ABSTRACT

In this paper, a new extended (CG) algorithms is proposed. It is in fact, a particular type of the Conjugate Gradient (CG) method which employs non-quadratic rational model, and based on inexact line searches. The Fletcher and Reeves restarting criterion was employed to the standard and New versions and gave dramatic savings in computational time. The new algorithms is were promising in general, seven non linear tests function with different versions were used.

أعطنا نماذج غير تربيعة كأساس الخوارزمية في التدرج المترافق

 الملخص

في هذا البحث تم استحداث خوارزمية جديدة موسمة في مجال التدرج المترافق لربط نماذج غير تربيعة والنموذج الجديد نموذج نسبي غير تربيعي Fletcher and Reeves المعتمد على خطوط بحث غير تامة. ولقد تم استعمال وسيلة استرجاع في الصيغتين القديمة والجديدة وكان لذلك أثر كبير في توفير الزمن المطلوب للحل. أثبتت الحسابات العددية اكفاءة الخوارزمية الجديدة مقارنة بالأخرى وباستعمال دوال غير خطية وبأبعاد مختلفة.

* Department of Mathematics/ College of Science/ Mosul University
** Department of Mathematics/College of Science/Kirkuk University

Received: 3/ 9 /2004 Accepted: 27/ 11 / 2004
1. Introduction:

Conjugate gradient (CG) algorithms form a class of CG-algorithms for minimizing a general differentiable function $f(x)$ $x \in \mathbb{R}^n$, whose gradient $g(x)$ can be calculated, are based on the following concept of conjugacy:

If $Q$ is a positive definite symmetric $n \times n$ matrix, then the directions $d_1, d_2, \ldots, d_n$ where $d_k \neq 0$ for $k=1,2,\ldots,n$, are mutually $Q$-conjugate if

$$d_i^T Q d_k = 0 \quad \text{for} \quad i \neq k$$

The classical algorithm in this category proposed by Fletcher and Reeves (9) and is based on the iterative scheme:

$$x_{k+1} = x_k + \lambda_k d_k, \quad k = 1,2,\ldots,n$$

Where the scalar $\lambda_k$ is the smallest positive local minimizer of the One dimensional problem

$$\min_{\lambda} f(x_k + \lambda d_k)$$

is a search direction generated by the equations: $d_k$ and

$$
\begin{align*}
  &d_1 = -g_1, \\
  &d_{k+1} = -g_{k+1} + \beta_k d_k, \\
  &\beta_k = \frac{g_{k+1}^T y_k}{d_k^T y_k}
\end{align*}
$$

If the method is applied to the quadratic function

$$q(x) = \frac{1}{2} x^T Q x$$

2. Extended Conjugate Gradient (ECG) method:

A more general model than the quadratic one is proposed in this paper as a basis for a CG algorithm. If $q(x)$ is a quadratic function, then a function $f$ is defined as a nonlinear scaling of $q(x)$ if the following condition holds:

$$f = F(q(x)), \quad \frac{dF}{dq} = F' > 0 \quad \text{and} \quad q(x) > 0 \quad \ldots \ldots \quad (1)$$

Where $x^*$ is the minimizer of $q(x)$ with respect to $x$ Spedicate( ).
The following properties are immediately derived from the above condition:

i) Every contour line to \( q(x) \) is a contour line of \( f \).

ii) If \( x^* \) is a minimizer of \( q(x) \), then it is a minimizer of \( f \).

That \( x^* \) is a global minimum of \( q(x) \) does not necessarily mean that it is a global minimum of \( f \).

Various authors have published related works in the area:

A conjugate method which minimizes the function \( f(x) = (q(x))^p \), and \( x \in \mathbb{R}^n \) in at most step has been described by Fried [10].

Another special case, namely

\[
F(q(x)) = \varepsilon_1 q(x) + \frac{1}{2} \varepsilon_2 Q^2(x)
\]

Where \( \varepsilon_1 \) and \( \varepsilon_2 \) are scalars, has been investigated by Boland & Kowalik [7].

