New Variable Metric Algorithm by The Mean of 2\textsuperscript{nd} Order Quasi-Newton Condition

Abbas Y. Al Bayati
profabbasalbayati@yahoo.com
College of Computer Sciences and Mathematics
University of Mosul, Iraq

Runak M. Abdullah
College of Science
University of Suleimani

Received on: 17/11/2006
Accepted on: 23/01/2007

ABSTRACT

In this paper a new class of Quasi-Newton update for solving unconstrained nonlinear optimization problem is proposed. In this work we suggested a new formula for the variable metric update with a new quasi-Newton condition used for the symmetric rank two formula.

Finally, a numerical study is reported in which the performance of this new algorithm is compared to that of various members of the unmodified family. Numerical experiments indicate that this new algorithm is effective and superior to the standard BFGS and DFP algorithms, with respect to the number of functions evaluations (NOF) and number of iterations (NOI).

Keywords: Unconstrained Optimization, Quasi-Newton Condition, Inexact Line Search.

1. Introduction

The Quasi-Newton family of variable metric formula introduced by Broyden [2]. It is the most efficient technique for minimizing a non–linear function \( f(x) \). Generating a sequence of points \( x_k \) and matrices \( H_k \) by the following procedure we have:

\[ d_k = -H_k g_k, \quad k = 0, 1, 2, \ldots \]  
\[ x_{k+1} = x_k + \alpha_k d_k \] 

where \( g_k \) is the gradient of \( f \) at \( x_k \).

The scalar \( \alpha_k \) is chosen to ensure, at least that

\[ f(x_{k+1}) < f(x_k) - \gamma \alpha_k d_k^T g_k \]
For some predetermined $\gamma$, or more likely in theoretical analysis it is chosen to minimize
\[
\min \quad f(x_k + \alpha_k d_k)
\]
so that
\[
g_{k+1}^T d_k = 0 ,
\]
this is an exact line search (ELS).

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 new matrix $H_{k+1}$ is given by
\[
H_{k+1} = H_k + \frac{v_k v_k^T}{y_k^T y_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \phi R_k^T R_k
\]
where
\[
y_k = g_{k+1} - g_k
\]
\[
v_k = x_{k+1} - x_k
\]
\[
R_k = \frac{v_k}{y_k^T y_k} - \frac{H_k y_k}{y_k^T H_k y_k}
\]
\[
H_{(0)} = I
\]
$\phi$ is parameter $\in (0,1)$.

Different values of the scalar $\phi$ in equation (6) correspond to different members of Broyden's Quasi-Newton family. It will be noted that $\phi = 0$ corresponds to the original variable metric algorithm introduced by Davidon [4] and Fletcher and Powell (DFP) [5]. In studying the theoretical behavior of these techniques 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. It has frequently been postulated that fact accounts for the region near the minimum has been found.

One of the best known VM method is the BFGS method that was proposed independently by Broyden, Fletcher, Goldfarb and Shanno. The BFGS update is defined by formulae (6) where $\phi = 1$.

Broyden (1970) [2] has shown that if search along all $k$ conjugate directions is necessary then, with analysis based on the error matrix
\[
K = G^{-\frac{1}{2}} H G^{-\frac{1}{2}}
\]
and determined a value of $\phi$ for which the sequence $H_k$ converges steadily to $G^{-1}$.

2. New Algorithm

In this section a new formula of preconditioned conjugate gradient (PCG) method is presented with inexact line search.

If $H_{k+1}$ is to be viewed as an approximation to $G^{-1}$, it is natural to require that
\[
H_{k+1} y_k = v_k
\]
Which is called the Quasi-Newton condition.

For the new algorithm we derive a new expression for the QN condition as follows [2]:

Let
New Variable Metric Algorithm by The Mean of 2nd Order Quasi-Newton Condition

\[ H_k y_k = \alpha_k v_k \] \hspace{1cm} \text{...}(8)

Where \( \alpha_k > 0 \), and

\[ H_{k+1} y_k = v_k \] \hspace{1cm} \text{...}(9)

Then we get the following relationship

\[ H_{k+1} = \frac{1}{\alpha_k} H_k \]

and we obtain the following results

\[ \frac{H_{k+1}}{H_k} = \frac{1}{\alpha_k} \] \hspace{1cm} \text{...}(10)

