A Descent Conjugate Gradient Method for Large Scale Unconstrained Optimization Problems with Application

Ahmad Alhawarat[1], Sultanah Masmali[2], Ibrahim M. Sulaiman[3, 4], Issam A. R. Moghrabi[5,*], Norazura Ahmad[3], Shahrina Ismail[6]

[1]Department of Mathematics, Faculty of Arts and Science, Amman Arab University, Amman 11953, Jordan

[2]Department of Mathematics, College of Science, Jazan University, Jazan, Saudi Arabia

[3]School of Quantitative Sciences, Universiti Utara Malaysia, Sintok, 06010, Kedah, Malaysia

[4]Faculty of Education and Art, Sohar University, Sohar 311, Oman

[5]Information Systems and Technology Department, Kuwait Technical College, Kuwait

[6]Financial Mathematics Program, Faculty of Science and Technology, Universiti Sains Islam Malaysia, Bandar Baru Nilai, 71800, Nilai, Negeri Sembilan, Malaysia

Emails: abadee2010@yahoo.com; sultanah@math.ksu.edu; sulaimancga@gmail.com; i.moghrabi@ktech.edu.kw; nhaslinda@ummaed.my

Abstract

In recent years, there has been a surge of attention to the Conjugate Gradient Method (CGM) and its applications. This is because the algorithm of CGM does not require the computation of the second derivative or an approximation during the iteration process. In this study, a four-term descent CGM is proposed by utilizing the famous Polak–Ribiere–Polyak (PRP) conjugate gradient formula. The direction of the proposed method achieves the descent property without line search consideration. In addition, the convergence properties are met to generate the stationary points. Findings from numerical experiments on unconstrained optimization and robotic motion control problems demonstrate that the novel approach outperforms some existing methods including the famous CG-Descent conjugate gradient method.

Keywords: Inexact line search; Conjugate gradient method; Descent condition; CG-Descent; Numerical comparison

1. Introduction

Assume we have the optimization problem as expressed below:

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

in which f: $R^n \rightarrow R$. The Conjugate Gradient Method (CGM) obtains the stationary point utilizing an iterative starting point x_1 provided below:

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

Usually, we employ the strong Wolfe-Powell (SWP) [1,2] line search in determining the step length, which is provided by the equations below:

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k \nabla f_k^T d_k, \tag{3}$$

and

DOI: https://doi.org/10.54216/IJNS.260220

$$\sigma \nabla f_k^T d_k \leq g(x_k + \alpha_k d_k)^T d_k \leq -\sigma \nabla f_k^T d_k, \quad (4)$$

in which $0 < \delta < \frac{1}{2}, \delta < \sigma < 1$.

The Weak Wolfe-Powell (WWP) line search is expressed in equation (3) as well as

$$\nabla f(x_k + \alpha_k d_k)^T d_k \geq \sigma \nabla f_k^T d_k. \quad (5)$$

The CGM's search direction is expressed below by:

$$d_k = \begin{cases} -\nabla f_k, & \text{if } k = 1, \\ -\nabla f_k + \beta_k d_{k-1}, & \text{if } k \geq 2, \end{cases} \quad (6)$$

in which $g_k = g(x_k) = \nabla f$, while β_k is recognized as the CG formula.

Table 1 displays the most widely employed classical CGM given the subsequent qualifications: Hestenses–Stiefel (HS) [3], Polak–Ribiere –Polyak (PRP) [4], Liu and Storey (LS) [5], Fletcher–Reeves (FR) [6], Fletcher (CD) [7] as well as Dai and Yuan (DY) [8]:

Table 1: Classical CGM groups

Not robust but efficient CGM	$\beta_k^{HS} = \frac{\nabla f_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}$	$\beta_k^{PRP} = \frac{\nabla f_k^T y_{k-1}}{\|g_{k-1}\|^2}$	$\beta_k^{LS} = -\frac{\nabla f_k^T y_{k-1}}{d_{k-1}^T g_{k-1}}$
Inefficient but robust CGM	$\beta_k^{FR} = \frac{\|\nabla f_k\|^2}{\|\nabla f_{k-1}\|^2}$	$\beta_k^{CD} = -\frac{\|\nabla f_k\|^2}{d_{k-1}^T g_{k-1}}$	$\beta_k^{DY} = \frac{\|\nabla f_k\|^2}{d_{k-1}^T g_{k-1}}$

in which $y_{k-1} = g_k - g_{k-1}$.

Dai and Liao [9], in the year 2001, introduced the conjugacy condition given below:

$$d_k^T y_{k-1} = -t \nabla f_k^T s_{k-1}, \quad (7)$$

Here, $s_{k-1} = x_k - x_{k-1}$and $t \geq 0$. When $t = 0$, equation (7) given below represents the classical conjugacy condition. Now, by implementing equations (6) as well as (7), [9] introduced the CG formula given below:

$$\beta_k^{DL} = \frac{\nabla f_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} - t \frac{\nabla f_k^T s_{k-1}}{d_{k-1}^T y_{k-1}} = \beta_k^{HS} - t \frac{\nabla f_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}. \quad (8)$$

Nevertheless, β_k^{DL} inherits a similar issue as β_k^{HS} and β_k^{PRP}, for instance, $\beta_k^{DL}$is not positive in general. Hence, [9] is employed to replace equation (8) with:

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

In the year 2006, Hager and Zhang [10,11] proposed the CG formula given below depending on equation (8), which is expressed by:

$$\beta_k^{HZ} = \max\{\beta_k^N, \eta_k\}, \quad (9)$$

in which $\beta_k^N = \frac{1}{d_k^T y_k}(y_k - 2d_k \frac{\|y_k\|^2}{d_k^T y_k})^T g_k$, $\eta_k = -\frac{1}{\|d_k\| \min\{\eta, \|\nabla f_k\|\}}$, while $\eta > 0$ denotes a constant value.

Provided that $t = 2\frac{\|y_k\|^2}{s_k^T y_k}$, we now have $\beta_k^N = \beta_k^{DY}$.

Furthermore, Andrei [12,13] introduced two Three-Term CG (TTCG) methods in the year 2013 building on equation (8) as given below:

DOI: https://doi.org/10.54216/IJNS.260220

$$d_k = -g_k + (\eta_k)d_{k-1} - \theta_k y_{k-1}, t_k = \left(1 + \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}}\right),$$

$$d_k = -g_k + (\eta_k)d_{k-1} - \theta_k y_{k-1}, t_k = \left(1 + \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}}\right),$$

where

$$\eta_k = \frac{\nabla f_k^T y_{k-1} - t\nabla f_k^T s_{k-1}}{y_{k-1}^T d_{k-1}}, \theta_k = \frac{\nabla f_k^T d_{k-1}}{y_{k-1}^T d_{k-1}}.$$

Moreover, Babaie-Kafaki and Ghanbari [14] provided another variation described by equation (8) in 2014, given by:

$$d_k = -\nabla f_k + (\eta_k)d_{k-1} - \theta_k y_{k-1}, t_k = \left(max\left(\varsigma, 1 - \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}}\right)\right),$$

where $\varsigma > 0$.

Similar to [12–14], Deng and Wan [15], in the year 2015, presented the CGM as given below:

$$d_k = -\nabla f_k + (\eta_k)d_{k-1} - \theta_k y_{k-1}, t_k = \left(max\left(0{,}1 - \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}}\right)\right).$$

In the year 2019, Liu et al. [16] recommended and utilized g_{k-1} as a search direction given by:

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

with the following assumptions given by:

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

In addition, Liu et al. [17] suggested the TTCG method, as expressed below:

$$d_k = -\nabla f_k + \left(\beta_k^{LS} - \frac{\|\nabla f_{k-1}\|^2 \nabla f_k^T s_{k-1}}{(a_{k-1}^T \nabla f_{k-1})^2}\right) d_{k-1} - \left(\frac{\nabla f_k^T d_{k-1}}{a_{k-1}^T y_{k-1}}\right) \nabla f_{k-1},$$

which is used in solving equation (1). Consequently, Liu et al. [17] recommended a new TTCG method given below:

$$d_k^{\text{MTTPRP}} = -\nabla f_k + \left(\beta_k^{PRP} - \frac{\nabla f_k^T s_{k-1}}{\|\nabla f_{k-1}\|^2}\right) d_{k-1} + \left(\frac{\nabla f_k^T d_{k-1}}{\|\nabla f_{k-1}\|^2}\right) g_{k-1}.$$

Moreover, Alhawarat et al. [18] presented the four-term CGM by employing the directions expressed below: $-\nabla f_k, d_{k-1}, y_{k-1}$, and s_{k-1}

$$d_k^{FTCGHS} = -\nabla f_k + \left(\beta_k^{HS} - t_k \frac{\nabla f_k^T s_{k-1}}{y_{k-1}^T d_{k-1}}\right) d_{k-1} - \left(\frac{\nabla f_k^T d_{k-1}}{y_{k-1}^T d_{k-1}}\right)(y_{k-1} + s_{k-1}). \quad (10)$$

The CGM may be employed in multiple research areas, such as mathematical problems in engineering, neural networks, image restoration, and many others. For further discussion regarding the CGM and its applications, we encourage the reader to read the following reference [19].