Another model has developed by Tassopoulos and Storey [14] as follows:

\[
F(q(x)) = \varepsilon_1 q(x) + \frac{1}{\varepsilon_2} q(x) : \varepsilon_2 > 0
\]

AL-Assady in [3] developed another model as follows (\( F(q(x)) = \ln(q(x)) \))

Al-Bayati [1] has been developed a new rational models which is defined as follows: \( F(q(x)) = \varepsilon_1 q(x)/\varepsilon_2 q(x) , \varepsilon_2 < 0 \).

Also Al-Bayati [2] developed an extended CG algorithm which is based on a general logarithmic model

\[
F(q(x)) = \log(\varepsilon q(x) - 1) , \varepsilon > 0
\]

Al-Assady & Huda [5] described their ECG algorithm which is based on the natural log function for the rational \( q(x) \) function

\[
F(q) = \log \left[ \frac{\varepsilon_1 q(x)}{\varepsilon_2 q(x) + 1} \right] , \varepsilon_2 < 0
\]
3. CONJUGATE GRADIENT METHOD WITH INEXACT LINE SEARCH:

In order to improve the local rate of convergence and the efficiency of the traditional CG-method several well-Know methods are discussed in this area. Among these methods Sloboda (13) defines a new generalized CG-algorithm for minimizing a strictly convex function of the general form

\[ f(x) = F(q(x)) \]  

(2)

We now list out-lines of sloboda extended CG method:

Algorithm (sloboda 1980)

Step (1): set \( k=1; \quad g_k = g_k \) and \( d_k = -g_k \)

Step (2): compute \( \lambda_k \) by exact line search and \( x_{k+1} = x_k + \lambda_k d_k \)

Step (3): compute \( g_{k+1/2} = g(x_k + \lambda_k d_k / 2) \)

Step (4): Test for convergence if achieved stop. If not continue

Step (5): If \( k=0 \) mod (n) go to step (1) else continue

Step (6): compute \( w_k = d_k^T g_k / d_k^T g_{k+1/2} \)

If \( w_k = 0 \) set \( i=i+1 \) go to step (1)

Step (7): \( d_{k+1} = -g_{k+1} + \beta_k d_k ; \beta = (g_{k+1} - g_k)g_{k+1} / d_k^T (g_{k+1} - g_k) \)

Step (8): set \( k=k+1 \) and go to step (2)

Step (9): If \( k \) EQ .n go to Step (1). If not continue.
4. The Derivation $p_i$:

The implementation of the extended CG method has been performed for general function $F(q(x)$ of the form of equation (2). The unknown quantities $P_i$ were expressed in terms of available quantities of the algorithm. The authors, introduced in [4] a new model, which can be written as:

$$f(x) = F(q(x) = \sin \left( \frac{\epsilon_i q(x) + 1}{\epsilon_2 q(x)} \right)$$

Solving equation (2) for $q$

$$\sin^{-1} f(x) \left( \frac{\epsilon_i q(x) + 1}{\epsilon_2 q(x)} \right) = \frac{\epsilon_i q(x) + 1}{\epsilon_2 q(x)} = \frac{1}{\epsilon_2 \ln[f(x) + \sqrt{1 - f(x)^2} - \epsilon_i]}$$

And using the expression for $p_i = \frac{f'(i-1)}{f'(i)}$

$$\rho_i = -\frac{\cos(\epsilon_i q_{i-1} + 1/\epsilon_2 q_{i-1})}{\cos(\epsilon_i q_i + 1/\epsilon_2 q_i)} \left( \frac{-1}{\epsilon_2 q^2_{i-1}} \right)$$

From the above equation we have
\[
\rho_i = \frac{\left[ if_{i-1} + \sqrt{1 - f_{i-1}^2} \right]^2 + 1 \left[ \ln(if_{i-1} + \sqrt{1 - f_{i-1}^2}) - \frac{e_1}{e_2} \right]^2}{if_{i-1} + \sqrt{1 - f_{i-1}^2} \left[ if_{i} + \sqrt{1 - f_{i}^2} \right]^2 + 1 \left[ \ln(if_{i} + \sqrt{1 - f_{i}^2}) - \frac{e_1}{e_2} \right]^2}
\]

