A new family of conjugate gradient coefficient with application

Norrtaili Shapiee¹, Mohd Rivaié², Mustafa Mamat¹*, Puspa Liza Ghazali³

¹Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu, Malaysia
²Department of Computer Sciences and Mathematic, Universiti Teknologi MARA, Terengganu, Malaysia
³Faculty of Economic and Management Science, Universiti Sultan Zainal Abidin, Terengganu, Malaysia
*Corresponding author E-mail: must@unisz.edu.my

Abstract

Conjugate gradient (CG) methods are famous for their utilization in solving unconstrained optimization problems, particularly for large scale problems and have become more intriguing such as in engineering field. In this paper, we propose a new family of CG coefficient and apply in regression analysis. The global convergence is established by using exact and inexact line search. Numerical results are presented based on the number of iterations and CPU time. The findings show that our method is more efficient in comparison to some of the previous CG methods for a given standard test problems and successfully solve the real life problem.

Keywords: conjugate gradient method; conjugate gradient coefficient; global convergence; line search; regression analysis.

1. Introduction

The conjugate gradient (CG) method plays an important role in solving unconstrained optimization problems. Generally, an unconstrained optimization problem is written as

$$\min f(x),$$

where \( f: \mathbb{R} \rightarrow \mathbb{R} \) is continuously differentiable. The CG method is an iterative method of the form

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, 2, \ldots$$

where \( x_k \) is the current iterate point, \( x_k \) is the current iterate point, \( \alpha_k \geq 0 \) is the stepsize and \( d_k \) is the search direction. The equation for calculating \( d_k \) is defined by

$$d_k = \begin{cases} -g_k & \text{if } k = 0, \\ -g_k + \beta_k d_{k-1} & \text{if } k \geq 1, \end{cases}$$

where \( g_k \) is the gradient of \( f(x) \) at the point \( x_k \). The parameter \( \beta_k \in \mathbb{R} \) is known as the CG coefficient. Some examples of known \( \beta_k \) are Hestenes-Stiefel (HS) [14], Dai-Yuan (DY) [32], Rivaie, Mustafa, Ismail and Leong (RMIL) [17], Polak and Ribiére (PR) [5], Liu and Storey (LS) [37], Conjugate Descent (CD) [29], Fletcher and Reeves (FR) [30]. The corresponding formulas for the \( \beta_k \) mentioned are written as follows:

$$\beta_k^{HS} = \frac{g_k^T (g_k - g_{k-1})}{g_{k-1}^T g_{k-1}}, \quad \beta_k^{DY} = \frac{g_k^T (g_k - g_{k-1})}{d_k^T d_{k-1}}, \quad \beta_k^{HS} = \frac{g_k^T (g_k - g_{k-1})}{d_k^T g_{k-1}}, \quad \beta_k^{FR} = \frac{g_k^T (g_k - g_{k-1})}{d_k^T g_{k-1}}.$$

These CG coefficients could likewise be divided into two groups. The first group comprises of PR, HS, RMIL and LS while in the second group we have FR, DY and CD. It is easy to see that the first group possesses the restart properties, whereas the second group does not have this characteristic [17-18]. In [20] has arranged the CG method into three distinct groups; the classical CG method, the scaled CG method and lastly the hybrid and parameterized CG methods. The classical CG method is the simplest and most straightforward to apply. However, it is difficult to find and produces a new CG method of this type [16].

As indicated to [34-35], all these methods are equivalent if the objective function is strictly convex quadratic. However, they behave distinctively when applied to general non quadratic functions. As mentioned in [33], the history of CG methods starts in [14] who initially proposed a CG method to solve a linear system of equation with a symmetric positive definite matrix. After that, in [30] applied the CG method to general unconstrained optimization problems. Nowadays, the CG methods are noted to be exceedingly valuable for solving large-scale unconstrained optimization problems since it needn’t the storage of matrices [1-2, 8, 10-13, 15, 21-28, 31, 38, 39].

