Investigation on Scaled CG-Type Algorithms for Unconstrained Optimization

Abbas Y.AL-Bayati  Khalil K. Abo  Salah G. Shareef
profabbasalbayati@yahoo.com
College of Computer sciences and Mathematics
University of Mosul, Iraq

Received on: 05/03/2006  Accepted on: 16/08/2006

ABSTRACT
In this paper, we describe two new algorithms which are modifications of the Hestens-stiefel CG-method. The first is the scaled CG-method (obtained from function and gradient-values) which improves the search direction by multiplying to a scalar obtained from function value and its gradient at two successive points along the iterations. The second is the Preconditioned CG-method which uses an approximation at Hessein of the minimizing function. These algorithms are not sensitive to the line searches. Numerical experiments indicate that these new algorithms are effective and superior especially for increasing dimensionalities.

Keywords: Unconstrained Optimization, Conjugate gradient algorithm, Hestens-stiefel method, Hessein matrix.
Where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is twice differentiable real valued function, is one of the most active areas in optimization community, virtually appearing in every human activity.

For solving these problems many efficient methods have been suggested. Excellent Presentations of these methods can be found, for example (Fletcher, 1987; Gill et-al 1981; Boyd and Vavdenberhe, 2003; Nocedal, 1992; Edwin 2001). The most useful algorithms classify in: The Conjugate gradient and its variants, Newton method and its extensions; the DFP variable metric method, many different QN methods. All these methods are iterative and consider iterations of the form:

$$x_{k+1} = x_k + \alpha_k d_k$$  \hspace{1cm} (2)

where $d_k$ is a descent direction i.e.

$$d_k^T g_k < 0$$  \hspace{1cm} (3)

Where $g_k = \nabla f(x_k)$ and $\alpha_k$ is a step length obtained by line search. Conjugate Gradient methods consider the search directions as:

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \quad k \geq 1$$  \hspace{1cm} (4)

where the scalar $\beta_k$ is chosen in such a manner that the method reduces to the linear Conjugate Gradient when the function is quadratic and line search is exact. The rest of the methods define the search directions by:

$$d_k = -G_k^{-1} g_k$$,  \hspace{1cm} (5)

where $G_k$ is a non-singular symmetric matrix. Mainly the matrix $G_k$ is selected as $G_k = I$ (identity matrix which gives the steepest descent method), $G_k = \nabla^2 f(x_k)$ (the Newton method) or an approximation of the Hessian $\nabla^2 f(x_k)$.

Different modifications are made to the CG-algorithm in different ways (see for example Hu and Story, 1990; Fletcher, 1993; Al-Baali 1985), most of these modifications are made to the search directions to improve then.

We end this general introduction by content of this paper which is organized as follows: In section(2) we review the Conjugate gradient, QN-methods and their Combinations, section(3) contains the development of the new algorithms, the last section includes the numerical results.

2. Review of the methods:
2-1 Conjugate Gradient Methods:
Non-linear Conjugate gradient (CG) is one of the most useful and the earliest techniques for solving large-scale non-linear optimization problems.

Many variants of this original scheme have been proposed, and some are widely used in practice, CG-methods only use the first order derivatives information of the objective function and need not update the Hessian matrix at each iteration. First, these are used to solve the general unconstrained optimization problems by Fletcher and Reeves (1964).

Conjugate gradient methods depend on the fact that for quadratic, if we search along a set of \( n \) mutually conjugate directions \( d_k, k = 1, 2, ..., n \) (by conjugate directions we mean that \( d_i^T G d_j = 0, \forall i \neq j \), where \( G \) is \( n \times n \) positive matrix), then we will find the minimum in at most \( n \)-steps if line searches are exacts i.e.

\[
d_k^T g_{k+1} = 0 \tag{6}
\]

Moreover, if we generate this set of directions by known gradients then each direction can be simply expressed as:

\[
d_{k+1} = -g_{k+1} + \beta_k d_k, \quad k \geq 1 \tag{7}
\]

Where

\[
\beta_k = \frac{g_{k+1}^T y_k}{d_k^T y_k} \tag{8}
\]

And

\[
y_k = g_{k+1} - g_k \tag{9}
\]

