Two efficient modifications of AZPRP conjugate gradient method with sufficient descent property

Zabidin Salleh¹*, Adel Almarashi² and Ahmad Alhawarat¹

Abstract
The conjugate gradient method can be applied in many fields, such as neural networks, image restoration, machine learning, deep learning, and many others. Polak–Ribiere–Polyak and Hestenes–Stiefel conjugate gradient methods are considered as the most efficient methods to solve nonlinear optimization problems. However, both methods cannot satisfy the descent property or global convergence property for general nonlinear functions. In this paper, we present two new modifications of the PRP method with restart conditions. The proposed conjugate gradient methods satisfy the global convergence property and descent property for general nonlinear functions. The numerical results show that the new modifications are more efficient than recent CG methods in terms of number of iterations, number of function evaluations, number of gradient evaluations, and CPU time.

MSC: 49M37; 65K05; 90C3

Keywords: Conjugate gradient method; Strong Wolfe–Powell line search; Polak–Ribiere–Polyak method; Global convergence

1 Introduction
We consider the following form for the unconstrained optimization problem:

\[ \min \{ f(x) : x \in \mathbb{R}^n \}, \quad (1.1) \]

where \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) is a continuously differentiable function and its gradient is denoted by \( g(x) = \nabla f(x) \). To solve (1.1) using the CG method, we use the following iterative method starting from the initial point \( x_0 \in \mathbb{R}^n \). Then

\[ x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, 2, \ldots, \quad (1.2) \]
where $\alpha_k > 0$ is the step size obtained by some line search. The search direction $d_k$ is defined by

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

(1.3)

where $g_k = g(x_k)$ and $\beta_k$ is known as the conjugate gradient method. To obtain the step-length $\alpha_k$, we have the following two line searches:

1. Exact line search

$$f(x_k + \alpha_k d_k) = \min f(x_k + \alpha d_k), \quad \alpha \geq 0.$$  

(1.4)

However, (1.4) is computationally expensive if the function has many local minima.

2. Inexact line search

To overcome the cost of using exact line search and obtain steps that are neither too long nor too short, we usually use inexact line search, in particular weak Wolfe–Powell (WWP) line search [1, 2] given as follows:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k,$$

(1.5)

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k.$$  

(1.6)

Another, strong, version of Wolfe–Powell (SWP) line search is given by (1.5) and

$$|g(x_k + \alpha_k d_k)^T d_k| \leq \sigma |g_k^T d_k|,$$

(1.7)

where $0 < \delta < \sigma < 1$.

The descent condition (downhill condition) plays an important role in the CG method, where the equation of the descent condition is given as follows:

$$g_k^T d_k < 0.$$  

(1.8)

Albaali [3] extended (1.8) to the following form:

$$g_k^T d_k \leq -c \|g_k\|^2, \quad k \geq 0 \text{ and } c > 0,$$

(1.9)

called the sufficient descent condition.

The steepest descent method is the simplest of the gradient methods for optimization functions in $n$ variables. From a current trial point $x_1$, for a function $f(x)$, one expects to find a vector close to a minimum by moving away from $x_1$ along the direction which causes $f(x)$ to decrease rapidly, i.e., $f(x_1) > f(x_2) > f(x_3) > \cdots$. This direction of steepest descent is given by the negative gradient, $-g_k$. Using contour lines, the minimum point of a function is obtained with two variables. For example, Fig. 1 shows contour lines for Booth function in two dimensions.

As we see in Fig. 2, the gradient $f'(x)$ is orthogonal with the contour lines, and for every $x$, the gradient point in the direction of the steepest increases $f(x)$. In Fig. 2, the gradient,
contours, and Booth function are plotted, which clearly portrays the function's minimum using the function or contour line graph. Despite the steepest descent method robustness, it is not efficient due to CPU time for large-dimensional functions. Thus, using the CG method will avoid the orthogonality between the $\nabla f$ and the search direction. Figure 3 shows the angle between the $\nabla f$ and $d_k$ using the CG method.

\[
\cos(\theta_k) = \left( -\frac{d_k^T g_k}{\|d_k\| \|g_k\|} \right).
\]
The most famous classical formulas of CG methods are Hestenes–Stiefel (HS) [3], Polak–Ribiére–Polyak (PRP) [4], Liu and Storey (LS) [5], Fletcher–Reeves (FR) [6], Fletcher (CD) [7], Dai and Yuan (DY) [8], given as follows:

\[
\begin{align*}
\beta_{HS}^k &= \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, & \beta_{PRP}^k &= \frac{g_k^T y_{k-1}}{\|g_k\|^2}, & \beta_{LS}^k &= -\frac{g_k^T y_{k-1}}{d_{k-1}^T g_k}, \\
\beta_{FR}^k &= \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, & \beta_{CD}^k &= -\frac{\|g_k\|^2}{d_{k-1}^T g_k}, & \beta_{DY}^k &= \frac{\|g_k\|^2}{d_{k-1}^T g_{k-1}},
\end{align*}
\]

where \( y_{k-1} = g_k - g_{k-1} \).

These methods are similar if we use exact line search and a function satisfying quadratic line search condition since \( g_k^T d_{k-1} = 0 \), which implies \( g_k^T d_k = -\|g_k\|^2 \) using (1.3). In addition, if the function is quadratic, then \( g_k^T g_{k-1} = 0 \).

The global convergence properties were studied by Zoutendijk [9] and Al-Baali [10]. The global convergence of the PRP method for a convex objective function under exact line search was proved by Polak and Ribere in [4]. Later, Powell [11] gave a counterexample showing a nonconvex function, in which PRP and HS can cycle infinitely without getting a solution. Powell emphasized the importance to achieve the global convergence of PRP and HS method, which should not be negative. Moreover, Gilbert and Nocedal [12] proved that nonnegative PRP, i.e., with \( \beta_k = \max\{\beta_{PRP}^k, 0\} \), is globally convergent under complicated line searches.