2. The new search direction

The descent condition provided in equation (10) is satisfied utilizing the Wolfe-Powell line search. Here, the new search direction fulfils the descent condition without requiring any line search by using β_k^{PRP}. Moreover, function evaluations, CPU time, gradient evaluations, as well as iteration numbers are substantially more efficient than equation (10) and other well-known approaches like CG-Descent and DL+. The modified search direction is provided below:

DOI: https://doi.org/10.54216/IJNS.260220

$$d_k^{FTCGPRP} = -\nabla f_k + \left(\beta_k^{PRP} - t\frac{\nabla f_k^T s_{k-1}}{\|\nabla f_{k-1}\|^2}\right) d_{k-1} - \left(\frac{\nabla f_k^T d_{k-1}}{\|\nabla f_{k-1}\|^2}\right)(y_{k-1} + s_{k-1}). \quad (11)$$

where $t > 0$. We make the subsequent assumptions for equation (11) in this paper:

$\eta_k = \beta_k^{PRP} - t\frac{\nabla f_k^T s_{k-1}}{\|\nabla f_{k-1}\|^2}$, and $\theta_k = \left(\frac{\nabla f_k^T d_{k-1}}{\|\nabla f_{k-1}\|^2}\right)$.

Provided that we apply an exact line search, equation (11) may be simplified to the PRP CGM. Correspondingly, we obtain the following results by employing exact line search properties:

$\nabla f_k^T d_{k-1} = 0$,

given that $\nabla f_k^T s_{k-1} = \alpha_k g_k^T d_{k-1}$. Here, we attain:

$$d_k^{FTCGPRP} = -\nabla f_k + (\beta_k^{PRP}) d_{k-1}.$$

The CGM stages to accomplish the stationary point implementing equation (11) as well as SWP line search with a stopping criterion $\|\nabla f_k\| \leq 10^{-6}$ listed in Algorithm 1.

Algorithm 1

Step 1. Fix an initial pointx_1, this point may be standard or arbitrary with regard to scientific functions. Here, the initial search direction refers to the negative gradient, for instance,$d_1 = -\nabla f_1$. Now let $k := 1$.

Step 2. Given that the stopping condition is accomplished, we can then stop.

Step 3. Compute the search direction d_k predicated on equation (2) utilizing equation (11).

Step 4. Compute the step size α_k employing the equations (3) and (4).

Step 5. Update x_{k+1} depending on equation (2).

Step 6. Let $k := k + 1$, we then repeat Step 2.

3. Global convergence analysis of the CGM with Algorithm 1

The assumption below is crucial to accomplish the convergence pertaining to CGM's analysis.

Assumption 1

A. The level set $\Omega = \{x | f(x) \leq f(x_1)\}$ is bounded, denoting the existence of a positive constant ρ given that:

$$\|x\| \leq \rho, \forall x \in \Omega.$$

B. Considering neighbourhoods $Q$of Ω,f resembles a continuously differentiable function, which possesses Lipschitz continuous gradient. In other words, $\forall x, y \in Q$, there appears a constant $L > 0$ provided that:

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|.$$

Furthermore, we may deduce from this Assumption the existence of a positive constant B given that:

$\|\nabla f(u)\| \leq B, \quad \forall u \in N$.

The descent condition (downhill condition) is expressed by:

$$\nabla f_k^T d_k < 0, \quad \forall k \geq 1. \quad (12)$$

It is significant in the global convergence analysis's proof and is relevant in the research of CGM. Equation (12) was then amended by Al-Baali [24] to the equivalent form given by:

DOI: https://doi.org/10.54216/IJNS.260220

$$g_k^T d_k \le -c\|g_k\|^2, \quad \forall k \ge 1, \tag{13}$$

in which $c > 0$.

3.1 The descent property with regard to the new search direction

Without utilizing any line search, we may establish that the search direction given in equation (11) meets the required descent condition (13) given in the theorem below.

Theorem 3.1 Let the sequences$\{x_k\}$ and$\{d_k\}$be produced employing equation (11) and equation (2), in whichα_k can be computed via any line search. Following that, the sufficient descent condition provided in equation (13) is proven true.

Proof. Multiplying (11) by g_k^T yields:

$$\begin{aligned}
\nabla f_k^T d_k &= -\|\nabla f_k\|^2 + \left(\frac{\nabla f_k^T y_{k-1}}{\|\nabla f_{k-1}\|^2} - t\frac{\nabla f_k^T s_{k-1}}{\|\nabla f_{k-1}\|^2}\right) g_k^T d_{k-1} - \left(\frac{\nabla f_k^T d_{k-1}}{\|\nabla f_{k-1}\|^2}\right) \nabla f_k^T (y_{k-1} + s_{k-1}) \\
&= -\|\nabla f_k\|^2 + \left(\frac{\nabla f_k^T y_{k-1}}{\|\nabla f_{k-1}\|^2}\right) \nabla f_k^T d_{k-1} - t\left(\frac{\nabla f_k^T s_{k-1}}{\|\nabla f_{k-1}\|^2}\right) \nabla f_k^T d_{k-1} - \left(\frac{\nabla f_k^T d_{k-1}}{\|\nabla f_{k-1}\|^2}\right) \nabla f_k^T y_{k-1} \\
&\quad - \left(\frac{\nabla f_k^T d_{k-1}}{\|\nabla f_{k-1}\|^2}\right) \nabla f_k^T s_{k-1} \\
&= -\|\nabla f_k\|^2 - t\alpha_k \left(\frac{\nabla f_k^T d_{k-1}}{\|\nabla f_{k-1}\|^2}\right) \nabla f_k^T d_{k-1} - \alpha_k \left(\frac{\nabla f_k^T d_{k-1}}{\|\nabla f_{k-1}\|^2}\right) \nabla f_k^T d_{k-1} \\
&= -\|\nabla f_k\|^2 - \alpha_k \left(\frac{\|\nabla f_k^T d_{k-1}\|^2}{\|\nabla f_{k-1}\|^2}\right)(t+1).
\end{aligned}$$

Therefore,

$$\nabla f_k^T d_k \le -\|\nabla f_k\|^2.$$

The proof is then complete. ■

The Zoutendijk condition [25] provided a valuable Lemma to assess the CGM's global convergence property. The following is the Lemma:

Lemma 3.1 Consider that Assumption 1 is valid. Next, recognize any CGM of the form (2), (6), and $\alpha_k$that meets the WWP line search given in equations (3) and (4) with regard to the descent direction. It follows that the subsequent condition implements:

$$\sum_{k=0}^{\infty} \frac{(\nabla f_k^T d_k)^2}{\|d_k\|^2} < \infty. \tag{14}$$

Furthermore, Dai et al. [9] proposed a beneficial theorem to gain the global convergence theorem with regard to the CGM, as stated in Theorem 4.2. below:

Theorem 3.2. We assume that Assumption 1 holds. Now, let any CGM in equations (2) as well as (6), in which $d_k$resembles the descent direction. Meanwhile, α_k is expressed following the SWP line search provided that:

$$\sum_{k\ge 1}^{\infty} \frac{1}{\|d_k\|^2} = \infty.$$

Hence,

DOI: https://doi.org/10.54216/IJNS.260220

$$\liminf_{k\to\infty}\|\nabla f_k\| = 0.$$

3.2 Global convergence of Algorithm 1 having general non-linear functions

The accompanying constraint concerning η_k is crucial in deciding our new search direction's convergence analysis. This constraint's main goal is to keep the CGM multiplayer from becoming non-negative:

$$\eta_k^+ = max\left\{0, \beta_k^{PRP} - t\frac{\nabla f_k^T s_{k-1}}{\|\nabla f_{k-1}\|^2}\right\}.$$

Therefore, equation (11) is provided below:

$$d_k = -g_k + \eta_k^+ d_{k-1} - \theta_k(y_{k-1} + s_{k-1}).$$

Lemma 3.2 Let us assume that Assumption 1 remains true, while the sequences $\{g_k\}$ as well as $\{d_k\}$ are produced employing Algorithm 1. Meanwhile, the step size α_k is measured using the SWP line searches given in equations (3) and (4) or WWP line searches provided in equations (3) and (5). This holds provided that the sufficient descent condition remains true. Furthermore, given that $\beta_k \geq 0$, there exists a constant $\gamma > 0$ provided that $\|g_k\| > \gamma$ with $\forall k \geq 1$. It follows that $d_k \neq 0$ such that:

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

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

Proof. Initially, provided that $d_k = 0$. Following from here, given the sufficient descent condition, we attain $\nabla f_k = 0$. Hence, we make an assumption that $d_k \neq 0$. This implies:

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

We now express:

$$u_k = w_k + \delta_k u_{k-1},$$

in which

$$w_k = \frac{-g_k - \theta_k(y_{k-1}+s_{k-1})}{\|d_k\|},\ \delta_k = \eta_k^+\frac{\|d_{k-1}\|}{\|d_k\|}.$$

Given that u_k denotes a unit vector, we can then say that:

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

Using the triangular inequality and $\delta_k \geq 0$, we now have:

$$\|u_k - u_{k-1}\| \leq (1+\delta_k)\|u_k - u_{k-1}\| = \|u_k - \delta_k u_{k-1} - (u_{k-1} - \delta_k u_k)\|, \tag{17}$$

$$\leq \|u_k - \delta_k u_{k-1}\| + \|u_{k-1} - \delta_k u_k\| = 2\|w_k\|.$$

We now express:

$$v = -\nabla f_k - \theta_k(y_{k-1} + s_{k-1}).$$

By employing the triangular inequality, we now achieve:

$$\|v\| \leq \|\nabla f_k\| + |\theta_k|\,\|y_{k-1} + s_{k-1}\|. \tag{18}$$

From Theorem 3.1, we may deduce the subsequent equation:

DOI: https://doi.org/10.54216/IJNS.260220

$$\nabla f_k^T d_k = -\|\nabla f_k\|^2 - \alpha_k \left(\frac{\|\nabla f_k^T d_{k-1}\|^2}{\|\nabla f_{k-1}\|^2} \right) (t+1)$$

Hence, we have:

$$|\theta_k| = \left| \frac{\nabla f_k^T d_{k-1}}{\|\nabla f_{k-1}\|^2} \right| \leq \frac{\|\nabla f_k\|^2}{\|\nabla f_{k-1}\|^2} + \alpha_k \left(\frac{\sigma^2 \|\nabla f_{k-1}\|^4}{\|\nabla f_{k-1}\|^4} \right) (t+1) = \frac{\|\nabla f_k\|^2}{\|\nabla f_{k-1}\|^2} + \alpha_k \sigma^2 (t+1) = \frac{\bar{\gamma}^2}{\gamma^2} + \alpha_k \sigma^2 (t+1) = \varpi.$$

in which ϖdenotes a positive constant.

Moreover, by employing Assumption 1, the triangular inequality yields:

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

Therefore, the inequality in (18) may be expressed by:

$$\|v\| \leq B + \varpi(\, 2B + 2\rho).$$

By considering

$\omega = B + \varpi(\, 2B + 2\rho)$,

we now have

$\|v\| \leq \omega$.

Referring to equation (17), the following inequality holds:

$$\|u_k - u_{k-1}\| \leq 2w.$$

By utilizing equations (15) as well as (16), we acquire the inequality expressed below:

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

The proof is then complete. ■

The property given below, indicated as Property*, was proposed by Gilbert and Nocedal in [26].

Property*

Provided that we have a method of the form (2) as well as (6). Given also that equation (16) holds true for all$k \geq 1$. It follows that the CGM possesses Property* such that there exist constants $b > 1$ as well as $\lambda > 0$ for all $k \geq 1$,$|\eta_k| \leq b$ and $\|s_k\| \leq \lambda$. We now acquire:

$$|\eta_k| \leq \frac{1}{2b}$$

Lemma 3.3 Provided that the CGM resembles the form given in equation (2). Meanwhile, equation (6) with η_k^+ has step size that fulfills SWP line search (4) and (5). Given that equation (16) remains true, consequently, η_k^+ holds Property*. Particularly, they assume that equation (16) holds. Thus, there exist $b > 1$and $\lambda > 0$for all $k \geq 1$ whereby$|\eta_k^+| \leq b$. Provided that $\|s_k\| \leq \lambda$, we have $|\eta_k^+| \leq \frac{1}{2b}$.

Proof. As an outcome, we set $b = \frac{2(L+t)\bar{\gamma}B}{\gamma^2} \geq 1$, while$\lambda = \frac{\gamma^2}{2b(L+t)\bar{\gamma}}$.

Utilizing SWP (4) as well as (5) together with equation (16), we acquire:

DOI: https://doi.org/10.54216/IJNS.260220

$$|\eta_k^+| \leq \left|\frac{\nabla f_k^T y_{k-1}}{\|\nabla f_{k-1}\|^2}\right| + t\left|\frac{\nabla f_k^T s_{k-1}}{\|\nabla f_{k-1}\|^2}\right| \leq \frac{L\|\nabla f_k\|\|s_{k-1}\| + t\|\nabla f_k\|\|s_{k-1}\|}{\gamma^2} \leq \frac{2(L+t)\bar{\gamma}B}{\gamma^2} = b > 1.$$

for $\|s_k\| \leq \lambda$, while

$$|\eta_k^+| \leq \left|\frac{\nabla f_k^T y_{k-1}}{\|\nabla f_{k-1}\|^2}\right| + t\left|\frac{\nabla f_k^T s_{k-1}}{\|\nabla f_{k-1}\|^2}\right| \leq \frac{L\|\nabla f_k\|\|s_{k-1}\| + t\|\nabla f_k\|\|s_{k-1}\|}{\gamma^2} \leq \frac{(L+t)\bar{\gamma}\lambda}{\gamma^2}$$

$|\eta_k^+| \leq \frac{1}{2b}$.

The proof is then complete. ■

The following Lemma and theorem are comparable to those provided in [20]. Lemma 3.4 is presented here without its proof, which can be identified in [20].

Lemma 3.4. Provided that Assumption 1 remains true. Moreover, consider that the sequences $\{d_k\}$ as well as $\{\nabla f_k\}$ are produced utilizing Algorithm 1, where α_k is calculated using the WWP line search. Note that the sufficient descent condition holds, taking into consideration that the method abides by Property*. Moreover, $\|g_k\| \geq \gamma$ for some$\lambda > 0$. Consequently, there exists $\lambda > 0$ given that for any$\Delta \in N$as well as any index k_0, there exists an index $k > k_0$ such that:

$$\left|\kappa_{k,\Delta}^{\lambda}\right| > \frac{\lambda}{2},$$

in which $\kappa_{k,\Delta}^{\lambda} = \{i \in N: k \leq i \leq k + \Delta\text{-}1, \|s_i\| > \lambda, N$, N resembles the positive integers set while $\left|\kappa_{k,\Delta}^{\lambda}\right|$ resembles the elements number in$\kappa_{k,\Delta}^{\lambda}$.

Theorem 3.4 Let Assumption 1 remains true. Suppose that sequences $\{\nabla f_k\}$ as well as $\{d_k\}$ are developed via Algorithm 1, where α_k is calculated using the sufficient descent condition and the WWP line search. Also, assume that Property* holds. Following from here, we obtain $\lim_{k\to\infty} inf\|\nabla f_k\| = 0$.

Proof. Depending on Lemma 3.2 as well as Lemma 3.4, the proof is executed by contradiction. We express $u_i := \frac{d_i}{\|d_i\|}$. For any two indexes, l, k in which $l \geq k$, we obtain the following:

$$x_l - x_{k-1} = \sum_{i=k}^{l}\|s_{i-1}\|u_{i-1} = \sum_{i=k}^{l}\|s_{i-1}\|u_{k-1} + \sum_{i=k}^{l}\|s_{i-1}\|(u_{i-1} - u_{k-1}),$$

in which$s_{i-1} = x_i - x_{i-1}$.

Considering the norms, we have:

$\sum_{i=k}^{l}\|s_{i-1}\| \leq \|x_l\| + \|x_{k-1}\| + \sum_{i=k}^{l}\|s_{i-1}\|\|u_{i-1} - u_{k-1}\|$.

Utilizing Assumption 1, we attain that the sequence $\{x_k\}$ is bounded. Moreover, there exists a positive constant η given that$\|x_k\| \leq \eta$, for all $k \geq 1$.Hence,

$\|x_l\| + \|x_{k-1}\| \leq 2\eta$,

which demonstrates that:

$$\sum_{i=k}^{l}\|s_{i-1}\| \leq 2\eta + \sum_{i=k}^{l}\|s_{i-1}\|\|u_{i-1} - u_{k-1}\|. \tag{19}$$

Consider$\lambda > 0$ as provided by Lemma 3.4. We can now express the notation as follows:

$\Delta := \left\lceil\frac{8\eta}{\lambda}\right\rceil$.

