https://doi.org/10.1186/s13660-024-03142-0

Journal of Inequalities and Applications
a SpringerOpen Journal

RESEARCH Open Access

A three-term conjugate gradient descent method with some applications

Ahmad Alhawarat[1], Zabidin Salleh[1], Hanan Alolaiyan[2], Hamid El Hor[3] and Shahrina Ismail[4*]

*Correspondence:
shahrinaismail@usim.edu.my
[4]Financial Mathematics Program, Faculty of Science and Technology, Universiti Sains Islam Malaysia, Bandar Baru Nilai, 71800, Nilai, Negeri Sembilan, Malaysia
Full list of author information is available at the end of the article

Abstract

The stationary point of optimization problems can be obtained via conjugate gradient (CG) methods without the second derivative. Many researchers have used this method to solve applications in various fields, such as neural networks and image restoration. In this study, we construct a three-term CG method that fulfills convergence analysis and a descent property. Next, in the second term, we employ a Hestenses-Stiefel CG formula with some restrictions to be positive. The third term includes a negative gradient used as a search direction multiplied by an accelerating expression. We also provide some numerical results collected using a strong Wolfe line search with different sigma values over 166 optimization functions from the CUTEr library. The result shows the proposed approach is far more efficient than alternative prevalent CG methods regarding central processing unit (CPU) time, number of iterations, number of function evaluations, and gradient evaluations. Moreover, we present some applications for the proposed three-term search direction in image restoration, and we compare the results with well-known CG methods with respect to the number of iterations, CPU time, as well as root-mean-square error (RMSE). Finally, we present three applications in regression analysis, image restoration, and electrical engineering.

Mathematics Subject Classification: 65K05; 90C30

Keywords: Conjugate gradient method; Image restoration; Global convergence; Unconstrained; Optimization

1 Introduction

In order to determine the stationary point of optimization problems, the nonlinear conjugate gradient (CG) method does not necessitate the second derivative or its approximation. Here, the form we consider in the present investigation is as follows:

$$\min f(x), \quad x \in R^n, \tag{1}$$

where $f : R^n \to R$ as well as the gradient $g(x) = \nabla f(x)$ is available. The following is how iterative approaches are usually applied to solve (1).

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

where α_k is obtained by an exact or inexact line search. Moreover, an inexact line search, for instance, a strong Wolfe–Powell (SWP) line [1, 2], is commonly used and may be expressed as the following:

$$f(x_k + \alpha_k d_k) \le f(x_k) + \delta \alpha_k g_k^T d_k, \tag{3}$$

and

$$\left|g(x_k + \alpha_k d_k)^T d_k\right| \le \sigma \left|g_k^T d_k\right|. \tag{4}$$

A weak Wolfe-Powell (WWP) line search is as given by Equation (3) and

$$g(x_k + \alpha_k d_k)^T d_k \ge \sigma g_k^T d_k \tag{5}$$

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

The following expresses the search direction, d_k pertaining to two terms

$$d_k = \begin{cases} -g_k, & k = 1, \\ -g_k + \beta_k d_{k-1}, & k \ge 2, \end{cases} \tag{6}$$

where $g_k = g(x_k)$, while β_k resembles the CG parameter. Here, the most well-known CG parameters are divided into two groups, the first of which is an efficient group defined as follows, which includes the Hestenses–Stiefel (HS) [3], Polak–Ribière–Polyak (PRP) [4], as well as Liu and Storey (LS) [5] methods.

$$\beta_k^{\mathrm{HS}} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, \qquad \beta_k^{PRP} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2}, \qquad \beta_k^{LS} = -\frac{g_k^T y_{k-1}}{d_{k-1}^T g_{k-1}},$$

where $y_{k-1} = g_k - g_{k-1}$. However, this group encounters a convergence problem if their values become negative [6]. In contrast, the second group is inefficient and exhibits strong global convergence. This category includes the Fletcher–Reeves (FR) [7], Fletcher (CD) [8], and Dai and Yuan (DY) [9] approaches, as defined by the following equations.

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \qquad \beta_k^{CD} = -\frac{\|g_k\|^2}{d_{k-1}^T g_{k-1}}, \qquad \beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T g_{k-1}}.$$

The subsequent conjugacy condition was put forth by Dai and Liao [10].

$$d_k^T y_{k-1} = -t g_k^T s_{k-1}, \tag{7}$$

where $s_{k-1} = x_k - x_{k-1}$ and $t \ge 0$. Pertaining to $t = 0$, the classical conjugacy condition is then expressed as Equation (8) becomes the classical conjugacy condition. They also presented the CG formula below [10], utilizing (6) and (7).

$$\beta_k^{DL} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} - t \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}} = \beta_k^{\mathrm{HS}} - t \frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}. \tag{8}$$

Table 1 Some recent three-term CG methods

Andrei [13, 14]	$t = (1 + \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}}), t = (1 + 2\frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}}).$
Babaie-Kafaki and Ghanbari [15]	$t = (\max(\varsigma, 1 - \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}}))\varsigma > 0.$
Deng and Wan [16]	$t = (\max(0, 1 - \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}})).$

Nonetheless, β_k^{DL} carries over the same issue as β_k^{PRP} and β_k^{HS}, e.g., β_k^{DL} is generally not nonnegative. Equation (8) was then replaced [10]:

$$\beta_k^{DL+} = \max\{\beta_k^{\mathrm{HS}}, 0\} - t\frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}.$$

Hager and Zhang [11, 12] provided the CG formula below, predicated in Eq. (8).

$$\beta_k^{\mathrm{HZ}} = \max\{\beta_k^N, \eta_k\}, \tag{9}$$

where $\beta_k^N = \frac{1}{d_k^T y_k}(y_k - 2d_k\frac{\|y_k\|^2}{d_k^T y_k})^T g_k, \eta_k = -\frac{1}{\|d_k\|\min\{\eta, \|g_k\|\}}$, while $\eta > 0$ is a constant. Note that $t = 2\frac{\|y_k\|^2}{s_k^T y_k}$ when $\beta_k^N = \beta_k^{DY}$.

Based on Equation (8), many researchers have suggested the three-term CG methods given below. Let the following equation represent the general form with regard to the three-term CG method:

$$d_k = -g_k + \eta_k d_{k-1} - \theta_k y_{k-1}, \tag{10}$$

where $\theta_k = \frac{g_k^T y_{k-1} - t g_k^T s_{k-1}}{y_{k-1}^T d_{k-1}}$ and $\eta_k = \frac{g_k^T d_{k-1}}{y_{k-1}^T d_{k-1}}$. We then obtain a wide variety of choices by replacing t in Eq. (10) with an appropriate term, as shown in Table 1.

By replacing y_{k-1} with g_{k-1}, Liu et al. [17] proposed the following three-term CG method:

$$d_k = -g_k + \left(\beta_k^{LS} - \frac{\|g_{k-1}\|^2 g_k^T d_{k-1}}{(d_{k-1}^T g_{k-1})^2}\right)d_{k-1} + \left(\frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}}\right)g_{k-1},$$

with the following assumption

$$\left(\frac{g_k^T d_{k-1}}{d_{k-1}^T g_{k-1}}\right) > \nu \in (0,1).$$

Liu et al. [17] demonstrated how nonconvex functions may address nonlinear monotone equations if the sufficient descent condition is met. Meanwhile, Liu et al. [18] created the three-term CG methods given below and solved Equation (1) by utilizing it in order to avoid using the condition $(\frac{g_k^T d_{k-1}}{d_{k-1}^T g_{k-1}}) > \nu \in (0,1)$.

$$d_k = -g_k + \left(\beta_k^{LS} - \frac{\|g_{k-1}\|^2 g_k^T s_{k-1}}{(d_{k-1}^T g_{k-1})^2}\right)d_{k-1} - \left(\frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}}\right)g_{k-1}.$$

Yao et al. [19] suggested a three-term CG with the following new choice of t given by

$$d_{k+1} = -g_{k+1} + \left(\frac{g_k^T y_k - t_k g_{k+1}^T s_k}{y_k^T d_k}\right)d_k + \frac{g_{k+1}^T d_k}{y_k^T d_k}y_k.$$

t_k was also chosen to meet the descent condition like the one below as per the SWP line search.

$$t_k > \frac{\|y_k\|^2}{y_k^T s_k}.$$

Another theorem put forth by Yao et al. [19] states that if t_k is close to $\frac{\|y_k\|^2}{y_k^T s_k}$, then the search direction produces a zigzag search path. Thus, they decided on the option t_k given below.

$$t_k = 1 + 2\frac{\|y_k\|^2}{y_k^T s_k}.$$

At the beginning of the CG method, a nonnegative CG formula with a new restart property was presented by Alhawarat et al. [20].