In terms of the known quantities such a function and gradient values, from

\[ g_i = F_i^\top Q(x_i - x^*) \]
\[ g_{i-1} = F_{i-1}^\top Q(x_{i-1} - x^*) \]

Where Q is the Hessian Matrix and \( x^* \) is the minimum point, we have:

\[
\rho_i = \frac{\left[ if_{i-1} + \sqrt{1 - f_{i-1}^2} \right]^2 + 1 \left[ \ln(if_{i-1} + \sqrt{1 - f_{i-1}^2}) - \frac{e_1}{e_2} \right]^2}{if_{i-1} + \sqrt{1 - f_{i-1}^2} \left[ if_{i} + \sqrt{1 - f_{i}^2} \right]^2 + 1 \left[ \ln(if_{i} + \sqrt{1 - f_{i}^2}) - \frac{e_1}{e_2} \right]^2}
\]

Furthermore

\[ g_{i-1}^\top (x_i - x^*) = g_{i-1}^\top (x_{i-1} + \lambda_{i-1}d_{i-1} - x^*) \]
\[ = g_{i-1}^\top (x_{i-1} - x^*) + \lambda_{i-1}g_{i-1}^\top d_{i-1} \]

\[ g_i^\top (x_i - x^*) = g_i^\top (x_i + \lambda_i d_i - x^*) \]
\[ = g_i^\top (x_i - x^*) \]

Since \( g_i^\top d_{i-1} = 0 \) Therefore, we can express \( \rho_i \) as follows:

\[
\rho_i = \frac{g_{i-1}^\top (x_{i-1} + \lambda_{i-1}d_{i-1} - x^*)}{g_i^\top (x - x^*)}
\]
From (3) and (4), it follows that: 

\[ \rho_i = \rho_i \left[ \frac{q_{i-1}}{q_i} \right] + \lambda_{i-1} g_i^T d_{i-1} / 2F_i q_i \]

Where 

\[ q_i = \frac{\left[ \text{if} + \sqrt{1 - f_i^2} \right]^2 + 1}{\text{if} + \sqrt{1 - f_i^2}} - \frac{\varepsilon_2}{\varepsilon_1} \left[ \text{ln} \left( \text{if} + \sqrt{1 - f_i^2} \right) - \frac{\varepsilon_1}{\varepsilon_2} \right] \]

\[ 2 \left[ \text{if} + \sqrt{1 - f_i^2} \right] \]

The quantities \( q_{i-1}/q_i \) and \( J_i q_i \) can be rewritten as:

\[ q_{i-1} = \frac{\ln \left[ \text{if}_{i-1} + \sqrt{1 - f_{i-1}^2} \right] - \frac{\varepsilon_1}{\varepsilon_2}}{q_i} \]

\[ f_i q_i = \frac{\left[ \text{if}_i + \sqrt{1 - f_i^2} \right]^2 + 1}{\text{if}_i + \sqrt{1 - f_i^2}} \frac{\left[ \text{ln} \left( \text{if}_i + \sqrt{1 - f_i^2} \right) - \frac{\varepsilon_1}{\varepsilon_2} \right]^2}{2 \left[ \text{if}_i + \sqrt{1 - f_i^2} \right]^2} \]

From the definition of \( \rho_i \), we have:

\[ \frac{\left[ \text{if}_i + \sqrt{1 - f_i^2} \right]^2 + 1}{\text{if}_i + \sqrt{1 - f_i^2}} \frac{\left[ \text{ln} \left( \text{if}_{i-1} + \sqrt{1 - f_{i-1}^2} \right) - \frac{\varepsilon_1}{\varepsilon_2} \right]^2}{\text{if}_{i-1} + \sqrt{1 - f_{i-1}^2}} = \]
using the following transformation:

\[
\frac{[\ln f_i + \sqrt{1-f_i^2}] + 1}{\ln f_i + \sqrt{1-f_i^2}} = x, \quad \ln \left(\frac{f_i + \sqrt{1-f_i^2}}{\varepsilon_2} \right) = y
\]

\[
\ln \left(\frac{f_i + \sqrt{1-f_i^2}}{\varepsilon_2} \right) = y + w \quad \text{and} \quad \ln \left(\frac{f_i + \sqrt{1-f_i^2}}{\varepsilon_2} \right) - \ln \left(\frac{f_{i-1} + \sqrt{1-f_{i-1}^2}}{\varepsilon_2} \right) = w
\]

c = \lambda_{i-1} g_{i-1}^T d_{i-1}

Then \(y = cw/xw + c\)