DOI: https://doi.org/10.54216/IJNS.260220

We can find an index k_0 by applying Lemma 3.2 in which:

$$\sum_{k\geq k_0}^{\infty}\|u_i - u_{i-1}\|^2 < \frac{1}{4\Delta}. \tag{20}$$

Given this k_0 and Δ, Lemma 3.4 provides an index $k \geq k_0$ in which:

$$\left|\kappa_{k,\Delta}^{\lambda}\right| > \frac{\Delta}{2}. \tag{21}$$

Afterwards, given the equation (20) and Cauchy-Schwarz inequality, we attain any index with regard to $i \in [k, k + \Delta - 1]$, in which:

$\|u_{i-1} - u_{k-1}\| \leq \sum_{j=k}^{i-1}\|u_j - u_{j-1}\|,$

$\leq (i-k)^{1/2}(\sum_{j=k}^{i-1}\|u_j - u_{j-1}\|^2)^{1/2},$

$$\leq \Delta^{1/2}(\frac{1}{4\Delta})^{1/2} = \frac{1}{2}.$$

From relations (19) and (21), utilizing$l = k + \Delta - 1$, we obtain:

$$2\eta \geq \frac{1}{2}\sum_{i=k}^{k+\Delta-1}\|s_{i-1}\| > \frac{\lambda}{2}\left|\kappa_{k,\Delta}^{\lambda}\right| > \frac{\lambda\Delta}{4}.$$

Hence, $\Delta < 8\eta/\lambda$, contradicting the Δ definition. The proof is then completed. ■

4. Numerical results and discussion

The study considered several test functions from Andrei [21] as presented in Table 2 to assess the efficiency of new search direction. Therefore, the following large notations are utilized in Table 2. Note that the term DIM refers to the dimension of the function and the metrics used to access the efficiency of the methods are ITER which stands for iterations number, NOF which represents function evaluation number, and CPU denoting CPU time. The performance of the proposed method was compared with other well known and powerful CG coefficients including MTTPRP [17], ADAP [17], CG-DESCENT [10], DPRP [28], and PRP-DC [27] formulas. Meanwhile, for the search strategy, the SWP line search was utilized to determine the step length with $\delta = 0.01$ as well as $\sigma = 0.1$ used as the parameter value of the proposed method while the default values reported in the classical methods were maintained.

For all algorithms, the gradient's norm was applied as the stopping criterion, primarily $\|\nabla f_k\| \leq 10^{-6}$ or when the iteration exceeds 2000. Note that the host computer uses an AMD A4-7210 APU Radeon R3 Graphics with 4 GB of installed RAM and Ubuntu 20.04.2.0 LTS as the operating system. The detailed experimental results are presented in Table 2 below.

DOI: https://doi.org/10.54216/IJNS.260220

Table 2: Numerical performance based on ITER, NOF, and CPU time

		FTCGPRP			MTTPRP			ADAP			DPRP			CG-DESCENT			PRP-DC		
FUNCTION	**DIM**	**ITER**	**NOF**	**CPU**	**ITER**	**NOF**	**CPU**	**ITER**	**NOF**	**CPU**	**ITER**	**NOF**	**CPU**	**ITER**	**NOF**	**CPU**	**ITER**	**NOF**	**CPU**
Trig White& Holst	2	3	10	0.001522	5	12	0.0061492	4	10	0.0035208	9	22	0.00131	4	8	0.00174	71	264	0.00098
Trig White & Holst	2	5	17	0.0015	10	20	0.0009197	9	20	0.0008796	8	41	0.00104	9	18	0.00107	125	479	0.00111
Linear Perturbed	5000	2	5	0.001823	2	5	0.0010683	2	5	0.001228	2	4	0.00283	2	5	0.0018	2	4	0.00146
Linear Perturbed	10000	2	5	0.001842	3	7	0.001856	3	7	0.0036527	2	4	0.00247	3	7	0.00143	3	5	0.00118
Linear Perturbed	50000	3	7	0.004804	3	7	0.0050941	3	7	0.0039809	***	***	***	3	7	0.00352	3	5	0.00389
Zirilli	2	8	25	0.001976	14	34	0.0021043	13	35	0.0019217	6	24	0.00245	9	24	0.00119	142	543	0.00204
Zirilli	2	7	17	0.000609	9	18	0.0016343	13	22	0.0007317	7	26	0.00054	10	19	0.0014	115	428	0.0011
Test	3	5	22	0.003308	***	***	***	20	103	0.0042947	7	49	0.00552	61	263	0.02114	112	452	0.01702
Test	3	31	118	0.010102	56	154	0.008196	25	64	0.0047338	11	59	0.00686	69	157	0.00681	300	###	0.04014
Diagonal 2	4	12	34	0.004709	18	26	0.0039497	18	24	0.0033059	5	16	0.00455	17	18	0.00304	168	638	0.0388
Diagonal 2	10	20	64	0.004567	30	43	0.0083089	29	45	0.0027465	***	***	***	50	51	0.00576	366	###	0.07963
Diagonal 2	20	29	99	0.006961	40	70	0.0042639	36	71	0.0065909	7	21	0.00305	103	104	0.00774	764	###	0.11506
Diagonal 2	50	51	176	0.00981	75	130	0.0096052	58	110	0.0093518	7	26	0.00511	80	136	0.00888	###	###	0.29121
Wayburn Seader 1	2	8	50	0.000473	12	47	0.0008535	9	37	0.0011408	23	115	0.00077	22	92	0.00043	139	516	0.00042
Wayburn Seader 1	2	11	67	0.000519	14	77	0.0004244	13	72	0.0003299	***	***	***	22	147	0.00102	218	815	0.00041
Trecanni	2	7	25	0.000996	11	36	0.0003748	10	34	0.0003666	***	***	***	8	28	0.00074	257	983	0.00042
Trecanni	2	9	32	0.000646	8	31	0.0003893	7	28	0.0005554	5	29	0.00038	11	42	0.00077	262	994	0.00044

DOI: https://doi.org/10.54216/IJNS.260220

Diagonal 1	4	10	21	0.002534	11	22	0.0028187	12	24	0.0035406	5	17	0.00246	14	25	0.00273	149	550	0.02274
Diagonal 1	10	21	45	0.004832	21	44	0.0032977	22	46	0.0041766	7	21	0.0038	22	47	0.00677	***	***	***
ARWHEAD	1000	3	9	0.000445	4	10	0.0007813	4	10	0.0008371	2	8	0.00128	4	10	0.00158	***	***	***
ARWHEAD	10000	3	9	0.000913	4	10	0.0017032	4	10	0.0011028	2	8	0.00139	4	10	0.00126	***	***	***
Price 4	2	25	107	0.00036	23	74	0.0007543	17	58	0.0005169	***	***	***	17	45	0.00033	25	112	0.00057
Price 4	2	52	233	0.000428	30	110	0.0006196	49	197	0.0005078	***	***	***	51	159	0.00039	348	###	0.00068
Shallow	1000	10	26	0.000623	8	21	0.0004291	8	21	0.0006586	21	90	0.00135	12	26	0.0009	476	###	0.00109
Shallow	10000	10	26	0.001201	8	21	0.001201	8	21	0.0011396	21	90	0.00172	12	26	0.00105	505	###	0.00098
Diagonal 3	2	5	18	0.002777	8	18	0.0036098	7	16	0.0037834	***	***	***	9	27	0.00267	99	377	0.01883
Diagonal 3	4	15	38	0.006771	16	35	0.0036414	15	33	0.0047317	10	50	0.00583	17	45	0.00366	238	908	0.04064
Hager	10	11	23	0.000393	12	25	0.0003682	12	25	0.0003647	6	15	0.0004	13	23	0.00045	185	700	0.00044
Hager	50	19	40	0.000545	19	40	0.0004076	19	40	0.0003454	6	18	0.00051	20	42	0.00091	***	***	***
Hager	100	24	51	0.000443	24	50	0.0003718	24	50	0.0008592	***	***	***	26	55	0.00077	***	***	***
Ext Tridiagonal 1	500	19	52	0.001274	16	43	0.0005693	18	41	0.0006094	68	222	0.00082	18	46	0.00068	263	###	0.00061
Ext Tridiagonal 1	1000	20	58	0.001106	16	43	0.0009535	18	41	0.000592	102	315	0.00085	34	69	0.00058	249	###	0.0006
Ext Tridiagonal 1	10000	22	65	0.001725	24	60	0.001543	21	46	0.0019442	222	614	0.00257	39	74	0.00143	299	###	0.0016
Ext Tridiagonal 1	50000	30	118	0.007114	24	60	0.0063323	21	46	0.0069458	91	280	0.00806	55	117	0.00533	240	958	0.0059
GenerlizedTridiag 2	2	11	61	0.007558	14	64	0.0088959	13	62	0.0087664	***	***	***	***	***	***	307	###	0.09005
Freudenstein & Roth	2	8	38	0.004474	***	***	***	11	49	0.0048485	***	***	***	12	62	0.0035	***	***	***