In this decade, a few new CG methods have been proposed. Some of the recent studies aim at creating a \( d_k \) that satisfies the sufficient descent condition and possess global convergence property. The earliest and well-known research on global convergence of CG methods is done by [6]. In that paper, the global convergence of FR method with exact line search is proven for general function. How-
ever, the PR method with exact line search is not globally convergent [7, 36]. It is well known that regression analysis regularly emerges in economics, finance, trade, meteorology, medicine biology, chemistry physics and etc. [39-43]. The classical regression model is defined by
\[ Y = h(X_1, X_2, ..., X_i + \varepsilon) \] (1.4)
where \( Y \) is the reaction variable, \( X_i \) is the indicator variable, \( i = 1, 2, ..., p, p > 0 \) is an integer constant and \( \varepsilon \) is the error term. The function \( h(X_1, X_2, ..., X_i) \) clarify the type of relationship that exist between \( Y \) and \( X = (X_1, X_2, ..., X_i) \). In this manner, we acquire the following linear regression model when \( h \) is a linear function
\[ Y = a_0 + a_1X_1 + a_2X_2 + ... + a_iX_i + \varepsilon \] (1.5)
which is the simplest regression model where \( a_0, a_1, ..., a_i \) are the regression parameters. The most important errand in regression analysis is to evaluate the parameter \( a = (a_0, a_1, ..., a_i) \) and the method of least squares is essential method to determine the parameter which is defined by
\[ \min_{a} S(a) = \sum_{i=1}^{n} (h - a_0 + a_1X_1 + a_2X_2 + ... + a_iX_i)^2 \] (1.6)
The contents of this paper are arranged into six sections. Section 2 introduces the new \( \beta \) and its algorithm. In Section 3, we show the proof of the sufficient descent condition and the global convergence property of our new method. Description of the problem by utilizing regression is displayed as a part of Section 4. Some fascinating numerical result is exhibited in Section 5, where our new method is compared any other CG method and Himmelblau’s function is displayed. Discussions on the result are also included. Finally, a short conclusion is presented in Section 6.

2. New CG coefficient

In this section we propose our new CG coefficient known as \( \beta^{\text{new}} \). The NRMI signifies Norrallti, Rivaie, Mustafa and Ismail. The \( \beta^{\text{new}} \) is given as
\[ \beta^{\text{new}} = \frac{\beta_i}{\beta^{\text{new}}_i} = \frac{\beta_i}{\beta^{\text{new}}_i} \] (2.1)
The general algorithm of CG method utilized as a part of this study is as per the followings:
Step 1: Initialization. Given \( x_0 \), set \( k = 0 \).
Step 2: Compute \( \beta \) based on (2.1).
Step 3: Compute \( d_k \) based on (1.3). If \( \|g_k\| = 0 \), then stop.
Step 4: Compute \( \alpha_k \). Based on exact and inexact line search.
For exact line search compute \( \alpha_k = \min_{\alpha} f(x_k + \alpha d_k) \).
For inexact line search compute \( \alpha_k \). \( f(x_k + \alpha d_k) \leq f(x_k) + \delta \alpha g_i^T d_i \), and
\[ \|g(x_k + \alpha d_k)^T d_k\| \leq \|\delta \alpha g_i^T d_i\| \]
Step 5: Update new point based on (1.2).
Step 6: Convergent test and stopping criteria. If \( f(x_{k+1}) < f(x_k) \) and \( \|g_k\| \leq \varepsilon \), then stop.

Otherwise, go to Step 1 with \( k = k + 1 \).

3. Convergent analysis

In this section, we initially showed the sufficient descent condition and later on the proof of global convergence properties.

3.1. Convergent analysis based on exact line search

First, the convergent properties of \( \beta^{\text{new}} \) will be studied. Firstly, we assume that every search direction \( d_k \) should satisfy the descent condition
\[ g_i^T d_k < 0 \] (3.1)
for all \( k \geq 0 \). If there exists a constant \( C > 0 \) for all \( k \geq 0 \), then the search directions satisfy following sufficient descent condition
\[ g_i^T d_k \leq -C\|g_i\| \] (3.2)

Theorem 1: Consider a CG method with the search direction (1.3) and \( \beta^{\text{new}} \) given as (2.1), then condition (3.2) holds for all \( k \geq 0 \).

Proof: If \( k = 0 \), then it is clear that \( g_i^T d_0 = -C\|g_i\| \). Hence, condition (3.2) holds true. We also need to show that for \( k \geq 1 \), condition (3.2) will also hold true. From (1.3), multiply by \( g_{k+1}^T \), then
\[ g_i^T d_{k+1} = g_i^T (-g_{k+1} + \beta^{\text{new}}_{k+1} d_k) = -\|g_{k+1}\| + \beta^{\text{new}}_{k+1} g_i^T d_k \] (3.3)
For exact line search, we realize that \( g_i^T d_k = 0 \). Thus,
\[ g_i^T d_{k+1} = -\|g_{k+1}\| \]
which implies that \( d_{k+1} \) is a sufficient descent direction. Hence, \( g_i^T d_k \leq -C\|g_i\| \) holds true. The proof is completed.