Therefore

\[
- \frac{f_{i-1}^2}{f_i^2} = \frac{\ln[f_i + \sqrt{1-f_i^2}] - \ln[f_{i-1} + \sqrt{1-f_{i-1}^2}]}{\ln[f_i + \sqrt{1-f_i^2}] + 1} \left[\ln[f_i + \sqrt{1-f_i^2}] - \ln[f_{i-1} + \sqrt{1-f_{i-1}^2}]\right] + \lambda_{i-1} g_{i-1}^T d_{i-1}
\]
5. New modification for the Sloboda method:

In order to improve the global rate of convergence of minimization algorithms when applied to more general functions than the quadratic variable metric (VM) matrices which may be used to accelerate the CG-algorithm (see for example Al-Byati and Al-assady (6)). In this section, a new expression for the new search direction $d_{k+1}$ of the Sloboda method is suggested which is invariant to non-linear scaling of non-quadratic function. Al-assady and Hassan (4) (to appear) are used to extend the sloboda method.

In use the new suggested algorithms require less vector storage’s than the sloboda algorithms.

We now give the outline of the new proposed modifications:

**New Algorithm:**

Step (1): set $k=1$; $g_k = g$ and $d_k = -g_k$

Step (2): compute $\lambda_k$ by ELS and $x_{k+1} = x_k + \lambda_k d_k$

Step (3): compute $g_{k+1}^* = g(x_k + \lambda_k d_k / 2)$

Step (4): Test for convergence if achieved stop.

If not continue

Step (5): If $k=0 \mod (n)$ go to step (1) else continue

Step (6): compute $g_{k+1} = w_k g_{k+1}^* - g_{k+1}$ where

$$w_k = d_k^T g_k / d_k^T g_{k+1}^*$$
If $g_{k+1} = 0$ set $I = I + 1$ go to step (1)

Step (7) compute

$$
\rho_i = \frac{\left[ \frac{f_{i-1} + \sqrt{1 - f_{i-1}^2}}{\ln \left( \frac{f_{i-1} + \sqrt{1 - f_{i-1}^2}}{\varepsilon_1} \right)} - 1 \right]^2 \ln \left( \frac{f_{i-1} + \sqrt{1 - f_{i-1}^2}}{\varepsilon_2} \right)}{\left( \frac{f_{i-1} + \sqrt{1 - f_{i-1}^2}}{\ln \left( \frac{f_{i-1} + \sqrt{1 - f_{i-1}^2}}{\varepsilon_1} \right)} - 1 \right)^2} + \frac{f_{i-1} + \sqrt{1 - f_{i-1}^2}}{\ln \left( \frac{f_{i-1} + \sqrt{1 - f_{i-1}^2}}{\varepsilon_2} \right)}
$$

Step (8): $d_{k+1} = -g_{k+1} + \beta_k d_k$; $\beta = (\rho_k g_{k+1} - g_k g_{k+1} / d^T (\rho_k g_{k+1} - g_k)$

Step (9): set $k = k + 1$ and go to step (2)

Step (10): If $k = N$ go to step (1)

6. Numerical Results and conclusion:

In order to test the effectiveness of the new algorithms that have been used to extent the Sloboda method, a number of function have been chosen and solved numerically by utilizing the new and established method.

The same line search was employed for all the methods. This was the cubic interpolation procedure described in Bunday (8).

It is found that the NEW method which modifies Sloboda algorithm is better than the previous algorithm shown in table.
Table (1): Which uses the H/S formula, presents a comparison between the results of the NEW methods and the Sloboda method. So we can show that the NEW method has less (NOI) and (NOF) than the classical Sloboda method. The NEW method improves the two measures of performances, vis (NOI) and (NOF) (67.17 %) and the ( 80.25 ) % for the H/S formula.

Table: Comparison between the different ECG – methods by using H/S formula.

