New Homotopy Conjugate Gradient for Unconstrained Optimization using Hestenes-Stiefel and Conjugate Descent

Salah Gazi Sharee, Hussein Ageel Khatab
Dept. of Mathematics, Faculty of Science, University of Zakho, Kurdistan Region-Iraq

Abstract: - In this paper, we suggest a hybrid conjugate gradient method for unconstrained optimization by using homotopy formula. We calculate the parameter \( \beta_k \) as a convex combination of \( \beta^{HS} \) (Hestenes Stiefel)[5] and \( \beta^{CD} \) (Conjugate descent)[3].

Keywords: - Unconstrained optimization, line search, conjugate gradient method, homotopy formula.

I. INTRODUCTION

Suppose that \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) is continuously differentiable function whose gradient is denoted by \( g(x) \). Consider the nonlinear following unconstrained optimization problem

\[
\min f(x) \quad x \in \mathbb{R}^n
\]

The iterates for solving (1.1)are given by

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

where \( \alpha_k \) is a positive size obtained by line search and \( d_k \) is a search direction. The search direction at the very first iteration is the steepest descent \( d_0 = -g_0 \), the directions along the iterations are computed according to:

\[
d_{k+1} = -g_{k+1} + \beta_k d_k , \quad k \geq 0
\]

where \( \beta_k \in R \) is known as conjugate gradient coefficient and different \( \beta_k \) will yield different conjugate gradient methods. Some well known formulas are given as follows:

\[
\beta_k^{PR} = \frac{g_k^T (g_{k-1} - g_k)}{|g_{k-1}|^2} \quad (1.4)
\]

\[
\beta_k^{FR} = \frac{g_{k+1}^T g_{k+1}}{|g_k|^2} \quad (1.5)
\]

\[
\beta_k^{HS} = \frac{\beta_k^{PR} g_{k+1} - g_k}{(g_{k+1} - g_k)^T d_k} \quad (1.6)
\]

\[
\beta_k^{DY} = \frac{\beta_k^{PR} g_{k+1} - g_k}{g_{k+1} - g_k} d_k \quad (1.7)
\]

\[
\beta_k^{CD} = \frac{\beta_k^{PR} g_{k+1} - g_k}{-d_k^T g_k} \quad (1.8)
\]

\[
\beta_k^{LS} = \frac{\beta_k^{PR} g_{k+1} - g_k}{d_k^T d_k} \quad (1.9)
\]

Where \( g_{k+1} \) and \( g_k \) are gradients \( \nabla f(x_{k+1}) \) and \( \nabla f(x_k) \) of \( \nabla f(x) \) at the point \( x_{k+1} \) and \( x_k \), respectively. \( \| \cdot \| \) denotes the Euclidian norm of vectors. The line search in the conjugate gradient algorithms often is based on the standard Wolfe conditions:

\[
f(x_k + \alpha_k d_k) - f(x_k) \leq \delta \alpha_k g_k^T d_k \quad (1.10)
\]

\[
g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k \quad (1.11)
\]

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

The strong Wolfe line search corresponds to:

\[
f(x_k + \alpha_k d_k) - f(x_k) \leq \delta \alpha_k g_k^T d_k \quad (1.12)
\]

\[
|g(x_k + \alpha_k d_k)^T d_k| \leq \sigma g_k^T d_k \quad (1.13)
\]

where \( 0 < \delta < \sigma < 1 \)[10]

II. HYBRID CONJUGATE GRADIENT ALGORITHMS

The hybrid conjugate gradient algorithms are combinations of different conjugate gradient algorithms. They are mainly purposed in order to avoid the jamming phenomenon.[1],these methods are an important class of conjugate gradient algorithms.[6],[9]
The methods of (FR)[4],(DY) [2] and (CD)[3] have strong convergence properties, but they may have modest practical performance due to jamming.

