A New Type of Conjugate Gradient Method
with a Sufficient Descent Property

Dr. Abbas Y. Al-Bayati*  Dr. Basim A. Hassan**

Abstract

This paper presents the development and implementation of a new unconstrained optimization method; based on the inexact line searches. Our new proposed Conjugate Gradient (CG) method always produces descent search directions and has been shown to be a global convergence. Our numerical results are promising in general by implementing ten nonlinear different test functions with different dimensions.

1. Introduction.

We consider the following unconstrained optimization problem:

$$\min \left\{ f(x) \mid x \in \mathbb{R}^n \right\}$$  \hspace{1cm} (1)

where $\mathbb{R}^n$ denotes an $n$-dimensional Euclidean space and $f: \mathbb{R}^n \rightarrow \mathbb{R}$ is smooth and nonlinear function.

It is a well-known, CG-method which is a line search method that takes the form:

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

where $d_k$ is a descent direction of $f(x)$ at $x_k$ and $\alpha_k$ is a step-size chosen by some kind of line search method and satisfies the Strong Wolfe (SW) conditions:

---

*Prof. \ College of Computers Sciences and Math. \ University of Mosul
**Lecture \ College of Computers Sciences and Math. \ University of Mosul

Received: 1/10/2011  \ Accepted: 21/12/2011
A New Type of Conjugate Gradient Method

\[ f(x_k + \alpha_k d_k) \leq f(x_k) + \delta_1 \alpha_k d_k^T g_k \] ...........(3)

\[ |g(x_k + \alpha_k d_k)^T d_k| \leq -\delta_2 d_k^T g_k \] ...........(4)

with \( 0 < \delta_1 < \delta_2 < 1 \). If \( x_k \) is the current iterate, we denote \( f(x_k) \) by \( f_k \), \( \nabla f(x_k) \) by \( g_k \), \( \nabla f(x_{k+1}) \) by \( g_{k+1} \), respectively. The search direction \( d_k \) is generally required to satisfy:

\[ g_{k+1}^T d_{k+1} < 0, \] ...........(5)

which guarantees that \( d_k \) is a descent direction of \( f(x) \) at \( x_k \) \([4, 8]\). In order to guarantee the global convergence, we sometimes require \( d_k \) to satisfy a **sufficient** descent condition:

\[ g_{k+1}^T d_{k+1} \leq -c\|g_{k+1}\|^2 \] ...........(6)

where \( c \) is a constant \([7]\). In line search methods, the well-known CG-method has the form \((2)\) in which

\[ d_{k+1} = \begin{cases} -g_0 & \text{if } k = 0 \\ -g_{k+1} + \beta_k d_k & \text{if } k > 0 \end{cases} \] ...........(7)

where

\[ \beta_k = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} \] ...........(8)

This method is called FRCG-method \([5]\). The above mentioned CG-method is equivalent to each other for minimizing strong convex quadratic functions under exact line searches; they have different performance when using them to minimize non-quadratic functions or using inexact line searches. For non-quadratic objective functions, the FRCG method has a global convergence property when exact line searches are used or Strong Wolfe line search \([2, 3]\) is used.

The structure of the paper is as follows. In section \(2\) we modify the standard FRCG-method and show that the search direction generated by this proposed FRCG-method at each iteration satisfies the sufficient descent condition. Section \(3\) establishes the global convergence property for the new class of CG-methods with \( |\beta_k| \leq \beta_k^{FR} \). Section \(4\) establishes some numerical results to show the effectiveness of the proposed CG-method and Section \(5\) gives brief conclusions and discussions.

2. **Modified Conjugate Gradient Method.**

In this section, we propose a modified FRCG-method in which the parameter \( \beta_k \) is defined on the basis of \( \beta_k^{FR} \) as follows:

\[ \beta_k^{MFR} = \frac{\|g_{k+1}\|^2}{\|g_k\|^2} - Min \left\{ \frac{\|g_{k+1}\|^2}{\|g_k\|^2}, \frac{\mu\|g_{k+1}\|}{\|g_k\|}, \frac{\mu\|g_{k+1}\|}{\|g_k\|} \right\} \] ...........(9)
where \( \mu \) is a parameter. Now we present the outline of the new proposed method as follows:

**2.1 Outline of the New Algorithm:**

**Step 0**: Given \( x_0 \in \mathbb{R}^n, c = 1 \times 10^{-4} \), \( \delta_1 \in (0,1) \), \( \delta_2 \in (0,1/2) \), \( d_0 = -g_0 \)

**Step 1**: Computing \( g_k \); if \( \| g_k \| \leq e \) then stop; else continue.

**Step 2**: Set \( \beta = \beta_k^{MPF} = \| g_{k+1} \|^2 - \min \{ \| g_k \|^2, \mu \| g_k \|^2 (g_k^T d_k) \} \),

**Step 3**: Set \( x_{k+1} = x_k + \alpha_k d_k \), (Use SW-condition to compute \( \alpha_k \))

**Step 4**: Compute \( d_{k+1} = -g_{k+1} + \beta_k d_k \),

**Step 5**: If \( k = n \) go to **Step 1** with new values of \( x_{k+1} \) and \( g_{k+1} \), if not, set \( k = k+1 \) and continue.