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

where $\|\cdot\|$ denotes the Euclidean norm, while μ_k can be represented as

$$\mu_k = \frac{\|x_k - x_{k-1}\|}{\|y_{k-1}\|}.$$

Similarly, Jiang et al. [21] suggested the CG method given by:

$$\beta_k^{JJSL} = \frac{g_k^T y_{k-1}}{g_{k-1}^T y_{k-1}}.$$

To improve the efficiency of prior methods, they constructed a restart criterion given as follows:

$$d_k = \begin{cases} -g_k, & k = 1, \\ -g_k + \beta_k^{JJSL} d_{k-1} + \frac{g_k^T d_{k-1}}{d_{k-1}^T (g_k - g_{k-1})} g_{k-1} & \text{if } 0 \le g_k^T g_{k-1} \le \|g_k\|^2 \le \|g_{k-1}\|^2, k \ge 2, \\ -g_k + \xi \frac{g_k^T g_{k-1}}{\|g_{k-1}\|^2} g_{k-1}, & k \ge 2, \text{ otherwise.} \end{cases}$$

where $0 < \xi < 1$.

Recently, a convex combination between two distinct search directions is presented by Alhawarat et al. [22] as follows:

$$d_k = \lambda d_k^(1)) + (1 - \lambda) d_k^(2))$$

where

$$0 \le \lambda \le 1,$$

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

and

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

The authors selected $\beta_k^((1))$ and $\beta_k^((2))$ given below:

$$\beta_k^((1)) = \begin{cases} \frac{\|g_k\|^2 - \mu_k |g_k^T g_{k-1}|}{d_{k-1}^((1) y_{k-1}} & \text{if } \|g_k\|^2 > \mu_k |g_k^T g_{k-1}|, \\ -t \frac{g_k^T s_{k-1}}{d_{k-1}^((1) y_{k-1}} & \text{otherwise}, \end{cases}$$

and

$$\beta_k^((2)) = \beta_k^{CG-DESCENT}.$$

The descent condition, also known as the downhill condition, given by

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

helps research CG methods and is crucial to the validation of global convergence analysis. Al-Baali [23] also utilized the subsequent version of (13) to demonstrate the FR method.

$$g_k^T d_k \leq -c \|g_k\|^2, \quad \forall k \geq 1, \tag{11}$$

where $c \in (0, 1)$. Next, the sufficient descent condition is given by Eq. (14) below. Moreover, it performs better than (13) because the quantity of $g_k^T d_k$ can be controlled using $\|g_k\|^2$.

2 Proposed modified search direction (3TCGHS) and motivation

The main motivation for researchers in CG methods is to propose a positive CG method with an efficiency similar to that of PRP or HS, with a global convergence. In the following modification, we utilize the new search direction g_{k-1} proposed by [17] with β_k^{HS} restricted to be nonnegative, as given below:

$$d_k^{\mathrm{Jiang}} = \begin{cases} -g_k, & k = 1, \\ -g_k + \beta_k^{\mathrm{HS}} d_{k-1} + \frac{g_k^T d_{k-1}}{d_{k-1}^T (g_k - g_{k-1})} g_{k-1}, & \text{if } \|g_k\|^2 > g_k^T g_{k-1}, k \geq 2, \\ -\mu_k \frac{g_k^T s_{k-1}}{d_{k-1}^T (g_k - g_{k-1})}, & \text{otherwise}. \end{cases} \tag{12}$$

where $\mu_k = \frac{\|x_k - x_{k-1}\|}{\|g_k - g_{k-1}\|}$.

The procedures acquired to determine the optimization function's stationary point are outlined in the following Algorithm 1.

3 Global convergence properties

The assumption that follows is considered as a condition for the objective function.

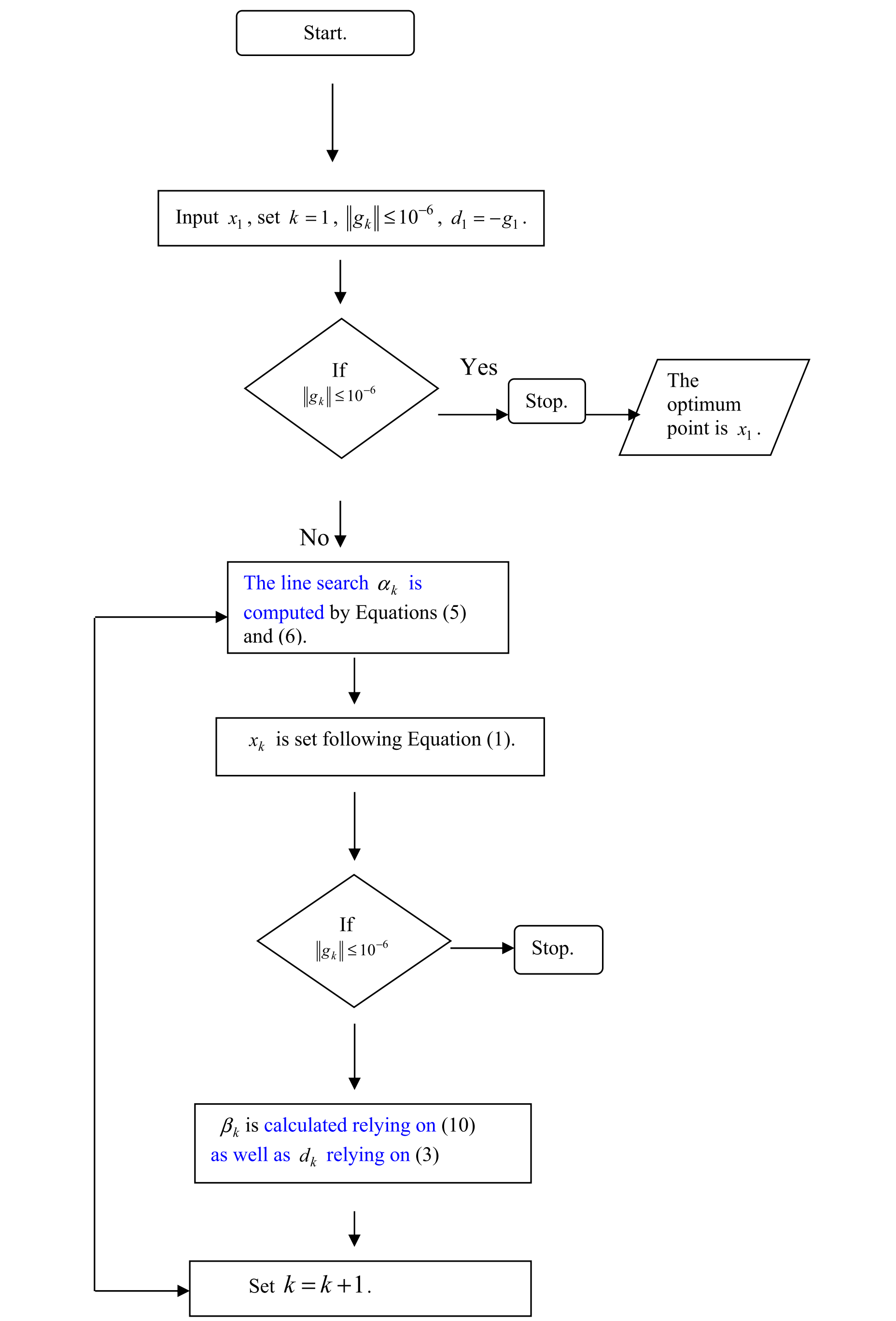


Figure 1 Algorithm 1

Assumption 1 I. The level set $\Psi = \{x \in R^n : f(x) \le f(x_1)\}$ is bounded. Here, a positive constant ρ exists; in this case

$$\|x\| \le \rho, \quad \forall x \in \Psi.$$

II. f is a continuous and differentiable function in some neighborhood W of Ψ, and its gradient is Lipchitz continuous, meaning that, for every $x, y \in W$, a constant $L > 0$ exists, in which case

$$\|g(x) - g(y)\| \leq L\|x - y\|.$$

As per this assumption, there must be a positive constant η; in this case

$$\|g(u)\| \leq \eta, \quad \forall \eta \in W.$$

The CG method's convergence properties are typically established using the following lemma proposed by Zoutendijk [24]. The method involves multiple line searches, including SWP and WWP line searches.