On the other hand, the methods of (PR)[8],(HS)[5] and (LS)[7] may not always be convergent, but the often have better computational performances.[1]

III. NEW HYBRID SUGGESTION

Our suggestion generates iterates $x_0, x_1, x_2, \ldots$ computed by means of the recurrence ($x_{k+1} = x_k + \alpha_k d_k$), where the steps $\alpha_k > 0$ is determined according to the Wolf line search condition (1.10) and (1.11), and the directions $d_k$ are generated by the rule:

$$d_{k+1} = -g_{k+1} + \beta_k^{\text{NEW}} d_k$$

Where

$$\beta_k^{\text{NEW}} = (1 - \theta) \beta_k^{\text{HS}} + \theta \beta_k^{\text{CD}}, \quad 0 \leq \theta \leq 1$$

$$\beta_k^{\text{NEW}} = (1 - \theta) \frac{(g^T_{k+1} y_k)}{d^T_k y_k} - \theta \frac{\|g_{k+1}\|^2}{d^T_k g_k}$$

Observe that, if $\theta = 0$, then $\beta_k^{\text{NEW}} = \beta^{\text{HS}}$, if $\theta = 1$, then $\beta_k^{\text{NEW}} = \beta^{\text{CD}}$.

On the other hand, if $0 < \theta < 1$, then we can find $\beta_k^{\text{NEW}}$ as follows:

We know that

$$d_{k+1} = -g_{k+1} + (1 - \theta) \frac{(g^T_{k+1} y_k)}{d^T_k y_k} d_k - \theta \frac{\|g_{k+1}\|^2}{d^T_k g_k} d_k$$

Our motivation is to choose the parameter $\theta_k$ in such a way so that the direction $d_{k+1}$ given (3.3) to be the Newton direction. Therefore

$$-\nabla^2 f(x_{k+1})^{-1} g_{k+1} = -g_{k+1} + (1 - \theta) \frac{(g^T_{k+1} y_k)}{d^T_k y_k} d_k - \theta \frac{\|g_{k+1}\|^2}{d^T_k g_k} d_k$$

Multiply both sides of above equation by $d_k^T \nabla^2 f(x_{k+1})$, we get

$$-d_k^T g_{k+1} = -d_k^T \nabla^2 f(x_{k+1}) g_{k+1} + (1 - \theta) d_k^T \nabla^2 f(x_{k+1}) \frac{(g^T_{k+1} y_k)}{d^T_k y_k} d_k$$

$$-d_k^T g_{k+1} = -d_k^T \nabla^2 f(x_{k+1}) g_{k+1} + (1 - \theta) \frac{(g^T_{k+1} y_k)}{d^T_k y_k} (d_k^T \nabla^2 f(x_{k+1}) d_k)$$

$$-\theta_k \frac{\|g_{k+1}\|^2}{d^T_k g_k} (d_k^T \nabla^2 f(x_{k+1}) d_k)$$

Since $d_k^T \nabla^2 f(x_{k+1}) = y_k$, then we have

$$-d_k^T g_{k+1} = -y_k^T g_{k+1} + (1 - \theta) (g^T_{k+1} y_k) - \theta_k \frac{\|g_{k+1}\|^2}{d^T_k g_k} y_k^T d_k$$

$$-d_k^T g_{k+1} = -\theta_k (g^T_{k+1} y_k) - \theta_k \frac{\|g_{k+1}\|^2}{d^T_k g_k} y_k^T d_k$$

Implies that

$$\theta_k = \frac{d_k^T g_{k+1}}{(g^T_{k+1} y_k) + \|g_{k+1}\|^2 (y_k^T d_k)}$$

Or

$$\theta_k = \frac{(d_k^T g_{k+1}) (d_k^T g_k)}{(g^T_{k+1} y_k) (d_k^T g_k) + \|g_{k+1}\|^2 (y_k^T d_k)}$$

3.1. Convergence of the new hybrid conjugate gradient algorithm

Theorem 3.1.1: Assume that $d_k$ is a descent direction and $\alpha_k$ in algorithm (1.2) and (3.2) where $\alpha_k$ is given by (2.4) is determined by the wolfe line search (1.10) and (1.11). If $0 < \theta < 1$, then the direction $d_{k+1}$ given by (3.3) is a descent direction.