Since the function is quadratic, i.e., the step size is obtained by exact line search (1.4), the CG method satisfies the conjugacy condition, i.e., \( d_i^T H d_j^T = 0 \), \( \forall i \neq j \). Using the mean value theorem and exact line search with equation (1.3), we can obtain \( \beta_{HS}^k \). From the quasi-Newton method, BFGS method, the limited memory (LBFGS) method, and equation (1.3), Dai and Liao [13] proposed the following conjugacy condition:

\[
d_k^T y_{k-1} = -t g_k^T s_{k-1}, \tag{1.10}
\]

where \( s_{k-1} = x_k - x_{k-1} \), and \( t \geq 0 \). In the case of \( t = 0 \), equation (1.10) becomes the classical conjugacy condition. By using (1.3) and (1.10), [13] proposed the following CG formula:

\[
\beta_{DL}^k = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} - t \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}. \tag{1.11}
\]
However, $\beta_{DL}^k$ faces the same problem as $\beta_{PRP}^k$ and $\beta_{HS}^k$, i.e., $\beta_{DL}^k$ is not nonnegative in general. Thus, [13] replaced equation (1.11) by

$$\beta_{DL}^{k+} = \max\{\beta_{HS}^k, 0\} - t \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}. \quad (1.12)$$

Moreover, Hager and Zhang [14, 15] presented a modified CG parameter that satisfies the descent property for any inexact line search with $g_k^T d_k \leq -(7/8)\|g_k\|^2$. This new version of the CG method is globally convergent whenever the line search satisfies the (WP) line search requirement. This formula is given as follows:

$$\beta_{HZ}^k = \max\{\beta_N^k, \eta_k\}, \quad (1.13)$$

where $\beta_N^k = \frac{1}{d_k^T y_k} (y_k - 2d_k \frac{\|g_k\|^2}{d_k^T y_k})^T g_k$, $\eta_k = -\frac{1}{\|d_k\| \min\{\|g_k\|\}}$, and $\eta > 0$ is a constant.

Note that if $t = 2\frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}}$, then $\beta_N^k = \beta_{DY}^k$.

In 2006, Wei et al. [16] gave a new positive CG method, which is quite similar to the original PRP method, which has global convergence under exact and inexact line search, that is,

$$\beta_{WYL}^k = \frac{g_k^T (g_k - \frac{\|g_k\|^2}{\|g_k - g_{k-1}\|^2} g_{k-1})}{\|g_{k-1}\|^2},$$

where $y_{k-1} = g_k - g_{k-1}$. From the WYL method, many modifications appeared, such as the following [17]:

$$\beta_{DPRP}^k = \frac{\|g_k\|^2 - \|g_k - g_{k-1}\|^2}{m|g_k| d_{k-1}| + \|g_{k-1}\|^2}, \quad m \geq 1 \quad [11]$$

and

$$\beta_{DHS}^k = \frac{\|g_k\|^2 - \|g_k - g_{k-1}\|^2}{m|g_k^T d_{k-1}| + d_{k-1}^T y_{k-1}}, \quad \text{where } m > 1.$$  

Alhawarat et al. [18] constructed the following CG method with a new restart criterion as follows:

$$\beta_{AZPRP}^k = \begin{cases} \frac{\|g_k\|^2 - \|g_k - g_{k-1}\|^2}{\|g_k - g_{k-1}\|^2}, & \|g_k\|^2 > \mu_k \|g_k^T g_{k-1}\|, \\ 0, & \text{otherwise,} \end{cases}$$

where $\mu_k = \frac{\|s_k\|}{\|y_k\|}$, $s_k = x_k - x_{k-1}$, $y_k = g_k - g_{k-1}$, and $\|\cdot\|$ denotes the Euclidean norm.

Besides, Kaelo et al. [19] proposed the following CG formula:

$$\beta_{PKT}^k = \begin{cases} \frac{\|g_k\|^2 - g_k^T g_{k-1}}{\max\{d_{k-1}^T y_{k-1}, d_{k-1}^T d_{k-1}\}}, & 0 < g_k^T g_{k-1} < \|g_k\|^2, \\ \|g_k\|^2 \max\{d_{k-1}^T y_{k-1}, d_{k-1}^T d_{k-1}\}, & \text{otherwise.} \end{cases}$$
2 Motivation and the new restarted formula

To improve the efficiency of $\beta_k^{AZPRP}$ in terms of function evaluation, gradient evaluation, number of iterations, and CPU time, we construct two new CG methods based on $\beta_k^{AZPRP}$, $\beta_k^{DPRP}$, and $\beta_k^{DHS}$ as follows:

$$
\beta_{k}^{A1} = \begin{cases} \frac{\|g_k\|^2 - \mu_k }{mg_k^T d_{k-1}} - \mu_k g_k^T d_{k-1}^T, & \text{if } \|g_k\|^2 > \mu_k |g_k^T g_{k-1}|, \\ -\mu_k g_k^T y_{k-1}, & \text{otherwise}, \end{cases} \quad (2.1)
$$

where

$$
\mu_k = \frac{\|s_{k-1}\|}{\|y_{k-1}\|}. \quad (2.2)
$$

The second modification is given as follows:

$$
\beta_{k}^{A2} = \begin{cases} \frac{\|g_k\|^2 - \mu_k }{mg_k^T d_{k-1}} - \mu_k g_k^T d_{k-1}^T, & \text{if } \|g_k\|^2 > \mu_k |g_k^T g_{k-1}|, \\ -\mu_k g_k^T y_{k-1}, & \text{otherwise}. \end{cases} \quad (2.3)
$$

Algorithm 2.1

3 The global convergence properties

Assumption 1

1. $f(x)$ is bounded from below on the level set $\Omega = \{x \in \mathbb{R}^n : f(x) \leq f(x_1)\}$, where $x_1$ is the starting point.
II. In some neighborhood $N$ of $\Omega$, $f$ is continuous and differentiable, and its gradient is Lipchitz continuous. That is, for any $x, y \in N$, there exists a constant $L > 0$ such that
\[
\|g(x) - g(y)\| \leq L\|x - y\|.
\]
The following is considered one of the most important lemmas used to prove the global convergence properties. For more details, the reader can refer to [9].

**Lemma 3.1** Suppose Assumption 1 holds. Considering the CG method of the form (1.3), where the search direction satisfies the sufficient descent condition and $\alpha_k$ exists by standard WWPlinesearch, we have
\[
\sum_{k=0}^{\infty} \frac{(g^T_k d_k)^2}{\|d_k\|^2} < \infty,
\] (3.1)
where (3.1) is known as the Zoutendijk condition. Inequality (3.1) also holds for the exact line search, the Armijo-Goldstein line search, and the SWP line search.

Substituting (1.9) into (3.1) yields
\[
\sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} < \infty.
\] (3.2)

Gilbert and Nocedal [11] presented an important theorem to find the global convergence of nonnegative PRP and nonnegative methods summarized by Theorem 3.3. Furthermore, they presented a nice property, called Property*, as follows:

**Property*** Consider a method of the form (1.1) and (1.2), and suppose $0 < \gamma \leq \|g_k\| \leq \bar{\gamma}$. We say that the method possesses Property* if there exist constant $b > 1$ and $\lambda > 0$ such that for all $k \geq 1$, we get $|\beta_k| \leq b$, and if $\|x_k - x_{k-1}\| \leq \lambda$, then
\[
|\beta_k| \leq \frac{1}{2b}.
\]
The following theorem plays a crucial role in the CG method given in [11].