\( \beta_k \) given in equation (8) is called the Hestenes-stiefel formula; clearly there are other different forms of \( \beta_k \) such as Polak-Ribiere formula which is derived from \( \beta_k \) in equation (8) as follows:

\[
g_{k+1}^T y_k = g_{k+1}^T (g_{k+1} - g_k) = g_{k+1}^T g_{k+1} - g_{k+1}^T g_k \]

\[
d_k^T y_k = d_k^T (g_{k+1} - g_k) = d_k^T g_{k+1} - d_k^T g_k \]

Assuming equation (6) and considering:

\[
d_k = -g_k + \beta_{k-1} d_{k-1}
\]

(see Edwin 2001) we get

\[
d_k^T g_k = -g_k^T g_k
\]

Hence

\[
\frac{g_{k+1}^T y_k}{d_k^T y_k} = \frac{g_{k+1}^T g_k}{g_k^T g_k} = \beta_{PR} \tag{10}
\]
The Fletcher Reeves formula can be derived from $\beta_{pr}$ by considering $g^T_{k+1}g_k = 0$ to obtain $\beta_{FR} = \frac{g^T_{k+1}g_{k+1}}{g^T_k g_k}$ .................(11), and there are many other forms. All these $\beta_k$ are equivalent on quadratics with exact line searches and starting with steepest descent direction, but when extended to general non-linear functions, the conjugate gradient algorithms with different $\beta_k$ are quite different in effacing and are considered to be not so efficient as the QN algorithms in general. The focus in this paper to the $\beta_k$ given in equation (8) since its original CG-method.

Finally, in many implementations of Conjugate gradient methods, the iteration (7) is restarted every n or (n+1) steps setting $\beta_k$ equal to zero i.e. taking steepest descent step. This ensures global convergence (Nocedal, 1992). However, different restarts are introduced (see Fletcher, 1987). One of the well-known restarts given by Powell (Powell 1977) is:

$$|g^T_{k+1}g_k| \geq 0.2\|g_k\|^2$$

.........................(12)

This criterion will be used later in our suggested algorithms.

2-2 Quasi-Newton Methods:
Quasi-Newton methods are probably the most popular general purpose algorithms for unconstrained optimization problems. Many QN-methods are advantageous due to their fast convergence and absence of second order derivatives computation.

For the QN-methods assume that at the $kth$ iteration at approximation point $x_k$ and $n \times n$ matrix $H_k$ are available. Then the methods proceed by generating a sequence of approximation points via the equation:

$$x_{k+1} - x_k + \alpha_k d_k$$

and

$$d_{k+1} = -H_k g_k$$

.........................(13)

Where $H_k$ is an approximation of $G_k^{-1}$ which is corrected or updated form iteration to iteration, In general, $H_k$ is symmetric and positive definite, there are different choices of $H_k$ (see Fletcher, 1987), we list here some most popular forms (Frandsen et. al. 2004)
Investigation on Scaled CG-Type...

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

is called Rank one Correction formula,

where \( v_k = x_{k+1} - x_k \) and \( y_k \) as in equation (9)

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

is the DFP formula

\[ H_{k+1}^{BFGS} = H_k + [1 + \frac{y_k^T H_k y_k}{y_k^T y_k}] \frac{v_k^T v_k}{y_k^T y_k} y_k - \frac{v_k^T v_k H_k + H_k y_k v_k^T}{y_k^T y_k} \] .................(16)

All three forms satisfy the Quasi-Newton Condition

\[ y_k = G_k v_k \] or \[ H_{k+1} y_k = v_k \]

and maintains positive definite matrices if \( H_0 \) is positive.

We conclude our discussion of the QN methods with the following theorems:

**Theorem (1):** If the DFP algorithm is applied to the quadratic with Hessian \( G = G^T \) we have

\[ H_{k+1} y_i = v_i \] ............................................. (17)

**Theorem (2):** Suppose that \( g_k \neq 0 \). In the DFP algorithm, if \( H_k \) is positive definite then so \( H_{k+1} \). For proof of the theorems (1) and (2) (see Dixon, 1972)

### 2-3 Preconditioned CG algorithm (PCG):

The Preconditioned CG methods (PCG) first appeared in paper by Axelsson (Axelsson, 1972). It was developed with object of accelerating the convergence of the CG-method by a transformation of variables while keeping the basic properties of the method. Such transformation was introduced by Allwright (Allwright, 1972), the symmetric positive definite matrix \( H \) can be factored in various ways for example as \( H = LL^T \) where \( L \) is lower triangular and non-singular (for more detail see Allwright, 1972).