**Theorem (2.2)**

Consider any iterative CG-method of the form (2) and (7), where \( \beta = \beta_k^{MPF} \) . If \( g_k \neq 0 \) for all \( k \geq 1 \), then:

\[
 g_{k+1}^T d_{k+1} \leq -c \| g_{k+1} \|^2 < 0 . \tag{10}
\]

**Proof.**

Firstly, for \( k = 0 \), it is easy to see that (10) is true since \( d_0 = -g_0 \).

Secondly, assume that:

\[
 g_k^T d_k \leq -c \| g_k \|^2 < 0 \text{ where } 0 < c < 1 \tag{11}
\]

holds for \( k \) when \( k \geq 1 \). Multiplying (7) by \( g_{k+1}^T \), we have

\[
 g_k^T d_{k+1} = -\| g_{k+1} \|^2 + \left[ \| g_{k+1} \|^2 - \min \left\{ \| g_k \|^2, \mu \| g_k \|^2 (g_k^T d_k) \right\} \right] g_k^T d_k \tag{12}
\]

If

\[
 \frac{\| g_{k+1} \|^2}{\| g_k \|^2} < \frac{\mu \| g_{k+1} \|^2 (g_k^T d_k)}{\| g_k \|^4} \tag{13}
\]

then

\[
 g_k^T d_{k+1} = -\| g_{k+1} \|^2 . \tag{13}
\]

If

\[
 \frac{\| g_{k+1} \|^2}{\| g_k \|^2} > \frac{\mu \| g_{k+1} \|^2 (g_k^T d_k)}{\| g_k \|^4} \tag{14}
\]

then

\[
 g_k^T d_{k+1} = -\| g_{k+1} \|^2 + \left[ \| g_{k+1} \|^2 - \frac{\mu \| g_{k+1} \|^2 (g_k^T d_k)}{\| g_k \|^4} \right] g_k^T d_k
\]

\[
 = -\| g_{k+1} \|^2 \left[ 1 - \frac{\| g_{k+1} \|^2}{\| g_k \|^2} \right] + \frac{\mu \| g_{k+1} \|^2 (g_k^T d_k)}{\| g_k \|^4} \tag{14}
\]

\[
 \leq -\| g_{k+1} \|^2 \left[ 1 - \frac{\| g_{k+1} \|^2}{\| g_k \|^2} \right] + \frac{\mu \| g_{k+1} \|^2 (g_k^T d_k)}{\| g_k \|^4} \]

from (4) and (11), we get:
3. Global convergence

In this section, we come to study the global convergence property of the new proposed Algorithm (2.1). For this, we are going to verify that Algorithm (2.1) is well defined. For the proof of the global convergence property, the following Assumption is needed.

**Assumption (3.1)**

i- The level set \( L = \{ x \in \mathbb{R}^n | f(x) \leq f(x_0) \} \) is bounded.

ii- In some neighborhood \( U \) and \( L \), \( f(x) \) is continuously differentiable and its gradient is Lipschitz continuous, namely, there exists a constant \( \mu > 0 \) such that:

\[
\|g(x_{k+1}) - g(x_k)\| \leq \mu \|x_{k+1} - x_k\| \quad \forall x_{k+1}, x_k \in U .
\]

We will see that it is possible to obtain the global convergence property if the parameter \( \beta_k \) is appropriately bounded in magnitude. We consider a method of the form (2) and (7), where \( \beta_k \) is any scalar such that:

\[
|\beta_k| \leq z\beta_k^{FR} , \quad z > 1
\]

for all \( k \geq 2 \), and where the step length satisfies the strong Wolfe conditions (3) – (4). Note that Zoutendijk's condition holds in this case. We show that any method of the form (2) and (7) is globally convergent if \( \beta_k \) satisfies (17). For the details of this theorem see [9].