These implies that the condition number

\[ K(H) = \frac{1}{\alpha_k} \] \hspace{1cm} \text{...}(11)

therefore, to compute a new formula, \( \alpha_k \) will be chosen as follows

\[ \alpha_k = \left( \frac{y_k^T H_k y_k}{v_k^T y_k} \right)^T \] \hspace{1cm} \text{Al-Bayati [1]} \hspace{1cm} \text{...}(12)

or \( \alpha_k \geq 2 \)

then we have

\[ H_{k+1} y_k = v_k (1 + \alpha_k) - H_k y_k \] \hspace{1cm} \text{...}(13)

Now dividing (13) by \( y_k \) and multiplying and dividing the first and the second terms of right hand side of equation(13) by \( v_k \) we get

\[ H_{k+1} = \frac{(1 + \alpha_k) v_k v_k^T}{y_k^T v_k} - \frac{H_k y_k v_k^T}{y_k^T v_k} \] \hspace{1cm} \text{...}(14)

To obtain the correctly approximation of \( G^{-1} \) we can rewrite equation (14) as follows

\[ H_{k+1} = H_k - \frac{H_k y_k v_k^T}{y_k^T v_k} + (1 + \alpha_k) \frac{v_k v_k^T}{y_k^T v_k} \] \hspace{1cm} \text{...}(15)

where \( \alpha_k \) defined by (12).

This is a new update \( H_{k+1} \) which satisfies a new QN-like condition.

3. Outlines of The New Suggested Algorithm

Step(1): set \( x_0, \epsilon, H_0 = I \)

Step(2): For \( k = 1 \) set \( d_1 = -H_1 g_1 \)

Step(3): Set \( x_{k+1} = x_k + \alpha_k d_k \) where \( \alpha_k \) is optimal step- size.

Step(4): If \( \|g_{k+1}\| \leq \epsilon \) stop , otherwise

Step(5): Compute \( v_k = x_{k+1} - x_k, y_k = g_{k+1} - g_k \)

Step(6): Compute \( H_{k+1} \) by equation (15).

Step(7): \( d_{k+1} = -H_{k+1} g_{k+1} + \beta_k d_k \) where \( \beta_k = \frac{y_k^T H_{k+1} g_{k+1}}{d_k^T y_k} \) and \( H_{k+1} \) is defined by (15).

Step(8): If restart criterion is satisfied, i.e. \( g_{k+1}^T d_{k+1} > 0 \) and \( g_k^T g_k > g_{k+1}^T g_{k+1} \) go to step(2). Else \( k = k + 1 \) and go to step(3).
Theorem (3.1):
Let \( f \) be given by
\[
    f(x) = \frac{1}{2} x^T G x + b^T x
\]
where \( G \) is symmetric positive definite. Choose an initial approximation \( H_0 = H \), where \( H \) is any symmetric positive definite matrix of appropriate order. Obtain \( H_{new}^* \) from \( H \) where \( d = -Hg \) is the search direction and assuming exact line searches then
\[
    H_{i+1}^* = Hg^*, \quad \text{for} \quad 0 \leq i < k \leq n, \quad g_{k+1} = g^*
\]

Proof:
Apply induction on \( i \). Let \( H_0 = H \), on the above assumptions so that
\[
    H_0g^* = Hg^*
\]
using formula(16) we have
\[
    H_{i+1}^* = H_k g^* - \frac{H_k y_k v_k^T g^* + (1 + \alpha) v_k v_k^T g^*}{y_k^T v_k}
\]
... (16)
since \( v_k^T g^* = 0 \) (ELS), equation (16) become
\[
    H_{i+1}^* = Hg^*
\]
we prove it is true for \( i+1 \) by using the following two standard properties (not proved here) which are satisfied in the case of a quadratic function and exact line searches).
(a): \( v_j^T g^* = 0, \quad \text{for} \quad j = 1, 2, \ldots, k \)
(b): \( g_j^T H_j g^* = 0, \quad \text{for} \quad j = 1, 2, \ldots, k \)
The orthogonality property satisfies when \( f \) is quadratic function and exact line searches is used. Hence \( H_{i+1} g^* = Hg^* \)

4. Numerical Results:
Seven well-known test functions (given in the appendix) were tested with different dimensions (4 \( \leq k \leq 1000 \)). All programs are written in FORTRAN 90 language and for all cases the stopping criterion is taken to be \( \|g_{k+1}\| < 1*10^{-5} \).

The line search routine used was cubic interpolation which uses function and gradient values and it is an adaptation of the routine published by Bundy [3].

The results are given in the Table (1) and specifically quoting the number of functions (NOF) and the number of iterations (NOI).