Buckley (Buckley, 1978, a and b) introduced an algorithm in which conjugate gradient and quasi-Newton search directions occur together and which can be interpreted as a conjugate gradient algorithm with changing...
metric. Many authors have extended this type of algorithms (see for example Al-Bayati, 1996).

The search direction to the preconditioned (PCG) method is defined by:

\[
d_1 = -H_1 g_1
\]

\[
d_{k+1} = -H g_k + \beta_k d_k \quad \text{for } k \geq 1
\]

\[
\beta_{HS} = \frac{g_{k+1}^T H y_k}{d_k^T H y_k}
\]

Where \( H \) is one of the forms given in equations (7) or (8) or (9). In this paper, our focus is to the \( H^{PDF} \).

3. Development of two suggested Algorithms

3-1 Scaled CG-method New 1 (say)

One of the reasons for inefficiency of conjugate gradient algorithms is that none of the \( \beta_k \) takes into consideration the effect of inexact line searches (Hu and Story, 1990). In order to do this and find an optimal \( \beta \), Liu and Story (1991) introduced an algorithm that finds an optimal \( \beta_{LS} \) by solving a quadratic function \( f \) as:

\[
F(x_{k+1}, d_{k+1}) = \min_{\alpha} f(x_{k+1} + \alpha d_{k+1})
\]

and

\[
\min_{\beta} F(x_{k+1}, -g_{k+1} + \beta d_k)
\]

an then takes

\[
d_{k+1} = \alpha^* (-g_{k+1} + \beta_{LS} d_k)
\]

Where \( \alpha^* \) and \( \beta_{LS} \) solve (19) and (20), respectively, the major drawback in Liu and Story CG algorithm is solving equation (20) at each iteration.

In this paper, a new form of CG-method presented with line search and so it develops conjugate search directions. This new approach will find the minimum of a \( n \) - dimensional quadratic function in at most \((n+1)\) function and gradient evaluations.

The motivation of the iteration

\[
x_{k+1} = x_k + \alpha_k d_k , \quad \alpha_k > 0
\]

is that the search direction \( d_i \) is chosen to ensure that the point \( x_{i+1} \) is a minimize of the model.
Investigation on Scaled CG-Type…

\[ f(x_k + \alpha_k d_k) = f(x_k) + g_k^T v_k + \frac{1}{2} v_k^T G v_k, \quad \ldots \] (22)

where \( v_k = x_{k+1} - x_k = \alpha_k d_k, \quad \ldots \) (23)

or \( f(x_{k+1}) = f(x_k) - \alpha_k g_k^T g_k + \frac{1}{2} \alpha_k^2 g_k^T \nabla^2 f(z) g_k, \quad \ldots \) (24)

where \( z \in [x_k, x_{k+1}] \). If the distance between \( x_k \) and \( x_{k+1} \) is small enough we can choose \( z = x_{k+1} \) and consider \( \gamma(x_{k+1})I \) as an approximation of the \( \nabla^2 f(x_{k+1}) \), where \( \gamma(x_{k+1}) \in R \). This is an approximation of the Hessian at a point \( x_{k+1} \) is computed using the local information from point \( x_k \) therefore we can write

\[ \gamma(x_{k+1}) = \frac{2}{g_k^T g_k} \left[ f(x_k) - f_k + \alpha_k g_k^T g_k \right] \quad \ldots \] (25)

where the step size \( \alpha_k \) is satisfying line search conditions (Wolfe-Powell conditions) such as

\[ f(x_k + \alpha_k d_k) \leq f(x_k) + \rho \alpha_k g_k^T d_k \quad \text{and} \quad |\nabla f(x_k + \alpha_k d_k) d_k| \leq -\sigma |\nabla f(x_k) d_k| \quad \ldots \] (26)

where \( 0 < \rho < \sigma < 1 \)

Then we take \( \gamma(x_{k+1})I \) as an approximation of \( G \). It is clear that if \( \gamma(x_{k+1}) \geq 0 \) this approximation will be positive definite hence to complete the method.

