A New Parameter Conjugate Gradient Method Based on Three Terms Unconstrained Optimization

Hussein Ageel Khatab 1 and Salah Gazi Shareef 2

1,2 Department of Mathematics, Faculty of Science, University of Zakho, Kurdistan Region, Iraq

1 husseinkhatab632@gmail.com, 2 salah.gazi2014@gmail.com

Abstract. In this paper, we suggest a new conjugate gradient method for solving nonlinear unconstrained optimization problems by using three term conjugate gradient method, We give a descent condition and the sufficient descent condition of the suggested method.

Keywords: Unconstrained Optimization; Conjugate Gradient Method; Descent Condition; Three Term Conjugate Gradient Method

1. Introduction

We consider the unconstrained optimization problem:

\[
\text{Min } f(x), x \in \mathbb{R}^n
\]  

where \( f: \mathbb{R}^n \rightarrow \mathbb{R} \) is a real-valued, continuously differentiable function.

A nonlinear conjugate gradient method generates a sequence \( \{x_k\} \), \( k \geq 1 \), starting from an initial guess \( x_1 \in \mathbb{R}^n \), using the recurrence

\[
x_{k+1} = x_k + \alpha_k d_k
\]  

Or \( v_k = \alpha_k d_k \)

Where, \( v_k = x_{k+1} - x_k \)

The positive step size \( \alpha_k \) is obtained by some line search, and \( d_k \) is a search direction, normally the search direction at the first iteration is the steepest descent direction, namely \( \alpha_1 = -g_1 \) and the other search directions can be defined as:

\[
d_{k+1} = -g_{k+1} + \beta_k d_k
\]  

Where \( g_k = \nabla f(x_k) \) and \( \beta_k \) is a scalar. There are many formulas for \( \beta_k \), for example, Hestenes-Stiefel (HS) [8], Conjugate descent (CD) [10], Polak-Ribiére-Polyak (PRP) [1], Dai and Yuan (DY) [11], (DPRP) [2], RMIL [9] and \( \beta_k^1 \) [12], these formulas are as follows:

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

(1.4)

\[
\beta_k^{CD} = -\frac{1}{d_k^T g_k}
\]  

(1.5)

\[
\beta_k^{PRP} = \frac{g_k^T (g_{k+1} - g_k)}{\|g_k\|^2}
\]  

(1.6)

\[
\beta_k^{DY} = \frac{\|g_k\|^2}{d_k^T (g_{k+1} - g_k)}
\]  

(1.7)

\[
\beta_k^{DPRP} = \beta_k^{PRP} - t \frac{g_k^T d_k \|y_k\|}{\|g_k\|^4}
\]  

(1.8)

where, \( t > 1/4 \)

\[
\beta_k^{RMIL} = \frac{g_k^T (g_{k+1} - g_k)}{\|d_k\|^2}
\]  

(1.9)

\[
\beta_k^1 = \frac{g_k^T y_k}{d_k^T y_k} - t \frac{g_k^T d_k}{\|d_k\|^2}
\]  

(1.10)

where, \( t > 0 \)
\[ y_k = \delta_k + \beta k s_k + \delta_k y_k \]

\[ d_{k+1} = -g_{k+1} - \beta_k s_k + \delta_k y_k \] (2.1)

where \( \beta_k = \frac{g_{k+1}^T s_k}{g_k^T s_k} \) and \( \delta_k = \frac{g_{k+1}^T y_k}{y_k^T s_k} \), see[4]. Also, There are many three term conjugate gradient algorithm suggested, for instance:

\[ d_{k+1} = -g_{k+1} - \beta_k s_k - \frac{g_{k+1}^T d_k}{g_k^T s_k} y_k \] (2.2)

\[ d_{k+1} = -g_{k+1} - \beta_k y_k - \frac{g_{k+1}^T d_k}{g_k^T s_k} y_k \] (2.3)

We suggest a new three term conjugate gradient as following:

\[ d_{k+1} = -g_{k+1} - \beta_0^N ew s_k - \delta k y_k \] (2.4)

where, \( \delta \in (0,1) \) and \( \mu = \frac{g_{k+1}^T d_k}{g_k^T s_k} \).