**Theorem (3.2)**

Suppose that Assumption (3.1) holds. Let \( \{g_k\} \) and \( \{d_k\} \) be generated by Algorithm (2.1), then we have:

\[
\sum_{k=1}^{\infty} \left( g_k^T d_k \right)^2 < \infty .
\]

**Proof:**

From Theorem (2.2) we have \( g_k^T d_k < 0 \) for all \( k+1 \). We also have from (4) and Assumption (3.1, ii) that:

\[
-(1 - \delta_2) d_k^T g_k \leq (g_{k+1} - g_k)^T d_k \leq \mu \alpha_k \|d_k\|^2 .
\]

Thus:

\[
\alpha_k \geq \frac{1 - \delta_2}{\mu} g_k^T d_k ,
\]

which combining (4), we get:
\[ f(x_k) - f(x_{k+1}) \geq -\delta_1 \alpha_k g_k^T d_k \geq \delta_1 \cdot \frac{1 - \delta_2 (g_k^T d_k)^2}{\mu \|d_k\|^2}. \] ..........(21)

Further, from Assumption (3.1, i) we have from [6] \{f(x_i)\} is a decreasing sequence and has a bound below in \( L \), and shows \( \lim_{k \to \infty} f(x_{k+1}) < \infty \), this shows:

\[ \infty > f(x_i) - \lim_{k \to \infty} f(x_{k+1}) = \sum_{k=1}^{\infty} f(x_k) - f(x_{k+1}) \geq \delta_1 \frac{1 - \delta_2 \sum_{k=1}^{\infty} (g_k^T d_k)^2}{\mu \|d_k\|^2}, \] ..........(22)

We can conclude that (18) holds.

**Theorem (3.3)**

Suppose that Assumptions (3.1) holds. Consider any method of the form (2) and (7), where \( \beta_k \) satisfies (12), and where the step length satisfies the strong Wolfe conditions (3) – (4) with \( 0 < \delta_1 < \delta_2 < 1 \) then:

\[ \lim \inf_{k \to \infty} \|g_k\| = 0. \] ..........(23)

**Proof:** See [9].

It is natural to ask if the bound \( |\beta_k| \leq \beta_k^{FR} \) can be replaced by

\[ |\beta_k| \leq c_2 \beta_k^{FR} \] ..........(24)

where \( c_2 > 1 \) is some suitable constant.

This theorem suggests the following globally convergent modification of the FR method. Applying relation (24) of the parameter (9) we get:

\[ \beta_{MFR} \leq \frac{\|g_{k+1}\|^2}{\|g_k\|^2} + \frac{\mu \|g_{k+1}\|^2}{\|g_k\|^2} |g_k^T d_k| \] ..........(25)

from (4) and (10) we get:

\[ \beta_{MFR} \leq \frac{\|g_{k+1}\|^2}{\|g_k\|^2} + \frac{\mu \|g_{k+1}\|^2}{\|g_k\|^2} \delta_2 c \|g_k\|^2 \]

\[ \leq \frac{\|g_{k+1}\|^2}{\|g_k\|^2} [1 + \mu \delta_2 c] \] ..........(26)

\[ \leq c_3 \beta_k^{FR} \]

4. **Numerical Results**

In this section, we have reported some numerical results obtained with the implementation of the new Algorithm (2.1) on a set of unconstrained optimization test problems. We have selected (10) large scale unconstrained optimization problems in extended or generalized form, for each test function we have considered numerical experiment with the number of variable \( n=100-1000 \). Using the strong Wolfe line search condition (3) and (4) with \( \delta_1 = 0.0001 \) and \( \delta_2 = 0.9 \) In all these
cases, the stopping criteria is the $\|g_k\| \leq 10^{-4}$. The programs are written in Fortran 90. The test functions are commonly used for unconstrained test problems with standard starting points and a summary of the results of these test functions was given in Table (4.1). We tabulate for comparison of these algorithms, the Number Of Function evaluations (NOF) and the Number Of Iterations (NOI).

**Table (4-1)**

<table>
<thead>
<tr>
<th>No.</th>
<th>n</th>
<th>New Algorithm (2.1)</th>
<th>FR Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NOF (NOI)</td>
<td>NOF (NOI)</td>
</tr>
<tr>
<td>Powell</td>
<td>100</td>
<td>208 (102)</td>
<td>209 (102)</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>662 (329)</td>
<td>679 (337)</td>
</tr>
<tr>
<td>Wood</td>
<td>100</td>
<td>548 (218)</td>
<td>864 (319)</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>2101 (1003)</td>
<td>7739 (2004)</td>
</tr>
<tr>
<td>Miele</td>
<td>100</td>
<td>224 (101)</td>
<td>212 (101)</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>746 (332)</td>
<td>804 (372)</td>
</tr>
<tr>
<td>Cantrel</td>
<td>100</td>
<td>151 (27)</td>
<td>152 (27)</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>163 (28)</td>
<td>164 (28)</td>
</tr>
<tr>
<td>Rosen</td>
<td>100</td>
<td>269 (103)</td>
<td>297 (103)</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>380 (159)</td>
<td>376 (143)</td>
</tr>
<tr>
<td>Wolfe</td>
<td>100</td>
<td>99 (49)</td>
<td>99 (49)</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>259 (129)</td>
<td>279 (139)</td>
</tr>
<tr>
<td>Sum</td>
<td>100</td>
<td>68 (14)</td>
<td>68 (14)</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>119 (27)</td>
<td>121 (27)</td>
</tr>
<tr>
<td>Penalty 2</td>
<td>100</td>
<td>207 (101)</td>
<td>207 (101)</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>421 (208)</td>
<td>463 (229)</td>
</tr>
<tr>
<td>Beale</td>
<td>100</td>
<td>75 (37)</td>
<td>75 (37)</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>85 (37)</td>
<td>85 (37)</td>
</tr>
</tbody>
</table>
5. Conclusions and Discussions.