Experimental results in table (1A) confirm that the new algorithm is superior to standard DFP and BFGS methods. Namely there are an improvement of the new suggested about % 34.87 NOI and % 23.82 NOF by using the new suggested VM-algorithm in this.
New Variable Metric Algorithm by The Mean of 2nd Order Quasi-Newton Condition

Table (1A). Comparison of DFP, BFGS and new algorithm

<table>
<thead>
<tr>
<th>Test Function</th>
<th>N</th>
<th>DFP</th>
<th>BFGS</th>
<th>New Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NOI</td>
<td>NOF</td>
<td>NOI</td>
<td>NOF</td>
</tr>
<tr>
<td>Powel</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>82</td>
<td>487</td>
<td>38</td>
<td>98</td>
</tr>
<tr>
<td>100</td>
<td>54</td>
<td>284</td>
<td>40</td>
<td>100</td>
</tr>
<tr>
<td>500</td>
<td>56</td>
<td>294</td>
<td>50</td>
<td>128</td>
</tr>
<tr>
<td>1000</td>
<td>40</td>
<td>233</td>
<td>36</td>
<td>101</td>
</tr>
<tr>
<td>Wood</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>243</td>
<td>861</td>
<td>235</td>
<td>670</td>
</tr>
<tr>
<td>100</td>
<td>48</td>
<td>158</td>
<td>81</td>
<td>183</td>
</tr>
<tr>
<td>500</td>
<td>76</td>
<td>333</td>
<td>82</td>
<td>188</td>
</tr>
<tr>
<td>1000</td>
<td>40</td>
<td>233</td>
<td>36</td>
<td>101</td>
</tr>
<tr>
<td>Cubic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>19</td>
<td>68</td>
<td>19</td>
<td>56</td>
</tr>
<tr>
<td>100</td>
<td>27</td>
<td>93</td>
<td>64</td>
<td>150</td>
</tr>
<tr>
<td>500</td>
<td>48</td>
<td>158</td>
<td>81</td>
<td>183</td>
</tr>
<tr>
<td>1000</td>
<td>76</td>
<td>333</td>
<td>82</td>
<td>188</td>
</tr>
<tr>
<td>Rosen</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>68</td>
<td>187</td>
<td>88</td>
<td>256</td>
</tr>
<tr>
<td>100</td>
<td>155</td>
<td>389</td>
<td>238</td>
<td>671</td>
</tr>
<tr>
<td>500</td>
<td>10</td>
<td>31</td>
<td>9</td>
<td>24</td>
</tr>
<tr>
<td>Dixon</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>26</td>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td>100</td>
<td>8</td>
<td>26</td>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td>500</td>
<td>8</td>
<td>26</td>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td>1000</td>
<td>8</td>
<td>26</td>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td>Shallow</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>26</td>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td>100</td>
<td>8</td>
<td>26</td>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td>500</td>
<td>8</td>
<td>26</td>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td>1000</td>
<td>8</td>
<td>26</td>
<td>6</td>
<td>18</td>
</tr>
<tr>
<td>Non-diagonal</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>48</td>
<td>115</td>
<td>76</td>
<td>170</td>
</tr>
<tr>
<td>100</td>
<td>74</td>
<td>175</td>
<td>86</td>
<td>194</td>
</tr>
<tr>
<td>500</td>
<td>78</td>
<td>188</td>
<td>86</td>
<td>200</td>
</tr>
<tr>
<td>1000</td>
<td>8</td>
<td>26</td>
<td>6</td>
<td>18</td>
</tr>
</tbody>
</table>

From Table (1A)

Percentage performance of the new algorithm against both DFP and BFGS algorithms for 100% DFP. We have:

Table (1B)

<table>
<thead>
<tr>
<th>Tools</th>
<th>DFP</th>
<th>BFGS</th>
<th>New</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOI</td>
<td>100%</td>
<td>65.13</td>
<td>34.87</td>
</tr>
<tr>
<td>NOF</td>
<td>100%</td>
<td>76.18</td>
<td>23.82</td>
</tr>
</tbody>
</table>

5. Appendix:

All the test functions used in this paper are from general literature: See [1] for the details of all these test functions.

1. Powell function (Generalized form)

   \[
   f = \sum_{i=1}^{4/3} \left[ (x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4 \right],
   \]

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

5. Dixon function
   \[ f = (1 - x_i)^2 + (1 - x_{i0})^2 + \sum_{i=2}^{n} (x_i^2 - x_{i+1})^2, \ x_0 = (-1;...)^T \]

6. Non-diagonal function
   \[ f = \sum_{i=1}^{n} [100(x_i - x_i^2)^2 + (1 - x_i)^2], \ x_0 = (-1;...)^T \]

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