Now, from equation (2.3) and equation (2.4), we have

\[ \frac{g_{k+1}^T y_k}{g_k^T s_k} d_k - \frac{g_{k+1}^T d_k}{g_k^T s_k} y_k = \beta_0^N ew d_k - \delta k \frac{g_{k+1}^T d_k}{g_k^T s_k} y_k \]

Multiplying both sides of above equation by \( d_k \), we obtain

\[ \frac{g_{k+1}^T y_k}{g_k^T s_k} d_k d_k - \frac{g_{k+1}^T d_k}{g_k^T s_k} d_k y_k = \beta_0^N ew d_k^T d_k - \delta k \frac{g_{k+1}^T d_k}{g_k^T s_k} d_k^T d_k y_k \]

This implies that

\[ \frac{g_{k+1}^T y_k}{g_k^T s_k} d_k d_k - \frac{g_{k+1}^T d_k}{g_k^T s_k} d_k y_k = \beta_0^N ew \frac{g_{k+1}^T d_k}{g_k^T s_k} d_k - \delta k \frac{g_{k+1}^T d_k}{g_k^T s_k} d_k^T d_k y_k \]

So,

\[ \beta_0^N ew = \beta_0^R PR + (\delta - 1) \frac{g_{k+1}^T d_k y_k}{d_k^T d_k g_k^T s_k} \] (2.5)

where, \( 0 < \delta < 1 \).

2.1 Algorithm of a new conjugate gradient method(\( \beta_0^N ew \))

Step(1): Select \( x_1 \) and \( \epsilon = 10^{-5} \).

Step(2): Set \( d_1 = -g_1 \), \( g_k = \frac{\nablaf(x_k)}{g_k} \), set \( k = 1 \).

Step(3): Compute the step length \( \alpha_k > 0 \) satisfying the Wolfe line search

\[ f(x_k + \alpha_k d_k) - f(x_k) \leq c_1 \alpha_k g_k^T d_k \]

\[ \alpha_k d_k \leq c_2 |g_k^T d_k| \]

where, \( 0 < \alpha_k < c_2 < 1 \).

Step(4): Compute

\[ x_{k+1} = x_k + \alpha_k d_k \]

\[ g_{k+1} = \frac{\nablaf(x_{k+1})}{g_k^T d_k} \] if \( g_{k+1} \leq \epsilon \), then stop.

Step(5): Compute \( \beta_0^N ew \) by (2.5)

Step(6): Compute \( d_{k+1} = -g_{k+1} - \beta_0^N ew d_k \)

Step(7): If \( g_{k+1} \leq 0.2 \) then go to step 2.

Else
Let \( k = k + 1 \) and go to step 3.

**Theorem 2.1:** Assume that the sequence \( \{x_k\} \) is generated by (1.2), then the search direction in (1.3) with new conjugate gradient method (2.5) satisfy the descent condition, i.e. \( d_{k+1}^T g_{k+1} \leq 0 \) with exact and inexact line search.

**Proof:** From (1.3) and (2.5) we have,
\[
d_{k+1} = -g_{k+1} + \left( \frac{\nabla^T y_k}{\|g_k\|^2} + (\delta - 1) \frac{\nabla^T y_k}{d_k^T d_k \|g_k\|^2} \right) d_k
\] (2.6)

Multiply both sides of above equation by \( g_{k+1} \), to obtain
\[
d_{k+1}^T g_{k+1} = -\|g_{k+1}\|^2 + \frac{\nabla^T y_k}{\|g_k\|^2} d_k^T g_{k+1} + (\delta - 1) \frac{(\nabla^T y_k)}{d_k^T d_k \|g_k\|^2} \|g_{k+1}\|^2 \leq 0.
\] (2.7)

If the step length \( \alpha_k \) is chosen by an exact line search which requires \( d_k^T g_k \neq 0 \). Then the proof is complete. If the step length \( \alpha_k \) is chosen by inexact line search which requires \( d_k^T g_k \neq 0 \)

The first two terms of equation (2.7) are less than or equal to zero because the parameter of (PRP) satisfies the descent condition, and the third term clearly is less than or equal to zero since \( d_k^T y_k > 0 \), so,
\[
d_{k+1}^T g_{k+1} = -\|g_{k+1}\|^2 + \frac{\nabla^T y_k}{\|g_k\|^2} d_k^T g_{k+1} + (\delta - 1) \frac{(\nabla^T y_k)}{d_k^T d_k \|g_k\|^2} \|g_{k+1}\|^2 \leq 0.
\]

**Theorem 2.2:** Assume that the sequence \( \{x_k\} \) is generated by (1.2), then the search direction in (1.3) with new conjugate gradient method (2.5) satisfy the sufficient descent condition, i.e. \( d_{k+1}^T g_{k+1} \leq -C \|g_{k+1}\|^2 \).