Proof:
From (3.3) and (3.4) we have
\[
d_{k+1} = -g_{k+1} + \left(1 - \frac{(d_{k}^T g_{k+1}) (d_{k}^T g_{k})}{(g_{k+1}^T y_k)(d_{k}^T g_{k}) + \|g_{k+1}\|^2 (y_k^T d_k)} \right) \left(\frac{g_{k+1}^T y_k}{d_{k}^T g_{k}}\right) d_{k} - \left(\frac{(d_{k}^T g_{k+1}) (d_{k}^T g_{k})}{(g_{k+1}^T y_k)(d_{k}^T g_{k}) + \|g_{k+1}\|^2 (y_k^T d_k)} \right) \left(\frac{g_{k+1}^T y_k}{d_{k}^T g_{k}}\right) d_{k} \tag{3.1.1}\]

Multiply both sides of (3.1.1) by \(g_{k+1}^T d_{k}\), we get
\[
g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + (g_{k+1}^T y_k) (g_{k+1}^T d_{k}) \tag{3.1.2}\]

\[
g_{k+1}^T d_{k+1} = -\|g_{k+1}\|^2 + (g_{k+1}^T y_k) (g_{k+1}^T d_{k}) \tag{3.1.3}\]

The prove is complete if the step length \(\alpha_k\) is chosen by an exact line search which requires \(d_{k}^T g_{k+1} = 0\). Now, if the step length \(\alpha_k\) is chosen by an inexact line search which requires \(d_{k}^T g_{k+1} \neq 0\), we know that the first two terms of equation (3.1.3) are less than or equal to zero because the algorithm of Hestenes – Siefel (HS) satisfies the descent condition (i.e.)

\[
-\|g_{k+1}\|^2 + \frac{(g_{k+1}^T y_k)(g_{k+1}^T d_{k})}{d_{k}^T y_k} \leq 0, \tag{3.1.4}\]

It remains to consider the third and fourth terms

\[
= -\frac{(d_{k}^T g_{k+1})(d_{k}^T g_{k})}{(g_{k+1}^T y_k)(d_{k}^T g_{k}) + \|g_{k+1}\|^2 (y_k^T d_k)} \left(\frac{g_{k+1}^T y_k}{d_{k}^T g_{k}}\right) d_{k} \tag{3.1.5}\]

We know that \((d_{k}^T g_{k+1})^2\) is greater than or equal to zero and \(d_{k}^T y_k > 0\). Consequently, we have

\[
\frac{-(d_{k}^T g_{k+1})^2}{d_{k}^T y_k} \leq 0
\]

Implies that

\[
g_{k+1}^T d_{k+1} \leq 0.
\]

Then the proof is completed.

**Theorem 3.1.2** - Assume that the conditions in theorem (3.1.1) hold and \(\frac{(g_{k+1}^T y_k)(g_{k+1}^T d_{k})}{y_k^T d_k} \leq \|g_{k+1}\|^2\). If there exists a constant \(c_1 > 0\), such that \(g_{k+1}^T d_{k+1} \leq -c_1 \|g_{k+1}\|^2\), then the direction \(d_{k+1}\) satisfies the sufficient descent condition.

**Proof:**

From (3.3) and (3.4) we have
\[
d_{k+1} = -g_{k+1} + \left(1 - \frac{(d_{k}^T g_{k+1}) (d_{k}^T g_{k})}{(g_{k+1}^T y_k)(d_{k}^T g_{k}) + \|g_{k+1}\|^2 (y_k^T d_k)} \right) \left(\frac{g_{k+1}^T y_k}{d_{k}^T g_{k}}\right) d_{k} - \left(\frac{(d_{k}^T g_{k+1}) (d_{k}^T g_{k})}{(g_{k+1}^T y_k)(d_{k}^T g_{k}) + \|g_{k+1}\|^2 (y_k^T d_k)} \right) \left(\frac{g_{k+1}^T y_k}{d_{k}^T g_{k}}\right) d_{k} \tag{4.2.1}
\]
Multiply both sides of (4.2.1) by \( g_{k+1}^T \) we get

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

(3.1.6)

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

(3.1.7)

\[
g_{k+1}^T d_{k+1} \leq \frac{-d_k^T g_{k+1}}{d_k^T y_k}
\]

(3.1.8)

Multiply and divided right hand side of above inequality by \( y_k^T g_{k+1} \), we get

\[
g_{k+1}^T d_{k+1} \leq \frac{-d_k^T g_{k+1}}{y_k^T g_{k+1}} \|g_{k+1}\|^2
\]

(3.1.9)

By hypothesis , (3.1.9) gives

\[
g_{k+1}^T d_{k+1} \leq \frac{-d_k^T g_{k+1}}{y_k^T g_{k+1}} \|g_{k+1}\|^2
\]