DOI: https://doi.org/10.54216/IJNS.260220

Freudenstein & Roth	2	7	27	0.002847	12	36	0.0043422	9	30	0.0033	***	***	***	7	31	0.00337	***	***	***
Rotated Ellipse 2	2	5	10	0.000487	8	15	0.0006893	4	8	0.0007624	4	14	0.00053	4	8	0.0017	130	490	0.00043
Rotated Ellipse 2	2	1	3	0.001074	1	3	0.0009132	1	3	0.0005117	1	3	0.00085	1	3	0.00058	1	3	0.00094
Diagonal 5	10	2	7	0.002612	4	8	0.0036338	6	9	0.0015323	2	8	0.00504	8	9	0.00166	85	322	0.03085
Diagonal 5	50000	2	7	0.056566	4	8	0.072009	7	10	0.090279	***	***	***	***	***	***	***	***	***
Diagonal 5	100000	2	7	0.12491	4	8	0.12772	8	11	0.17323	***	***	***	10	11	0.18546	***	***	***
Diagonal 4	1000	5	10	0.002653	5	10	0.0031683	4	8	0.0032843	4	8	0.00276	4	8	0.00281	457	###	0.13626
Diagonal 4	10000	5	10	0.007706	5	10	0.019143	4	8	0.0090883	4	8	0.00885	4	8	0.00649	512	###	0.71533
Diagonal 4	100000	5	10	0.057878	5	10	0.066496	4	8	0.047125	4	8	0.06339	4	8	0.04324	373	###	6.0543
Raydan 2	50	114	275	0.014172	61	153	0.0098579	82	184	0.01109	***	***	***	62	115	0.00964	***	***	***
Raydan 2	80	146	354	0.017942	224	500	0.036642	115	278	0.020765	***	***	***	69	154	0.01274	***	***	***
Raydan 1	100	2	5	0.003911	4	7	0.0027369	3	5	0.0019608	1	5	0.00235	5	9	0.00226	6	8	0.0026
Raydan 1	10000	3	7	0.004152	4	7	0.0030333	3	5	0.0029293	1	5	0.00325	5	9	0.00377	6	8	0.00294
Raydan 1	5000	3	7	0.007589	4	7	0.0069466	3	5	0.0062849	***	***	***	5	9	0.00979	6	8	0.012
Six Hump	2	5	17	0.00053	7	20	0.0005575	7	20	0.0007097	***	***	***	6	20	0.00075	***	***	***
Six Hump	2	5	17	0.000347	7	20	0.000754	7	20	0.0007983	***	***	***	6	20	0.00044	***	***	***
Deckkers-Aarts	2	9	92	0.000565	***	***	***	***	***	***	***	***	***	12	101	0.0008	***	***	***
Deckkers-Aarts	2	12	101	0.000681	***	***	***	14	136	0.0005676	***	***	***	14	125	0.00037	***	***	***
Brent	2	1	3	0.000553	1	3	0.0007159	1	3	0.0004954	1	3	0.00061	1	3	0.00058	1	3	0.00043
Brent	2	1	3	0.000573	1	3	0.0005161	1	3	0.0005166	1	3	0.00081	1	3	0.00105	1	3	0.00056

DOI: https://doi.org/10.54216/IJNS.260220

Quadratic penalty 2	50	8	36	0.004728	***	***	***	***	***	***	***	***	***	8	34	0.00697	***	***	***
Quadratic penalty 2	100	7	33	0.006378	***	***	***	***	***	***	9	30	0.00432	9	38	0.00531	***	***	***
Quadratic penalty 1	100	6	15	0.003343	7	16	0.0030547	7	16	0.0021633	***	***	***	6	14	0.00304	***	***	***
Quadratic penalty 1	500	6	15	0.004447	7	16	0.0029442	7	16	0.0036463	6	18	0.00555	7	16	0.0071	***	***	***
Quadratic penalty 1	1000	6	15	0.005943	7	16	0.00446	7	16	0.0042491	6	19	0.00623	7	16	0.0039	***	***	***
Quadratic penalty 1	2000	6	15	0.005967	7	16	0.0048772	7	16	0.0052611	***	***	***	7	16	0.0083	***	***	***
Ext Block-Diagonal	1000	13	44	0.000564	13	35	0.0005811	16	39	0.0005532	***	***	***	30	71	0.00048	226	866	0.0005
Ext Block-Diagonal	50000	13	44	0.004113	15	40	0.0043601	19	45	0.0039358	***	***	***	33	77	0.00333	250	958	0.00317
Ext Block-Diagonal	100000	14	46	0.006791	15	40	0.0071825	19	45	0.0067635	***	***	***	33	77	0.007	254	972	0.00728
DENSCHNF	1000	9	41	0.000822	10	45	0.0008444	9	41	0.0007287	8	42	0.00086	7	37	0.00098	214	829	0.00055
DENSCHNF	10000	9	41	0.001618	10	45	0.0018512	9	41	0.0018762	8	42	0.00309	11	44	0.00165	224	866	0.0018
DENSCHNF	50000	9	41	0.004147	10	45	0.0050502	9	41	0.006857	8	42	0.00901	10	42	0.00601	233	903	0.00594
DENSCHNF	100000	9	41	0.008397	10	45	0.011331	9	41	0.010653	8	42	0.01569	10	42	0.01114	239	926	0.00895
Matyas	2	1	8	0.000369	3	9	0.0004185	7	23	0.0004573	1	8	0.00045	63	128	0.00034	72	277	0.00078
Matyas	2	1	8	0.000387	3	9	0.0009606	7	23	0.0005084	1	8	0.00089	59	120	0.00046	67	257	0.00049
Sum Square	10	5	50	0.001159	8	65	0.0004486	8	65	0.0003408	2	21	0.00119	***	***	***	8	18	0.00113

DOI: https://doi.org/10.54216/IJNS.260220

Sum Square	20	6	72	0.000426	9	91	0.0006673	9	91	0.0003685	2	24	0.00079	***	***	***	25	48	0.00051
Sum Square	50	8	124	0.000646	16	203	0.0006074	6	73	0.000428	2	27	0.0005	***	***	***	12	27	0.00054
Ext DENSCHNB	10000	1	3	0.002211	1	3	0.0022097	1	3	0.0014525	1	3	0.00337	1	3	0.00271	1	3	0.00157
Ext DENSCHNB	50000	1	3	0.003383	1	3	0.0034638	1	3	0.0029875	1	3	0.00495	1	3	0.00277	1	3	0.00254
Ext DENSCHNB	100000	1	3	0.00525	1	3	0.0063888	1	3	0.0059404	1	3	0.00944	1	3	0.00439	1	3	0.00618
Diagonal 7	100	3	7	0.001965	3	7	0.00204	3	7	0.0030945	***	***	***	3	7	0.00154	4	7	0.00327
Diagonal 7	1000	3	7	0.004315	3	7	0.0034086	3	7	0.0039132	***	***	***	3	7	0.00287	4	7	0.00316
Diagonal 7	5000	3	7	0.009988	3	7	0.0082896	3	7	0.0080778	***	***	***	3	7	0.01673	4	7	0.00746
Diagonal 6	1000	2	5	0.002834	3	6	0.0054482	3	6	0.003908	2	8	0.00456	4	8	0.00376	4	6	0.00367
Diagonal 6	10000	2	5	0.007181	3	6	0.010636	3	6	0.014777	2	8	0.00961	4	8	0.01546	4	6	0.00994
Diagonal 6	50000	2	5	0.019538	3	6	0.025714	3	6	0.026285	2	8	0.03552	4	8	0.02857	4	6	0.02339
FLETCHCR	1000	4	24	0.001168	6	29	0.0006886	6	29	0.0007074	***	***	***	5	28	0.00097	5	28	0.00135
FLETCHCR	10000	4	24	0.001137	6	29	0.0013543	6	29	0.0021462	***	***	***	4	26	0.0123	5	28	0.00092
FLETCHCR	50000	4	24	0.003454	6	29	0.0037354	6	29	0.0031499	***	***	***	5	29	0.00355	5	28	0.00339
Extended Penalty	2000	3	9	0.000428	4	10	0.0005531	4	10	0.0005476	***	***	***	5	12	0.00076	***	***	***
Extended Penalty	3000	3	9	0.000485	4	10	0.000824	4	10	0.0008353	***	***	***	***	***	***	***	***	***
Extended Penalty	5000	3	9	0.001305	4	10	0.0008642	4	10	0.0007432	***	***	***	***	***	***	***	***	***
Power	2	6	19	0.000723	12	25	0.0008437	12	25	0.0008599	2	9	0.00065	15	31	0.00075	11	16	0.00035
Power	2	6	19	0.000523	14	29	0.0003443	14	29	0.000605	2	9	0.00036	18	37	0.00037	11	16	0.00035
Extended Hiebert	2	26	157	0.000716	29	155	0.0006532	39	217	0.0003725	***	***	***	40	199	0.00047	384	###	0.00042