<table>
<thead>
<tr>
<th>Test Function</th>
<th>N</th>
<th>New NOI (NOF)</th>
<th>Sloboda NOI (NOF)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHALLO</td>
<td>2</td>
<td>7 (20)</td>
<td>9 (23)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>7 (21)</td>
<td>8 (21)</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>10 (30)</td>
<td>8 (20)</td>
</tr>
<tr>
<td>DIXON</td>
<td>2</td>
<td>6 (13)</td>
<td>6 (17)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>10 (25)</td>
<td>12 (26)</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>18 (40)</td>
<td>19 (41)</td>
</tr>
<tr>
<td>POWELL</td>
<td>20</td>
<td>49 (126)</td>
<td>60 (158)</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>120 (270)</td>
<td>112 (384)</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>178 (410)</td>
<td>119 (251)</td>
</tr>
<tr>
<td>WOOD</td>
<td>100</td>
<td>102 (210)</td>
<td>205 (423)</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>150 (310)</td>
<td>402 (817)</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>105 (214)</td>
<td>103 (213)</td>
</tr>
<tr>
<td>NON-DIAGONAL</td>
<td>4</td>
<td>20 (60)</td>
<td>23 (63)</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>21 (61)</td>
<td>21 (61)</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>16 (45)</td>
<td>23 (67)</td>
</tr>
<tr>
<td>CANTRAL</td>
<td>100</td>
<td>35 (230)</td>
<td>43 (347)</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>47 (270)</td>
<td>46 (393)</td>
</tr>
<tr>
<td></td>
<td>400</td>
<td>42 (180)</td>
<td>47 (410)</td>
</tr>
<tr>
<td>WOLFE</td>
<td>4</td>
<td>11 (26)</td>
<td>12 (27)</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>30 (71)</td>
<td>35 (71)</td>
</tr>
<tr>
<td>Total NOI (NOF)</td>
<td>837 (2560)</td>
<td>1246 (3190)</td>
<td></td>
</tr>
</tbody>
</table>
Appendix

1. Generalized Powell Functions:
\[ F(x) = \sum_{i=1}^{n/4} (x_{4i-9} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^2 + 10(x_{4i-9} - x_{4i})^4 \]
\[ x_0 = (3,1,0,1)^T \]

2. Generalized Cantreal Functions:
\[ F(x) = \sum_{i=1}^{n/4} \left[ \exp(x_{4i-3}) - x_{4i-2} \right]^2 + 100(x_{4i-2} - x_{4i-1})^6 + \left[ (a \tan(x_{4i-1} - x_{4i}))^4 + x_{4i-1}^8 \right] \]
\[ x_0 = (1,2,2,2)^T \]

3. Wod Functions:
\[ F(x) = \sum_{i=1}^{n/4} 100(x_{4i-2} + x_{4i-3}^2)^2 + (1 - x_{4i-3})^2 + 90(x_4 - x_{4i-1})^2 + (1 - x_{4i-1})^2 + 1.0 \]
\[ x_0 = (-3,-1,-3,-1)^T \]

4. Non-Diagonal Functions:
\[ F(x) = \sum_{i=2}^{n} 100(x_i - x_0^2)^2 + (1 - x_i)^2 \]
\[ x_0 = (-1,............)^T \]

5. Dixon Functions:
\[ F(x) = (1 - x_1)^2 + (1 - x_0)^2 + \sum_{i=2}^{n} (x_i - x_{i-1}) \]
\[ x_0 = (-1,............)^T \]

6. Wolfe Functions:
\[ F(x) = (-x_1(3 - x_1 / 2) + 2x_2 - 1)^2 + \sum_{i=1}^{n-1} (x_{i-1} - x_i(3 - x_i / 2) + 2x_{i+1} - 1)^2 + (x_n - x_1(3x_n / 2) - 1)^2 \]
\[ x_0 = (-1,............)^T \]

7. Shallo Functions:
\[ F(x) = \sum_{i=1}^{n/2} (x_{2i-1}^2 - x_{2i})^2 + (1 - x_{2i-1})^2 \]
\[ x_0 = (-2,-2,........)^T \]
REFERENCES