In this paper, we have proposed a modified CG method for solving unconstrained minimization problems. The computational experiments show that the new approaches given in this paper are successful.

Table (4.1) gives a comparison between the new-algorithm and the Fletcher-Reeves (FR) algorithm for convex optimization, this table indicates, see Table (4.2), that the new algorithm saves $(52.38\%)$ NOI and $(68.98\%)$ NOF, overall against the standard Fletcher-Reeves (FR) algorithm, especially for our selected group of test problems.

Table (4.2): Relative efficiency of the new Algorithm (2.1)

<table>
<thead>
<tr>
<th>Tools</th>
<th>NOI</th>
<th>NOF</th>
</tr>
</thead>
<tbody>
<tr>
<td>FR Algorithm</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Algorithm (2.1)</td>
<td>47.62%</td>
<td>31.02%</td>
</tr>
</tbody>
</table>

Appendix.

1. Generalized powell function:

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

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

2. Generalized d wood function:

\[
f(x) = \sum_{i=1}^{n} 4(x_{4i-2} - x_{4i-3})^2 + (1 - x_{4i-3})^2 + 90(x_{4i} - x_{4i-1})^2 + (1 - x_{4i-1})^2 + 10.1((x_{4i-2} - 1)^2 + (x_{4i} - 1)^2 + 19.8((x_{4i-2} - 1) + (x_{4i} - 1))
\]

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

3. Miele function:

\[
f(x) = \sum_{i=1}^{n} [\exp(x_{4i-3}) - x_{4i-2}]^2 + 100(x_{4i-2} - x_{4i-1})^6 + [\tan(x_{4i-1} - x_{4i})]^4 + x_{4i-3}^8 + (x_i - 1)^2
\]

Starting po int : $(1, 2, 2, 2, ...............)\,^T$

4. Cantrell function:

\[
f(x) = \sum_{i=1}^{n} [\exp(x_{4i-3}) - x_{4i-2}]^4 + 100(x_{4i-2} - x_{4i-1})^6 + [\tan^{-1}(x_{4i-1} - x_{4i})]^3 + x_{4i-3}^8
\]

Starting po int : $(1, 2, 2, 2, ............ ...)\,^T$

<table>
<thead>
<tr>
<th>Helical</th>
<th>100</th>
<th>(42)</th>
<th>(42)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>214</td>
<td>234</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(105)</td>
<td>(115)</td>
</tr>
<tr>
<td>1000</td>
<td></td>
<td>466</td>
<td>1124</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(231)</td>
<td>(560)</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>7465</td>
<td>14251</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3345)</td>
<td>(4849)</td>
</tr>
</tbody>
</table>
5. Rosenbrock function:

\[ f(x) = \sum_{i=1}^{n/2} \left( 100 \left( x_{2i} - x_{2i-1}^2 \right)^2 + \left( 1 - x_{2i-1} \right)^2 \right) \]

Starting point: \((-1.2,1,-1.2,1,\ldots)^T\)

6. Welfe function:

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

Starting point: \((-1, \ldots)^T\)

7. Sum of Quartics function:

\[ f(x) = \sum_{i=1}^{n} (x_i - 1)^4 \]

Starting point: \((2, \ldots)^T\)

8. Penalty 2 function:

\[ f(x) = \sum_{i=1}^{d} e^{-(x(i)-1)^2} + (x(i)^2 - 0.25)^2 \]

Starting point: \((1,2, \ldots)^T\)

9. Beale function:

\[ f(x) = (1.5 - x_1(1 - x_2))^2 + (2.25 - x_1(1 - x_2)^2) + (2.652 - x_1(1 - x_2)^2) \]

Starting point: \((0,0, \ldots)^T\)

10. Helical valley function:

\[ f(x) = 100((x_3 - 10\theta)^2 + (r - 1)^2) + x_3^2 \]  
where \( \theta = \begin{cases} (2\pi)^{-1} \tan(x_2/x_1) & \text{for } x_1 > 0 \\ 0.5 + (2\pi)^{-1} \tan(x_2/x_1) & \text{for } x_1 < 0 \end{cases} \)

\[ r = (x_1^2 + x_2^2)^{1/2} \]

Starting point: \((-1,0,0, \ldots)^T\)
References