DOI: https://doi.org/10.54216/IJNS.260220

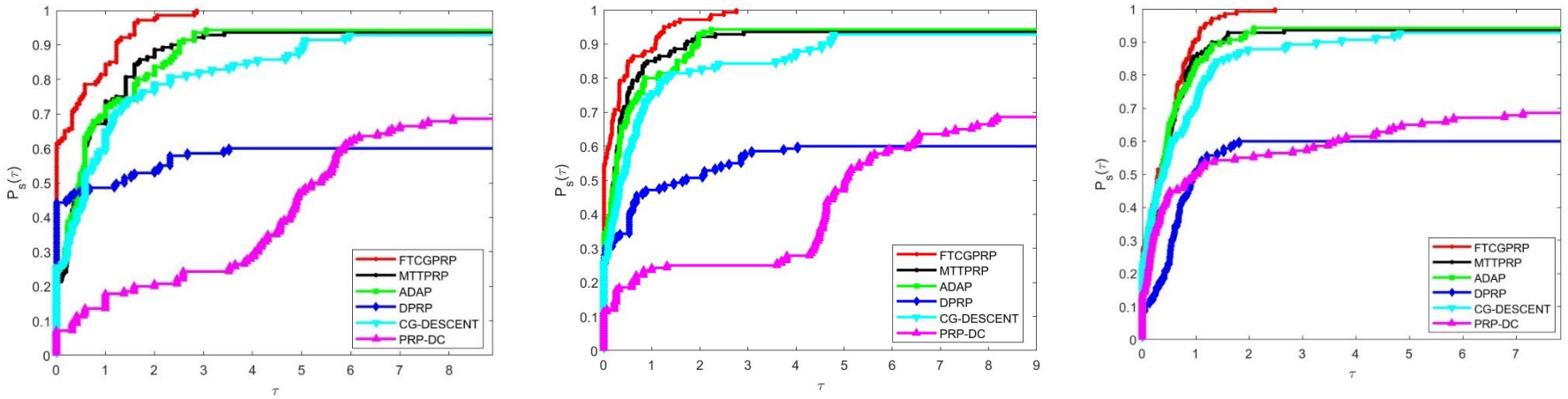
Extended Hiebert	2	29	176	0.001157	36	226	0.0006515	30	157	0.0005866	***	***	***	42	212	0.00055	***	***	***
Zettl	2	9	25	0.000524	12	31	0.0003373	12	28	0.0005023	11	46	0.0004	20	35	0.00071	328	###	0.00039
Zettl	2	18	46	0.000642	16	41	0.0003846	22	52	0.0003627	13	63	0.00069	29	57	0.00104	383	###	0.00041
Zettl	2	18	46	0.00083	16	41	0.0007749	22	52	0.0007935	13	63	0.00151	29	57	0.00062	383	###	0.00052
Diagonal 9	2	9	21	0.00394	14	54	0.0040428	11	34	0.0041378	***	***	***	13	37	0.00757	***	***	***
Diagonal 9	4	49	102	0.015305	***	***	***	***	***	***	7	18	0.00273	120	245	0.01805	***	***	***
Diagonal 8	1000	3	7	0.003169	3	7	0.0033962	3	7	0.0026874	***	***	***	3	7	0.00665	***	***	***
Diagonal 8	10000	3	7	0.022527	3	7	0.014815	3	7	0.015398	***	***	***	3	7	0.01573	***	***	***
Diagonal 8	50000	3	7	0.056113	3	7	0.060824	3	7	0.056212	***	***	***	3	7	0.04625	***	***	***
QUARTC	100	7	25	0.00393	8	28	0.0066379	14	72	0.009521	3	36	0.00561	76	402	0.03766	133	532	0.05549
QUARTC	10000	7	25	0.05593	8	28	0.074144	16	89	0.21173	3	36	0.07231	92	579	1.1589	148	591	1.1764
QUARTC	50000	7	25	0.23106	8	28	0.27576	17	98	0.97275	3	36	0.32982	98	657	6.0986	153	612	5.6921
QUARTC	100000	7	25	0.44943	8	28	0.53006	17	98	1.9121	3	36	0.65335	100	685	12.7225	155	619	11.7179
Quadratic QF2	50	2	6	0.002151	3	7	0.0017993	3	7	0.0028982	***	***	***	4	9	0.00185	3	6	0.00209
Quadratic QF2	1000	1	4	0.002185	3	7	0.0022269	3	7	0.0023962	1	4	0.00128	4	9	0.00203	3	6	0.00268
Quadratic QF2	5000	1	4	0.004725	3	7	0.0046031	3	7	0.0048911	1	4	0.00322	4	9	0.01089	3	6	0.00508
Quadratic QF1	2	4	7	0.00227	3	6	0.0035542	3	6	0.0036471	2	4	0.00188	3	6	0.00249	54	198	0.00798
Quadratic QF1	4	5	21	0.002705	8	26	0.0024643	8	25	0.0024289	***	***	***	9	36	0.00427	13	22	0.00306
El-Attar-Vidyasagar	2	13	74	0.000334	17	106	0.000444	13	73	0.0004025	***	***	***	12	70	0.00074	***	***	***

DOI: https://doi.org/10.54216/IJNS.260220

El-Attar-Vidyasagar	2	12	90	0.000445	13	83	0.0006934	15	103	0.0005307	***	***	***	16	99	0.00081	***	***	***
Booth	2	2	5	0.00074	2	5	0.0006896	2	5	0.0004084	8	25	0.00058	2	5	0.00035	382	###	0.00043
Booth	2	2	5	0.001035	2	5	0.0006293	2	5	0.0005081	23	82	0.00042	2	5	0.00048	352	###	0.00045
BIGGSBI	2	1	3	0.002157	1	3	0.0025687	1	3	0.0027188	1	3	0.00257	1	3	0.00193	1	3	0.00225
BIGGSBI	2	1	3	0.002607	1	3	0.0019027	1	3	0.002567	1	3	0.00258	1	3	0.00186	1	3	0.00194
ENGVAL1	4	15	44	0.001926	23	51	0.0012592	23	51	0.001536	***	***	***	18	42	0.00126	351	###	0.00104
ENGVAL1	10	19	49	0.000467	22	48	0.000337	25	54	0.0004764	***	***	***	23	53	0.00045	***	***	***
ENGVAL1	20	18	48	0.000889	27	58	0.0003642	28	60	0.0007013	***	***	***	24	55	0.00083	***	***	***
ENGVAL1	50	20	50	0.000586	25	54	0.0004923	27	58	0.0003662	***	***	***	23	53	0.0009	***	***	***
DQDRTIC	4	4	9	0.000581	4	9	0.0003997	4	9	0.0007821	2	5	0.00041	4	9	0.00045	4	6	0.00089
DQDRTIC	4	3	7	0.000532	3	7	0.0003871	3	7	0.0007692	2	5	0.00039	3	7	0.00053	4	6	0.0005
QUARTICM	1000	7	25	0.000541	8	28	0.0005735	15	80	0.0006076	3	36	0.00083	79	486	0.00061	140	560	0.00073
QUARTICM	10000	7	25	0.001646	8	28	0.001666	16	89	0.0017351	3	36	0.00226	92	579	0.00167	148	591	0.00157
QUARTICM	50000	7	25	0.005598	8	28	0.0056884	17	98	0.0056053	3	36	0.00813	98	657	0.0054	153	612	0.00534
QUARTICM	100000	7	25	0.01033	8	28	0.012052	17	98	0.01138	3	36	0.01432	100	685	0.01109	155	619	0.01088
HIMMELBH	100	5	11	0.000353	5	11	0.000384	5	11	0.0006445	6	16	0.00044	5	11	0.00085	***	***	***
HIMMELBH	500	5	11	0.000426	5	11	0.0007518	5	11	0.0010509	***	***	***	5	11	0.00046	***	***	***
HIMMELBH	50	5	11	0.001191	5	11	0.0004129	5	11	0.0004061	***	***	***	5	11	0.00079	***	***	***
HIMMELBH	200	5	11	0.00069	5	11	0.0003946	5	11	0.000409	***	***	***	5	11	0.00038	***	***	***
Extended Maratos	50000	16	75	0.002124	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***

DOI: https://doi.org/10.54216/IJNS.260220

Extended Maratos	100000	16	75	0.004258	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***
Extended Maratos	200000	16	75	0.009899	20	95	0.0094063	***	***	***	***	***	***	***	***	***	***	***	***
Extended Maratos	500000	16	75	0.035	20	95	0.042376	***	***	***	***	***	***	25	121	0.03795	***	***	***
DENSCHNA	1000	6	14	0.000563	7	14	0.0009163	6	12	0.0021638	30	101	0.00081	9	17	0.00062	293	###	0.00068
DENSCHNA	10000	6	14	0.001469	7	14	0.0022933	7	14	0.0015117	30	101	0.00258	9	17	0.0013	325	###	0.00143
DENSCHNA	50000	11	35	0.004558	12	33	0.0048206	11	30	0.0052208	25	99	0.00743	16	41	0.00457	462	###	0.0052
DENSCHNA	100000	6	14	0.009121	7	14	0.010775	7	14	0.011294	30	101	0.01472	9	17	0.01181	348	###	0.009
DENSCHNA	200000	6	14	0.020116	7	14	0.020577	7	14	0.023489	30	101	0.03212	7	14	0.02213	346	###	0.01885
DENSCHNA	500000	8	16	0.073339	7	14	0.076923	7	14	0.066273	14	50	0.09799	6	12	0.06135	***	***	***



A **B** **C**

Figure 1. Performance profile based on: number of Iteration (A); Number of function evaluation (B); and CPU time (C)

DOI: https://doi.org/10.54216/IJNS.260220

Figure 1 illustrate the findings, which were calculated by utilizing the performance measure developed by Dolan and More [22]. Regarding number of iteration numbers and function evaluation (see; Figures 1(A) and (b)), FTCGPRP slightly performed better that MTTPRP, ADAP, CG-DESCENT, and outperformed both DPRP and PRP-DC formulas. The poor performance of the DPRP and PRP-DC method can be due to their poor scaling. However, by looking at the left-hand side of Figure 1(C), which plots the performance of the methods based on CPU time, it can be seen that FTCGPRP, MTTPTP, and ADAP methods competed until they reached 70% of the solved problems. At this stage, the performance of the methods was differentiated with the proposed FTCGPRP having advantage over both MTTPTP and ADAP method under the Wolfe line search strategy.

5 . Application of FTCGPRP method to Robotic model

Robotic manipulators play an important part in automation across several industrial and engineering applications, from research and medical procedures to manufacturing and assembly. The planar robotic manipulators, operating within two-dimensional planes are among the fundamental because of their wide range of applications and simplicity. However, one of the major challenges in the robotic manipulators domain is the real-time motion control of a system with three degrees of freedom (DOF).This challenge entails optimizing a time-dependent nonlinear system and most of the optimization algorithms (see; [29,30]),are only tailored for static nonlinear optimization and thus, may not effectively address the complexities of time-varying nonlinear optimization (TVNO) problems [31,32]. The primary difference between TVNO problems and static nonlinear ones lies in the fact that TVNO problems evolve over time, emphasizing the importance of the time derivative for accurate real-time solutions.

Therefore, this section will be investigating the application of proposed method in tackling the problem of real-time motion control for a three-degrees-of-freedom planar robotic manipulator. This problem is first presented with a discrete-time kinematic model at a specified position level, as noted in [31-33]:

$$\Upsilon(\Phi_k) = \Upsilon_k \tag{22}$$

where $\Upsilon(\Phi_k)$ represents function of the kinematicmapping and $\Phi_k \in \mathbb{R}^2$is the joint angle vector parameter with a familiar characteristic that can be formulated as follows:

$$\Upsilon(\Phi) = \begin{bmatrix} \iota_1 \cos(\Phi_1) + \iota_2 \cos(\Phi_1 + \Phi_2) + \iota_3 \cos(\Phi_1 + \Phi_2 + \Phi_3) \\ \iota_1 \sin(\Phi_1) + \iota_2 \sin(\Phi_1 + \Phi_2) + \iota_3 \sin(\Phi_1 + \Phi_2 + \Phi_3) \end{bmatrix}. \tag{23}$$

The function $\Upsilon(\cdot)$ maps the joint parameters to the planar configuration of the robotic system. Accordingly, the chain distances are indicated byι_i (for $i = 1,2,3$). More so, $\Upsilon(\Phi)$defines the end-effector position vector. Defining $t_k \in \mathbb{R}^2$ as the vector representing the desired path at time$t_k \in [0, t_f]$, the objective is to minimize the following nonlinear least squares problem:

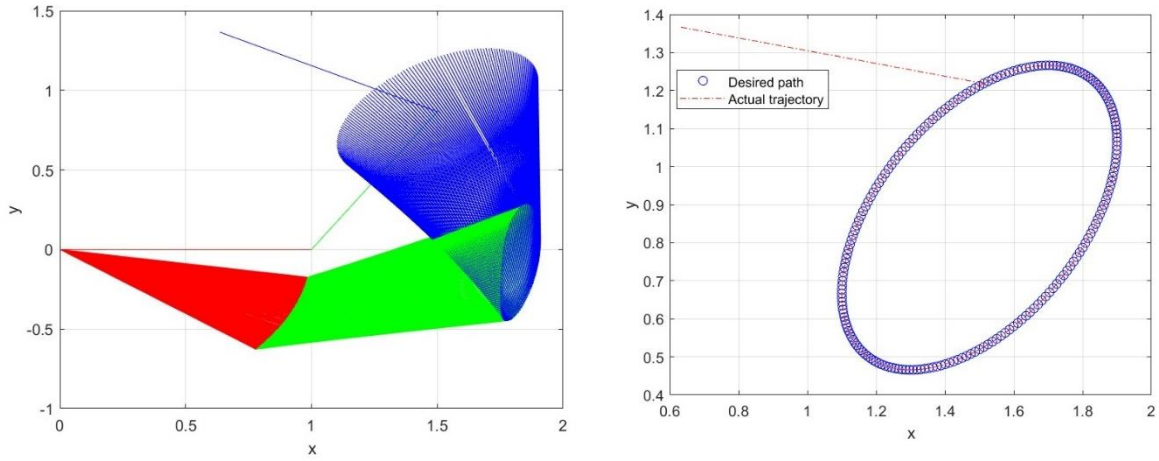
$$\min_{\Phi \in \mathbb{R}^2} \frac{1}{2} \left\| \Upsilon(\Phi) - \Upsilon_{t_k} \right\|^2 \tag{24}$$

with the route of Lissajous curve at time t_k defined as:

$$\Upsilon_{t_k} = \begin{bmatrix} \frac{3}{2} + \frac{2}{5} \sin\left(\frac{\pi t_k}{5}\right) \\ \frac{\sqrt{3}}{2} + \frac{2}{5} \sin\left(\frac{\pi t_k}{5} + \frac{\pi}{5}\right) \end{bmatrix} \tag{25}$$

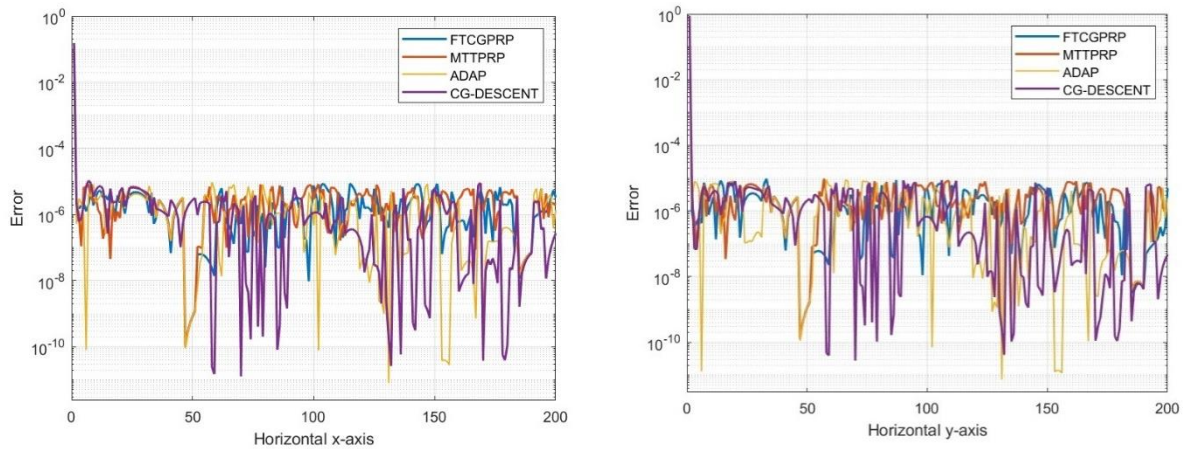
More so, the end effector vector monitors the proper track associated with a Lissajous curve as described in the following figures.

DOI: https://doi.org/10.54216/IJNS.260220



(A) **(B)**

Figure 2. Synthesized robot trajectories of Lissajous curve Υ_{t_k} (A) and end-effector trajectory and desired path of Lissajous curveΥ_{t_k} (B)



(A) **(B)**

Figure 3. Tracking residual error of Lissajous curve Υ_{t_k} on x–axis (A) and tracking residual error of Lissajous curveΥ_{t_k} on y–axis (B)

The experimental report starts by initializing the chain at time $t = 0$, with the end effector δk set to$\omega_0 = [\omega_2, \omega_2, \omega_3] = [0, \pi/3\,, \pi/2]$, and dividing it into 200 equal units. The simulated results for the proposed FTCGPRP method and other existing methods including MTTPRP, ADAP, and CG-DESCENT are shown in Figures 2 and 3. The findings indicate that these methods accurately generate the robot trajectories as depicted in Figure 2, with an error of 10^{-6} shown in Figure 3. By observing the errors from Figure 3, it can be seen that all the methods performed very well at some points with CG-DESCENT algorithm having advantage over all the methods. This result suggests that the methods are competitive for the overall errors analysis.

5. Conclusion

In this study, we utilize the following directions to propose a novel four-term CGM, as indicated in equation (11) given by:

DOI: https://doi.org/10.54216/IJNS.260220

$-\nabla f_k, d_{k-1}, y_{k-1}, s_{k-1}$.

The properties are sent down to the search direction

1- The four-term CG technique is demonstrated.
2- Without utilizing any line search, it fulfils the sufficient descent condition.
3- Convergence analysis with regard to non-linear functions, in general, is produced.
4- The numerical results on both unconstrained optimization and robotic motion problems portray that the recent adjustment surpasses existing approaches such as MTTPRP, ADAP, CG-DESCENT, DPRP, and PRP-DC.
5- Various values of t can be utilized to create a new search direction.

In the future, we wish to enhance the SWP line search by using different sigma values to reduce CPU time and gradient evolutions for both the FTCGHS and FTCGPRP techniques. We also aim to utilize the CGM in deep learning and machine learning as an application with regard to the method. The reader might find further information in the following references [23,26].

AMS Subject Classifications: 65K05; 49M37; 90C3

Availability of data and material

The information is contained inside the publication.

Competing interests

There are no conflicting interests declared by the writers.

Authors' contributions

All of the writers participated fairly and meaningfully in drafting this work. All writers reviewed and accepted the completed manuscript.

Acknowledgments

The Researchers Supporting Project number (RSP2024R317), King Saud University, Riyadh, Saudi Arabia, supports this project. The authors want to state their appreciation to the reviewers and editor for any comments that have helped in improving this manuscript. The authors want to express their appreciation to the reviewers and editor for any comments that have helped improve this manuscript.

References

[1] P. Wolfe, "Convergence conditions for ascent methods," *SIAM Review*, vol. 11, pp. 226–235, 1969.

[2] P. Wolfe, "Convergence conditions for ascent methods. II: Some corrections," *SIAM Review*, vol. 13, pp. 185–188, 1971.

[3] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *J. Res. Natl. Bur. Stand.*, vol. 49, pp. 409–436, 1952.

[4] E. Polak and G. Ribiere, "Note sur la convergence de méthodes de directions conjuguées," *Rev. Française d'Informatique et de Recherche Opérationnelle. Série Rouge*, vol. 3, pp. 35–43, 1969.

[5] Y. Liu and C. Storey, "Efficient generalized conjugate gradient algorithms, part 1: Theory," *J. Optim. Theory Appl.*, vol. 69, pp. 129–137, 1991.

[6] R. Fletcher, "Function minimization by conjugate gradients," *Comput. J.*, vol. 7, pp. 149–154, 1964.

[7] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. New York: Wiley, 2000.

[8] Y.-H. Dai and Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," *SIAM J. Optim.*, vol. 10, pp. 177–182, 1999.

[9] Y.-H. Dai and L.-Z. Liao, "New conjugacy conditions and related nonlinear conjugate gradient methods," *Appl. Math. Optim.*, vol. 43, pp. 87–101, 2001.

[10] W. W. Hager and H. Zhang, "Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent," *ACM Trans. Math. Softw.*, vol. 32, no. 1, pp. 113–137, 2006.

[11] W. W. Hager and H. Zhang, "The limited memory conjugate gradient method," *SIAM J. Optim.*, vol. 23, pp. 2150–2168, 2013.

DOI: https://doi.org/10.54216/IJNS.260220

[12] N. Andrei, "A simple three-term conjugate gradient algorithm for unconstrained optimization," *J. Comput. Appl. Math.*, vol. 241, pp. 19–29, 2013.

[13] G. Zhou, Y. Yang, M. Cao, and S. Qu, "A new spectral three-term conjugate gradient method with random parameter based on modified secant equation and its application to low-carbon supply chain optimization," *J. Math.*, vol. 2022, Art. ID 8939770, 2022.

[14] S. Babaie-Kafaki and R. Ghanbari, "Two modified three-term conjugate gradient methods with sufficient descent property," *Optim. Lett.*, vol. 8, pp. 2285–2297, 2014.

[15] S. Deng and Z. Wan, "A three-term conjugate gradient algorithm for large-scale unconstrained optimization problems," *Appl. Numer. Math.*, vol. 92, pp. 70–81, 2015.

[16] J.-K. Liu, J.-Ł. Xu, and L.-Q. Zhang, "Partially symmetrical derivative-free Liu–Storey projection method for convex constrained equations," *Int. J. Comput. Math.*, vol. 96, pp. 1787–1798, 2019.

[17] J.-K. Liu, Y.-X. Zhao, and X.-L. Wu, "Some three-term conjugate gradient methods with the new direction structure," *Appl. Numer. Math.*, vol. 150, pp. 433–443, 2020.

[18] A. Alhawarat, G. Alhamzi, I. Masmali, and Z. Salleh, "A descent four-term conjugate gradient method with global convergence properties for large-scale unconstrained optimisation problems," *Math. Probl. Eng.*, vol. 2021, Art. ID 6651234, 2021.

[19] A. Alhawarat, Z. Salleh, and I. A. Masmali, "A convex combination between two different search directions of conjugate gradient method and application in image restoration," *Math. Probl. Eng.*, vol. 2021, Art. ID 6697025, 2021.

[20] J. C. Gilbert and J. Nocedal, "Global convergence properties of conjugate gradient methods for optimization," *SIAM J. Optim.*, vol. 2, pp. 21–42, 1992.

[21] N. Andrei, "An unconstrained optimization test functions collection," *Adv. Model. Optim.*, vol. 10, pp. 147–161, 2008.

[22] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Math. Program.*, vol. 91, pp. 201–213, 2002.

[23] Z. Salleh, G. Alhamzi, I. Masmali, and A. Alhawarat, "A modified Liu and Storey conjugate gradient method for large scale unconstrained optimization problems," *Algorithms*, vol. 14, no. 1, Art. ID 18, 2021.

[24] M. Al-Baali, "Descent property and global convergence of the Fletcher–Reeves method with inexact line search," *IMA J. Numer. Anal.*, vol. 5, pp. 121–124, 1985.

[25] G. Zoutendijk, "Nonlinear programming, computational methods," in *Integer and Nonlinear Programming*, J. Abadie, Ed. Amsterdam: North-Holland, 1970, pp. 37–86.

[26] S. Sra, S. Nowozin, and S. J. Wright, Eds., *Optimization for Machine Learning*. Cambridge, MA: MIT Press, 2012.

[27] N. Andrei, "A modified Polak–Ribière–Polyak conjugate gradient algorithm for unconstrained optimization," *Optim.*, vol. 60, no. 12, pp. 1457–1471, 2011.

[28] H. Mohammad, I. M. Sulaiman, and M. Mamat, "Two diagonal conjugate gradient-like methods for unconstrained optimization," *J. Ind. Manag. Optim.*, vol. 20, no. 1, pp. 170–187, 2024.

[29] E. G. Birgin and J. M. Martinez, "A spectral conjugate gradient method for unconstrained optimization," *Appl. Math. Optim.*, vol. 43, pp. 117–128, 2001.

[30] Y. Narushima and H. Yabe, "Conjugate gradient methods based on secant conditions that generate descent search directions for unconstrained optimization," *J. Comput. Appl. Math.*, vol. 236, pp. 4303–4317, 2012.

[31] Y. Zhang et al., "General four-step discrete-time zeroing and derivative dynamics applied to time-varying nonlinear optimization," *J. Comput. Appl. Math.*, vol. 347, pp. 314–329, 2019.

[32] M. M. Yahaya et al., "New Generalized Quasi-Newton Algorithm," *IEEE Access*, vol. 10, pp. 10816–10826, 2022.

[33] R. B. Yunus et al., "A Modified Structured Spectral HS Method," *Mathematics*, vol. 11, no. 14, Art. ID 3215, 2023.

DOI: https://doi.org/10.54216/IJNS.260220