Malaysian Journal of Mathematical Sciences 20(3): 865–882 (2026)
https://doi.org/10.47836/mjms.20.3.01

Malaysian Journal of Mathematical Sciences

Journal homepage: https://mjms.upm.edu.my

A DL-like Hybrid of DY Conjugate Gradient Algorithm for Unconstrained Optimization Problems with Application in Mode Function

Ayinde, S. A.*[1], Mehamdia, A. E.[2], Adelodun, J. F.[3], and Oyeniyi, K. B.[4]

[1]*Department of Mathematics, University of Ibadan, Ibadan, Nigeria*
[2]*Department of Mathematics, University M'Hamed, Bougara of Boumerdes, Algeria*
[3]*Department of Basic Sciences , Babcock University, Ilishan-Remo, Ogun State, Nigeria*
[4]*Department of Mechanical Engineering, Adeseun Ogundoyin Polytechnic, Eruwa, Oyo State, Nigeria.*

E-mail: ayindes@babcock.edu.ng
**Corresponding author*

Received: 16 July 2025
Accepted: 19 December 2025

Abstract

Conjugate gradient methods have come a long way in solving system of equations and large scaled optimization problems. In this paper, a hybrid conjugate gradient method which incorporates the DY update parameter and herein constructed conjugate gradient solver that uses Dai-Liao conjugacy conditions is proposed for nonlinear unconstrained optimization problems. It is shown that this proposed method possesses descent search direction at every iteration and thereby converges globally under standard Wolfe line search conditions. Proofs of its robustness and efficiency were demonstrated through numerical tests and application to mode function when compared with other known optimization conjugate gradient algorithms.

Keywords: global convergence; unconstrained optimization; standard Wolfe conditions; descent direction; step length.

1 Introduction

Optimization problems are decision making problems and are found in diverse fields, especially in theoretical and applied fields like engineering, economics, social sciences etc. Most of these optimization problems are convertible to unconstrained optimization problems. The Majority of known optimization algorithms require a lot of memory to store and calculate matrices involved. These optimization algorithms fail to solve large scale optimization problems as a result of this requirement. On the other hand conjugate gradient methods generate search direction without storing matrices, hence do not require large memory thereby provide a very easy, robust and efficient scheme for the solution of large-scale unconstrained optimization problems. This work seeks an efficient and robust scheme to solve nonlinear unconstrained optimization problems:

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

provided, f is smooth.

The following iteration,

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

is used in the conjugate gradient algorithm scheme required, where $\alpha_k > 0$ is a step-length found from a line search scheme with the direction d_k given by

$$d_k = \begin{cases} -g_0, & if \;\; k = 0; \\ -g_k + \beta_k d_{k-1}, & otherwise. \end{cases} \tag{3}$$

$\beta_k \in \mathbb{R}$ in (3) is a formula called CG coefficient or update parameter while g_k is the derivative of f at x_k. It should be noted that (2) and (3) with CG coefficient is called CG method. Different CG methods arise as a result of carefully constructed β_k.

Several authors and researchers have made numerous contributions in this area of research. These contributions spanning many decades saw to the emergence of the well known classical conjugate gradient methods. These classical methods were formed from the following CG coefficients.

$$\begin{cases} \beta_k^{HS} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, \\ \beta_k^{FR} = \frac{||g_k||^2}{||g_{k-1}||^2}, \\ \beta_k^{PRP} = \frac{g_k^T y_{k-1}}{||g_k||^2}, \\ \beta_k^{LS} = \frac{-g_k^T y_{k-1}}{d_{k-1}^T g_{k-1}}, \\ \beta_k^{DY} = \frac{||g_k||^2}{d_{k-1}^T y_{k-1}}. \end{cases} \tag{4}$$

Convergence of any conjugate gradient method depends on the following conditions known as Wolfe line search conditions

$$f(x_k) - f(x_k + \alpha_k d_k) \geq -\delta \alpha_k g_k^T d_k, \tag{5}$$

and

$$g(x_k + \alpha_k d_k)^T d_k \geq \sigma g_k^T d_k, \tag{6}$$

or

$$|g(x_k + \alpha_k d_k)^T d_k| \leq -\sigma g_k^T d_k, \tag{7}$$

for $0 < \delta \leq \sigma < 1$.