Lemma 3.1 *Let Assumption 1 hold. If α_k satisfies the WWP line search with the descent condition (9), then take any form of (2) and (3). Then, the inequality that follows holds.*

$$\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty. \tag{13}$$

As can be seen from the following theorem, the new formula fulfills the descent condition (9).

Theorem 3.1 *Let the sequences $\{x_k\}$ and $\{d_k^{\text{Jiang}}\}$ be developed by Equations (2) and (12), and consider the line search method obtained using Equations (3) and (4). The sufficient descent condition (11) is then satisfied.*

Proof Multiply (12) by g_k^T to obtain

$$\begin{aligned} g_k^T d_k^{\text{Jiang}} &= -\|g_k\|^2 + \frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T (g_k - g_{k-1})} d_{k-1} + \frac{g_k^T d_{k-1}}{d_{k-1}^T (g_k - g_{k-1})} g_k^T g_{k-1}, \\ &= -\|g_k\|^2 + \frac{\|g_k\|^2}{d_{k-1}^T (g_k - g_{k-1})} g_k^T d_{k-1}. \end{aligned}$$

Using a SWP line search, we obtain

$$\leq -\|g_k\|^2 + \frac{\|g_k\|^2}{(1-\sigma) g_k^T d_{k-1}} \sigma g_k^T d_{k-1} = -\|g_k\|^2 + \frac{\sigma \|g_k\|^2}{(1-\sigma)}.$$

If $\sigma \leq \frac{1}{2}$,

$$g_k^T d_k^{\text{Jiang}} \leq -c||g_k||^2.$$

The proof is now complete. □

Theorem 3.2 *Let sequence $\{x_k\}$ be generated by Equation (2), where $d_k = -\mu_k \frac{g_k^T s_{k-1}}{d_{k-1}^T (g_k - g_{k-1})}$ is the step length obtained by SWP line search. Afterwards, condition (11) for a sufficient descent holds.*

Proof After multiplying (2) by g_k^T and substituting $d_k = -\mu_k \frac{g_k^T s_{k-1}}{d_{k-1}^T(g_k-g_{k-1})}$, we acquire

$$g_k^T d_k = -\|g_k\|^2 - \mu_k \frac{\alpha_{k-1} g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} g_k^T d_{k-1},$$
$$g_k^T d_k = -\|g_k\|^2 - \mu_k \frac{\alpha_{k-1} \|g_k^T d_{k-1}\|^2}{d_{k-1}^T y_{k-1}} < 0.$$

This completes the proof. □

Gilbert and Nocedal [25] outlined a property known as Property* to perform a specialized role in studies on CG formulas related to the PRP method. The property is described below.

Property* *Consider a method of the form* (2) *and* (6), *and let*

$$0 < \gamma \leq \|g_k\| \leq \bar{\gamma}. \tag{14}$$

We claim that the method contains Property (*) *provided that constants* $b > 1$ *and* $\lambda > 0$ *exist, where for every* $k \geq 1$, *we acquire* $|\beta_k| \leq b$. *Meanwhile, if* $\|x_k - x_{k-1}\| \leq \lambda$, *then*

$$|\beta_k| \leq \frac{1}{2b}.$$

The lemma below illustrates that β_k^{HS} *inherits Property**. *The proof is similar to that given by Gilbert and Nocedal* [25].

Lemma 3.2 *Let Assumption* 1 *hold and consider any form of Equations* (2) *and* (3). *Consequently,* β_k^{HS} *fulfills Property**.

Proof Let $b = \frac{2\bar{\gamma}}{(1-\sigma)c\gamma^2} > 1$, and $\lambda = \frac{(1-\sigma)c\gamma^4}{2L\bar{\gamma}b}$. Then, using (14) and SWP line search, we obtain

$$\left|\beta_k^{\mathrm{HS}}\right| \leq \left|\frac{g_k^T(g_k - g_{k-1})}{d_{k-1}^T(g_k - g_{k-1})}\right| \leq \frac{\|g_k\|^2 + |g_k^T g_{k-1}|}{c(1-\sigma)\|g_{k-1}\|^2} = \frac{2\bar{\gamma}}{c(1-\sigma)\gamma^2} = b.$$

If $\|x_{k+1} - x_k\| \leq \lambda$ holds with Assumption 1, then

$$\begin{aligned}\left|\beta_k^{\mathrm{HS}}\right| &\leq \frac{g_k^T(g_k - g_{k-1})}{d_{k-1}^T(g_k - g_{k-1})} \leq \frac{\|g_k\|\|g_k - g_{k-1}\|}{c(1-\sigma)\|g_{k-1}\|^2} \leq \frac{L\|g_k\|\|x_k - x_{k-1}\|}{c(1-\sigma)\|g_{k-1}\|^2} \\ &\leq \frac{L\bar{\gamma}\lambda}{c(1-\sigma)\gamma^2} = \frac{1}{2b}.\end{aligned}$$

□

Lemma 3.3 *Via Algorithm* 1, *let Assumption* 1 *hold while the sequences* $\{g_k\}$ *and* $\{d_k^{\mathrm{Jiang}}\}$ *are developed. The step size* α_k *is determined by utilizing the SWP line search to ensure that the sufficient descent condition is met. Provided that* $\beta_k \geq 0$, *then a constant* $\gamma > 0$ *exists,* *where* $\|g_k\| > \gamma$ *for every* $k \geq 1$. *Afterwards*, $d_k \neq 0$ *and*

$$\sum_{k=0}^{\infty} \|u_{k+1} - u_k\|^2 < \infty, \tag{15}$$

where $u_k = \frac{d_k}{\|d_k\|}$.

Proof First, given that $d_k = 0$, we can acquire $g_k = 0$ from the sufficient descent condition. Hence, we can assume that $d_k \neq 0$ and

$$\bar{\gamma} \geq \|g_k\| \geq \gamma > 0, \quad \forall k \geq 1. \tag{16}$$

We provide definitions as below:

$$u_k = w_k + \delta_k u_{k-1},$$
$$\eta_k = \frac{g_k^T(g_k - g_{k-1})}{d_{k-1}^T(g_k - g_{k-1})},$$

where

$$w_k = \frac{-g_k + \frac{g_k^T d_{k-1}}{d_{k-1}^T(g_k - g_{k-1})} g_{k-1}}{\|d_k\|}, \qquad \delta_k = \eta_k \frac{\|d_{k-1}\|}{\|d_k\|}.$$

Since u_k denotes a unit vector, we have

$$\|w_k\| = \|u_k - \delta_k u_{k-1}\| = \|\delta_k u_k - u_{k-1}\|.$$

By the triangular inequality and $\delta_k \geq 0$, we obtain

$$\begin{aligned} \|u_k - u_{k-1}\| \leq (1 + \delta_k)\|u_k - u_{k-1}\| &= \left\| u_k - \delta_k u_{k-1} - (u_{k-1} - \delta_k u_k) \right\|. \\ &\leq \|u_k - \delta_k u_{k-1}\| + \|u_{k-1} - \delta_k u_k\| = 2\|w_k\|. \end{aligned} \tag{17}$$

We now define

$$\nu = -g_k + \frac{g_k^T d_{k-1}}{d_{k-1}^T(g_k - g_{k-1})} g_{k-1}.$$

Utilizing the triangular inequality, we establish

$$\|\nu\| \leq \|g_k\| + \left| \frac{g_k^T d_{k-1}}{d_{k-1}^T(g_k - g_{k-1})} \right| \|g_{k-1}\|. \tag{18}$$