3.1.2. Global convergent properties

Next, we will show that CG methods with \( \beta^{\text{new}} \) converge globally. However, we first need to simplify our new \( \beta^{\text{new}} \), so that our convergence proof will be markedly easier [16]. From (2.1), we realize that
\[ \beta_{k+1} = \frac{g_{k+1}^T (g_{k+1} - g_k)}{g_i^T (g_i - d_i)} \]
Hence, we get
\[ \beta_{k+1} \leq \frac{\|g_{k+1}\|}{\|g_i\|} \] (3.4)
The following basic assumptions are always needed in the analysis of global convergence properties of CG methods.
Assumption 1:
(i) $f$ is bounded below on the level set $R^*$ and is continuous and differentiable in a neighborhood $N$ of the level set $\ell = \{x| f(x) \leq f(x_0)\}$ at the initial point $x_0$.
(ii) The gradient $g(x)$ is Lipschitz continuous in $N$, so a constant $L > 0$ exists such that $\|g(x) - g(y)\| \leq L\|x - y\|$, for any $x, y \in N$.

Under this assumption, we have the following lemma which was proven by [6]. This lemma also holds for the exact minimization rule, the Goldstein and the Wolfe rule as shown in [10].

Lemma 1: Suppose that Assumption 1 holds true. Consider any CG method of the form (1.3), where $d_k$ is a descent search direction and $\alpha_k$ satisfies the exact minimization rule. Then, the following condition also known as the Zoutendijk condition holds

$$\sum_{k=0}^{\infty} \left( \frac{g_k^* d_k}{\|d_k\|} \right)^2 < \infty.$$  

The proof of this lemma can be seen from [9]. By using Lemma 1, we can obtain the following convergent theorem of the CG method using (3.4).

Theorem 2: Suppose that Assumption 1 holds true. Consider any CG method of the form (1.3) and (1.2), where $\alpha_k$ is obtained by the exact minimization rule. Also, suppose that Assumption 1 and the descent condition hold true. Then, either

$$\lim_{k \to \infty} g_k = 0 \quad \text{or} \quad \sum_{k=0}^{\infty} \left( \frac{g_k^* d_k}{\|d_k\|} \right)^2 < \infty.$$  

Proof: To prove Theorem 2, we use the contradiction approach. That is if Theorem 2 is not true, then a constant $c > 0$ exists such that

$$\|g_k\| \geq c.$$  

Rewriting (1.3) as

$$d_{k+1} + g_{k+1} = \beta_k d_k,$$

and squaring both sides of the equation, we obtain

$$\|d_{k+1}\|^2 = (\beta_k)^2 \|d_k\|^2 - 2g_k^* d_{k+1} - \|g_{k+1}\|^2.$$  

Dividing both side by $\left( \frac{g_k^* d_k}{\|d_k\|} \right)^2$, then,

$$\|d_{k+1}\|^2 = \left( \frac{\beta_k}{\|d_k\|^2} \right) \|d_k\|^2 - \frac{2}{\|d_k\|^2} \left( \frac{g_k^* d_k}{\|d_k\|} \right)^2 - \left( \frac{g_k^* d_k}{\|d_k\|} \right)^2.$$  

By completing the square,

$$\|d_{k+1}\|^2 = \left( \frac{\beta_k}{\|d_k\|^2} \right) \|d_k\|^2 - \left( \frac{1}{\|d_k\|^2} + \frac{g_k^* d_k}{\|d_k\|} \right)^2 + \frac{1}{\|d_k\|^2} \left( \frac{g_k^* d_k}{\|d_k\|} \right)^2.$$  

Applying (3.4) in (3.7) yields

$$\frac{\|d_{k+1}\|^2}{\left( \frac{g_k^* d_k}{\|d_k\|} \right)^2} \leq \left( \frac{\|g_k\|^2}{\|d_k\|^2} \right) \frac{1}{\|d_k\|^2} + \frac{1}{\|g_k\|^2}.$$  

This implies

$$\sum_{k=0}^{\infty} \left( \frac{g_k^* d_k}{\|d_k\|} \right)^2 \geq c^2 \sum_{k=0}^{\infty} \frac{1}{\|g_k\|^2} = \infty.$$  

Therefore, from (3.9) and (3.5), it follows that

$$\sum_{k=0}^{\infty} \left( \frac{g_k^* d_k}{\|d_k\|} \right)^2 = \infty.$$  

This contradicts the Zoutendijk condition in Lemma 1. Therefore, the proof is completed.