Since the introduction of classical CG methods, a number of drawbacks were noticed to be associated with these methods. For instance: LS method [22] is known to have better computation but has poor convergence rate. FR method [12] which happens to have better convergence property fails to converge for certain values of σ. In the case of PRP method [26] which was believed to have better computation, cycles without converging under exact line search. HS method [16] which is believed to have good computation like PRP method has issues with search direction. As a result of deficiency associated with global convergence of these methods, Dai and Yuan [8] introduced a method called DY with an update parameter that satisfies the following requirement:

$$-||g_k||^2 + \beta_k g_k^T d_{k-1} < 0,$$

with the assumption that $\beta_k > 0$. A parameter τ_k was defined as,

$$\tau_k = \frac{||g_k||^2}{\beta_k}.$$

So also by letting $\tau_k = d_{k-1}^T y_{k-1}$, the following update parameter was proposed.

$$\beta_k = \frac{||g_k||^2}{d_{k-1}^T y_{k-1}}.$$

DY method converges globally under weak Wolfe line search conditions (5) and (6), but has the same deficiency like FR method under conditions (5) and (7) known as strong Wolfe line search conditions. In order to reduce or remove these deficiencies, a new wave of CG methods is being considered, namely hybrid CG methods. A number of authors have proposed hybrid methods to solve (1). For instance, Hafaidia et al. [13] introduced a hybrid method for (1). In a different twist, a monotone operator equation was solved by a hybrid method proposed by Kumam et al. [19]. Ayinde et al. in [29] constructed a compound update parameter from which a number of hybrid methods can be formed.

The evolution of hybrid CG algorithms started in 1990, with the introduction of the update parameter

$$\beta_k = min\{\beta_k^{FR}, \beta_k^{PRP}\},$$

by Touati-Ahmed and Storey [31] with the aim of eliminating the deficiency associated with FR method and to combine the good numerical performance of the PRP method and the nice global convergence property of the FR method. In 2001, Dai and Yuan [8] followed suit in like manner by proposing two hybrid CG methods that use the following update parameters:

$$\beta_k = max\big\{-c\beta_k^{DY}, min\{\beta_k^{HS},\ \beta_k^{DY}\}\big\},$$

and

$$\beta_k = max\big\{0, min\{\beta_k^{HS},\ \beta_k^{DY}\}\big\}.$$

The two methods combined the global convergence of DY CG method with the computation strength of HS CG method because it is known that HS CG method behaves like PRP CG method in practical computation. Moreover,the same denomination that formulas β_k^{HS} and β_k^{DY} have coupled with other reasons like restart strategy and search direction issue also influenced their choice of these methods. The aforementioned hybrid CG methods belong to a class that is based on projection concept. Another promising category is the class that is based on linear combination of known CG methods. Some of the methods in this class include: [3] where DY method was incorporated by using DL approach, [17] with the introduction of a hybrid method with a search direction that is a linear combination of LS and conjugate descent (CD) update parameters and [33] that presented a non-negative CG method that modified PRP to give

$$\beta_k^{AMZR} = \frac{g_k^T(\frac{||g_{k-1}||}{||g_k||}g_k - g_{k-1})}{\frac{||g_{k-1}||}{||g_k||}(g_{k-1}^T g_{k-1})},$$

Another prominent method among the class of hybrid CG methods that is based on linear combination is the one proposed by Dai and Liao [7]. They ensured that the conjugacy condition,

$$d_k^T y_{k-1} = -t g_k^T s_{k-1}, \tag{8}$$

holds, if the line search is inexact and came up with a hybrid CG methods that use

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

and

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

where $t \in [0, \infty)$. β_k^{DL+} was proposed to ensure global convergence of their method.
Further to this, Cheng et al. [6] added the second term of DL method to

$$\beta_k^{MLS} = \frac{g_k^T y_{k-1}^*}{-d_{k-1}^T g_{k-1}},$$

to propose a method that satisfied sufficient descent condition; Hager and Zhang [14] made a special modification to DL method by using a particular t given by

$$t = 2\frac{||y_{k-1}||^2}{s_{k-1}^T y_{k-1}},$$

In similar manner, Babaie-Kafaki and Ghanbari [5] made their modification to DL method by constructing two different values for parameter t namely:

$$t_k^* = \frac{s_k^T y_k}{||s_k||^2} + \frac{||y_k||}{||s_k||},$$

and

$$t_k = \frac{||y_k||}{||s_k||}.$$

Yabe and Takano [32] made further development to DL method with different modifications of the secant equations. Another efficient DL-type methods was developed and considered in [23] where different technique was used. Li et al. [21] introduced new conjugacy for the solution of (1).