Using the SWP Equations (4) and (5), we can obtain the following two inequalities.

$$d_{k-1}^T y_{k-1} \geq (\sigma - 1) g_{k-1}^T d_{k-1},$$
$$\left| \frac{g_k^T d_{k-1}}{d_{k-1}^T y_{k-1}} \right| \leq \left(\frac{\sigma}{1 - \sigma} \right).$$

Hence, the inequality in Eq. (18) can be expressed in the following way:

$$\|v\| \leq B\left(1 + \left(\frac{\sigma}{1-\sigma}\right)\right).$$

Let

$$T = B\left(1 + \left(\frac{\sigma}{1-\sigma}\right)\right).$$

Then, $\|v\| \leq T$. From Eq. (17), we have $\|u_k - u_{k-1}\| \leq 2w$.
By Eqs. (16) and (15), we acquire what is presented below

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

This completes the proof. □

By Lemmas 4.1 and 4.2 in [10], we are able to obtain the following outcome:

Theorem 3.3 *Using the CG method in Eq.* (12), *let* (2) *and* (3) *generate the sequences* $\{x_k\}$ *and* $\{d_k^{\text{Jiang}}\}$, *and let the step size satisfy* (4) *and* (5). *Utilizing Lemmas* 3.2, 3.3, *and Lemmas* 4.1 *and* 4.2 *in* [10], *we acquire such findings of* $\lim\ \inf_{k\to\infty} \|g_k\| = 0$.

4 Numerical results and discussions

In this section, we provide some numerical findings to validate the efficiency for the proposed search direction. Details are provided in the Appendix. We used 166 test functions from the CUTEr library [26]. The functions can be downloaded in .SIF file format from the URL below.

https://www.cuter.rl.ac.uk/Problems/mastsif.shtml

We modified the code from CG-Descent 6.8 to implement the proposed search direction and DL+ method. The following website has the code available for download.

https://people.clas.ufl.edu/hager/software/

With an AMD A4-7210 CPU and 4 GB of RAM, the host computer was running Ubuntu 20.04 to carry out the necessary computations. We compared the modified search direction d_k^{Jiang} with DL+ methods, and we used a SWP line search to acquire the step length with $\sigma = 0.1$ and $\delta = 0.01$ for 3TCGHS and DL+ and the previously mentioned approximate Wolfe-Powell line search for CG-Descent. Figures 1–4 present all outcomes via a performance measure first used by Dolan and More [27]. We utilize an SWP line search together with $\sigma = 0.1$ and $\delta = 0.01$ for d_k^{Jiang} method and DL+. From Figs. 1–4, it may be observed that the new search direction strongly outperformed DL+ in terms of the number of iterations, function evaluation, CPU time, and a number of gradient evaluations. The subsequent big notations are used in the Appendix:

No. iter: Number of iterations.

No. function: Number of function evaluations

No. gradient: Number of gradient evaluations

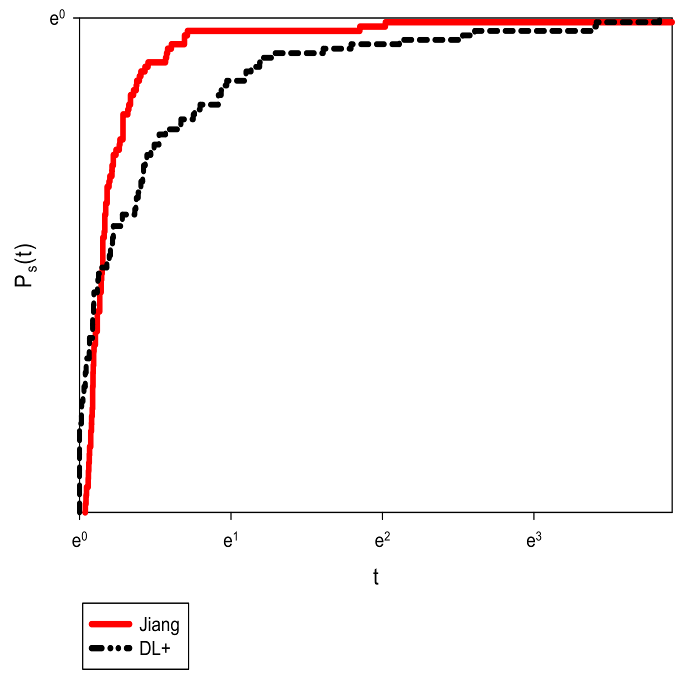


Figure 2 Graph of the number of iterations

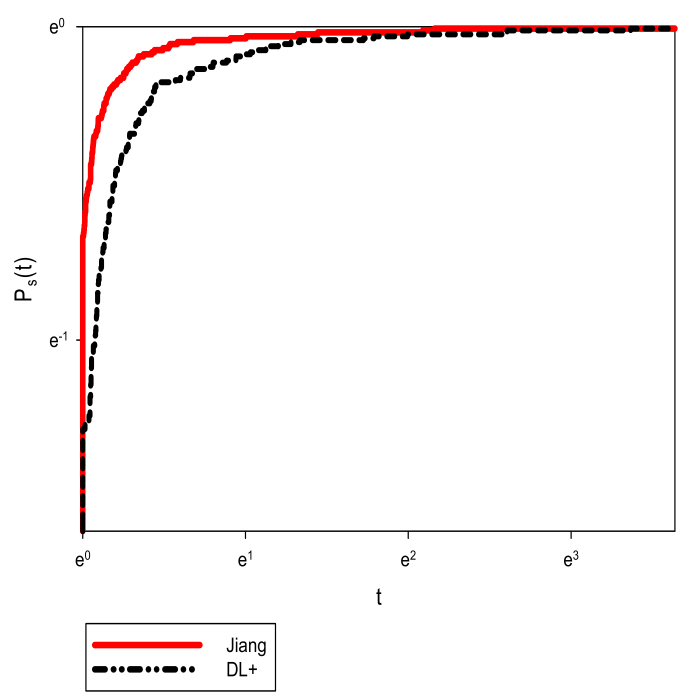


Figure 3 Graph of the number of function evaluations

4.1 Application to image restoration

To the original images, we applied Gaussian noise with a standard deviation of 25%. Next, we used 3TCGHS as well as the β_k^{DL+} (Dai–Liao) CG algorithm to restore these images.

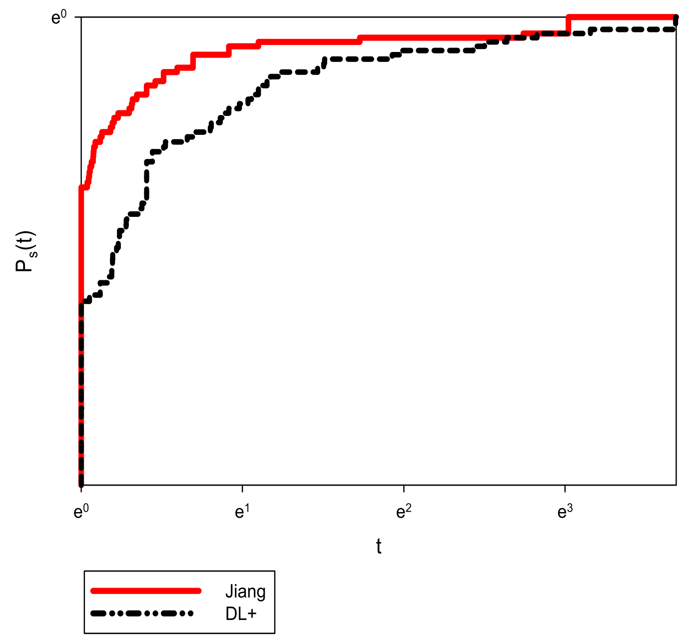


Figure 4 Graph of CPU time

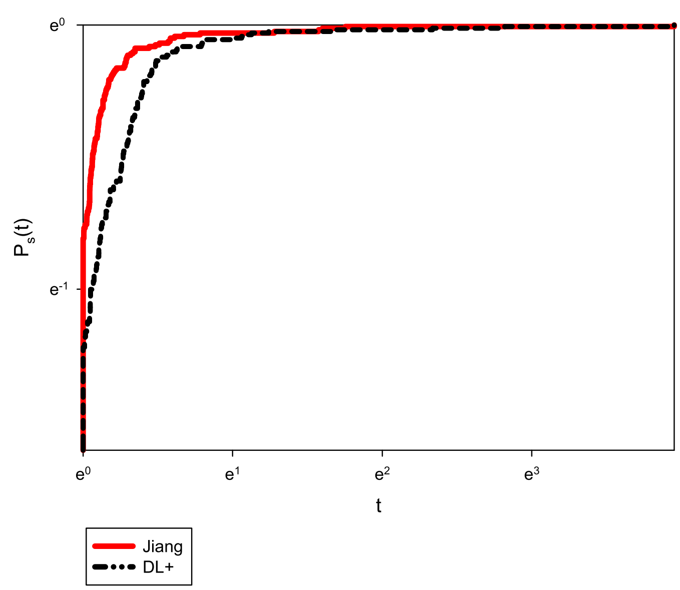


Figure 5 Graph of the number of number of gradient evaluations

Take note that we made use of the (Dai–Liao) CG algorithm and 3TCGHS.

$$\beta_k^{DL+} = \max\left(0, \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}\right) - t\frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}.$$

If the descent condition was met; if not, we employed the steepest-descent approach to restart the algorithm. We utilized the root-mean-square error (RMSE) between the

Table 2 Numerical outcomes from images with Gaussian noise with a 25% standard deviation added to the original images using the Dai-Liao CG method as well as 3TCGHS

Image	Algorithm	Number of iterations	CPU time in seconds	RMSE
Moon 128 pixels	Dai-Liao (DL+)	171	1.189E+001	00.0654
	3TCGHS	173	9.922E+000	00.0652
Baboon 128 pixels	Dai-Liao(DL+)	156	1.198E+001	00.1492
	3TCGHS	154	1.095E+001	00.1487
Cameraman 128 pixels	Dai-Liao(DL+)	162	1.217E+001	00.1146
	3TCGHS	160	9.891E+000	00.1143
Coins 128 pixels	Dai-Liao(DL+)	132	9.422E+000	00.0834
	3TCGHS	133	8.141E+000	00.0836
Moon 256 pixels	Dai-Liao(DL+)	167	5.497E+001	00.0351
	3TCGHS	166	4.917E+001	00.0351
Baboon 256 pixels	Dai-Liao(DL+)	169	1.453E+002	00.1058
	3TCGHS	169	1.433E+002	00.1058
Cameraman 256 pixels	Dai-Liao(DL+)	164	9.627E+001	00.0889
	3TCGHS	164	9.411E+001	00.0887
Coins 256 pixels	Dai-Liao(DL+)	128	5.845E+001	00.0504
	3TCGHS	129	5.539E+001	00.0505
Moon 512 pixels	Dai-Liao(DL+)	155	2.071E+002	00.0183
	3TCGHS	154	1.866E+002	00.0182
Baboon 512 pixels	Dai-Liao(DL+)	173	8.931E+002	00.1070
	3TCGHS	177	8.636E+002	00.1070
Cameraman 512 pixels	Dai-Liao(DL+)	146	3.931E+002	00.0534
	3TCGHS	146	3.600E+002	00.0534
Coins 512 pixels	Dai-Liao(DL+)	129	2.964E+002	00.0326
	3TCGHS	128	2.682E+002	00.0326
Moon 1024 pixels	Dai-Liao(DL+)	151	7.440E+002	00.0080
	3TCGHS	150	6.448E+002	00.0080
Cameraman 1024 pixels	Dai-Liao(DL+)	126	1.210E+003	00.0028
	3TCGHS	125	1.116E+003	00.0026
Coins 1024 pixels	Dai-Liao(DL+)	120	9.912E+002	00.0017
	3TCGHS	114	9.612E+002	00.0015

restored image and the original true image to assess the quality of the restored image.

$$RMSE = \frac{\|\varsigma - \varsigma_k\|_2}{\|\varsigma\|}.$$

The restored image is denoted by ς_k and the true image by ς. The RMSE determines the quality of the restored image, in which lower values correspond to higher quality. The criteria for stopping is

$$\frac{\|x_{k+1} - x_k\|_2}{\|x_k\|_2} < \omega.$$

In this context, $\omega = 10^{-3}$. Note that if $\omega = 10^{-4}$ or $\omega = 10^{-6}$, then RMSE remains constant, meaning that a fixed RMSE can vary in the number of iterations.

Table 2 compares 3TCGHS with the Dai-Liao CG algorithm through a series of numerical experiments. The RMSE, CPU time, and the number of iterations are all compared. It may be observed that the 3TCGHS method performed better than Dai–Liao with respect to CPU time, RMSE, and the number of iterations for most experimental tests.

Table 3 shows the outcomes of restoring destroyed images using Algorithm 1, indicating that it can be considered an efficient approach.

Table 3 Restoration of destroyed images of Coins, Cameraman,d Moon, an Baboon by reducing z via Algorithm 1

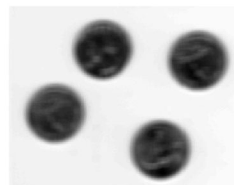
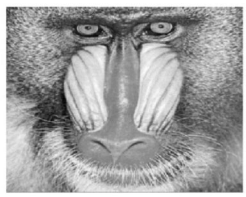
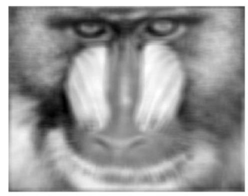
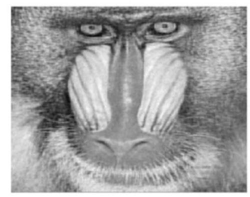
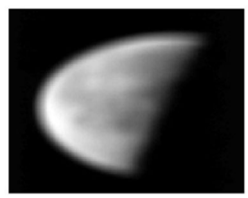
Image	Original image	Image with Gaussian noise	Resorted image
Coins (256 pixels)			
Baboon (256 pixels)			
Moon (128 pixels)			
Cameraman (128 pixels)			

Table 4 Data on demand and price

x:Price ($)	y:Demand
1	5
2	3.5
2	3
2.3	2.7
2.5	2.4
2.6	2.5
2.8	2
3	1.5
3.3	1.2
3.5	1.2

4.2 Application to a regression problem

Table 4 shows data on the prices and demand for some commodities over several years. The data is similar to that used by [28].

The relation between x and y is parabolic; thus, the regression function can be defined as follows:

$$r = w_2x^2 + w_1x + w_0, \tag{19}$$

where w_0, w_1, and w_2 are the regression parameters. We aim to solve the equation given below using the least square method.

$$\min Q = \sum_{j=1}^{n} \left(y_j - \left(w_0 + w_1 x_j + w_2 x_j^2\right)\right)^2.$$

This equation is able to be modified to the following unconstrained optimization problem.

$$\min \sum_{j=1}^{n} f(w)_{w \in R^3} = \sum_{j=1}^{n} \left(y_j - \psi\left(1 + x_j + x_j^2\right)^T\right)^2.$$

Next, we can use Algorithm 1 to get the subsequent outcomes. $w_2 = 0.1345, w_1 = -2.1925, w_0 = 7.0762$.

4.3 Solving system of linear equations in electrical engineering

The main challenge is solving complex systems of linear equations generated from linear circuits with many components. The first CG formula was suggested by Hestenes and Steifel [3] in 1952 to solve the linear equation systems. The linear equation system is presented in the format

$$Qx = b.$$

In the case where the matrix Q is symmetric and positive definite, it may be regarded as a method for resolving a corresponding quadratic function.

$$\min f(x) = \frac{1}{2} x^T Q x - b^T x.$$

To see the similarities between the above equations, differentiate $f(x)$ in relation to x and make the gradient zero. In other words,

$$\nabla f(x) = Qx - b = 0.$$

The following example illustrates using the CG method to solve a linear equation system generated from the circuit.