Theorem 3: Suppose that Assumption 1 holds true. Consider any CG methods of the form (1.3) and (1.2), where $\alpha_k$ is obtained by the exact minimization rule. Also, suppose that Assumption 1 and the descent condition hold true. Then, either

$$\lim_{k \to \infty} g_k = 0 \quad \text{or} \quad \sum_{k=0}^{\infty} \left( \frac{g_k^* d_k}{\|d_k\|} \right)^2 < \infty.$$  

Proof: From (3.6) and (3.4)

$$\|d_{k+1}\|^2 = \left( \frac{\|g_k\|^2}{\|d_k\|^2} \right) \|d_k\|^2 - 2g_k^* d_{k+1} - \|g_{k+1}\|^2.$$  

We have already proved that condition (3.2) holds. Therefore, we know that

$$g_{k+1}^* d_{k+1} \leq -c \|g_{k+1}\|^2.$$
Hence, from (3.10)

$$
\|d_{k+1}\|^2 = \frac{g_k^T (g_k - g_{k+1})^2}{\|g_k\|^2} + 2C \|s_k\|^2 - \|s_k\| (1 - 2C)
$$

(3.11)

Multiply both sides of (3.11) with \(\frac{\|g_k\|^2}{\|d_{k+1}\|^2}\), then

$$
\|d_{k+1}\|^2 \frac{\|g_k\|^2}{\|d_{k+1}\|^2} = \frac{g_k^T (g_k - g_{k+1})^2}{\|g_k\|^2} + 2C \|s_k\|^2 - \|s_k\| (1 - 2C)
$$

$$
= \|g_k\|^2 (2C - 1) + \|s_k\|^2 \frac{\|d_{k+1}\|^2}{\|g_k\|^2}
$$

$$
\|d_{k+1}\|^2 \frac{\|g_k\|^2}{\|d_{k+1}\|^2} \leq \|g_k\|^2 (2C - 1) + \|s_k\|^2 \frac{\|d_{k+1}\|^2}{\|g_k\|^2}
$$

(3.12)

Based on Theorem 2, we know that \(\lim_{k \to \infty} \frac{(g^T, d_{k+1})^2}{\|d_{k+1}\|^2} < 0\). This will imply that if Theorem 3 is not true, then we have

$$
\lim_{k \to \infty} \frac{(g^T, d_{k+1})^2}{\|d_{k+1}\|^2} = \infty
$$

From (3.12), we get \(\infty \leq \frac{\|g_k\|^2}{\|d_{k+1}\|^2}\) and \(\infty \leq \frac{\|s_k\|^2}{\|d_{k+1}\|^2}\) respectively. Hence, Theorem 3 holds true for sufficiently large \(k\).

3.2. Convergent analysis based on inexact line search

In this section, the convergent properties of \(\beta_{k+1}^{\text{swp}}\) will be studied based on the inexact line search in terms of strongly Wolfe line search. We will also show that this CG coefficient will possess sufficient descent conditions under this line search. Under this inexact line search, we have

$$
f(x_k + \alpha_i d_i) \leq f(x_k) + \alpha_i g^T_i d_i.
$$

(3.13)

and

$$
g(x_k + \alpha_i d_i)^T d_i \leq -\sigma g^T_i d_i.
$$

(3.14)

The following theorem which illustrates that the formula possesses the sufficient descent condition under the strongly Wolfe line search.

From (2.1), we realize that

$$
\beta_{k+1}^{\text{swp}} = \frac{g^T_k (g_k - g_{k+1})}{\|g_k\|^2} = \frac{g^T_k (g_k - g_{k+1})}{\|g_k\|^2}.
$$

Hence, we get

$$
\beta_{k+1}^{\text{swp}} \leq \frac{\|g_k\|^2}{\|g_k\|^2}.
$$

(3.15)

Theorem 4: Let the sequences \(\{r_k\}\) and \(\{d_k\}\) in the general algorithm, and let the stepsize \(\alpha_i\) be determined by the SWP line search (3.13) and (3.14). If \(\sigma \in (0, 1)\), then the sufficient descent condition (3.2) holds.