**Theorem 3.1** Considering any CG method of the form (1.2) and (1.3), suppose the following conditions hold:
I. $\beta_k > 0$.
II. The sufficient descent condition is satisfied.
III. The Zoutendijk condition holds.
IV. Property* is true.
V. Assumption 1 is satisfied.
Then, the iterates are globally convergent, i.e., $\lim_{k \to \infty} \|g_k\| = 0$.

### 3.1 The global convergence properties of $\beta^{A1}_k$

**Theorem 3.2** Suppose that Assumption 1 holds. Then, by considering the CG method of the form (1.2), (1.3), and (2.1), where $\alpha_k$ is computed by (1.5) and (1.6) and the sufficient
descent condition holds, we multiply (1.2) by \( g_k^T \), which yields

\[
g_k d_k = -\|g_k\|^2 + \frac{\|g_k\|^2 - \mu_k g_k^T g_k}{m |g_k^T d_k-1| + \|g_k-1\|^2} |g_k^T d_k|
\]

\[
\leq -\|g_k\|^2 + \frac{\|g_k\|^2 - \mu_k g_k^T g_k}{m |g_k^T d_k-1| + \|g_k-1\|^2} |g_k^T d_k|
\]

\[
\leq -\|g_k\|^2 + \frac{\|g_k\|^2 - \mu_k g_k^T g_k}{m |g_k^T d_k-1|} |g_k^T d_k|
\]

\[
\leq -\|g_k\|^2 + \frac{\|g_k\|^2 - \mu_k g_k^T g_k}{m |g_k^T d_k-1|} |g_k^T d_k|
\]

\[
\leq \|g_k\|^2 \left( -1 + \frac{1}{m} \right).
\]

**Theorem 3.3** Suppose that Assumption 1 holds. Consider the CG method of the form (1.2), (1.3), and (2.3), where \( \alpha_k \) is computed by (1.5) and (1.6), then \( \beta^{A1}_k \) satisfies Property*.

**Proof** Let \( \lambda = \frac{\gamma^2}{2(L+1)\gamma b} \) and

\[
\beta^{A1}_k = \frac{\|g_k\|^2 - \mu_k g_k^T g_k - \|g_k\|^2}{m |g_k^T d_k-1| + \|g_k-1\|^2} \leq \frac{\|g_k\|^2 + |g_k^T g_k|}{\|g_k-1\|^2} \leq \frac{2\gamma^2}{\gamma^2} = b > 1.
\]

To show that \( \beta^{A1}_k \leq \frac{1}{\gamma b} \), we have the following two cases:

**Case 1:** \( \mu_k > 1 \)

\[
\beta^{A1}_k = \frac{\|g_k\|^2 - \mu_k g_k^T g_k - \|g_k\|^2}{m |g_k^T d_k-1| + \|g_k-1\|^2} \leq \frac{\|g_k\|^2 - |g_k^T g_k|}{\|g_k-1\|^2} \leq \frac{L\lambda \gamma}{\gamma^2}.
\]

**Case 2:** \( \mu_k < 1 \)

To satisfy Property* for \( \beta^{A1}_k \) with \( \mu_k < 1 \), we need the following inequality:

\[
\|w_k\| + \|v_k\| \leq L\|w_k + v_k\|,
\]

where \( w_k = g_k - \frac{1}{L} g_k - 1 \), and \( v_k = \frac{1}{L} g_k - 1 \), which yields

\[
|\beta^{A1}_k| \leq \frac{\|g_k\|^2 - \mu_k g_k^T g_k - \|g_k\|^2}{m |g_k^T d_k-1| + \|g_k-1\|^2} \leq \frac{\|g_k\|^2 - |g_k^T g_k|}{\|g_k-1\|^2} \leq \frac{L\lambda \gamma}{\gamma^2}.
\]

Using (3.3), we obtain

\[
\left\| \frac{g_k - 1}{L} g_k - 1 \right\| \leq L \left\| g_k - 1 \right\| + \frac{1}{L} g_k - 1 \leq (L + 1) \|g_k - 1\|,
\]

\[
|\beta^{A1}_k| \leq \frac{(L + 1)\|g_k\| \|g_k - 1\|}{\|g_k-1\|^2} \leq L (L + 1)\lambda \gamma \frac{1}{\gamma^2}.
\]
Thus, in all cases
\[ |\beta_k^{A1}| \leq \frac{L \bar{\gamma}}{\gamma^2} \leq \frac{L(L + 1) \bar{\gamma}}{\gamma^2} \leq \frac{1}{2b}. \]

The proof is completed. \( \square \)

**Theorem 3.4** Suppose that Assumption 1 holds. Consider the CG method of the form (1.2), (1.3), and (2.3), where \( \alpha_k \) is computed by (1.5) and (1.6), then \( \lim_{k \to \infty} \|g_k\| = 0. \)

**Proof** We will apply Theorem 3.1. Note that the following properties hold for \( \beta_k^{A1} \):

i. \( \beta_k^{A1} > 0 \).

ii. \( \beta_k^{A1} \) satisfies Property\( ^* \) using Theorem 3.3.

iii. \( \beta_k^{A1} \) satisfies the descent property using Theorem 3.2.

iv. Assumption 1 holds.

Thus, all properties in Theorem 3.1 are satisfied, which leads to \( \lim_{k \to \infty} \|g_k\| = 0. \) \( \square \)

### 3.2 The global convergence properties of \( \beta_k^{A2} \)

**Theorem 3.5** Suppose Assumption 1 holds. Consider the CG method of the form (1.2), (1.3), and (2.3), where \( \alpha_k \) is computed by (1.5) and (1.6), and where the sufficient descent condition holds for \( \beta_k^{A2} \). Since \( d_{k-1}^T y_{k-1} \geq 0 \), we obtain

\[
g_k d_k = -\|g_k\|^2 + \frac{\|g_k\|^2 - \mu_k |g_k^T g_{k-1}|}{m |g_k^T d_{k-1}| + d_{k-1}^T y_{k-1}} g_k^T d_{k-1}
\leq -\|g_k\|^2 + \frac{\|g_k\|^2}{m |g_k^T d_{k-1}|} |g_k^T d_{k-1}|
\leq -\|g_k\|^2 + \frac{\|g_k\|^2}{m |g_k^T d_{k-1}|} |g_k^T d_{k-1}|
\leq \|g_k\|^2 \left( -1 + \frac{1}{m} \right). \]

**Theorem 3.6** Suppose that Assumption 1 holds. Consider the CG method of the form (1.2), (1.3), and (2.3), where \( \alpha_k \) is computed by (1.5) and (1.6), then the iterates \( \beta_k^{A2} \) satisfy Property\( ^* \).

**Proof** Let \( \lambda = \frac{(1-\sigma)\bar{\gamma}^2}{2(1+\gamma)} \) and

\[
\beta_k^{A2} = \frac{\|g_k\|^2 - \mu_k |g_k^T g_{k-1}|}{m |g_k^T d_{k-1}| + d_{k-1}^T y_{k-1}} \leq \frac{\|g_k\|^2 - \mu_k |g_k^T g_{k-1}|}{d_{k-1}^T y_{k-1}}
\leq \frac{\|g_k\|^2 - \mu_k |g_k^T g_{k-1}|}{(1-\sigma)c \|g_{k-1}\|^2} \leq \frac{\|g_k\|^2 + |g_k^T g_{k-1}|}{(1-\sigma)c \|g_{k-1}\|^2}
\leq \frac{\|g_k\|(\|g_k\| + \|g_{k-1}\|)}{(1-\sigma)c \|g_{k-1}\|^2} \leq \frac{2\bar{\gamma}^2}{(1-\sigma)c \gamma^2} = b > 1.
\]

To show that \( \beta_k^{A2} \leq \frac{1}{2b} \), we have the following two cases:
Case $\mu_k > 1$

$$
\beta_{k}^{A2} = \frac{\|g_k\|^2 - \mu_k |g_k^T g_{k-1}|}{m |g_k^T d_{k-1}| + \hat{d}_k^T Y_{k-1}} \leq \frac{\|g_k\|^2 - |g_k^T g_{k-1}|}{\hat{d}_k^T Y_{k-1}}
$$

$$
\leq \frac{\|g_k\| \Vert g_k - g_{k-1}\|}{(1-\sigma)c\|g_{k-1}\|^2} \leq \frac{L\bar{\gamma}}{(1-\sigma)c\gamma^2}.
$$

Case $\mu_k < 1$

To satisfy Property* for $\beta_{k}^{A1}$ with $\mu_k < 1$, we need property (3.3) which gives

$$
|\beta_{k}^{A2}| \leq \frac{|\|g_k\|^2 - \mu_k |g_k^T g_{k-1}||}{m |g_k^T d_{k-1}| + \hat{d}_k^T Y_{k-1}}
$$

$$
\leq \frac{\|g_k\|^2 - \frac{1}{L} |g_k^T g_{k-1}|}{\hat{d}_k^T Y_{k-1}}
$$

$$
\leq \frac{\|g_k\| \Vert g_k - g_{k-1}\|}{(1-\sigma)c\|g_{k-1}\|^2}.
$$

Using (3.3), we obtain

$$
\|g_k - \frac{1}{L} g_{k-1}\| \leq L \|g_k - \frac{1}{L} g_{k-1} + \frac{1}{L} g_k - g_{k-1}\| \leq (L + 1) \|g_k - g_{k-1}\|,
$$

$$
|\beta_{k}^{A2}| \leq \frac{(L + 1) \|g_k\| \|g_k - g_{k-1}\|}{(1-\sigma)c\|g_{k-1}\|^2} \leq L \frac{(L + 1)\bar{\gamma}}{(1-\sigma)c\gamma^2}.
$$

Thus, in all cases

$$
|\beta_{k}^{A2}| \leq \frac{L\bar{\gamma}}{(1-\sigma)c\gamma^2} \leq \frac{L (L + 1)\bar{\gamma}}{(1-\sigma)c\gamma^2} \leq \frac{1}{2b}.
$$

**Theorem 3.7** Suppose that Assumption 1 holds. Consider the CG method of the form (1.2), (1.3), and (2.3), i.e., $\beta_{k}^{A2}$, where $\alpha_k$ is computed by (1.5) and (1.6), then $\lim_{k \to \infty} \|g_k\| = 0$.

**Proof** We will apply Theorem 3.1. Note that the following properties hold for $\beta_{k}^{A2}$:

i. $\beta_{k}^{A2} > 0$.

ii. $\beta_{k}^{A2}$ satisfies Property* by using Theorem 3.6.

iii. $\beta_{k}^{A2}$ satisfies the descent property by using Theorem 3.5.

iv. Assumption 1 holds.

Thus all properties in Theorem 3.1 are satisfied, which leads to $\lim_{k \to \infty} \|g_k\| = 0$.

If the condition $\|g_k\|^2 > \mu_k |g_k^T g_{k-1}|$ does not hold for $\beta_{k}^{A1}$ and $\beta_{k}^{A2}$, then the CG method will be restarted using $\beta_{k}^{D-H} = -\mu_k \frac{g_k^T g_{k-1}}{\hat{d}_k^T Y_{k-1}}$.

The following two theorems show that the CG method with $\beta_{k}^{D-H}$ has the descent and convergence properties.

**Theorem 3.8** Let sequences $\{x_k\}$ and $\{d_k\}$ be obtained using Eqs. (1.2) and (1.3), which is computed by SWP line search in Eqs. (1.5) and (1.7), then the descent condition holds for $\{d_k\}$ with $\beta_{k}^{D-H}$.
Proof. By multiplying Eq. (1.3) with \( g_k^T \), and substituting \( \beta_k^{D-H} \), we obtain
\[
g_k^T d_k = -\|g_k\|^2 - t \frac{g_k^T s_k - 1}{d_k^T y_{k-1}} g_k^T d_{k-1}
\]
\[
= -\|g_k\|^2 - t \alpha_k \frac{\|g_k^T d_{k-1}\|^2}{d_k^T y_{k-1}} \leq -\|g_k\|^2.
\]

Letting \( c = 1 \), we then obtain
\[
g_k^T d_k \leq -c\|g_k\|^2,
\]
which completes the proof. \( \square \)

**Theorem 3.9** Assume that Assumption 1 holds. Consider the conjugate gradient method in (1.2) and (1.3) with \( \beta_k^{D-H} \) a descent direction and \( \alpha_k \) obtained by the strong Wolfe line search. Then, \( \lim \inf_{k \to \infty} \|g_k\| = 0 \).

Proof. We will prove this theorem by contradiction. Suppose Theorem 3.4 is not true. Then, a constant \( \varepsilon > 0 \) exists such that
\[
\|g_k\| \geq \varepsilon, \quad \forall k \geq 1. \tag{3.4}
\]

By squaring both sides of (1.2), we obtain
\[
\|d_k\|^2 = \|g_k\|^2 - 2\beta_k g_k^T d_{k-1} + \beta_k^2 \|d_{k-1}\|^2
\]
\[
\leq \|g_k\|^2 + 2\beta_k \|g_k^T d_{k-1}\| + \beta_k^2 \|d_{k-1}\|^2
\]
\[
\leq \|g_k\|^2 + 2 \|g_k\| \|s_k\| \frac{1}{L(1-\sigma)} g_k^T d_{k-1} + \frac{1}{L^2} \|s_{k-1}\|^2
\]
\[
\leq \|g_k\|^2 + \frac{2 \|g_k\| \|s_k\| \sigma}{L(1-\sigma)} + \frac{1}{L^2} \|s_{k-1}\|^2
\]
\[
\|d_k\|^2 \leq \frac{1}{\|g_k\|^2} \left( 1 + \frac{2 \|s_k\|}{L(1-\sigma)} + \frac{1}{L^2} \|s_{k-1}\|^2 \right)
\]
\[
\|d_k\|^2 \leq \frac{1}{\|g_k\|^2} \left( 1 + \frac{2 \|s_k\|}{L(1-\sigma)} + \frac{1}{L^2} \|s_{k-1}\|^2 \right)
\]
\[
\|d_k\|^2 \leq \frac{1}{\|g_k\|^2} \left( 1 + \frac{2 \|s_k\|}{L(1-\sigma)} + \frac{1}{L^2} \|s_{k-1}\|^2 \right)
\]
\[
\|d_k\|^2 \leq \frac{1}{\|g_k\|^2} \left( 1 + \frac{2 \|s_k\|}{L(1-\sigma)} + \frac{1}{L^2} \|s_{k-1}\|^2 \right)
\]
\[
\|d_k\|^2 \leq \frac{1}{\|g_k\|^2} \left( 1 + \frac{2 \|s_k\|}{L(1-\sigma)} + \frac{1}{L^2} \|s_{k-1}\|^2 \right)
\]

Let
\[
\|g_k\|^q = \min\{\|g_k\|^2, \|g_k\|^3, \|g_k\|^4\}, \quad q \in \mathbb{N},
\]
then
\[
\|d_k\|^2 \leq \frac{1}{\|g_k\|^q} \left( 1 + \frac{2 \|s_k\|}{L(1-\sigma)} + \frac{1}{L^2} \|s_{k-1}\|^2 \right).
\]
Also, let
\[ R = \left( 1 + \frac{2}{L} \frac{\lambda}{(1 - \sigma)^2} + \frac{1}{\lambda^2 (1 - \sigma)^2} \right), \]
then
\[
\frac{\|d_k\|^2}{\|g_k\|^2} \leq \frac{R}{\|g_k\|^q} \leq \sum_{i=1}^{k} \frac{1}{\|g_i\|^q},
\]
\[
\frac{\|g_k\|^q}{\|d_k\|^2} \geq \frac{\epsilon^q}{kR}.
\]
Therefore,
\[
\sum_{k=0}^{\infty} \frac{\|g_k\|^4}{\|d_k\|^2} = \infty.
\]

4 Numerical results and discussions
To analyze the efficiency of the new CG method, several test functions are selected from CUTE [20], as shown in the Appendix. These functions can be obtained from the following website:

http://ccpforge.cse.rl.ac.uk/gf/project/cutest/wiki/

In the Appendix, the following notations are defined as follows:
- No. iter means the number of iterations.
- No. fun. Eva means the number of function evaluations.
- No. Grad. Eva means the number of gradient evaluations.

Figure 4: Performance profile based on the number of iteration.
The comparison was made with respect to CPU time, the number of function evaluations, the number of iterations, and the number of gradient evaluations. The SWP line search is employed with the following parameters of $\delta = 0.01$ and $\sigma = 0.1$. The modified CG-Descent 6.8 with zero memory is employed to obtain the result for $\beta_k^{A1}$, $\beta_k^{A2}$. The code can be downloaded from the Hager webpage:

http://users.clas.ufl.edu/hager/papers/Software/
A minimum time of 0.02 seconds is used for all algorithms. The host computer is an Intel® Dual-Core CPU with 2 GB of DDR2 RAM. The results are shown in Figs. 4, 5, 6, and 7, in which a performance measure introduced by Dolan and Moré [21] was employed.

It is clear that based on the left-hand side of Figs. 4, 5, 6, and 7, the CG method A1 is above the other curves. Therefore, it is the most efficient method among related AZPRP methods. However, CG method A2 is not as efficient as A1. Still, it is more efficient than AZPRP with respect to CPU time, the number of function evaluations, gradient evaluations, and the number of iterations. In addition, as an application of the CG method in image restoration, the reader can refer to the following references [22–24].

5 Conclusion
In this paper, we proposed two efficient conjugate gradient methods related to the AZPRP method. The two methods satisfied global convergence properties and the descent property when SWP line searches were employed. Furthermore, our numerical results showed that the new methods are more efficient than the AZPRP method with respect to the number of iterations, gradient evaluations, function evaluations, and CPU time.
## Appendix

| Function | A1 CG method | A2 CG method | AZPRPG method |
|----------|--------------|--------------|---------------|
| No. Iter | No. funEva. | No. GradEva. | CPU Time      |
| AKIVA    | 2            | 8            | 20            | 15             | 0.02 |
| ALLINITU | 4            | 9            | 18            | 15             | 0.02 |
| ARGLINA  | 200          | 6            | 16            | 12             | 0.02 |
| ARWHEAD  | 6            | 3            | 16            | 12             | 0.02 |
| BARD     | 5000         | 2             | 32            | 32             | 6   |
| BOOKE    | 2            | 11           | 33            | 26             | 0.02 |
| BORG     | 6            | 24           | 64            | 44             | 0.02 |
| BOX10,000| 103          | 180          | 143           | 0.47           |
| BDQRTIC  | 5000         | 84           | 157           | 115            | 0.44 |
| BEALE    | 2            | 11           | 33            | 26             | 0.02 |
| BIGGS6   | 6            | 24           | 64            | 44             | 0.02 |
| BOX3     | 3            | 31           | 02            | 3              |
| BOX10,000| 7            | 25           | 21            | 0.14           |
| BROWN    | 200          | 51           | 11            | 6              |
| BROWNL   | 6            | 24           | 64            | 44             | 0.02 |
| BROWNBS  | 4            | 318          | 635           | 393            | 0.7 |
| BROYDN   | 5000         | 84           | 157           | 115            | 0.44 |
| CHAINOO  | 5000         | 318          | 635           | 393            | 0.7 |
| CHANNO   | 2            | 10           | 46            | 39             | 0.02 |
| CUBE     | 21           | 74           | 8             | 3              |
| CURVY10  | 2            | 17           | 48            | 34             | 0.02 |
| CURVY20  | 10,000       | 50,576       | 70,093        | 81,622         |
| DECOWU   | 10,000       | 74,906       | 97,403        | 165,28         |
| DECHNA   | 2            | 313          | 637           | 367            | 0.02 |
| DENSCHE  | 3            | 14           | 46            | 40             | 0.02 |
| Function     | Dim | A1 CG method |          |          |          | A2 CG method |          |          |          | AZPRP CG method |          |          |          |          |
|--------------|-----|--------------|----------|----------|----------|--------------|----------|----------|----------|-----------------|----------|----------|----------|----------|
|              |     | No. Iter     | No. funEva | No. GradEva | CPU Time | No. Iter | No. funEva | No. GradEva | CPU Time | No. Iter | No. funEva | No. GradEva | CPU Time |
| DENSCHNF     | 2   | 9            | 31        | 26        | 0.02     | 9         | 31         | 26         | 0.02     | 9         | 31         | 26         | 0.02     |
| DIXMAANA     | 3000| 6            | 15        | 11        | 0.02     | 6         | 15         | 11         | 0.02     | 6         | 15         | 11         | 0.02     |
| DIXMAANB     | 3000| 6            | 16        | 12        | 0.02     | 6         | 16         | 12         | 0.02     | 6         | 16         | 12         | 0.02     |
| DIXMAANC     | 3000| 6            | 14        | 9         | 0.02     | 6         | 14         | 9          | 0.02     | 6         | 14         | 9          | 0.02     |
| DIXMAAND     | 3000| 7            | 17        | 12        | 0.02     | 7         | 17         | 12         | 0.02     | 6         | 15         | 11         | 0.02     |
| DIXMAANE     | 3000| 218          | 245       | 419       | 0.23     | 256       | 283        | 495       | 0.3      | 218       | 242        | 422       | 0.3      |
| DIXMAANG     | 3000| 170          | 345       | 178       | 0.16     | 129       | 263        | 137       | 0.12     | 174       | 353        | 182       | 0.14     |
| DIXMAANH     | 3000| 176          | 358       | 185       | 0.16     | 186       | 377        | 194       | 0.14     | 173       | 353        | 184       | 0.14     |
| DIXMAANI     | 3000| 2994         | 3083      | 5909      | 3.28     | 3174      | 3248       | 6824      | 3.7      | 3264      | 3359       | 6443      | 3.3      |
| DIXMAANJ     | 3000| 363          | 731       | 371       | 0.28     | 345       | 695        | 353       | 0.31     | 384       | 773        | 392       | 0.31     |
| DIXMAANK     | 3000| 304          | 613       | 312       | 0.25     | 398       | 801        | 406       | 0.31     | 401       | 806        | 408       | 0.34     |
| DIXMAANL     | 3000| 342          | 691       | 353       | 0.27     | 379       | 765        | 390       | 0.31     | 430       | 867        | 441       | 0.49     |
| DIXON3DQ     | 10000| 10,000      | 10,007    | 19        | 0.78     | 10,000    | 10,007     | 19,995    | 19.12    | 19,995    | 19,995     | 19,12     | 0.78     |
| DJTL         | 2   | 75           | 1163      | 1148      | 0.02     | 75        | 1163       | 1148      | 0.02     | 75        | 1163       | 1148      | 0.02     |
| DQDRTIC      | 5000| 5            | 11        | 6         | 0.02     | 5         | 11         | 6         | 0.02     | 5         | 11         | 6         | 0.02     |
| DQEELT       | 5000| 15           | 32        | 18        | 0.03     | 15        | 32         | 18        | 0.02     | 15        | 32         | 18        | 0.02     |
| EDENSCH      | 2000| 31           | 70        | 54        | 0.05     | 34        | 74         | 67        | 0.06     | 30        | 67         | 57        | 0.08     |
| EG2          | 1000| 3            | 8         | 5         | 0.02     | 3         | 8          | 5         | 0.02     | 3         | 8          | 5         | 0.02     |
| EIGENALS     | 2550| 8785         | 16,438    | 9953      | 141      | 17,061    | 32,415     | 18,795    | 280      | 11,275    | 20,477     | 13,384    | 194      |
| EIGENBLS     | 2550| 19,480       | 38,968    | 19,489    | 284      | 234       | 548        | 335       | 5.14     | 34599     | 69,207     | 34,609    | 589      |
| EIGENCLS     | 2652| 11,704       | 23,434    | 11,740    | 180.05   | 9800      | 19,062     | 10,385    | 162.48   | 9838      | 18,888     | 10,710    | 185      |
| ENVAL1       | 5000| 23           | 45        | 40        | 0.05     | 23        | 47         | 41        | 0.06     | 23        | 45         | 40        | 0.06     |
| ENVAL2       | 5    | 26           | 73        | 55        | 0.02     | 26        | 73         | 55        | 0.02     | 26        | 73         | 55        | 0.02     |
| ERRINROS     | 50  | 102,326      | 201,278   | 1E+05     | 2.78     | 104       | 260        | 178       | 0.02     | 82,469    | 2E+05      | 86,569    | 2.08     |
| EXPFIT       | 2   | 9            | 22        | 22        | 0.02     | 9         | 29         | 22        | 0.02     | 9         | 29         | 22        | 0.02     |
| EXTROSNB     | 1000| 2359         | 5279      | 3112      | 0.8      | 69        | 193        | 145       | 0.03     | 2205      | 4964       | 2945      | 0.86     |
| FLETCHBV2    | 5000| 1            | 1         | 1         | 0.02     | 1         | 1          | 1         | 0.02     | 1         | 1          | 1         | 0.02     |
| FLETCHCR     | 1000| 71           | 153       | 85        | 0.03     | 29        | 68         | 43        | 0.05     | 88        | 178        | 114       | 0.03     |
| FMINSRF2     | 5625| 426          | 875       | 453       | 1.31E+00 | 1803      | 3546       | 1849      | 4.59E+00 | 459       | 940        | 486       | 1.58E+00 |
| FMINSURF     | 5625| 562          | 1140      | 584       | 1.67     | 1327      | 2705       | 1384      | 4.08     | 548       | 1118       | 575       | 2.06     |
| Function   | A1 CG method | A2 CG method | AZPRP CG method |
|------------|--------------|--------------|-----------------|
| FREUROTH 5000 | 34 | 86 | 0.17 |
| GENHUMPS 1000 | 21 | 54 | 0.09 |
| GENROSE 2000 | 1831 | 30,411 | 46,977 |
| GROWTHLS 3 | 109 | 199 | 165 |
| HATFLDFL | 10 | 4 | 2 |
| HEART6LS 1000 | 375 | 1137 | 876 |
| HELIX 2 | 23 | 60 | 0.02 |
| HUMPS 2 | 45 | 223 | 179 |
| JIMACK 2 | 17 | 21 | 1.6 |
| KOWOSB 4000 | 1641 | 1088 | 55 |

Continued
| Function      | Dim | A1 CG method | A2 CG method | AZPRP CG method |
|--------------|-----|--------------|--------------|-----------------|
|              |     | No. Iter | No. fun Eva | No. Grad Ev  | No. fun Eva | No. Grad Ev | CPU Time | No. Iter | No. fun Eva | No. Grad Ev | CPU Time |
| NCB20B       | 500 | 4994    | 7577        | 10892       | 80.22 | 99      | 216       | 202       | 1.59 | 4912    | 7503        | 10461     | 79.45    |
| NCB20        | 5010| 908    | 1965        | 1477        | 11.94 | 216     | 470       | 371       | 3.05 | 1074    | 2459        | 1545      | 13.16    |
| NONCVXU2     | 5000| 6864   | 13,196      | 7400        | 16.33 | 6580    | 12,710    | 7032      | 15.19 | 7477    | 14009       | 8428      | 17.9     |
| NONDIA       | 5000| 7      | 25          | 19          | 0.02  | 7       | 25        | 19        | 0.03 | 7      | 25          | 19        | 0.03     |
| NONDQUAR     | 5000| 616    | 1372        | 868         | 0.78  | 1423    | 3001      | 1663      | 1.75 | 2562    | 5235        | 2743      | 2.95     |
| OSBORNEA     | 5   | 82     | 230         | 174         | 0.02  | 82      | 230       | 174       | 0.02 | 82     | 230         | 174       | 0.02     |
| OSBORNEB     | 11  | 57     | 134         | 84          | 0.02  | 57      | 134       | 84        | 0.02 | 57     | 134         | 84        | 0.02     |
| OSCIPATH     | 10  | 2950,29 | 781,729    | 5E+05      | 2.16  | 3E+05  | 781,729   | 534,425   | 2.23 | 2950,29 | 8E+05       | 5E+05     | 2.23     |
| PALMER1C     | 8   | 12     | 27          | 28          | 0.02  | 12      | 27        | 28        | 0.02 | 12     | 27          | 28        | 0.02     |
| PALMER1D     | 7   | 10     | 24          | 23          | 0.02  | 10      | 24        | 23        | 0.02 | 10     | 24          | 23        | 0.02     |
| PALMER2C     | 8   | 11     | 22          | 22          | 0.02  | 11      | 22        | 22        | 0.02 | 11     | 22          | 22        | 0.02     |
| PALMER3C     | 8   | 11     | 21          | 21          | 0.02  | 11      | 21        | 21        | 0.02 | 11     | 21          | 21        | 0.02     |
| PALMER4C     | 8   | 11     | 21          | 21          | 0.02  | 11      | 21        | 21        | 0.02 | 11     | 21          | 21        | 0.02     |
| PALMER5C     | 6   | 6      | 13          | 7           | 0.02  | 6       | 13        | 7         | 0.02 | 6      | 13          | 7         | 0.02     |
| PALMER6C     | 8   | 11     | 24          | 24          | 0.02  | 11      | 24        | 24        | 0.02 | 11     | 24          | 24        | 0.02     |
| PALMER7C     | 8   | 11     | 20          | 20          | 0.02  | 11      | 20        | 20        | 0.02 | 11     | 20          | 20        | 0.02     |
| PALMER8C     | 8   | 11     | 19          | 19          | 0.02  | 11      | 19        | 19        | 0.02 | 11     | 19          | 19        | 0.02     |
| PARKH        | 15  | 740    | 1513        | 1404        | 35.83 | 15      | 59        | 134       | 0.02 | 15     | 59          | 134       | 0.02     |
| PENALTY1     | 1000| 15     | 61          | 56          | 0.02  | 41      | 164       | 144       | 0.02 | 43     | 168         | 146       | 0.02     |
| PENALTY2     | 200 | 215    | 263         | 421         | 0.03  | 212     | 247       | 404       | 0.05 | 200    | 243         | 386       | 0.03     |
| PENALTY3     | 200 | 99     | 330         | 275         | 2.06  | 32      | 105       | 86        | 0.64 | 83     | 278         | 236       | 1.86     |
| POWELL5G     | 5000| 20     | 49          | 34          | 0.01  | 34      | 84        | 58        | 0.02 | 28     | 72          | 49        | 0.03     |
| POWER        | 10,000| 456   | 933         | 488         | 0.81  | 325     | 890       | 637       | 0.89 | 544    | 1119        | 592       | 1        |
| QUARTC       | 5000| 15     | 32          | 18          | 0.02  | 15      | 32        | 18        | 0.03 | 15     | 32          | 18        | 0.02     |
| ROSENBR      | 2   | 28     | 84          | 65          | 0.02  | 28      | 84        | 65        | 0.02 | 28     | 84          | 65        | 0.02     |
| Function     | Dim | A1 CG method | A2 CG method | AZPRPCG method |
|--------------|-----|--------------|--------------|----------------|
|              |     | No. Iter | No. fun Eva | No. Grad Eva | CPU Time | No. Iter | No. fun Eva | No. Grad Eva | CPU Time | No. Iter | No. fun Eva | No. Grad Eva | CPU Time |
| S308         | 2   | 7        | 21          | 17           | 0.02     | 7        | 21          | 17           | 0.02     | 7        | 21          | 17           | 0.02     |
| SCHMVETT     | 5000| 41       | 71          | 58           | 0.25     | 41       | 71          | 58           | 0.22     | 37       | 67          | 50           | 0.17     |
| SENSORS      | 100 | 46       | 116         | 75           | 0.61     | 35       | 97          | 69           | 0.55     | 51       | 131         | 88           | 0.78     |
| SINEVAL      | 2   | 46       | 181         | 153          | 0.02     | 46       | 181         | 153          | 0.02     | 46       | 181         | 153          | 0.02     |
| SINOQUAD     | 5000| 13       | 44          | 38           | 0.08     | 14       | 45          | 39           | 0.08     | 14       | 51          | 44           | 0.11     |
| SISPER       | 2   | 5        | 19          | 19           | 0.02     | 5        | 19          | 19           | 0.02     | 5        | 19          | 19           | 0.02     |
| SNAIL        | 2   | 61       | 251         | 211          | 0.02     | 61       | 251         | 211          | 0.02     | 61       | 251         | 211          | 0.02     |
| SPARSINE     | 5000| 22,466   | 22,744      | 44,664       | 83.92    | 21,468   | 21,760      | 42,654       | 83       | 21,700   | 22,006      | 43,104      | 84.5     |
| SPARSQUR     | 10,000| 37     | 158         | 148          | 0.91     | 52       | 205         | 188          | 0.84     | 34       | 143         | 136          | 0.84     |
| SPMSRTLS     | 4999| 216      | 439         | 229          | 0.47     | 252      | 501         | 275          | 0.61     | 213      | 435         | 224          | 0.47     |
| SROSENBR     | 5000| 9        | 23          | 15           | 0.02     | 9        | 23          | 15           | 0.02     | 9        | 23          | 15           | 0.02     |
| STRATEC      | 10  | 170      | 419         | 283          | 6.11     | 170      | 419         | 283          | 6.2      | 170      | 419         | 283          | 6.17     |
| TESTQUAD     | 5000| 1543     | 1550        | 3081         | 1.25E+00 | 1515     | 3025        | 1.25         | 1.52E+00 | 1573     | 1580        | 3141         | 1.34E+00 |
| TOINTGOR     | 50  | 118      | 214         | 152          | 0.02     | 123      | 220         | 163          | 0.02     | 120      | 215         | 155          | 0.02     |
| TOINTGSS     | 5000| 4        | 9           | 5            | 0.02     | 4        | 9           | 5            | 0.02     | 4        | 9           | 5            | 0.02     |
| TOINTPSP     | 50  | 143      | 336         | 254          | 0.02     | 26       | 101         | 90           | 0.02     | 140      | 326         | 245          | 0.02     |
| TOINTQOR     | 50  | 29       | 36          | 53           | 0.02     | 29       | 36          | 53           | 0.02     | 29       | 36          | 53           | 0.02     |
| TQUARTIC     | 5000| 12       | 44          | 36           | 0.03     | 11       | 37          | 29           | 0.03     | 11       | 38          | 30           | 0.03     |
| TRIDIA       | 5000| 783      | 790         | 1561         | 0.89     | 782      | 789         | 1559         | 0.91     | 783      | 790         | 1561         | 0.89     |
| VARDIM       | 9   | 23       | 18          | 18           | 0.02     | 10       | 24          | 17           | 0.02     | 10       | 23          | 18           | 0.02     |
| VAREIGVL     | 50  | 24       | 51          | 29           | 0.02     | 24       | 51          | 29           | 0.02     | 23       | 49          | 28           | 0.02     |
| VIBBREAM     | 8   | 98       | 255         | 174          | 0.02     | 98       | 255         | 174          | 0.02     | 98       | 255         | 174          | 0.02     |
| WATSON       | 12  | 53       | 124         | 78           | 0.02     | 49       | 127         | 88           | 0.02     | 37       | 130         | 78           | 0.02     |
| WOODS        | 4000| 24       | 60          | 40           | 0.05     | 23       | 59          | 40           | 0.03     | 22       | 57          | 41           | 0.03     |
| YFITU        | 3   | 68       | 208         | 167          | 0.02     | 68       | 208         | 167          | 0.02     | 68       | 208         | 167          | 0.02     |
| ZANGWL2      | 2   | 1        | 3           | 2            | 0.02     | 1        | 3           | 2            | 0.02     | 1        | 3           | 2            | 0.02     |
Acknowledgements
The authors are grateful for all this support and improve our paper; also, we would like to thank the University of Malaysia Terengganu (UMT) for funding this paper.