Multiply and divided right hand side by \( y_k^T g_{k+1} \) we get

\[
g_{k+1}^T d_{k+1} \leq \frac{-d_k^T g_{k+1}}{(y_k^T g_{k+1})^2} \|g_{k+1}\|^2
\]

(3.1.10)

Now, if \( y_k^T g_{k+1} \neq 0 \),

Let \( c_1 = \frac{(d_k^T y_k)(y_k^T g_{k+1})}{(y_k^T g_{k+1})^2} \)

Then , (3.1.10) gives

\[
g_{k+1}^T d_{k+1} = -c_1 \|g_{k+1}\|^2
\]

If \( y_k^T g_{k+1} < 0 \) and we know that \( d_k^T y_k > 0 \), then,

\[
(d_k^T y_k)(y_k^T g_{k+1}) < d_k^T y_k
\]

Then , (3.1.10) gives

\[
g_{k+1}^T d_{k+1} \leq -\frac{d_k^T y_k}{(y_k^T g_{k+1})^2} \|g_{k+1}\|^2
\]

Let \( c_1 = \frac{d_k^T y_k}{(y_k^T g_{k+1})^2} \)

Hence

\[
g_{k+1}^T d_{k+1} \leq -c_1 \|g_{k+1}\|^2
\]

Then the proof is completed.

3.2 Theorem of global convergence

Since the new hybrid conjugate gradient algorithm is satisfies the sufficient descent condition by using wolfe conditions , then the new hybrid conjugate gradient algorithm is satisfies the global convergence property.
3.3 Algorithm of New Hybrid Conjugate Gradient algorithm

step (1) :- set k=0, select the initial point $x_k$.

step (2) :- $g_k = \nabla f(x_k)$, If $g_k = 0$, then stop.

else

set $d_k = -g_k$.

step (3) :- compute $\alpha_k > 0$ satisfying the Wolfe line search condition to minimize $f(x_{k+1})$.

step (4) :- $x_{k+1} = x_k + \alpha_k d_k$.

step (5) :- $g_{k+1} = \nabla f(x_{k+1})$, If $g_{k+1} = 0$, then stop.

step (6) :- compute $\theta_k$ as in (3.4).

step (7) :- if $0 < \theta_k < 1$, then compute $\beta_{k+1}^{NEW}$ as in (3.2). If $\theta_k \geq 1$, then set $\beta_{k+1}^{NEW} = \beta_{k+1}^{CD}$. If $\theta_k \leq 0$, then set $\beta_{k+1}^{NEW} = \beta_{k+1}^{HS}$.

step (8) :- $d_{k+1} = -g_{k+1} + \beta_{k+1}^{NEW} d_k$.

step (9) :- If $k=n$ then go to step 2, else $k=k+1$ and go to step 3.

3.4 NUMERICAL RESULTS:

This section is devoted to test the implementation of the new formula. We compare the hybrid algorithm with standard Hestenes – Stiefel (HS) and conjugate direction (CD). The comparative tests involve well-known nonlinear problems (standard test function) with different dimension $4 \leq n \leq 5000$, all programs are written in FORTRAN95 language and for all cases the stopping condition is $\|g_{k+1}\|_\infty \leq 10^{-5}$. The results are given in the below table is specifically quote the number of functions NOF and the number of iteration NOI. Experimental results in below table confirm that the new CG method is superior to standard CG method with respect to the NOI and NOF.