**Proof:** From (1.3) and (2.5) we have
\[
d_{k+1} = -g_{k+1} + \left( \frac{\nabla^T y_k}{\|g_k\|^2} + (\delta - 1) \frac{\nabla^T y_k}{d_k^T d_k \|g_k\|^2} \right) d_k
\] (2.8)

Multiply both sides of above equation by \( g_{k+1} \), to obtain
\[
d_{k+1}^T g_{k+1} = -\|g_{k+1}\|^2 + \frac{\nabla^T y_k}{\|g_k\|^2} d_k^T g_{k+1} + (\delta - 1) \frac{(\nabla^T y_k)}{d_k^T d_k \|g_k\|^2} \|g_{k+1}\|^2 \leq 0.
\] (2.9)

Now, since the parameter of (PRP) satisfies the descent condition, then the above equation becomes
\[
d_{k+1}^T g_{k+1} = (\delta - 1) \frac{(\nabla^T y_k)}{d_k^T d_k \|g_k\|^2} \|g_{k+1}\|^2 \leq 0
\]
\[
= (\delta - 1) \frac{(\nabla^T y_k)}{d_k^T d_k \|g_k\|^2} \|g_{k+1}\|^2 \|g_{k+1}\|^2
\]
\[
= \|g_{k+1}\|^2 \|g_{k+1}\|^2 \leq 0.
\]

Let \( C = (1 - \delta) \frac{(\nabla^T y_k)}{d_k^T d_k \|g_k\|^2} \) which is positive, then
\[
d_{k+1}^T g_{k+1} \leq -C \|g_{k+1}\|^2.
\] (2.10)

**3. Numerical Results**

This section is devoted to test the implementation of new method. We compare our method with Conjugate Gradient methods (PRP), (HS), RMIL and \( \beta_k \) the comparative tests involve Well-known nonlinear problems (standard test function) with different dimensions \( 4 \leq n \leq 5000 \), all programs are written in FORTRAN90 language and for all cases the stopping condition is \( \|g_{k+1}\| \leq 10^{-5} \), the results given in table (1) and table (2) specifically quote the number of function NOF and the number of iteration NOI. More experimental results in table (1) and table (2) confirm that the new CG is superior to standard CG (PRP), standard CG (HS), RMIL and \( \beta_k \) with respect to the NOI and NOF.
**Table (1): Comparative Performance of the Three Algorithms (PRP, HS and New Conjugate Gradient Method)**

<table>
<thead>
<tr>
<th>Test function</th>
<th>N</th>
<th>Algorithm of PRP</th>
<th>Algorithm of HS</th>
<th>New algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cubic</td>
<td>4</td>
<td>15</td>
<td>45</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>16</td>
<td>47</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>16</td>
<td>47</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>16</td>
<td>47</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>16</td>
<td>47</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>16</td>
<td>47</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>16</td>
<td>47</td>
<td>13</td>
</tr>
<tr>
<td>Extended PSC1</td>
<td>4</td>
<td>7</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>6</td>
<td>16</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>6</td>
<td>16</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>7</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>7</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>7</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>7</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td>GCentral</td>
<td>4</td>
<td>22</td>
<td>159</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>22</td>
<td>159</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>22</td>
<td>159</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>22</td>
<td>159</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>23</td>
<td>171</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>23</td>
<td>171</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>30</td>
<td>270</td>
<td>28</td>
</tr>
<tr>
<td>Miele</td>
<td>4</td>
<td>37</td>
<td>116</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>37</td>
<td>116</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>44</td>
<td>148</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>44</td>
<td>148</td>
<td>33</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>44</td>
<td>148</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>50</td>
<td>180</td>
<td>46</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>50</td>
<td>180</td>
<td>54</td>
</tr>
<tr>
<td>OSP</td>
<td>4</td>
<td>8</td>
<td>45</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>13</td>
<td>57</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>37</td>
<td>147</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>49</td>
<td>176</td>
<td>49</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>110</td>
<td>341</td>
<td>112</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>161</td>
<td>493</td>
<td>156</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>276</td>
<td>844</td>
<td>256</td>
</tr>
<tr>
<td>Powell</td>
<td>4</td>
<td>35</td>
<td>87</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>35</td>
<td>87</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>35</td>
<td>87</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>35</td>
<td>87</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>35</td>
<td>87</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>35</td>
<td>87</td>
<td>41</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>35</td>
<td>87</td>
<td>41</td>
</tr>
<tr>
<td>Wood</td>
<td>4</td>
<td>29</td>
<td>67</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>29</td>
<td>67</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>29</td>
<td>67</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>30</td>
<td>69</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>30</td>
<td>69</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>30</td>
<td>69</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>30</td>
<td>69</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>1734</td>
<td>5922</td>
<td>1679</td>
</tr>
</tbody>
</table>