We must consider the situation when \( \gamma(x_{k+1}) < 0 \), i.e. if

\[ f(x_{k+1}) - f(x_k) + \alpha_k g_k^T g_k < 0 \]

In this case we can change the step size \( \alpha_k \) as \( \alpha_k + \eta_k \) s.t.

\[ f(x_{k+1}) - f(x_k) + (\alpha_k + \eta_k) g_k^T g_k > 0 \quad \ldots \] (27)

to get a value for \( \eta_k \). Let us select a \( \delta_k > 0 \) small enough and consider

\[ \eta_k = \frac{1}{g_k^T g_k} [f(x_k) - f(x_{k+1}) - \alpha_k g_k^T g_k + \delta] \quad \ldots \] (28)

then the new value of \( \gamma(x_{k+1}) \) can be computed from

\[ \gamma(x_{k+1}) = \frac{2}{g_k^T g_k} \frac{1}{(\alpha_k + \eta_k)^2} [f(x_{k+1}) - f(x_k) + (\alpha_k + \eta_k) g_k^T g_k] \quad \ldots \] (29)

then the new algorithm can be obtained by multiplying the search direction to scalar \( \frac{1}{\gamma(x_{k+1})} \) as follows:

17
The outlines of the Self-Scaling CG-method (new1):

Step (1): Set initial point $x_1$ and scalar $\varepsilon$.

Step (2): For $k = 1$ set $d_1 = -g_1$.

Step (3): For $k \geq 1$ compute $x_{k+1} = x_k + \alpha_k d_k$ where $\alpha_k$ is obtained by line search procedure.

Step (4): If $\|g_k\| \leq \varepsilon$ stop. Otherwise go to step (5).

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

Step (6): Compute $\gamma(x_{k+1})$ from equations (25) or (28 and 29).

Step (7): Set the new $\beta_k$ to

$$
\beta_k = \frac{1}{\gamma_{k+1}} \frac{g_k^T v_k}{d_k^T y_k}
$$

Step (8): set $d_{k+1} = -\frac{1}{\gamma_{k+1}} g_{k+1} + \beta_k d_k$.

Step (9): If restart satisfied (restart Powell 1977) go to step (2) else $k = k + 1$ go to step (3).

3-2 Self-Scaling PCG method (New2):

In this section, a new PCG method for solving unconstrained optimization problems is proposed.

This new PCG algorithm considered here has an additional property of being invariant under scaling of the function or of its variables where the objective function is twice continuously differentiable and search direction is descent i.e. $g_k^T d_k < 0$ also we assume that line search is exact i.e. $g_k^T d_k = 0$.

Let $d_k = -H_k g_k + \beta_k d_{k-1}$ where $\beta_k = \beta(y_k d_k, H_k g_{k-1})$. where the matrix $H_k$ is an approximation of $G^{-1}$ the inverse of Hessian of the objective function $f(x)$.

One important feature of PCG method is the choice of $H_k$. The method requires $H_k$ to be positive definite to deduce directions.

Let $H_k = \gamma_{k+1} I$ as an approximation to the inverse Hessian where $I$ and identity matrix and $\gamma_{k+1}$ computed from (25), if $\gamma_{k+1} > 0$ and from (29) if $\gamma_{k+1} < 0$ and satisfy the condition

$$
H_k = \arg \min_{\gamma} \|H_k y_k - v_k\|_2 
$$

……………………………..(30)
Investigation on Scaled CG-Type…

Where $\|\cdot\|_2$ usual Euclidian norm, then the new2 algorithm can be organized as follows:

**The outlines of the Self-Scaling PCG method (new2):**

Step(1): Set $x_1, e, H_1 = I$.

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

Step(3): Set $x_{k+1} = x_k + \alpha_k d_k$ for $k \geq 1$, where $\alpha_k$ optimal step size.

Step(4): If $\|g_k\| \leq \varepsilon$ stop. Otherwise

Check if restart equation (12) is satisfied then set $x_k = x_1$ go to step (2).

otherwise go to step (5).

Step(5): Compute $y_k = g_{k+1} - g_k$ and $\nu_k = x_{k+1} - x_k$ and $\gamma_k$ form equation (25).

Step(6): Compute $H_k = \gamma_k I$ satisfying (30) and $\beta_k = \frac{g_k^T H_k y_k}{d_k^T y_k}$.

Step(7): Set $d_{k+1} = -H_k g_{k+1} + \beta_k d_k$ go to step (3).

4. **Numerical Results:**

All the algorithms described in this paper namely:

1. The standard HSCG method.
2. Preconditioned CG with (DFP and BFGS).
3. New1 and New2 proposed algorithms are coded in double precision FORTRAN 90. The complete set of results is given in Table (1) and Table (2). In our numerical comparison, the number of function evaluations NOF and number of iterations NOI are considered. The actual convergence criterion employed was $\|g_{k+1}\| < 1 \times 10^{-5}$ for all the algorithms. Well-known test functions with different dimensions n are employed in this comparisons.