Example 1 [29, 30]. Consider the circuit shown in Fig. 6. To create the loop equations, use loop analysis. Then, Algorithm 1 is applied to find the solution for the unknown currents.

Kirchhoff's Current Law (often abbreviated as KCL) asserts that all currents entering and leaving a node must sum up to zero algebraically. This law describes the flow of charge into and out of a wire junction point or node. The circuit in Fig. 6 has four loops; thus, Kirchhoff's Loop Equations can be written as follows:

$$\begin{aligned}
&14\mathrm{L}_1 - 3\mathrm{L}_2 - 3\mathrm{L}_3 + 0\mathrm{L}_4 = 0,\\
&-3\mathrm{L}_1 + 10\mathrm{L}_2 + 0\mathrm{L}_3 - 3\mathrm{L}_4 = 0,\\
&-3\mathrm{L}_1 + 0\mathrm{L}_2 + 10\mathrm{L}_3 - 3\mathrm{L}_4 = 0,
\end{aligned}$$

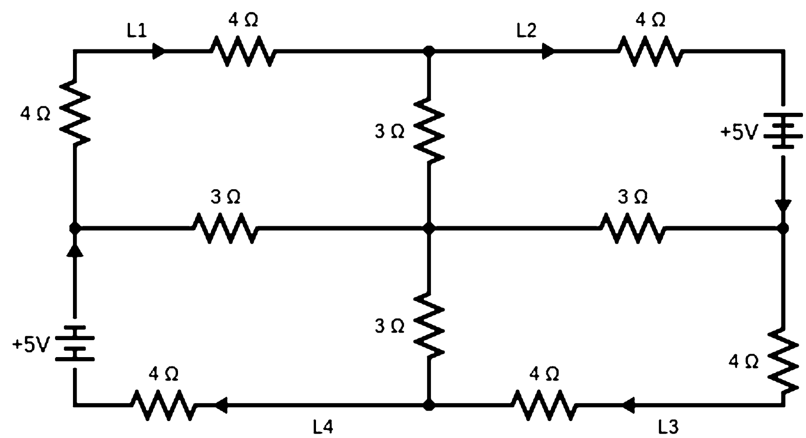


Figure 6 The circuit of Example 1

$$0 - 3L_2 - 3L_3 + 14L_4 = 0,$$

where the following is one way to write the system of equations:

$$\begin{bmatrix} 14 & -3 & -3 & 0 \\ -3 & 10 & 0 & -3 \\ -3 & 0 & 10 & -3 \\ 0 & -3 & -3 & 14 \end{bmatrix} \begin{bmatrix} L1 \\ L2 \\ L3 \\ L4 \end{bmatrix} = \begin{bmatrix} 0 \\ -5 \\ 5 \\ 0 \end{bmatrix}.$$

Thus, we can write the system $Qx = b$ as follows:

$$Q = \begin{bmatrix} 14 & -3 & -3 & 0 \\ -3 & 10 & 0 & -3 \\ -3 & 0 & 10 & -3 \\ 0 & -3 & -3 & 14 \end{bmatrix}, \qquad x = \begin{bmatrix} L1 \\ L2 \\ L3 \\ L4 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \qquad b = \begin{bmatrix} 0 \\ -5 \\ 5 \\ 0 \end{bmatrix}$$

where Q denotes positive definite and symmetric matrix. Thus, we have the following form:

$$f(x) = \frac{1}{2} x^T Q x - b^T x,$$

i.e.,

$$f(x_1, x_2, x_3, x_4) = \frac{1}{2} [x_1 x_2 x_3 x_4] \begin{bmatrix} 14 & -3 & -3 & 0 \\ -3 & 10 & 0 & -3 \\ -3 & 0 & 10 & -3 \\ 0 & -3 & -3 & 14 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} - [0\ 5\ -\ 5\ 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}.$$

After simple calculations, we compute the following function:

$$f(x_1, x_2, x_3, x_4) = 7x_1^2 + 5x_2^2 + 5x_3^2 - 3x_1 x_2 - 3x_3 x_1 - 3x_2 x_4 - 3x_3 x_4 + 5x_2 - 5x_3. \tag{20}$$

Using Algorithm 1, we can find the following solution for Eq. (20)

$$x_1=0, x_2=-0.5, x_3=0.5, x_4=0,$$

and the function value is

$$f=-2.5.$$

5 Conclusion

We have outlined a three-term CG method in the present research that satisfies both the convergence analysis and the descent condition via an SWP line search. Moreover, we have presented numerical results with different values of sigma, showing that the new search direction strongly outperformed alternative approaches with regard to the number of iterations as well as was very competitive when it came to the number of functions, gradients, and CPU time evaluated. Additionally, we have offered an application of the new search direction of image restoration, regression analysis, and solving linear systems in electrical engineering. Algorithm 1 demonstrates its efficiency in restoring destroyed images from degraded pixel data. In addition, using Algorithm 1 to solve a system of linear equations is easier than other traditional methods. In regression analysis, we found Algorithm 1 useful for obtaining the value of regression parameters. In future research, we intend to utilize CG methods in machine learning, mathematical problems in engineering, and neural networks.

Appendix

Function	Dim	No. iter 3TCGHS	No. function 3TCGHS	No. gradient 3TCGHS	CPU time 3TCGHS	No. iter DL+	NO. function DL+	No. gradient DL+	CPU time DL+
AKIVA	2	9	21	13	00.03	8	20	15	00.03
ALLINITU	4	11	25	15	00.03	9	25	18	0.03
ARGLINB	200	2	105	104	0.06	5	73	72	0.09
ARGLINC	200	2	106	104	00.03	5	79	78	0.06
ARWHEAD	5000	8	16	10	0.03	6	16	12	00.03
BARD	3	16	33	17	00.03	12	32	22	00.03
BDEXP	5000	4	9	4	00.03	2	7	7	00.03
BDQRTIC	5000	141	299	278	0.55	168	363	359	0.63
BEALE	2	13	30	18	00.03	11	33	26	00.03
BIGGS3	6	78	172	100	00.03	79	207	144	00.03
BIGGS5	6	78	172	100	00.03	79	207	144	00.03
BIGGS6	6	26	58	33	00.03	24	64	44	00.03
BIGGSB1	5000	2500	2507	4995	2.4	8328	8335	16651	10.86
BOX2	3	10	22	12	00.03	10	23	14	00.03
BOX3	3	10	22	12	00.03	10	23	14	00.03
BOX	10000	7	23	18	0.11	7	25	21	0.09
BRKMCC	2	5	11	6	00.03	5	11	6	00.03
BROYDNBDLS	10	26	54	28	00.03	25	57	34	00.03
BROWNAL	200	2	106	104	0.05	10	29	21	00.03
BROWNBS	2	12	26	17	00.03	10	24	18	00.03
BROWNDEN	4	15	31	19	00.03	16	38	31	00.03
BRYBND	5000	48	103	58	0.17	149	317	174	0.55
CAMEL6	2	11	36	28	00.03	6	22	18	00.03
CHNROSNB	50	281	556	295	00.03	1009	1998	1180	0.01