**Table (2): Comparative Performance of the two Algorithms (RMIL, $\beta^1_k$ and New Conjugate Gradient Method)**

<table>
<thead>
<tr>
<th>Test function</th>
<th>N</th>
<th>Algorithm of RMIL</th>
<th>$\beta^1_k$</th>
<th>New algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cubic</td>
<td>4</td>
<td>16</td>
<td>47</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>16</td>
<td>47</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>16</td>
<td>47</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>16</td>
<td>47</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>16</td>
<td>47</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>16</td>
<td>47</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>5000</td>
<td>16</td>
<td>47</td>
<td>13</td>
</tr>
<tr>
<td>Extended PSC1</td>
<td>4</td>
<td>7</td>
<td>18</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>6</td>
<td>16</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>6</td>
<td>16</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>7</td>
<td>18</td>
<td>7</td>
</tr>
</tbody>
</table>
500  7  18  7  18  6  16
1000  7  18  7  18  6  16
5000  7  18  7  18  6  16

**GCentral**
4  33  197  22  159  21  147
10  34  208  22  159  21  147
50  39  265  22  159  22  159
100  43  315  22  159  22  159
500  48  380  23  171  22  159
1000  51  420  23  171  22  159
5000  56  489  28  248  23  173

**Miele**
4  52  164  33  104  29  88
10  59  194  33  104  29  88
50  67  229  33  104  35  119
100  79  273  33  104  35  119
500  90  317  33  104  36  121
1000  90  317  33  104  36  121
5000  106  395  33  104  43  159

**OSP**
4  8  45  8  44  8  44
10  13  57  13  58  13  58
50  39  152  34  134  32  113
100  60  210  49  180  47  166
500  236  745  107  328  109  333
1000  471  F  146  448  156  480
5000  F  F  257  763  275  833

**Powell**
4  F  F  38  108  29  74
10  F  F  38  108  29  74
50  F  F  40  122  29  74
100  F  F  40  122  29  74
500  F  F  40  122  29  74
1000  F  F  40  122  32  92
5000  F  F  40  122  32  92

**Wood**
4  96  199  30  68  27  63
10  101  209  30  68  27  63
50  103  213  30  68  28  65
100  118  243  30  68  28  65
500  128  263  30  68  28  65
1000  128  263  30  68  28  65
1000  148  303  30  68  28  65

**Total**
6655  12016  1630  5580  1569  5322

**Notes:**
1- $F > \frac{1000}{2}$.
2- We took $F = 500$ for summation.

**Table (3): Percentage of Improving of the New Method**

<table>
<thead>
<tr>
<th>Algorithm of PRP</th>
<th>New Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOI 100%</td>
<td>90.48442907657%</td>
</tr>
<tr>
<td>NOF 100%</td>
<td>89.8682877406%</td>
</tr>
<tr>
<td>Algorithm of HS New Algorithm</td>
<td></td>
</tr>
<tr>
<td>NOI 100%</td>
<td>93.4448412388%</td>
</tr>
<tr>
<td>NOF 100%</td>
<td>90.881147541%</td>
</tr>
<tr>
<td>Algorithm of RMI New Algorithm</td>
<td></td>
</tr>
<tr>
<td>NOI 100%</td>
<td>23.5762504523%</td>
</tr>
<tr>
<td>NOF 100%</td>
<td>44.290454061%</td>
</tr>
<tr>
<td>$\beta_1^* $ New Algorithm</td>
<td></td>
</tr>
<tr>
<td>NOI 100%</td>
<td>96.2576687117%</td>
</tr>
<tr>
<td>NOF 100%</td>
<td>95.376344086%</td>
</tr>
</tbody>
</table>

**4. Conclusion**

In this paper, we suggested a new conjugate gradient method for unconstrained optimization. Implemented and tested to some extent, while numerical tests were carried out on low and high dimensionality problems, and comparisons were made amongst different test functions with inexact line search. Some of the numerical results have been reported.
A New Parameter Conjugate Gradient Method Based on Three Terms

References:


