Research Article

A Descent Four-Term Conjugate Gradient Method with Global Convergence Properties for Large-Scale Unconstrained Optimisation Problems

Ahmad Alhawarat 1, Ghaliah Alhamzi 2, Ibitsam Masmali 3, and Zabidin Salleh 1

1Department of Mathematics, Faculty of Ocean Engineering Technology and Informatics, University Malaysia Terengganu, Kuala Terengganu, Malaysia
2Department of Mathematics and Statistics, College of Science, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia
3Department of Mathematics, College of Science, Jazan University, Jazan, Saudi Arabia

Correspondence should be addressed to Zabidin Salleh; zabidin@umt.edu.my

Received 23 June 2021; Revised 31 July 2021; Accepted 5 August 2021; Published 16 August 2021

Copyright © 2021 Ahmad Alhawarat et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The conjugate gradient method is a useful method to solve large-scale unconstrained optimisation problems and to be used in some applications in several fields such as engineering, medical science, image restorations, neural network, and many others. The main benefit of the conjugate gradient method is not using the second derivative or its approximation, such as Newton’s method or its approximation. Moreover, the algorithm of the conjugate gradient method is simple and easy to apply. This study proposes a new modified conjugate gradient method that contains four terms depending on popular two- and three-term conjugate gradient methods. The new algorithm satisfies the descent condition. In addition, the new CG algorithm possesses the convergence property. In the numerical results part, we compare the new algorithm with famous methods such as CG-Descent. We conclude from numerical results that the new algorithm is more efficient than other popular CG methods such as CG-Descent 6.8 in terms of number of function evaluations, number of gradient evaluations, number of iterations, and CPU time.

1. Introduction

To solve large-scale unconstrained optimisation problems, we prefer to use conjugate gradient (CG) since it is efficient and robust and does not use the second derivative. Consider the following problem:

$$\min f(x) \quad \mathbb{R}^n,$$

where $f: \mathbb{R}^n \to \mathbb{R}$ is a continuous and differentiable function, and the gradient is available. The CG method generates a sequence of $x_k$ given by the equation

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

The step length $\alpha_k$ was obtained using line search. $d_k$ is a search direction of the CG method defined as follows:

$$d_k = \begin{cases} -g_k, & \text{if } k = 1, \\ -g_k + \beta_k d_{k-1}, & \text{if } k \geq 2, \end{cases}$$

where $g_k = \nabla f(x_k)$ is known as the CG formula.

To obtain the step length, we use the strong Wolfe-Powell (SWP) [1, 2] line search given by the following equations:

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

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

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

The weak Wolfe–Powell (WWP) line search is given by equation (4) and
\[ g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k. \] (6)

The popular classical formulas of CG methods Hestenes–Stiefel (HS) [3], Polak–Ribiere–Polyak (PRP) [4], Liu–Storey (LS) [5], Fletcher–Reeves (FR) [6], Fletcher (CD) [7], and Dai–Yuan (DY) [8] are as follows:

\[ \beta_k^{HS} = \frac{g_k^T y_{k-1}}{d_k^T y_{k-1}}, \]
\[ \beta_k^{PRP} = \frac{g_k^T y_{k-1}}{\|g_k\|^2}, \]
\[ \beta_k^{LS} = \frac{g_k^T y_{k-1}}{d_k^T y_{k-1}}, \]
\[ \beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \]
\[ \beta_k^{CD} = \frac{\|g_k\|^2}{d_k^T y_{k-1}}, \]
\[ \beta_k^{DY} = \frac{\|g_k\|^2}{d_k^T y_{k-1}}, \]

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

Powell [9] proposed an example to show that a function does not satisfy the convergence even if the exact line search is employed with the PRP formula. Gilbert and Nocedal [10] used Powell’s suggestion for the PRP formula. They proved that if \( \beta_k^{PPR} = \max\{0, \beta_k^{PPR}\} \), the WWP line search is employed and the sufficient descent Wolfe-Powell line search condition is satisfied; then \( \beta_k^{PPR} \) is globally convergent. Moreover, Zoutendijk [11] showed that the CG formula with CG method and the exact line search has convergence analysis. Finally, Al-Baali [12] proved that the CG method with the FR coefficient is globally convergent when \( \sigma \leq 1/2 \) SWP is employed.

In 2006, Wei et al. [13] presented a nonnegative CG formula that can satisfy the descent property and global convergence property as follows:

\[ \beta_k^{WYL} = \frac{g_k^T (g_k - \|g_k\| \|g_{k-1}\| g_{k-1}}{\|g_{k-1}\|^2}, \] (8)

where \( y_{k-1} = g_k - g_{k-1}. \) In 2016, Alhawarat et al. [14] presented the following nonnegative CG formula with new restart property:

\[ \beta_k^{AZPRP} = \begin{cases} \frac{\|g_k\|^2 - \mu_k g_k^T g_{k-1}}{\|g_{k-1}\|^2}, & \text{if } \|g_k\|^2 > \mu_k g_k^T g_{k-1}, \\ 0, & \text{otherwise,} \end{cases} \] (9)

where \( \| \cdot \| \) represents the Euclidean norm and \( \mu_k \) is defined as follows:

\[ \mu_k = \frac{\|x_k - x_{k-1}\|}{\|y_{k-1}\|}. \] (10)

In 2020, Kaelo et al. [15] proposed a nonnegative CG formula and compared the new formula with \( \beta_k^{AZPRP} \). The proposed equation is as follows:

\[ \beta_k^{PKT} = \begin{cases} \frac{g_k^T g_k - g_k^T g_{k-1}}{\max\{d_k^T y_{k-1}, -g_k^T d_{k-1}\}} & \text{if } 0 < g_k^T g_{k-1} < \|g_k\|^2, \\ \frac{\|g_k\|^2}{\max\{d_k^T y_{k-1}, -g_k^T d_{k-1}\}}, & \text{otherwise.} \end{cases} \] (11)

In 2001, Dai and Liao [16] proposed the following conjugacy condition:

\[ d_k^T y_{k-1} = -t g_k^T s_{k-1}, \] (12)

where \( s_{k-1} = x_k - x_{k-1} \) and \( t \geq 0 \). In the case of \( t = 0 \), equation (13) becomes the classical conjugacy condition. By using equations (2) and (12), Dai and Liao [16] proposed the following CG formula:

\[ \beta_k^{DL} = \frac{g_k^T y_{k-1}}{d_k^T y_{k-1}} - \frac{t g_k^T s_{k-1}}{d_k^T y_{k-1}} = \beta_k^{HS} - \frac{1}{d_k^T y_{k-1}}. \] (13)

However, \( \beta_k^{DL} \) inherits the same problem as \( \beta_k^{PPR} \) and \( \beta_k^{HS} \); that is, \( \beta_k^{DL} \) is not nonnegative in general. Thus, Gilbert and Nocedal [10] replaced equation (13) by

\[ \beta_k^{DL^+} = \max\{\beta_k^{HS}, 0\} - \frac{g_k^T s_{k-1}}{d_k^T y_{k-1}}. \] (14)

In 2006, Hager and Zhang [17, 18] presented the following CG formula based on (13):

\[ \beta_k^{HZ} = \max\{\beta_k^N, \eta_k\}, \] (15)

where

\[ \beta_k^N = \frac{1}{d_k^T y_{k-1}} y_{k-1}^2 (\|g_k\|^2 - d_k^T y_{k-1})^T g_k, \]
\[ \eta_k = -\frac{1}{d_k^T y_{k-1}} \min \{\eta, g_k^T y_{k-1}\}, \]
\[ \text{and } \eta > 0 \text{ is a constant.} \]

Note that if \( t = 2 \|g_k\|^2 / d_k^T y_{k-1} \), then \( \beta_k^N = \beta_k^{DY} \).

Based on equation (13), Andrei [19, 20] proposed two three-term CG (TTCG) methods in 2013 as follows:

\[ d_k = -g_k + \frac{g_k^T y_{k-1} - t g_k^T s_{k-1}}{y_{k-1}^T d_{k-1}} d_{k-1} - \frac{g_k^T d_{k-1}}{y_{k-1}^T d_{k-1}} y_{k-1} \]
\[ t = \left(1 + \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}}\right) \]
\[ d_k = -g_k + \frac{g_k^T y_{k-1} - t g_k^T s_{k-1}}{y_{k-1}^T d_{k-1}} d_{k-1} - \frac{g_k^T d_{k-1}}{y_{k-1}^T d_{k-1}} y_{k-1} \]
\[ t = \left(1 + \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}}\right) \] (16)
the following CG method:
\[ d_k = -g_k + \left( g_k^T y_{k-1} - t_k g_k^T s_k \right) d_{k-1} - \frac{g_k^T d_{k-1}}{y_{k-1}^T d_{k-1}} y_{k-1} d_{k-1}, \]
\[ t = \left( \max \left( \zeta, 1 - \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}} \right) \right). \]

(17)

where \( \zeta > 0 \).

Similar to [19–21], Deng and Wan [22] in 2015 presented the following CG method:
\[ d_k = -g_k + \left( g_k^T y_{k-1} - t_k g_k^T s_k \right) d_{k-1} - \frac{g_k^T d_{k-1}}{y_{k-1}^T d_{k-1}} y_{k-1} d_{k-1}, \]
\[ t = \left( \max \left( 0, 1 - \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}} \right) \right). \]

(18)

In 2019, Liu et al. [23] proposed
\[ d_k^{\text{MTTPR}} = -g_k + \left( p_k^{\text{PRP}} - \frac{g_k^T s_{k-1}}{\|g_{k-1}\|^2} \right) d_{k-1} + \left( \frac{g_k^T d_{k-1}}{\|g_{k-1}\|^2} \right) g_{k-1}, \]
\[ d_k^{\text{MTTHS}} = -g_k + \left( p_k^{\text{HS}} - \frac{\|g_{k-1}\|^2 g_k^T s_{k-1}}{(d_{k-1}^T y_{k-1})^2} \right) d_{k-1} + \left( \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} \right) g_{k-1}, \]
\[ d_k^{\text{ADAP}} = -g_k + \left( g_k^T y_{k-1} - t_k \frac{g_k^T s_{k-1}}{t_{k-1}} \right) d_{k-1} + \left( \frac{g_k^T d_{k-1}}{t_{k-1}} \right) g_{k-1}, \]
\[ t_{k-1} = \max \left\{ -d_{k-1} g_{k-1}, \frac{\|g_{k-1}\|^2}{d_{k-1}^T y_{k-1}} \right\}. \]

(22)

In 2020, Yao et al. [25] proposed three terms of CG with a new choice of \( t \) as follows:
\[ d_{k+1} = -g_{k+1} + \left( g_k^T y_k - t_k \frac{g_k^T s_k}{y_k^T d_k} \right) d_k + \frac{g_k^T d_k}{y_k^T d_k} y_k d_k. \]

(23)

Based on the SWP line search, Yao et al. [25] selected \( t_k \) to satisfy the descent condition as follows:
\[ t_k > \frac{\|y_k\|^2}{y_k^T s_k}. \]

(24)

Yao et al. [25] also proposed a theorem stating that if \( t_k \) is close to \( (\|y_k\|^2/y_k^T s_k) \), then the search direction results in a zigzag search path. Therefore, they selected the following choice for \( t_k \):
\[ d_k = -g_k + \left( p_k^{\text{LS}} - \frac{g_k^T s_{k-1}}{\|g_{k-1}\|^2} \right) d_{k-1} + \left( \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} \right) g_{k-1}, \]
\[ \text{with the following assumption:} \]
\[ \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} > \nu \in (0, 1). \]

(20)

Moreover, Liu et al. [23] proved the sufficient descent condition for no convex functions to solve nonlinear monotone equations. To avoid using the condition \( \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} > \nu \in (0, 1) \), Liu et al. [24] constructed the following three-term CG method:
\[ d_k = -g_k + \left( p_k^{\text{LS}} - \frac{g_k^T s_{k-1}}{\|g_{k-1}\|^2} \right) d_{k-1} + \left( \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} \right) g_{k-1}, \]
\[ \text{and they used it to solve equation (1). Liu et al. [24] proposed wide choices using equation (21) as follows:} \]
\[ t_k = 1 + 2 \frac{\|y_k\|^2}{y_k^T s_k}. \]

(25)

As an application of the CG method, we can restore actual images from damaged images by minimising the \( \varepsilon \) algorithm with an efficient number of iterations and CPU time. Moreover, we can test the quality using the root mean square error (RMSE) between the actual image (original) and the restored image as follows:
\[ \text{rmse} = \frac{\|\zeta - \zeta_k\|^2}{\|\zeta\|^2}. \]

(26)

Here, \( \zeta \) and \( \zeta_k \) are the actual image and restored image, accordingly. Moreover, the CG method can be applied in several fields: neural network, image restoration, medical science, machine learning, finance and economics, and many other fields. The reader can refer to [26–36] for more about the CG method and its applications.
2. The New Search Direction

In the literature review, we note that the general form of the three-term CG method can be written as follows:

\[ d_k = -g_k + \eta_k d_{k-1} - \theta_k y_{k-1}, \]  

(27)

where \( \theta_k \) and \( \eta_k \) are CG parameters. Based on equation (13), we extend equation (27) to the following form:

\[ d_k^{FTCGHS} = -g_k + \left( \beta^k_{HS} - t_k \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}} \right) d_{k-1} - \left( \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} \right) (y_{k-1} + s_{k-1}), \]  

(28)

where \( t_k = (\|s_k\|/\|y_{k-1}\|) \).

In this paper, we assume the following for equation (28):

\[ \eta_k = \beta^k_{HS} - t_k \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}, \]  

(29)

\[ \theta_k = \frac{\left( \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} \right)}{\left( \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}} \right)}. \]

Note that equation (28) can be reduced to the HSCG method if we use an exact line search.

By using exact line search, we obtain

\[ g_k^T d_{k-1} = 0, \]  

(30)

and since \( g_k^T s_{k-1} = a_k g_k^T d_{k-1} \), we obtain

\[ d_k^{FTCGHS} = -g_k + \left( \beta^k_{HS} \right) d_{k-1}. \]  

(31)

Note that \( \eta_k \) is similar to \( \beta^k_{HS} \) with different values of \( t_k \), where \( \beta^k_{HS} \) inherits the conjugacy condition with inexact line search. For \( \theta_k \), it is worth noting that it contains the term \( g_k^T d_{k-1} \) and is divided by the denominator of the HS method. Meanwhile, for \( (y_{k-1} + s_{k-1}) \), the direction \( y_{k-1} \) is an essential term in obtaining the descent condition when multiplied by negative gradient. It is also useful in terms of efficiency, i.e., if this term goes to zero, the search direction will restart using the steepest descent method, avoiding equation (28) to cycle without reaching the solution.

Figure 1 describes the steps of the CG method to obtain the stationary point using SWP line search and equation (28) with stopping criteria of \( \|g_k\| \leq 10^{-6} \).

3. Global Convergence Analysis of the CG Method with Figure 1

The following assumption is crucial to satisfy the convergence analysis of CG methods.

**Assumption 1**

(A). The level set \( \Omega = \{ x | f(x) \leq f(x_k) \} \) is bounded; that is, a positive constant \( \rho \) exists such that

\[ \|x\| \leq \rho, \quad \forall x \in \Omega. \]  

(32)

(B). In some neighbourhood \( Q \) of \( \Omega \), \( f \) is continuously differentiable, and its gradient is Lipschitz continuous; that is, for all \( x, y \in Q \), there exists a constant \( L > 0 \) such that

\[ \|g(x) - g(y)\| \leq L\|x - y\|. \]  

(33)

In addition to this assumption, we can conclude that there exists a positive constant \( B \) such that

\[ \|g(u)\| \leq B, \quad \forall u \in N. \]  

(34)

The descent condition (downhill condition),

\[ g_k^T d_k < 0, \quad \forall k \geq 1, \]  

(35)

is useful in studying the CG method and plays an important role in the proof of global convergence analysis. Abaali [12] modified (35) to the following form and used it to prove the FR method:

\[ g_k^T d_k \leq -c\|g_k\|^2, \quad \forall k \geq 1, \]  

(36)

where \( c \in (0, 1) \). Equation (36) is the sufficient descent condition. Moreover, using equation (36) is better than equation (35) since we can control the quantity of \( g_k^T d_k \) by using \( \|g_k\|^2 \).

3.1. The Descent Property of the New Search Direction

The following theorem shows that the search direction in equation (28) ensures that the sufficient descent condition (35) is satisfied with the SWP line search.

**Theorem 1.** Let the sequences generate (1) and (28), computed by SWP line search. Then, the sufficient descent condition holds.

**Proof.** Multiplying (28) by \( g_k^T \) yields

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

\[ = \|g_k\|^2 + \left( \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} \right) g_k^T d_{k-1} - \left( \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}} \right) g_k^T d_{k-1} - \left( \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} \right) g_k^T y_{k-1} - \left( \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} \right) g_k^T s_{k-1}. \]
\[ -\|g_k\|^2 - \alpha_k \left( g_k^T d_{k-1} \right) g_k^T d_{k-1} - \alpha_k \left( g_k^T d_{k-1} \right) g_k^T d_{k-1} \]

\[ = -\|g_k\|^2 - \alpha_k \left( g_k^T d_{k-1} \right) \left( d_{k-1}^T y_{k-1} \right) \]

\[ = -\|g_k\|^2 \]

From Wolfe–Powell search, we obtain

\[ d_{k-1}^T y_{k-1} > 0, \]

and thus

\[ g_k^T d_k \leq -\|g_k\|^2. \]

Zoutendijk [11] presented a useful lemma for analysing the global convergence property of the CG method. The lemma is given as follows.

**Lemma 1.** Let Assumption 1 hold. Consider any CG method in the form of (1) and (2), and \( \alpha_k \) satisfies the WWP line search of (5) and (6), in which the search direction is descent. Then, the following condition holds:
\[ \lim_{k \to \infty} \frac{g_k}{d_k} = 0. \]  

3.2. Global Convergence of Figure 1 with Convex Functions.
Dai and Liao [16] presented a useful theorem to obtain the

global convergence theorem of the CG method as follows.

**Theorem 2.** Suppose that Assumption 1 holds. Consider any

conjugate gradient method in the form of equations (1) and (2), where \(d_k\) is a descent direction and \(\alpha_k\) is obtained from the

strong Wolfe line search. If

\[ \sum_{k=1}^{\infty} \frac{1}{\|d_k\|^2} = \infty, \]

then

\[ \liminf_{k \to \infty} \|g_k\| = 0. \]

The following theorem shows that the new search direction strongly satisfies the convergence analysis with convex functions.

**Theorem 3.** Suppose that Assumption 1 holds. Consider the

CG method in the forms of equations (2) and (28) and \(d_k\) as a
descent direction, where \(\alpha_k\) is obtained using equations (3) and (4). If \(f(x)\) is a uniformly convex function, then

\[ \liminf_{k \to \infty} \|g_k\| = 0. \]

\[ \|d_k\| \leq \|g_k\| + \left( \frac{L \alpha_k}{\omega \alpha_k} \right) \|d_k\|^2 + \left( \frac{\alpha_k}{\omega \alpha_k} \right) \|d_k\|^2 \]

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

\[ \leq \|g_k\| + \frac{\|g_k\|}{\omega} (2L + t + 1). \]

By using Assumption 1, we now have

\[ \|d_k\| \leq \frac{B}{\omega} (2L + t + 1). \]

3.3. Global Convergence of Figure 1 with General Nonlinear Functions. The following restriction for \(\eta_{k^*}\) is very important to establish the convergence analysis for our new search direction. The main importance of this restriction is to avoid the CG method multiplayer, which will be nonnegative.

\[ \eta_{k^*} = \max \left\{ 0, \beta_{k^*} \right\} \]

Thus, equation (28) becomes

\[ d_k = -g_k + \eta_k d_{k-1} - \theta_k (y_{k-1} + s_{k-1}). \]
Lemma 2. Assume that Assumption 1 holds and the sequences \( \{g_k\} \) and \( \{d_k\} \) are generated using Figure 1, where the step size \( \alpha_k \) is computed via the SWP line search of (4) and (5) or weak Wolfe–Powell line search of (4) and (6), such that the sufficient descent condition holds. If \( \beta_k \geq 0 \), there exists a constant \( \gamma > 0 \), such that \( \|g_k\| > \gamma \) for all \( k \geq 1 \). Then, \( d_k \neq 0 \) and

\[
\sum_{k=0}^{\infty} \|u_{k+1} - u_k\|^2 < \infty,
\]

where \( u_k = (d_k/\|d_k\|) \).

Proof. First, if \( d_k = 0 \), then, from the sufficient descent condition, we obtain \( g_k = 0 \). Thus, we suppose that \( d_k \neq 0 \) and

\[
\forall \geq \|g_k\| \geq \gamma > 0, \quad \forall k \geq 1.
\]

We define

\[
u = -g_k - \theta_k(y_{k-1} + s_{k-1}).
\]

Using the triangular inequality, we obtain

\[
\|\nu\| \leq \|g_k\| + \|\theta_k\| \|y_{k-1} + s_{k-1}\|.
\]

Now, utilising the equations of SWP of (4) and (5), we can conclude the two following inequalities given by

\[
d_{k-1}^T y_{k-1} \geq (\sigma - 1)g_{k-1}^T d_{k-1},
\]

\[
|\theta_k| = \left| \frac{g_{k-1}^T d_{k-1}}{d_{k-1}^T y_{k-1}} \right| \leq \left( \frac{\sigma}{1 - \sigma} \right) = \xi.
\]

where \( \zeta \) is some positive constant. Since \( |\theta_k| \leq \zeta \) with \( \alpha_k \geq 1 \), then \( |\theta_k| \leq \zeta \) with \( \alpha_k < 1 \). Thus \( |\theta_k| \leq \zeta \) for all \( \alpha_k > 0 \).

By using the triangular inequality and Assumption 1, we obtain

\[
\|y_{k-1} + s_{k-1}\| \leq \|y_{k-1}\| + \|s_{k-1}\| \leq 2B + 2\rho.
\]

Let \( M = \max(\xi, \zeta) \). Thus, the inequality in (60) can be written as follows:

\[
\|u_k - u_{k-1}\| \leq (1 + \delta_k)\|u_k - u_{k-1}\| = \|u_k - \delta_k u_{k-1} - (u_{k-1} - \delta_k u_k)\|
\]

\[
\leq \|u_k - \delta_k u_{k-1}\| + \|u_{k-1} - \xi u_k\| = 2\|u_k\|.
\]

Also, by using weak Wolfe–Powell line search of (4) and (6), we obtain

\[
d_{k-1}^T y_{k-1} \geq (\sigma - 1)g_{k-1}^T d_{k-1}.
\]

Suppose that \( \alpha_k \geq 1 \); thus

\[
\|\nu\| \leq B + M(2B + 2\rho).
\]

Let

\[
T = B + M(2B + 2\rho),
\]

then

\[
\|\nu\| \leq T.
\]

From equation (54), we now have
| Function     | Dim | No. Iter. | No. Fun.Ev | No. Grad.Ev | CPU time |
|--------------|-----|----------|-----------|-------------|----------|
| AKIVA        | 2   | 8        | 20        | 15          | 0.02     |
| ALLINITU     | 4   | 9        | 25        | 18          | 0.02     |
| ARGGLNC      | 200 | 2        | 98        | 98          | 0.08     |
| ARWHEAD      | 5000| 7        | 26        | 22          | 0.03     |
| BARD         | 3   | 12       | 32        | 22          | 0.02     |
| BDEXP        | 5000| 2        | 7         | 7           | 0.02     |
| BEALE        | 2   | 11       | 33        | 26          | 0.02     |
| BIGGS3       | 6   | 79       | 207       | 15          | 0.02     |
| BIGGS5       | 6   | 79       | 207       | 15          | 0.02     |
| BIGGS6       | 6   | 24       | 64        | 44          | 0.02     |
| BIGGBS1      | 5000| 2500     | 2507      | 4995        | 2.47     |
| BOX2         | 3   | 10       | 23        | 14          | 0.02     |
| BOX3         | 3   | 10       | 23        | 14          | 0.02     |
| BOX          | 10000| 7        | 24        | 20          | 0.08     |
| BRKMCC       | 2   | 5        | 11        | 6           | 0.02     |
| BROWNAL      | 200 | 8        | 19        | 12          | 0.02     |
| BROWNDEN     | 4   | 16       | 38        | 31          | 0.02     |
| BROYDN7D     | 5000| 54       | 100       | 76          | 0.26     |
| BRYBND       | 5000| 32       | 86        | 62          | 0.08     |
| CAMEL6       | 2   | 6        | 22        | 18          | 0.02     |
| CHEVRSSB     | 5000| 299      | 207       | 15          | 0.02     |
| CUBE         | 2   | 17       | 48        | 39          | 0.02     |
| CURLY10      | 10000| 49915   | 70576     | 79203       | 175.06   |
| CURLY20      | 10000| 67275    | 90273     | 11160       | 369.84   |
| CURLY30      | 10000| 73708    | 98014     | 123274      | 514      |
| DENSCHNA     | 2   | 6        | 16        | 12          | 0.02     |
| DENSCHNB     | 2   | 6        | 18        | 15          | 0.02     |
| DENSCHNC     | 2   | 11       | 36        | 31          | 0.02     |
| DENSCHND     | 3   | 14       | 46        | 40          | 0.02     |
| DENSCHNE     | 3   | 12       | 43        | 38          | 0.02     |
| DIXMAANA     | 3000| 8        | 19        | 13          | 0.02     |
| DIXMAANB     | 3000| 19       | 111       | 110         | 0.02     |
| DIXMAAND     | 3000| 26       | 153       | 148         | 0.02     |
| DIXMAANE     | 3000| 254      | 286       | 481         | 0.02     |
| DIXMAANH     | 3000| 162      | 334       | 177         | 0.02     |
| DIXMAANI     | 3000| 2811     | 2893      | 5548        | 0.02     |
| DIXMAANJ     | 3000| 315      | 634       | 321         | 0.02     |
| DIXMAANK     | 3000| 275      | 555       | 282         | 0.02     |
| DIXON3DQ     | 10000| 10000   | 10000     | 199995      | 0.02     |
| DJTL         | 2   | 75       | 1163      | 1148        | 0.02     |
| DQDRTIC      | 5000| 5        | 11        | 6           | 0.02     |
| ECKERLE4LS.SIF | 3   | 2        | 6         | 4           | 0.02     |
| EGGCRATE     | 2   | 6        | 15        | 10          | 0.02     |
| EIGENALS     | 2550| 10197    | 18439     | 12170       | 172.55   |
| EIGENBLS     | 2550| 19030    | 38071     | 19044       | 311.06   |
| EIGENCLS     | 2652| 9958     | 18370     | 11529       | 179.77   |
| ELATVIDU     | 2   | 8        | 32        | 29          | 0.02     |
| ENGLV1       | 5000| 24       | 53        | 46          | 0.02     |
| ENGLV2       | 3   | 26       | 73        | 55          | 0.02     |
| ENSOLS       | 9   | 22       | 47        | 27          | 0.02     |
| EXTRON 3B    | 10000| 227     | 490       | 280         | 0.02     |
| exp2         | 2   | 7        | 16        | 9           | 0.02     |
| FBRAINLS     | 2   | 9        | 27        | 21          | 0.03     |
| FBRAIN2LS    | 4   | 79       | 259       | 204         | 0.47     |
| FLETCHV2     | 5000| 1        | 1         | 1           | 0.02     |
| FLETCHCR     | 2200| 10000    | 254       | 254         | 0.02     |

Table 1: The set of test functions with the numerical results.
| Function    | Dim | No. Iter. | No. Fun.Ev | No. Grad.Ev | CPU time | No. Iter. | No. Fun.Ev | No. Grad.Ev | CPU time | No. Iter. | No. Fun.Ev | No. Grad.Ev | CPU time |
|-------------|-----|-----------|------------|-------------|----------|-----------|------------|-------------|----------|-----------|------------|-------------|----------|
| GENHUMPS    | 5000| 8         | 107        | 106         | 0.37      | 6475      | 12964      | 6493        | 19.89    | 4938      | 14763      | 10180       | 25.27    |
| GROWTHILS   | 3   | 109       | 431        | 369         | 0.02      | 156       | 456        | 319         | 0.02     | 109       | 431        | 369         | 0.02     |
| GULF        | 3   | 33        | 95         | 72          | 0.02      | 37        | 84         | 48          | 0.02     | 33        | 95         | 72          | 0.02     |
| HAHN1LS     | 7   | 5         | 56         | 53          | 0.02      | 37        | 121        | 86          | 0.02     | 5         | 56         | 53          | 0.02     |
| HAIRY       | 2   | 17        | 82         | 68          | 0.02      | 36        | 99         | 65          | 0.02     | 17        | 82         | 68          | 0.02     |
| HATFDL      | 3   | 17        | 49         | 37          | 0.02      | 20        | 43         | 24          | 0.02     | 17        | 49         | 37          | 0.02     |
| HATFDLF     | 3   | 13        | 37         | 30          | 0.02      | 30        | 72         | 45          | 0.02     | 13        | 37         | 30          | 0.02     |
| HEART6LS    | 6   | 375       | 1137       | 876         | 0.02      | 684       | 1576       | 941         | 0.02     | 375       | 1137       | 876         | 0.02     |
| HIELOW      | 3   | 13        | 30         | 21          | 0.03      | 14        | 30         | 16          | 0.02     | 13        | 30         | 21          | 0.05     |
| HILBERTA    | 2   | 2         | 5          | 3           | 0.02      | 2         | 5          | 3           | 0.02     | 2         | 5          | 3           | 0.02     |
| HILBERTB    | 10  | 4         | 9          | 5           | 0.02      | 4         | 9          | 5           | 0.02     | 4         | 9          | 5           | 0.02     |
| HIELOW      | 3   | 13        | 30         | 21          | 0.03      | 14        | 30         | 16          | 0.02     | 13        | 30         | 21          | 0.05     |
| HILBERTA    | 2   | 2         | 5          | 3           | 0.02      | 2         | 5          | 3           | 0.02     | 2         | 5          | 3           | 0.02     |
| HIELOW      | 3   | 13        | 30         | 21          | 0.03      | 14        | 30         | 16          | 0.02     | 13        | 30         | 21          | 0.05     |
| HILBERTA    | 2   | 2         | 5          | 3           | 0.02      | 2         | 5          | 3           | 0.02     | 2         | 5          | 3           | 0.02     |
| HIELOW      | 3   | 13        | 30         | 21          | 0.03      | 14        | 30         | 16          | 0.02     | 13        | 30         | 21          | 0.05     |
| HILBERTA    | 2   | 2         | 5          | 3           | 0.02      | 2         | 5          | 3           | 0.02     | 2         | 5          | 3           | 0.02     |
| HIELOW      | 3   | 13        | 30         | 21          | 0.03      | 14        | 30         | 16          | 0.02     | 13        | 30         | 21          | 0.05     |
| HILBERTA    | 2   | 2         | 5          | 3           | 0.02      | 2         | 5          | 3           | 0.02     | 2         | 5          | 3           | 0.02     |
| HIELOW      | 3   | 13        | 30         | 21          | 0.03      | 14        | 30         | 16          | 0.02     | 13        | 30         | 21          | 0.05     |
| HILBERTA    | 2   | 2         | 5          | 3           | 0.02      | 2         | 5          | 3           | 0.02     | 2         | 5          | 3           | 0.02     |
| HIELOW      | 3   | 13        | 30         | 21          | 0.03      | 14        | 30         | 16          | 0.02     | 13        | 30         | 21          | 0.05     |
| HILBERTA    | 2   | 2         | 5          | 3           | 0.02      | 2         | 5          | 3           | 0.02     | 2         | 5          | 3           | 0.02     |
| HIELOW      | 3   | 13        | 30         | 21          | 0.03      | 14        | 30         | 16          | 0.02     | 13        | 30         | 21          | 0.05     |
| HILBERTA    | 2   | 2         | 5          | 3           | 0.02      | 2         | 5          | 3           | 0.02     | 2         | 5          | 3           | 0.02     |
| HIELOW      | 3   | 13        | 30         | 21          | 0.03      | 14        | 30         | 16          | 0.02     | 13        | 30         | 21          | 0.05     |
| HILBERTA    | 2   | 2         | 5          | 3           | 0.02      | 2         | 5          | 3           | 0.02     | 2         | 5          | 3           | 0.02     |
| HIELOW      | 3   | 13        | 30         | 21          | 0.03      | 14        | 30         | 16          | 0.02     | 13        | 30         | 21          | 0.05     |
| HILBERTA    | 2   | 2         | 5          | 3           | 0.02      | 2         | 5          | 3           | 0.02     | 2         | 5          | 3           | 0.02     |
| HIELOW      | 3   | 13        | 30         | 21          | 0.03      | 14        | 30         | 16          | 0.02     | 13        | 30         | 21          | 0.05     |
| HILBERTA    | 2   | 2         | 5          | 3           | 0.02      | 2         | 5          | 3           | 0.02     | 2         | 5          | 3           | 0.02     |
By using equations (53) and (54), we obtain the following:

$$
\sum_{k=0}^{\infty} \|u_{k+1} - u_k\|^2 \leq 4 \sum_{k=0}^{\infty} \|w_k\|^2 \leq 4T^2 \sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2} < \infty,
$$

which completes the proof.

The following property, which is referred to as Property 1, was presented by Gilbert and Nocedal in [10].

**Property 1.** Consider a method of the form of (2) and (3). Assume that (54) is satisfied for all $k \geq 1$. Then, the CG method has Property 1 if there exist constants $b > 1$ and $\lambda > 0$ such that, for all $k \geq 1$, $|\eta_k| \leq b$ and if $\|s_k\| \leq \lambda$, we obtain $|\eta_k^*| \leq (1/2b)$.

**Lemma 3.** Consider the CG method of the form of (1) and (2) with $\eta_k^*$, where the step size satisfies SWP line search of (4) and (5). If equation (54) holds, then $\eta_k^*$ possesses Property'. Namely, suppose that (54) holds; then there exist $b > 1$ and $\lambda > 0$ such that, for all $k \geq 1$, we have $|\eta_k^*| \leq b$ and if $\|s_k\| \leq \lambda$, we obtain $|\eta_k^*| \leq (1/2b)$.

**Proof.** As a result, set $b = (2(L + t)\eta B/(1 - \sigma)^2) \geq 1$ and $\lambda = ((1 - \sigma)c\gamma^2/2b(L + t)\eta B)$. Using SWP of (4) and (5) with equation (54), we obtain

$$
|\eta_k^*| \leq \left\| \frac{g_T^T y_{k-1}}{d_{k-1}^T y_{k-1}} \right\| + \left\| \frac{g_T^T s_{k-1}}{d_{k-1}^T y_{k-1}} \right\| \leq \frac{L}{(1 - \sigma)^2} \sum_{k=1}^{\infty} \|g_k\| \|s_k\| \leq \frac{2(L + t)B}{(1 - \sigma)^2} = b > 1.
$$
Also, if \( \| s_k \| \leq \lambda \), we obtain
\[
|\eta_k^*| = \frac{\| y_k^T y_{k-1} \|}{\| d_{k-1} \|} + t \frac{\| y_{k-1} s_{k-1} \|}{\| d_{k-1} \|} \leq \frac{L \| g_k \| \| s_{k-1} \| + f_n \| g_{k-1} \| \| s_{k-1} \|}{(1 - \sigma) \gamma^2}
\]
\[
\leq \frac{(L + t) \| g_{k-1} \|}{(1 - \sigma) \gamma^2},
\]
(71)

where
\[
|\eta_k^*| \leq \frac{1}{2b},
\]
(72)

By using Lemma 4.1 and Lemma 4.2 in [10], we obtain the following result.

**Theorem 4.** Let the sequences \( \{ x_k \} \) and \( \{ d_k \} \) be generated by (2) and (3) with the CG method in equation (28), where the step size satisfies (4) and (5). Then, we obtain the results by using Lemmas 2 and 3, as well as Lemmas 4.1 and 4.2 in [10], where \( \liminf_{k \to \infty} \| g_k \| = 0 \).

4. **Numerical Results and Discussion**

To test the new search direction (28), we selected some test functions in Table 1 from CUTeR [37]. A comparison with other popular and strong CG coefficients is employed. The comparison includes the CG-Descent 6.8 and DL + CG formula based on the CPU time, number of iterations, number of function evaluations, and \( t \) number of gradient evaluations. We use the SWP line search to obtain the step...
The norm of the gradient was employed as the stopping criterion, specifically \( \|g_k\| \leq 10^{-6} \) for all algorithms. The host computer is AMD A4-7210 APU Radeon R3 Graphics, where the installed memory is 4 GB with operating system Ubuntu 20.04.2.0 LTS. The results are shown in Figures 2–5, in which a performance measure introduced by Dolan and More [38] was employed. In Table 1, the following notations are used:

- No. Iter indicates number of iterations
- No. Fun.Ev indicates number of function evaluations
- No. Grad.Ev indicates number of gradient evaluations

Figure 2 shows that FTCGHS strongly outperformed all methods in the number of iterations. Meanwhile, Figure 3 presents the number of function evaluations, and we can note that the FTCGHS method also strongly outperforms CG-Descent 6.8 and DL+ methods. From Figures 4 and 5, we can note that FTCGHS method slightly outperforms CG-Descent and DL+, since we use SWP line search. However, let the SWP line search in equation (5) be extended as follows:

\[
\sigma_1 g_k^T d_k \leq g(x_k + \alpha_k d_k)^T d_k \leq \sigma_2 g_k^T d_k,
\]

\( 0 < \delta < \sigma_1 \leq \sigma_2 < 1 \). (73)

Then, the results in Figures 4 and 5 will be more efficient. For more about equation (73), the reader can refer to [39].

In addition, we present an example for two-dimensional three-hump camel back function, which is given by the following formula:

\[
f(x) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} - x_1x_2 + x_2^2.
\] (74)

Here, we have the following:

- Number of variables (n): 2
- Initial points: (1, 1), (5, 5), (10, 10), and (15, 15)

This function has three local minima, one of which is global. Therefore, this function is a multimodal function usually used to test global minima. The global minimum is \( x^* = (0, 0) \) and the function value is \( f(x^*) = 0 \). This function looks like the back of an upside-down camel. See Figure 6 for the three-dimensional graph.
5. Conclusions

In this paper, we proposed a new four-term CG method as presented in equation (28) by using the following direction:

\[ g_k, a_{k-1}, y_{k-1}, s_{k-1}. \]

The search direction inherits the following properties:

1. It presents four-term CG method.
2. It satisfies the sufficient descent condition.
3. The convergence analyses for convex and nonconvex functions are obtained.
4. The numerical results show that the new modification outperforms popular methods such as CG-Descent and DL+ methods.
5. The new search direction can be developed by using different values of \( t \).

In addition, we will attempt to extend the SWP line search in the future by different values of sigma to reduce the number of gradient evolutions and CPU time.

Data Availability

The data used to support the findings of this study are available within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors’ Contributions

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

Acknowledgments

The authors would like to acknowledge the University of Malaysia Terengganu (UMT) for funding this study. Also, the authors would like to thank Professor Dr. William Hager for publishing the code of CG-Descent. This study was partially supported by the Universiti Malaysia Terengganu, Centre of Research and Innovation Management.

References

[1] P. Wolfe, “Convergence conditions for ascent methods,” SIAM Review, vol. 11, no. 2, pp. 226–235, 1969.
[2] P. Wolfe, “Convergence conditions for ascent methods. II: some corrections,” SIAM Review, vol. 13, no. 2, pp. 185–188, 1971.
[3] M. R. Hestenes and E. Stiefel, “Methods of conjugate gradients for solving linear systems,” Journal of Research of the National Bureau of Standards, vol. 49, no. 6, pp. 409–436, 1952.
[4] E. Polak and G. Ribiere, “Note sur la convergence de méthodes de directions conjuguées,” ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique, vol. 3, pp. 35–43, 1969.
[5] Y. Liu and C. Storey, “Efficient generalized conjugate gradient algorithms, part 1: theory,” Journal of Optimization Theory and Applications, vol. 69, no. 1, pp. 129–137, 1991.
[6] R. Fletcher and C. M. Reeves, “Function minimization by conjugate gradients,” The Computer Journal, vol. 7, no. 2, pp. 149–154, 1964.
[7] R. Fletcher, Practical Method of Optimisation. Unconstrained Optimisation, Wiley, Hoboken, NJ, USA, 1997.
[8] Y. H. Dai and Y. Yuan, “A nonlinear conjugate gradient method with a strong global convergence property,” SIAM Journal on Optimization, vol. 10, no. 1, pp. 177–182, 1999.
[9] M. J. D. Powell, “Nonconvex minimization calculations and the conjugate gradient method,” Lecture Notes in Mathematics, vol. 1066, pp. 122–141, 1984.
[10] J. C. Gilbert and J. Nocedal, “Global convergence properties of conjugate gradient methods for optimization,” SIAM Journal on Optimization, vol. 2, no. 1, pp. 21–42, 1992.
[11] G. Zoutendijk, “Nonlinear programming,” Computational Methods. Integer Nonlinear Program, vol. 143, pp. 37–86, 1970.
[12] M. Al-Baali, “Descent property and global convergence of the Fletcher-Reeves method with inexact line search,” IMA Journal of Numerical Analysis, vol. 5, no. 1, pp. 121–124, 1985.
[13] Z. Wei, S. Yao, and L. Liu, “The convergence properties of some new conjugate gradient methods,” Applied Mathematics and Computation, vol. 183, no. 2, pp. 1341–1350, 2006.
[14] A. Alhawarat, Z. Salleh, M. Mamat, and M. Rivaie, “An efficient modified Polak-Ribiére-Polyak conjugate gradient method with global convergence properties,” Optimization Methods and Software, vol. 32, no. 6, pp. 1299–1312, 2017.
[15] P. Kaelo, P. Mtagulwa, and M. V. Thuto, “A globally convergent hybrid conjugate gradient method with strong Wolfe conditions for unconstrained optimization,” Mathematical Sciences, vol. 14, no. 1, pp. 1–9, 2020.
[16] Y.-H. Dai and L. Z. Liao, “New conjugacy conditions and related nonlinear conjugate gradient methods,” Applied Mathematics and Optimization, vol. 43, no. 1, pp. 87–101, 2001.
[17] W. W. Hager and H. Zhang, “A new conjugate gradient method with guaranteed descent and an efficient line search,” SIAM Journal on Optimization, vol. 16, no. 1, pp. 170–192, 2005.
[18] W. W. Hager and H. Zhang, “The limited memory conjugate gradient method,” SIAM Journal on Optimization, vol. 23, no. 4, pp. 2150–2168, 2013.
[19] N. Andrei, "A simple three-term conjugate gradient algorithm for unconstrained optimization," *Journal of Computational and Applied Mathematics*, vol. 241, pp. 19–29, 2013.

[20] N. Andrei, "On three-term conjugate gradient algorithms for unconstrained optimization," *Applied Mathematics and Computation*, vol. 219, no. 11, pp. 6316–6327, 2013.

[21] S. Babaie-Kafaki and R. Ghanbari, "Two modified three-term conjugate gradient methods with sufficient descent property," *Optimization Letters*, vol. 8, no. 8, pp. 2285–2297, 2014.

[22] S. Deng and Z. Wan, "A three-term conjugate gradient algorithm for large-scale unconstrained optimization problems," *Applied Numerical Mathematics*, vol. 92, pp. 70–81, 2015.

[23] J. K. Liu, J. L. Xu, and L. Q. Zhang, "Partially symmetrical derivative-free Liu-Storey projection method for convex constrained equations," *International Journal of Computer Mathematics*, vol. 96, no. 9, pp. 1787–1798, 2019.

[24] J. K. Liu, X. Zhao, and X. L. Wu, "Some three-term conjugate gradient methods with the new direction structure," *Applied Numerical Mathematics*, vol. 150, pp. 433–443, 2020.

[25] G. Yuan, J. Lu, and Z. Wang, "The PRP conjugate gradient algorithm with a modified WWP line search and its application in the image restoration problems," *Applied Numerical Mathematics*, vol. 152, pp. 1–11, 2020.

[26] S. Yao, Y. Wu, J. Yang, and J. Xu, "A three-term gradient descent method with subspace techniques," *Mathematical Problems in Engineering*, vol. 2021, Article ID 8867309, 7 pages, 2021.

[27] A. Alhawarat, N.-T. Trung, and Z. Salleh, "Conjugate gradient Method: a developed version to resolve unconstrained optimization problems," *Journal of Computer Science*, vol. 16, no. 9, pp. 1220–1228, 2020.

[28] Z. Dai and J. Kang, "Some new efficient mean-variance portfolio selection models," *International Journal of Finance & Economics*, 2021.

[29] Z. Dai, J. Kang, and F. Wen, "Predicting stock returns: a risk measurement perspective," *International Review of Financial Analysis*, vol. 74, Article ID 101676, 2021.

[30] Z. Dai and F. Wen, "Some improved sparse and stable portfolio optimization problems," *Finance Research Letters*, vol. 27, pp. 46–52, 2018.

[31] I.A.Masmali, Z. Salleh, Z. Salleh, and A. Alhawarat, "A decent three term conjugate gradient method with global convergence properties for large scale unconstrained optimization problems," *AIMS Mathematics*, vol. 6, no. 10, pp. 10742–10764, 2021.

[32] I. Bongartz, A. R. Conn, N. Gould, and P. L. Toint, "CUTE: constrained and unconstrained testing environment," *ACM Transactions on Mathematical Software*, vol. 21, no. 1, pp. 123–160, 1995.

[33] E. D. Dolan and J. J. More, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.

[34] M. Al-Baali, Y. Narushima, and H. Yabe, "A family of three-term conjugate gradient methods with sufficient descent property for unconstrained optimization," *Computational Optimization and Applications*, vol. 60, no. 1, pp. 89–110, 2015.