Following the performance of Dai-Liao method (DL method) and the global convergence property of DY method, we construct a hybrid CG method that incorporates DY update parameter adapted from Dai-Liao conjugacy condition thereby enhancing the global convergence and all other nice properties inherited from DY and DL CG methods. The update parameter for our method is constructed in section 2 with the inclusion of algorithm for the method. Section 3 discusses both the descent and convergence properties of the new method. We showcase the numerical strength of our method by testing it with other known methods in section 4. Section 5 compares our method with other methods in the application to mode function. Conclusion of this work is given in section 6.

2 New Hybrid CG Method and its Algorithm

2.1 Proposed update parameter.

The proposed method starts with the search direction

$$d_k = g_k + \beta_k d_{k-1}. \tag{9}$$

Pre-multiply (9) by y_{k-1}

$$d_k^T y_{k-1} = g_k^T y_{k-1} + \beta_k d_{k-1}^T y_{k-1}. \tag{10}$$

(10),

$$\beta_k = \frac{d_k^T y_{k-1} + g_k^T (g_k - g_{k-1})}{d_{k-1}^T y_{k-1}}, \tag{11}$$

or

$$\beta_k = \frac{d_k^T y_{k-1} + ||g_k||^2 - g_k^T g_{k-1}}{d_{k-1}^T y_{k-1}}. \tag{12}$$

Using (8), Equation (12) becomes

$$\beta_k = \frac{-t g_k^T s_{k-1} + ||g_k||^2 - g_k^T g_{k-1}}{d_{k-1}^T y_{k-1}}, \tag{13}$$

that is

$$\beta_k = \beta_k^{DY} - \frac{t g_k^T s_{k-1} + g_k^T g_{k-1}}{d_{k-1}^T y_{k-1}}. \tag{14}$$

The following conjugate coefficient is proposed

$$\beta_k^{AS} = \beta_k^{DY} - \frac{t(g_k^T s_{k-1} + g_k^T g_{k-1})}{d_{k-1}^T y_{k-1}}. \tag{15}$$

We achieve the best result when t multiplies the numerator of the second term in equation (15). Procedure (2) - (3) with β_k^{AS} is called AS CG method.

A number of authors find reasons to focus on the values of t because different values of t result in different computation experience. Hagar and Zhang [15] computed values of t for arbitrary k-th iteration. To some authors like Babaie-Kafaki [4], constant setting of t by trial and error style gives encouraging computation results. In this work, we use the same approach taken in [7]. $t = 0.1$ gives the best result for the computation experiment performed here.

The following are typical for update parameter.

$$s_{k-1} = x_k - x_{k-1}, \tag{16}$$

and

$$s_{k-1} = \alpha_{k-1} d_{k-1}. \tag{17}$$

The following relation also suffices for a constant $R > 0$. See [30].

$$R||s_{k-1}||^2 \leq \alpha_{k-1} d_{k-1}^T y_{k-1} \geq L||s_{k-1}||^2. \tag{18}$$

2.2 Algorithm for the method

Step 1: With the initial point $x_0 \in \mathbb{R}^n$, we set $k = 0$ and $d_0 = -g_0$. If $||g_k|| \leq \epsilon$ where $\epsilon > 0$ stop.
Step 2: Find $\alpha_k > 0$ by conditions (5) and (7).
Step 3: Let $x_{k+1} = x_k + \alpha_k d_k$, generate g_k, if $||g_k|| \leq \epsilon$, then stop.
Step 4: Compute β_k^{AS} from (15), generate d_k by (3).
Step 5: Put $k = k + 1$ and go to step 2.

