-322.
- Fletcher,R.and Powell,M.J.D.,(1963): A Rapidly Convergent Descent Method for Minimization, Computer Journal Vol. 6,pp.163-168.
- Huang,H.Y., (1970): Unified Approach to Quadratically Convergent Algorithms for Function Minimization.JOTA,5,PP.405-423.

Appendix

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

1. *Generalized powell function* :

$$f(x)=\sum_{i=1}^{n/4}(x_{4i-3}-10x_{4i-2})^2+5(x_{4i-1}-x_{4i})^2+(x_{4i-1}-2x_{4i})^2+10(x_{4i-9}-x_{4i})^4+(x_{4i-2}-2x_{4i-1}-x_{4i})^2)$$

$Starting\ \ po\mathrm{int}\ :(3,1,0,1,......................)^T$

2. *Generalized Wood function* :

$$f(x)=\sum_{i=1}^{n/4}100(x_{4i-2}+x_{4i-3}^2)^2+(1-x_{4i-3})^2+90(x_{4i}-x_{4i-1}^2)^2+(1-x_{4i-1}^2)^2+1.0$$

$Starting\ \ po\mathrm{int}\ :\ (-3,-1,-3,-1,...............)^T$

3. Generalized Rosen Brock Banana function:

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

$$x_0=[-1.2,1.,...,-1.2,1]$$

4. Generalized Non diagonal function:

$$f(x)=\sum_{i=2}^{n}[100(x_1-x_i^2)^2+(1-x_i)^2 \ ,$$

$$x_0=[-1.,...,-1.].$$

5. Generalized Beale Function:

$$f(x)=\sum_{i=1}^{n/2}\left[1.5-x_{2i}+(1-x_{2i})\right]^2+\left[2.25-x_{2i-1}(1-x_{2i}^2)\right]^2+\left[2.625-x_{2i-1}(1-x_{2i}^2\right]^2 ,$$

$$x_0=[-1.,-1.,...,-1.,-1.] .$$

6. Diagonal 6 Function:

$$f(x)=\sum_{i=1}^{n}(\exp(x_i)-(1+x_i)) ,$$

$$x_0=[1.,1.,...,1.,1.] .$$

List of symbols

Symbol	**Meaning**
n	is the dimensions of the problems
K	is the K-th step of iterations
F	is the twice differentiable real value function
g	is the n ×1 gradient vector of f(x)
d	is the n ×1 search direction vector
G	is the n ×n Hessian matrix
H	is the n ×n approximation to G $^{-1}$ matrix
v	is the n ×1 difference vector between two successive points
λ	is the positive scalar which minimizes f(x- λHg)
ELS	is the exact line search
ILS	is the inexact line search
QN	is the Quasi-Newton
VM	is the Variable metric
CG	is the Conjugate Gradient
NOF	is the number of function evaluations
NOI	is the number of iterations
c	is a scalar
b	is the constant vector having n component
G	is nxn symmetric and constant

Journal of Kirkuk University –Scientific Studies , vol.3, No.2,2008

تطوير جديد لتحديث Leverrier للمتري المتغير لحل مسائل الامثلية غير المقيدة

عباس يونس البياتي و صلاح غازي شريف

كلية علوم الحاسبات والرياضيات ــ جامعة الموصل

الخلاصة

في هذا البحث تم اقتراح تحديث جديد للمتري المتغير لحل مسائل غير خطية في الامثلية غير المقيدة . لقد تم اشتقاق الطريقة الجديدة باستخدام تحسين Faddeev من اجل تطوير طريقة Leverrier . اختبرت هذه الخوارزمية الجديدة باستخدام ٨ دوال غير خطية وبأبعاد مختلفة وقورنت مع خوارزمية الــBFGS القياسية.