Function	Dim	No. iter 3TCGHS	No. function 3TCGHS	No. gradient 3TCGHS	CPU time 3TCGHS	No. iter DL+	NO. function DL+	No. gradient DL+	CPU time DL+
CLIFF	2	13	58	40	00.03	10	46	39	0.01
COSINE	10000	14	77	70	0.26	12	52	43	0.2
CUBE	2	30	82	61	00.03	17	48	34	00.03
CURLY10	10000	46959	66455	74448	164.98	68087	88635	105434	240.64
CURLY20	10000	72787	95616	122794	391	88068	112535	142271	540.5
CURLY30	10000	77907	101195	132628	571.64	93324	121387	183418	712.27
DENSCHNA	2	8	18	11	00.03	6	16	12	00.03
DENSCHNB	2	7	15	8	00.03	6	18	15	00.03
DENSCHNC	2	11	26	16	00.03	11	36	31	00.03
DENSCHNE	3	16	49	36	00.03	12	43	38	00.03
DENSCHNF	2	8	17	9	00.03	9	31	26	00.03
DIXMAANA	3000	7	15	8	00.03	6	15	11	00.03
DIXMAANB	3000	6	13	7	00.03	6	15	11	00.03
DIXMAANC	3000	7	15	8	00.03	6	14	9	00.03
DIXMAAND	3000	10	21	11	00.03	7	17	12	00.03
DIXMAANF	3000	160	321	161	0.16	247	499	255	0.27
DIXMAANG	3000	167	335	168	0.19	348	701	356	0.38
DIXMAANH	3000	170	395	233	0.19	332	671	343	0.45
DIXMAANI	3000	3589	3673	7096	3.94	3522	3623	6953	4.66
DIXMAANJ	3000	360	721	361	0.3	476	957	486	0.56
DIXMAANK	3000	280	561	281	0.27	425	854	432	0.49
DIXON3DQ	10000	10000	10007	19995	19.48	15258	15265	30511	37.63
DQDRTIC	5000	5	11	6	00.03	15	32	18	00.03
ECKERLE4LS.SIF	3	3	7	4	00.03	2	6	4	00.03
EDENSCH	2000	28	54	42	0.05	27	66	54	0.03
EGGCRATE	2	6	15	10	00.03	6	15	10	00.03
EG2	1000	2	30	29	00.03	3	13	10	00.03
EIGENALS	2550	9104	17012	10328	152.7	9534	18450	18540	185.64
EIGENBLS	2550	27364	54733	27369	408.39	22540	45340	24700	350.43
EIGENCLS	2652	10972	21962	10992	174.08	13450	26740	18450	203.45
ELATVIDU	2	9	21	13	00.03	8	32	29	00.03
ENGVAL2	3	28	65	41	00.03	26	73	55	00.03
EXPFIT	2	9	21	13	00.03	9	29	22	00.03
EXTROSNB	1000	156	347	203	0.06	7182	12662	10680	2.39
exp2	2	7	16	9	00.03	7	16	9	00.03
FBRAINLS	2	9	23	16	00.03	9	27	21	00.03
FLETCHCR	1000	102	210	110	0.03	199	412	217	0.08
FMINSRF2	5625	378	805	451	1.37	733	1545	826	2.34
FMINSURF	5625	471	964	502	1.53	1245	2567	1342	4.08
GROWTHLS	3	135	425	333	00.03	109	431	369	00.03
GULF	3	35	85	52	00.03	33	95	72	00.03
HAIRY	2	15	62	50	00.03	17	82	68	00.03
HATFLDD	3	20	44	25	00.03	17	49	37	00.03
HATFLDE	3	18	49	32	00.03	13	37	30	00.03
HATFLDFLS	3	56	143	97	00.03	48	156	125	00.03
HEART8LS	8	237	531	309	00.03	253	657	440	00.03
HIELOW	3	15	33	19	0.03	13	30	21	0.05
HILBERTA	2	2	5	3	00.03	2	5	3	00.03
HILBERTB	10	4	9	5	00.03	4	9	5	00.03
HIMMELBB	2	8	25	19	00.03	4	18	18	00.03
HIMMELBF	4	22	50	30	00.03	23	59	46	00.03
HIMMELBG	2	7	20	15	00.03	7	22	17	00.03
HIMMELBH	2	7	16	9	00.03	5	13	9	00.03
HYDCAR6LS.SIF	29	33	70	38	00.03	1001	2027	1174	0.03
INDEF	5000	1	46	147	0.44	1	46	147	0.42
INTEQNELS.SIF	12	7	15	8	00.03	6	13	7	00.03
JENSMP	2	15	33	22	00.03	12	47	41	00.03
JIMACK	3549	8352	16706	8354	1196	11978	23971	12235	1732.9
JUDGE	2	9	21	12	00.03	9	24	18	00.03
KOWOSB	4	18	44	28	00.03	16	46	32	00.03

Function	Dim	No. iter 3TCGHS	No. function 3TCGHS	No. gradient 3TCGHS	CPU time 3TCGHS	No. iter DL+	NO. function DL+	No. gradient DL+	CPU time DL+
KSSLS	1000	2	5	3	0.08	6	19	16	0.55
LIARWHD	5000	16	41	27	0.03	15	41	31	0.03
LOGHAIRY	2	26	114	94	00.03	26	196	179	00.03
LSC1LS	3	36	98	66	00.03	31	108	89	00.03
LSC2LS	3	44	112	78	00.03	37	106	86	00.03
LUKSAN11LS	100	989	2004	1018	0.05	2434	5355	3048	0.13
LUKSAN12LS	98	118	274	165	00.03	252	529	407	0.01
LUKSAN13LS	98	102	186	163	00.03	142	279	243	00.03
LUKSAN14LS	98	151	302	205	00.03	157	313	201	00.03
LUKSAN15LS	100	26	57	40	00.03	27	60	45	00.03
LUKSAN16LS	100	28	55	39	00.03	35	72	53	00.03
MANCINO	100	10	21	11	0.06	11	23	12	0.06
MGH09LS	4	51	139	102	00.03	25	82	72	00.03
MGH10LS	3	1098	2724	3703	00.03	1082	4052	4968	00.03
MGH17LS	5	57	204	170	00.03	84	323	365	00.03
MISRA1DLS.SIF	2	26	115	100	00.03	22	90	84	00.03
MODBEALE.SIF	20000	180	374	242	3.7	224	473	304	4.89
MOREBV	5000	161	168	317	0.31	117	124	229	0.23
MSQRTALS	1024	2726	5458	2733	8.36	8953	17316	9581	28.81
MSQRTBLS	1024	2252	4510	2259	6.3	5786	11558	5818	17.72
NCB20	5010	6619	13128	7244	64.03	11026	20505	15341	129.2
NONCVXU2	5000	9816	16811	12915	25	54585	94397	84907	182.92
NONDIA	5000	7	25	20	0.01	7	25	19	0.03
OSBORNEB	11	62	135	77	00.03	57	134	84	00.03
OSCIPATH	10	301809	689800	405691	2.06	295029	781729	534425	2.42
PALMER1C	8	12	27	28	00.03	12	27	28	00.03
PALMER1D	7	10	24	23	0	10	24	23	00.03
PALMER2C	8	12	21	22	00.03	11	21	22	00.03
PALMER3C	8	11	21	21	00.03	11	21	21	00.03
PALMER4C	8	11	21	21	00.03	11	21	21	00.03
PALMER5C	6	6	13	7	00.03	6	13	7	00.03
PALMER6C	8	11	24	24	00.03	11	24	24	00.03
PALMER7C	8	11	20	20	00.03	11	20	20	00.03
PALMER8C	8	11	19	19	00.03	11	19	19	00.03
PARKCH	15	336	714	393	10.3	412	982	611	16.02
PENALTY1	1000	22	61	45	00.03	14	51	43	00.03
PENALTY2	200	200	234	375	0.03	337	480	758	0.06
PENALTY3	200	93	283	229	1.83	102	346	290	2.19
PENALTY3	200	93	283	229	1.91	102	346	290	2.22
POWELLBSLS	2	62	195	254	0	50	211	234	00.03
POWELLSG	5000	22	46	25	0.03	36	92	65	0.05
POWER	10000	324	701	386	0.66	356	733	391	0.58
POWERSUM	4	5	11	6	00.03	4	10	6	00.03
PRICE3	2	14	30	17	00.03	10	25	17	00.03
PRICE4	2	10	23	13	00.03	9	30	23	00.03
QING	100	68	136	86	00.03	85	179	96	00.03
RAT43LS	4	43	126	91	00.03	44	156	122	00.03
ROSENBR	2	32	78	51	00.03	28	84	65	00.03
ROSENBRTU.SIF	2	44	155	124	00.03	37	175	153	00.03
S308	2	8	19	12	00.03	7	21	17	00.03
SCHMVETT	5000	2	106	104	0.31	59	103	88	0.28
SENSORS	100	23	62	45	0.41	24	71	53	0.47
SINQUAD	5000	15	42	34	0.09	13	46	38	0.09
SISSER	2	6	18	14	00.03	5	19	19	00.03
SNAIL	2	89	232	162	00.03	61	251	211	00.03
SPMSRTLS	4999	217	440	224	0.66	310	633	327	0.81
SROSENBR	5000	9	20	12	00.03	9	23	15	00.03
STREG	4	80	222	168	00.03	60	218	180	00.03
STRATEC	10	222	507	303	6.72	170	419	283	6.3
TESTQUAD	5000	1496	1503	2987	1.61	20325	20361	40674	21.61