Funding
This study was partially supported by the Universiti Malaysia Terengganu, Centre of Research and Innovation Management.

Availability of data and materials
The data is available inside the paper.

Declarations

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
The authors contributed equally and significantly in writing this paper. All authors read and approved the final manuscript.

Author details
1 Department of Mathematics, Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu, 21030 Kuala Nerus, Terengganu, Malaysia. 2 Department of Mathematics, Faculty of Science, Jazan University, Jazan, Saudi Arabia.

Publisher's Note
Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 17 March 2021 Accepted: 22 December 2021 Published online: 10 January 2022

References
1. Wolfe, P.: Convergence conditions for ascent methods. SIAM Rev. 11(2), 226–235 (1969)
2. Wolfe, P.: Convergence conditions for ascent methods. II: some corrections. SIAM Rev. 13(2), 185–188 (1971)
3. Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. 49(6), 409–436 (1952)
4. Polak, E., Ribiere, G.: Note sur la convergence de méthodes de directions conjuguées. ESAIM: Math. Model. Numer. Anal. 3(1), 35–43 (1969)
5. Liu, Y., Storey, C.: Efficient generalised conjugate gradient algorithms, part 1: theory. J. Optim. Theory Appl. 69(1), 129–137 (1991)
6. Fletcher, R., Reeves, C.M.: Function minimisation by conjugate gradients. Comput. J. 7(2), 149–154 (1964)
7. Fletcher, R.: Practical Method of Optimisation. Unconstrained Optimisation, edn. (1997)
8. Dai, Y.H., Yuan, Y.: A non-linear conjugate gradient method with a strong global convergence property. SIAM J. Optim. 10(1), 177–182 (1999)
9. Zoutendijk, G.: Non-linear programming, computational methods. In: Integer and Non-linear Programming, pp. 37–86 (1970)
10. Al-Baali, M.: Descent property and global convergence of the Fletcher–Reeves method with inexact line search. IMA J. Numer. Anal. 8(1), 121–124 (1988)
11. Powell, M.J.: Non-convex minimisation calculations and the conjugate gradient method. In: Numerical Analysis, pp. 122–141. Springer, Berlin (1984)
12. Gilbert, J.C., Nocedal, J.: Global convergence properties of conjugate gradient methods for optimisation. SIAM J. Optim. 2(1), 21–42 (1992)
13. Dai, Y.H., Liao, L.Z.: New conjugacy conditions and related non-linear conjugate gradient methods. Appl. Math. Optim. 43(1), 87–101 (2001)
14. Hager, W.W., Zhang, H.: A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. Optim. 16(1), 170–192 (2005)
15. Hager, W.W., Zhang, H.: The limited memory conjugate gradient method. SIAM J. Optim. 23(4), 2150–2168 (2013)
16. Wei, Z., Yao, S., Liu, L.: The convergence properties of some new conjugate gradient methods. Appl. Math. Comput. 183(2), 1341–1350 (2006)
17. Dai, Z., Wen, F.: Another improved Wei–Yao–Liu non-linear conjugate gradient method with sufficient descent property. Appl. Math. Comput. 210(4), 7421–7430 (2012)
18. Alhawarat, A., Salleh, Z., Mamat, M., Rivaie, M.: An efficient modified Polak–Ribière–Polyak conjugate gradient method with global convergence properties. Optim. Methods Softw. 32(6), 1299–1312 (2017)
19. Kaelo, P., Mtagulwa, P., Thuto, M.V.: A globally convergent hybrid conjugate gradient method with strong Wolfe conditions for unconstrained optimisation. Math. Sci. 14(1), 1–9 (2020)
20. Bongartz, I., Conn, A.R., Gould, N., Toint, P.L.: CUTE: constrained and unconstrained testing environment. ACM Trans. Math. Softw. 21(1), 123–160 (1995)
21. Dolan, E.D., Moré, J.J.: Benchmarking optimisation software with performance profiles. Math. Program. 91(2), 201–213 (2002)
22. Alhawarat, A., Salleh, Z., Masmali, I.A.: A convex combination between two different search directions of conjugate gradient method and application in image restoration. Math. Probl. Eng. 2021, Article ID 9941757 (2021). https://doi.org/10.1155/2021/9941757
23. Guessab, A., Driouch, A.: A globally convergent modified multivariate version of the method of moving asymptotes.
   Appl. Anal. Discrete Math. 15(2), 519–535 (2021)
24. Guessab, A., Driouch, A., Nouisser, O.: A globally convergent modified version of the method of moving asymptotes.
   Appl. Anal. Discrete Math. 13(3), 905–917 (2019)