**Table (1a) Comparison of HSCG and New1**

<table>
<thead>
<tr>
<th>Test Fun.</th>
<th>N</th>
<th>HSCG</th>
<th>New1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NOI</td>
<td>NOF</td>
</tr>
<tr>
<td>Dixon</td>
<td>4</td>
<td>13</td>
<td>28</td>
</tr>
<tr>
<td>Powell (4)</td>
<td>4</td>
<td>50</td>
<td>114</td>
</tr>
<tr>
<td>Rosen</td>
<td>4</td>
<td>28</td>
<td>68</td>
</tr>
<tr>
<td>Cubic</td>
<td>4</td>
<td>16</td>
<td>42</td>
</tr>
<tr>
<td>Wood</td>
<td>4</td>
<td>31</td>
<td>67</td>
</tr>
<tr>
<td>Dixon</td>
<td>10</td>
<td>22</td>
<td>46</td>
</tr>
<tr>
<td>Powell (4)</td>
<td>20</td>
<td>34</td>
<td>78</td>
</tr>
</tbody>
</table>
Clearly there is an improvement of 45% in NOI and 41% in NOF for our new proposed algorithms.

Table (2a) Comparison of PCG methods with DFP, BFGS and New1

<table>
<thead>
<tr>
<th>Test Fun.</th>
<th>N</th>
<th>PCG with DFP</th>
<th>PCG with BFGS</th>
<th>New2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NOI</td>
<td>NOF</td>
<td>NOI</td>
</tr>
<tr>
<td>Powell (4)</td>
<td>4</td>
<td>22</td>
<td>79</td>
<td>21</td>
</tr>
<tr>
<td>Cubic</td>
<td>4</td>
<td>19</td>
<td>88</td>
<td>18</td>
</tr>
<tr>
<td>Dixon</td>
<td>4</td>
<td>10</td>
<td>31</td>
<td>14</td>
</tr>
<tr>
<td>Wood</td>
<td>4</td>
<td>54</td>
<td>149</td>
<td>37</td>
</tr>
<tr>
<td>Rosen</td>
<td>4</td>
<td>23</td>
<td>63</td>
<td>34</td>
</tr>
<tr>
<td>Powell (4)</td>
<td>20</td>
<td>36</td>
<td>135</td>
<td>38</td>
</tr>
<tr>
<td>Cubic</td>
<td>20</td>
<td>37</td>
<td>93</td>
<td>18</td>
</tr>
<tr>
<td>Dixon</td>
<td>10</td>
<td>31</td>
<td>90</td>
<td>22</td>
</tr>
<tr>
<td>Wood</td>
<td>20</td>
<td>130</td>
<td>353</td>
<td>84</td>
</tr>
<tr>
<td>Rosen</td>
<td>20</td>
<td>68</td>
<td>187</td>
<td>34</td>
</tr>
<tr>
<td>Powell (4)</td>
<td>100</td>
<td>82</td>
<td>387</td>
<td>71</td>
</tr>
<tr>
<td>Cubic</td>
<td>100</td>
<td>48</td>
<td>153</td>
<td>18</td>
</tr>
<tr>
<td>Wood</td>
<td>100</td>
<td>243</td>
<td>861</td>
<td>251</td>
</tr>
</tbody>
</table>

(1b) percentages of improving the New1 method

<table>
<thead>
<tr>
<th>Tools</th>
<th>HSCG method</th>
<th>New1 method</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOI</td>
<td>100%</td>
<td>55%</td>
</tr>
<tr>
<td>NOF</td>
<td>100%</td>
<td>59%</td>
</tr>
</tbody>
</table>
Investigation on Scaled CG-Type…

<table>
<thead>
<tr>
<th>Tools</th>
<th>PCG with PDF</th>
<th>PCG with PFGS</th>
<th>New2 method</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOI</td>
<td>100%</td>
<td>65%</td>
<td>39%</td>
</tr>
<tr>
<td>NOF</td>
<td>100%</td>
<td>62%</td>
<td>28%</td>
</tr>
</tbody>
</table>

Clearly, self-scaling techniques are very effective in unconstrained optimization algorithms. The two different approaches used in this paper proved to be very effective, especially for high dimension functions.

Conclusions:

Clearly, our numerical results indicate that there are improvements of proposed self-scaling techniques over standard DFP and BFGS algorithms.
REFERENCES