Function	Dim	No. iter 3TCGHS	No. function 3TCGHS	No. gradient 3TCGHS	CPU time 3TCGHS	No. iter DL+	NO. function DL+	No. gradient DL+	CPU time DL+
THURBERLS	7	112	260	208	00.03	105	259	216	00.03
TOINTGOR	50	123	222	153	00.03	192	348	270	00.03
TOINTGSS	5000	4	9	5	00.03	4	9	5	00.03
TOINTPSP	50	157	303	241	00.03	145	313	250	00.03
TOINTQOR	50	29	36	53	00.03	49	56	93	00.03
TQUARTIC	5000	12	38	28	00.03	11	41	34	0.03
TRIDIA	5000	782	789	1559	1.08	4699	4721	9408	6.38
TRIGON1.SIF	10	22	45	23	00.03	19	41	22	00.03
TRIGON2.SIF	10	26	52	28	00.03	22	57	43	00.03
VANDANMSLS.SIF	22	6	13	7	00.03	5	11	6	00.03
IM	200	10	21	11	00.03	9	20	15	00.03
VAREIGVL	50	23	48	26	00.03	28	71	51	00.03
VESUVIALS	8	1184	1856	3176	1.22	1262	1954	3155	1.22
VESUVIOULS	8	70	166	125	0.06	79	211	173	0.09
WAYSEA1	2	70	146	79	00.03	11	55	50	00.03
WAYSEA2	2	16	36	21	00.03	9	28	23	00.03
WOODS	4000	32	78	53	00.03	24	62	41	0.03
YATP1CLS	1E+05	18	45	31	5.89	17	48	36	7.12
YATP2CLS	1E+05	8	23	17	2.02	7	27	22	2.67
YFITU	3	71	176	116	00.03	68	208	167	0.03
ZANGWIL2	2	1	3	2	00.03	1	3	2	00.03

Acknowledgements
This project is supported by the Researchers Supporting Project number (RSP2024R317), King Saud University, Riyadh, Saudi Arabia. The authors greatly appreciate the editors and reviewers for any advice and insights that could help enhance this work. We express our gratitude to Dr. William Hager for sharing his CG method code.

Author contributions
Ahmad Alhawarat wrote the main text with proofs. Zabidin Salleh review the proofs. Hanan Alolaiyan prepared the figures and computer software. Hamid El hor and Shahrina Ismail did the application part.

Funding
The third author would like to thank King Saud University for its partial support in this paper. The corresponding author would like to express here sincere gratitude to Universiti Sains Islam Malaysia (USIM) for the financial support it has given us in this research.

Data Availability
No datasets were generated or analysed during the current study.

Declarations

Competing interests
The authors declare no competing interests.

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, College of Science, King Saud University, Riyadh, 11451, Saudi Arabia. [3]Sidel–Simulations, D-63303, Dreieich, Germany. [4]Financial Mathematics Program, Faculty of Science and Technology, Universiti Sains Islam Malaysia, Bandar Baru Nilai, 71800, Nilai, Negeri Sembilan, Malaysia.

Received: 4 January 2024 Accepted: 27 April 2024 Published online: 28 May 2024

References
1. Wolfe, P.: Convergence conditions for ascent methods. SIAM Rev. **11**, 226–235 (1969)
2. Wolfe, P.: Convergence conditions for ascent methods. II: some corrections. SIAM Rev. **13**, 185–188 (1971)
3. Hestenes, M.R., Stiefel, E.: Methods of conjugate gradients for solving linear systems. J. Res. Natl. Bur. Stand. **49**, 409–436 (1952)
4. Polak, E., Ribiere, G.: Note sur la convergence de méthodes de directions conjuguées. Rev. Fr. Inform. Rech. Opér. Sér. Rouge **3**, 35–43 (1969)
5. Liu, Y., Storey, C.: Efficient generalized conjugate gradient algorithms, part 1: theory. J. Optim. Theory Appl. **69**, 129–137 (1991)
6. Powell, M.J.D.: Nonconvex minimization calculations and the conjugate gradient method. In: Numerical Analysis: Proceedings of the 10th Biennial Conference, pp. 122–141. Springer, Berlin (1984)

7. Fletcher, R.: Function minimization by conjugate gradients. Comput. J. **7**, 149–154 (1964)
8. Fletcher, R.: Practical Methods of Optimization. Wiley, New York (2000)
9. Dai, Y.H., Yuan, Y.: A nonlinear conjugate gradient method with a strong global convergence property. SIAM J. Optim. **10**, 177–182 (1999)
10. Dai, Y.H., Liao, L.Z.: New conjugacy conditions and related nonlinear conjugate gradient methods. Appl. Math. Optim. **43**, 87–101 (2001)
11. Hager, W.W., Zhang, H.: A new conjugate gradient method with guaranteed descent and an efficient line search. SIAM J. Optim. **16**, 170–192 (2006)
12. Hager, W.W., Zhang, H.: The limited memory conjugate gradient method. SIAM J. Optim. **23**, 2150–2168 (2013)
13. Andrei, N.: A simple three-term conjugate gradient algorithm for unconstrained optimization. J. Comput. Appl. Math. **241**, 19–29 (2013)
14. Andrei, N.: On three-term conjugate gradient algorithms for unconstrained optimization. Appl. Math. Comput. **219**, 6316–6327 (2013)
15. Babaie-Kafaki, S., Ghanbari, R.: Two modified three-term conjugate gradient methods with sufficient descent property. Optim. Lett. **8**, 2285–2297 (2014)
16. Deng, S., Wan, Z.: A three-term conjugate gradient algorithm for large-scale unconstrained optimization problems. Appl. Numer. Math. **92**, 70–81 (2015)
17. Liu, J.K., Xu, J.Ł., Zhang, L.Q.: Partially symmetrical derivative-free Liu–Storey projection method for convex constrained equations. Int. J. Comput. Math. **96**, 1787–1798 (2019)
18. Liu, J.K., Zhao, Y.X., Wu, X.L.: Some three-term conjugate gradient methods with the new direction structure. Appl. Numer. Math. **150**, 433–443 (2020)
19. Yao, S., Ning, L., Tu, H., Xu, J.: A one-parameter class of three-term conjugate gradient methods with an adaptive parameter choice. Optim. Methods Softw. **35**, 1051–1064 (2020)
20. 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**, 1299–1312 (2017)
21. Jiang, X., Jian, J., Song, D., Liu, P.: An improved Polak–Ribière–Polyak conjugate gradient method with an efficient restart direction. Comput. Appl. Math. **40**, 1–24 (2021)
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**, 1–15 (2021)
23. Al-Baali, M.: Descent property and global convergence of the Fletcher—Reeves method with inexact line search. IMA J. Numer. Anal. **5**, 121–124 (1985)
24. Zoutendijk, G.: Some algorithms based on the principle of feasible directions. In: Nonlinear Programming, pp. 93–121. Elsevier, Amsterdam (1970)
25. Gilbert, J.C., Nocedal, J.: Global convergence properties of conjugate gradient methods for optimization. SIAM J. Optim. **2**, 21–42 (1992)
26. Bongartz, I., Conn, A.R., Gould, N., Toint, P.L.: CUTE: constrained and unconstrained testing environment. ACM Trans. Math. Softw. **21**, 123–160 (1995)
27. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
28. Yuan, G., Wei, Z.: Non monotone backtracking inexact BFGS method for regression analysis. Commun. Stat. Methods **42**, 214–238 (2013)
29. Bakr, M.: Nonlinear optimization in electrical engineering with applications in Matlab. The Institution of Engineering and Technology (2013)
30. Mishra, S.K., Ram, B.: Introduction to Unconstrained Optimization with R. Springer, Berlin (2019)

Publisher's Note