### Table: Comparative Performance of the three algorithms (Standard HS, CD and New formula)

<table>
<thead>
<tr>
<th>Test fun.</th>
<th>N</th>
<th>NOI</th>
<th>NOF</th>
<th>NOI</th>
<th>NOF</th>
<th>NOI</th>
<th>NOF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>HS</td>
<td>CD</td>
<td>NEW</td>
<td>CD</td>
<td>NEW</td>
<td>CD</td>
</tr>
<tr>
<td>Powell</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>65</td>
<td>170</td>
<td>3102</td>
<td>2077</td>
<td>30</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>276</td>
<td>3102</td>
<td>134002</td>
<td>6942</td>
<td>108</td>
<td>242</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>1062</td>
<td>*</td>
<td>*</td>
<td>270117</td>
<td>502</td>
<td>1011</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>1332</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>241</td>
<td>532</td>
</tr>
<tr>
<td>3000</td>
<td>1816</td>
<td>*</td>
<td>*</td>
<td>249</td>
<td>568</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>2074</td>
<td>*</td>
<td>*</td>
<td>409</td>
<td>913</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wood</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>59</td>
<td>353</td>
<td>709</td>
<td>26</td>
<td>61</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>61</td>
<td>928</td>
<td>2115</td>
<td>26</td>
<td>61</td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>63</td>
<td>1008</td>
<td>2277</td>
<td>27</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>63</td>
<td>561</td>
<td>1165</td>
<td>27</td>
<td>63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3000</td>
<td>63</td>
<td>500</td>
<td>1038</td>
<td>27</td>
<td>63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>63</td>
<td>517</td>
<td>1111</td>
<td>27</td>
<td>63</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cubic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>43</td>
<td>540</td>
<td>1104</td>
<td>11</td>
<td>34</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>40</td>
<td>801</td>
<td>2060</td>
<td>11</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>40</td>
<td>1501</td>
<td>5287</td>
<td>11</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>40</td>
<td>*</td>
<td>*</td>
<td>11</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3000</td>
<td>40</td>
<td>*</td>
<td>*</td>
<td>11</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>40</td>
<td>*</td>
<td>*</td>
<td>11</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rosen</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>66</td>
<td>611</td>
<td>1262</td>
<td>23</td>
<td>64</td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>52</td>
<td>461</td>
<td>1374</td>
<td>21</td>
<td>62</td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>*</td>
<td>461</td>
<td>1374</td>
<td>21</td>
<td>62</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>*</td>
<td>2063</td>
<td>6612</td>
<td>21</td>
<td>62</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3000</td>
<td>*</td>
<td>4036</td>
<td>13523</td>
<td>21</td>
<td>62</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>21</td>
<td>62</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
New Homotopy for Unconstrained Optimization using Hestenes Stiefel and Conjugate descent

<table>
<thead>
<tr>
<th>Mile</th>
<th>4</th>
<th>100</th>
<th>500</th>
<th>1000</th>
<th>3000</th>
<th>5000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>28</td>
<td>142</td>
<td>501</td>
<td>1001</td>
<td>1442</td>
<td>1660</td>
</tr>
<tr>
<td></td>
<td>101</td>
<td>346</td>
<td>1108</td>
<td>2312</td>
<td>3252</td>
<td>3688</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>149</td>
<td>501</td>
<td>998</td>
<td>1270</td>
<td>1418</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>355</td>
<td>1092</td>
<td>2290</td>
<td>2834</td>
<td>3130</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Non Digonal</th>
<th>4</th>
<th>100</th>
<th>500</th>
<th>1000</th>
<th>3000</th>
<th>5000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>23</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>61</td>
<td>60</td>
<td>59</td>
<td>59</td>
<td>59</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td>249</td>
<td>249</td>
<td>249</td>
<td>249</td>
<td>249</td>
<td>249</td>
</tr>
<tr>
<td></td>
<td>558</td>
<td>558</td>
<td>558</td>
<td>558</td>
<td>558</td>
<td>558</td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>59</td>
<td>59</td>
<td>59</td>
<td>59</td>
<td>59</td>
<td>59</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Woolf</th>
<th>4</th>
<th>100</th>
<th>500</th>
<th>1000</th>
<th>3000</th>
<th>5000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12</td>
<td>49</td>
<td>99</td>
<td>99</td>
<td>99</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>113</td>
<td>141</td>
<td>141</td>
<td>8001</td>
<td>365</td>
</tr>
<tr>
<td></td>
<td>49</td>
<td>3893</td>
<td>8001</td>
<td>19675</td>
<td>19675</td>
<td>19675</td>
</tr>
<tr>
<td></td>
<td>99</td>
<td>1879</td>
<td>8393</td>
<td>3130</td>
<td>3130</td>
<td>3130</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
<td>27</td>
</tr>
</tbody>
</table>

IV. CONCLUSION

In this paper we have presented a new hybrid conjugate gradient method in which a famous parameter $\beta_k$ is computed as a convex combination of $\beta_k^{HS}$ and $\beta_k^{CD}$ and comparative numerical performances of a number of well known conjugate gradient algorithms Hestenes Stiefel (HS) and Conjugate descent (CD). We saw that the performance profile of our method was higher than those of the well established conjugate gradient algorithms HS and CD.

REFERENCES