Proof: From (3.15) and using (3.14), we get

$$
\beta_{k+1}^{\text{swp}} g^T_k d_k \leq \|g_k\|^2 \sigma g^T_k d_k.
$$

(3.16)

By (1.3), we have

$$
\|d_{k+1}\|^2 = 1 + \beta_{k+1}^{\text{swp}} g^T_k d_k.
$$

(3.17)

We prove the descent property of \(\{d_k\}\) by induction. Since \(g^T_k d_k = -\|g_k\|^2 < 0\), if \(g_k \neq 0\), now we suppose that \(d_i, i = 1, 2, \ldots, k\) are all descent directions, for example \(d_i^T g_i < 0\). By (3.16), we get

$$
\beta_{k}^{\text{swp}} g^T_k d_k \leq \|g_k\|^2 \sigma g^T_k d_k.
$$

(3.18)

that is

$$
\|g_k\|^2 \sigma g^T_k d_k \leq \beta_{k}^{\text{swp}} g^T_k d_k \leq \|g_k\|^2 \sigma g^T_k d_k.
$$

(3.19)

However, from (3.17) together with (3.19), we deduce

$$
-1 - \sigma g^T_k d_k \leq g^T_k d_k \leq -1 - \sigma g^T_k d_k.
$$

(3.20)

and we know \(\|g_k\| \leq \frac{1}{\|g_k\|}\).

Replace (3.20), we get

$$
-1 - \sigma g^T_k d_k \leq g^T_k d_k \leq -1 - \sigma g^T_k d_k.
$$

(3.21)

Repeating this process and using the fact \(d_i^T g_i = -\|g_i\|^2\) imply

$$
-\sum_{i=0}^{k} \sigma \leq g^T_k d_k \leq -2 + \sum_{i=0}^{k} \sigma.
$$

(3.22)

Then, (3.22) can be written as

$$
-\frac{1}{1-\sigma} \|g_k\| \leq g^T_k d_k \leq \frac{1}{1-\sigma}.
$$

(3.23)

Thus, by induction \(g^T_k d_k < 0\), holds for all \(k \geq 0\). Denote \(c = 2 - 1/(1-\sigma)\), then \(c \in (0, 1)\) and (3.23) turns out to be

$$
c - 2 \leq -\frac{g^T_k d_k}{\|g_k\|} \leq c.
$$

(3.24)

This implies that (3.2) holds. The proof is complete.
4. Description of the problem

In this section, the detailed description of the problem considered is given below.

Problem 1: In Table 1, there is data of some kind of commodity between year and index of road deaths in 2004 until 2014:

<table>
<thead>
<tr>
<th>No.</th>
<th>Year (p)</th>
<th>Index of Road Deaths (d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2004</td>
<td>4.51</td>
</tr>
<tr>
<td>2</td>
<td>2005</td>
<td>4.18</td>
</tr>
<tr>
<td>3</td>
<td>2006</td>
<td>3.98</td>
</tr>
<tr>
<td>4</td>
<td>2007</td>
<td>3.73</td>
</tr>
<tr>
<td>5</td>
<td>2008</td>
<td>3.63</td>
</tr>
<tr>
<td>6</td>
<td>2009</td>
<td>3.55</td>
</tr>
<tr>
<td>7</td>
<td>2010</td>
<td>3.40</td>
</tr>
<tr>
<td>8</td>
<td>2011</td>
<td>3.21</td>
</tr>
<tr>
<td>9</td>
<td>2012</td>
<td>3.05</td>
</tr>
<tr>
<td>10</td>
<td>2013</td>
<td>2.90</td>
</tr>
<tr>
<td>11</td>
<td>2014</td>
<td>2.66</td>
</tr>
</tbody>
</table>

From a statistical point of view, we can conclude that there will be possible changes in the index of road deaths although will occur road deaths per year. In summary, there will be an abatement in the index of road deaths for the years ahead and our main objective is to determine the function between the index of road deaths and the year which is the regression equation for $p$.

From the given data above, one can observe that there exists a linear relationship between the year and the index of road deaths with the regression equation given by $d = a_0 + a_1 p$, where $a_0$ and $a_1$ signifying the regression parameters. Solving the above regression equation includes finding the value of $a_0$ and $a_1$ by the method of least squares that minimized the problem

$$
\min Q = \sum_{i=1}^{n} \left[ d_i - (a_0 + a_1 p_i) \right]^2. \tag{4.1}
$$

We can now change the above least square problem into an unconstrained optimization problem as

$$
\min f(a) = \sum_{i=1}^{n} \left[ d_i - a(1, p_i) \right]^2. \tag{4.2}
$$

Solving the above problem (4.2) utilizing the linear least squares method yields the solution $a = (-4.5325, -0.1675)$. In this context, we utilize our proposed a new conjugate gradient to solve this problem solving regression problem compare with the linear least squares method [3].

5. Results and discussion

In this section, we carry out some numerical tests for FR, HS, NRMI and RMIL methods. Some test problems in [19] are selected to analyze the efficiency of $\beta^{\text{rand}}$ based on the number of iterations and CPU time. The stopping criteria are set to be when $\Vert s_i \Vert \leq 10^{-6}$. As suggested by [9], a test point should not be restricted to a point that is too close to the solution point. The best selected initial points should be based on the random number generator. However, we believe that this approach, will add to the complexity of the computer programming hence leading to high CPU time. Therefore, in this research we have selected four different initial points for each of the test problems. We start from a point that is close to the solution point and then move to the one that is furthest from it. These four initial points will also allow us to test the global convergence properties and the robustness of our method at the same time [17]. A list of problem functions and the initial points selected are shown in Table 2.

All the problems mentioned in Table 2 are solved using MATLAB2011b subroutine programming by using the exact and inexact line searches to obtain the stepsize. In some cases, the computation stops due to the line search failing to find a positive stepsize, thus the test is considered a failure. In order to describe the method's performance and to determine the best method, we use the Performance Profile introduced by [4]. We present Himmelblau’s function (Fig. 2), which is a multimodal function to test the performance of the optimization algorithms. The function is defined as follows:

$$
f(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2.
$$

This function is a summation of two squared terms. Each term inside the bracket can be considered as an error term. The first term calculate the distinction between the term $(x_1^2 + x_2)$ and 11 and the second term calculates the contrast between the term $(x_1 + x_2^2)$ and 7. Since the goal is to minimize these squared contrasts (or deviations of the two terms from 11 and 7 separately), the optimum solution will be a set of values of $x_1$ and $x_2$, fulfilling the following two equations.

$$
x_1^2 + x_1 = 11, \quad x_1 + x_2^2 = 7.
$$

It has four indistinguishable four minima in this range (one in each quadrant). The locations of all the minima can be discovered logically. However, because they are roots of cubic polynomials, when written in terms of radicals, the expressions are fairly complicated.

Many engineering design problems expect to find a set of design parameters fulfilling various objectives simultaneously. In these problems, a mathematical expression for each objective is typically written and the distinction of the expression from the objective is calculated. The distinctions are then squared and included to form an overall objective function, which must be minimized. In this manner, the above Himmelblau’s function work looks like the mathematical expression of an objective function in many engineering design problems. We utilized different initial points with general algorithm of CG method and Himmelblau’s function. Every initial point gave a different solution point for every CG method, as shown in Table 2.

![Index of road deaths versus year for linear trend line](image-url)
Table 2: A list of problem functions

<table>
<thead>
<tr>
<th>No.</th>
<th>Function</th>
<th>n</th>
<th>Initial Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Six hump</td>
<td>2</td>
<td>(2,2), (10,10), (36,36), (52,52)</td>
</tr>
<tr>
<td>2</td>
<td>Three hump</td>
<td>2</td>
<td>(5.5), (31,31), (41,41), (55,55)</td>
</tr>
<tr>
<td>3</td>
<td>Booth</td>
<td>2</td>
<td>(3.3), (27.27), (73.73), (100,100)</td>
</tr>
<tr>
<td>4</td>
<td>Treccani</td>
<td>2</td>
<td>(1.1), (42.42), (62.62), (91.91)</td>
</tr>
<tr>
<td>5</td>
<td>Zettl</td>
<td>2</td>
<td>(19.19), (31.31), (65.65), (95.95)</td>
</tr>
<tr>
<td>6</td>
<td>Leon</td>
<td>2</td>
<td>(3.3), (7.7), (10,10), (12.12)</td>
</tr>
<tr>
<td>7</td>
<td>Matyas</td>
<td>2</td>
<td>(2.2), (32.32), (62.62), (92.92)</td>
</tr>
<tr>
<td>8</td>
<td>Extended wood</td>
<td>4</td>
<td>(21.21, ..., 21), (35.35, ..., 35), (51.51, ..., 51), (85.85, ..., 85)</td>
</tr>
<tr>
<td>9</td>
<td>Quartic</td>
<td>4</td>
<td>(1, 1, ..., 1), (21, 21, ..., 21), (41, 41, ..., 41), (61, 61, ..., 61)</td>
</tr>
<tr>
<td>10</td>
<td>Colville</td>
<td>4</td>
<td>(7, 7, ..., 7), (17, 17, ..., 17), (23, 23, ..., 23), (77, 77, ..., 77)</td>
</tr>
<tr>
<td>11</td>
<td>Powell</td>
<td>4</td>
<td>(14, 14, ..., 14), (21, 21, ..., 21), (84, 84, ..., 84), (94, 94, ..., 94)</td>
</tr>
<tr>
<td>12</td>
<td>Quadratic QF2</td>
<td>2,4</td>
<td>(16,16, ..., 16), (33,33, ..., 33), (52,52, ..., 52), (82,82, ..., 82)</td>
</tr>
<tr>
<td>13</td>
<td>Extended White and Holz</td>
<td>2,4</td>
<td>(3, 3, ..., 3), (7, 7, ..., 7), (10, 10, ..., 10), (12, 12, ..., 12)</td>
</tr>
<tr>
<td>14</td>
<td>Rosenbrock</td>
<td>2,4</td>
<td>(3, 3, ..., 3), (39, 39, ..., 39), (53, 53, ..., 53), (61, 61, ..., 61)</td>
</tr>
<tr>
<td>15</td>
<td>Extended Denschna</td>
<td>2,4</td>
<td>(3, 3, ..., 3), (5, 5, ..., 5), (12, 12, ..., 12), (50, 50, ..., 50)</td>
</tr>
<tr>
<td>16</td>
<td>Shallow</td>
<td>2,4</td>
<td>(11, 11, ..., 11), (18, 18, ..., 18), (60, 60, ..., 60), (64, 64, ..., 64)</td>
</tr>
<tr>
<td>17</td>
<td>Extended Triagonal</td>
<td>2,4</td>
<td>(34, 34, ..., 34), (41, 41, ..., 41), (72, 72, ..., 72), (93, 93, ..., 93)</td>
</tr>
<tr>
<td>18</td>
<td>Extended Beale</td>
<td>2,4</td>
<td>(2, 2, ..., 2), (6, 6, ..., 6), (8, 8, ..., 8), (11, 11, ..., 11)</td>
</tr>
<tr>
<td>19</td>
<td>Diagonal 4</td>
<td>2,4</td>
<td>(4, 4, ..., 4), (21, 21, ..., 21), (50, 50, ..., 50), (84, 84, ..., 84)</td>
</tr>
<tr>
<td>20</td>
<td>Quadratic QF1</td>
<td>2,4</td>
<td>(1,1,1,1,14,14,14,14), (31,31,31,31), (64,64,64,64)</td>
</tr>
</tbody>
</table>

Table 3: The initial points corresponding the optimal points and number of iteration with the Himmelblau function by using exact line search

<table>
<thead>
<tr>
<th>Initial Point</th>
<th>Function Value With Optimal Solution</th>
<th>Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1)</td>
<td>f(3,2) = 0</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 4: The initial points corresponding the optimal points and number of iteration with the Himmelblau function by using inexact line search

<table>
<thead>
<tr>
<th>Initial Point</th>
<th>Function Value With Optimal Solution</th>
<th>Iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,1)</td>
<td>f(3.2) = 0</td>
<td>7</td>
</tr>
</tbody>
</table>

Every function in Table 2 is used to test four CG methods which are FR, HS, RMIL and NRMI to show their performance. The result each CG methods are shown below for exact and inexact line search in terms of iteration and CPU time.

Fig. 2: Himmelblau’s function

Fig. 3: Performance profile relative to the number of iterations (exact)
different initial points are those chosen to test the efficiency of our method.

### Table 5: Numerical results for problem 1

<table>
<thead>
<tr>
<th>i.p</th>
<th>Exact</th>
<th>Inexact</th>
</tr>
</thead>
<tbody>
<tr>
<td>(5,5)</td>
<td>0.1341</td>
<td>0.4615</td>
</tr>
<tr>
<td>(8,8)</td>
<td>0.0737</td>
<td>0.443</td>
</tr>
<tr>
<td>(17,17)</td>
<td>0.6766</td>
<td>0.4439</td>
</tr>
<tr>
<td>(32,32)</td>
<td>0.5683</td>
<td>0.4588</td>
</tr>
</tbody>
</table>

### Table 6: Numerical results for problem 1 based on exact line search in terms of iteration

<table>
<thead>
<tr>
<th>i.p</th>
<th>b</th>
<th>ε</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>(5,5)</td>
<td>4.5325</td>
<td>0.3120590941191467</td>
<td>0.312059094653934</td>
</tr>
<tr>
<td>(8,8)</td>
<td>4.5325</td>
<td>0.3120590941195026</td>
<td>0.312059094653934</td>
</tr>
<tr>
<td>(17,17)</td>
<td>4.5325</td>
<td>0.3120590941195714</td>
<td>0.312059094653934</td>
</tr>
<tr>
<td>(32,32)</td>
<td>4.5325</td>
<td>0.3120590941195767</td>
<td>0.312059094653934</td>
</tr>
</tbody>
</table>

### Table 7: Numerical results for problem 1 based on exact line search in terms of CPU time

<table>
<thead>
<tr>
<th>i.p</th>
<th>b</th>
<th>ε</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>(5,5)</td>
<td>4.5325</td>
<td>0.3120590941196463</td>
<td>0.3120590941196463</td>
</tr>
<tr>
<td>(8,8)</td>
<td>4.5325</td>
<td>0.3120590941195714</td>
<td>0.3120590941195714</td>
</tr>
<tr>
<td>(17,17)</td>
<td>4.5325</td>
<td>0.3120590941195026</td>
<td>0.3120590941195026</td>
</tr>
<tr>
<td>(32,32)</td>
<td>4.5325</td>
<td>0.312059094119467</td>
<td>0.312059094119467</td>
</tr>
</tbody>
</table>

### Table 8: Numerical results for problem 1 based on inexact line search in terms of iteration

<table>
<thead>
<tr>
<th>i.p</th>
<th>b</th>
<th>ε</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>(5,5)</td>
<td>4.5325</td>
<td>0.3120590940195767</td>
<td>0.3120590940195767</td>
</tr>
<tr>
<td>(8,8)</td>
<td>4.5325</td>
<td>0.3120590954314233</td>
<td>0.3120590954314233</td>
</tr>
<tr>
<td>(17,17)</td>
<td>4.5325</td>
<td>0.3120590954314233</td>
<td>0.3120590954314233</td>
</tr>
<tr>
<td>(32,32)</td>
<td>4.5325</td>
<td>0.3120590954314233</td>
<td>0.3120590954314233</td>
</tr>
</tbody>
</table>

From both figures, the left side represents the method that is fastest in solving all of the test problems. On the other hand, the right side shows the method that successfully solved all the test problems. For exact line search, Fig. 3 and Fig. 4 show that NRMI has the best performance as it can solve all 100% of the test problems and the NRMI curve appears above FR, HS and RMIL curves. The HS and FR method solve only 95.38% of the problems, while a recently proposed method RMIL, method solves 97.69% of the test problems. For inexact line search, Fig. 5 and Fig. 6 show that HS has the best performance as it can solve 97.31%. The NRMI can solve 96.92%, the FR can solve 95.38% and the RMIL can solve 91.92% of the test problems. Therefore, we can say that NRMI is considered superior and more effective compared to FR, HS, and RMIL in exact line search and HS is considered superior and more effective compared to FR, HS and RMIL in inexact line search.

The application of our proposed new conjugate gradient method implemented with the exact and inexact line search rule and different initial points are chosen to test the efficiency of our method.

### Table 9: Numerical results for problem 1 based on inexact line search in terms of CPU time

<table>
<thead>
<tr>
<th>i.p</th>
<th>b</th>
<th>ε</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>(5,5)</td>
<td>4.5325</td>
<td>0.3120590940195767</td>
<td>0.3120590940195767</td>
</tr>
<tr>
<td>(8,8)</td>
<td>4.5325</td>
<td>0.3120590954314233</td>
<td>0.3120590954314233</td>
</tr>
<tr>
<td>(17,17)</td>
<td>4.5325</td>
<td>0.3120590954314233</td>
<td>0.3120590954314233</td>
</tr>
<tr>
<td>(32,32)</td>
<td>4.5325</td>
<td>0.3120590954314233</td>
<td>0.3120590954314233</td>
</tr>
</tbody>
</table>

The numerical results show the performance of the algorithm considered in this application, the numerical results demonstrate that our algorithm has made significant performance. The performance from the implementation of problem 1 indicates that regardless of the different initial points, our proposed method was able to solve the problem.

### 6. Conclusion

In this paper, we propose a new classical CG method known as $\beta_{\text{new}}$ and solve real life regression problems. The convergent
properties of this $\beta_{k}\text{max}$ have been studied. For an algorithm to converge, it must fulfill the sufficient descent condition and global convergence properties. Based on the theoretical proof and the numerical result, we can say that this $\beta_{k}\text{max}$ converges globally and performs better than other standard CG methods. Numerical results in application show successful for the test problems consider and can further be employed for application in regression analysis.

Acknowledgement

We would like to thank The Ministry of Higher Education of Malaysia for the funding of this research under Ministry of Education (Sponsorship MyPhD KPM).

References