3 Descent and Convergence Properties

The following assumptions are required to establish the descent and convergence properties of the optimization algorithm proposed.

3.1 Assumptions

1. The set

$$U = \{x | f(x) \leq f(y)\}, \tag{19}$$

with $y \in \mathbb{R}^n$ is bounded, that is, we can find a constant D where

$$||x|| \leq D. \tag{20}$$

2. Given a neighborhood W of U, f is continuous and differentiable in W and its gradient $g(x)$ with Lipschitz constant L satisfies Lipschitz continuity

$$||g(x) - g(y)|| \leq L||x - y||, \tag{21}$$

for any $x, y \in W$. In addition,

$$||g(x)|| \leq c_1, \tag{22}$$

for any $x \in W$.

Theorem 3.1. *Let assumption 3.1 hold. Suppose that α_k is computed by (5) and (7) for the CG method AS. Then, d_k satisfies*

$$g_k^T d_k \leq -c||g_k||^2, \tag{23}$$

for each $k \geq 0$ and constant c.

Proof. For $k = 0$, from step 1 of Algorithm 2.2, we have

$$d_0^T g_0 = -||g_0||^2, \tag{24}$$

the result follows. Also, pre multiply (9) by g_k and replacing β_k with β_k^{AS},

$$d_k^T g_k = -||g_k||^2 + \beta_k^{AS} g_k^T d_{k-1}. \tag{25}$$

Let $\beta_k^{AS} = 0$, then

$$d_k^T g_k = -||g_k||^2 + \beta_k^{AS} g_k^T d_{k-1} = -||g_k||^2 < 0. \tag{26}$$

Furthermore, we always assume that $\beta_k^{AS} \neq 0$. Therefore,

$$d_k^T g_k = -||g_k||^2 + (\frac{||g_k||^2 - t(g_k^T s_{k-1} + g_{k-1}^T g_k)}{d_{k-1}^T y_{k-1}}) g_k^T d_{k-1}. \tag{27}$$

From (7),

$$|g_k^T d_{k-1}| \leq -\sigma g_{k-1}^T d_{k-1}, \quad \sigma \in (0, 1]. \tag{28}$$

Since $y_{k-1} = g_k - g_{k-1}$, the denominator in (27) becomes

$$d_{k-1}^T y_{k-1} = g_k^T d_{k-1} - g_{k-1}^T d_{k-1}. \tag{29}$$

Whereby,

$$d_{k-1}^T y_{k-1} \leq |g_k^T d_{k-1}| - g_{k-1}^T d_{k-1}. \tag{30}$$

By (28),

$$d_{k-1}^T y_{k-1} \leq -\sigma g_{k-1}^T d_{k-1} - g_{k-1}^T d_{k-1}. \tag{31}$$

That is,

$$d_{k-1}^T y_{k-1} \leq -(1 + \sigma) g_{k-1}^T d_{k-1}. \tag{32}$$

Hence,

$$d_k^T g_k \leq -||g_k||^2 + (\frac{||g_k||^2 - t(g_k^T s_{k-1} + g_{k-1}^T g_k)}{-(1 + \sigma) g_{k-1}^T d_{k-1}}) - \sigma g_{k-1}^T d_{k-1}. \tag{33}$$

$$d_k^T g_k \leq -||g_k||^2 + (\frac{||g_k||^2 - t(g_k^T s_{k-1} + g_{k-1}^T g_k)}{(1 + \sigma)}) \sigma. \tag{34}$$

Since $t > 0$ is the requirement, then

$$d_k^T g_k \leq -||g_k||^2 + (\frac{||g_k||^2}{(1 + \sigma)}) \sigma. \tag{35}$$

Hence,

$$d_k^T g_k \leq -||g_k||^2 + \frac{\sigma}{1+\sigma}||g_k||^2. \tag{36}$$

Therefore,

$$d_k^T g_k \leq -(\frac{1}{1+\sigma})||g_k||^2. \tag{37}$$

This is the required result. □

We next consider results on global convergence for the new method.

Lemma 3.1. *Given Equation (2) with descent direction d_k, let Assumption 3.1 be satisfied where α_k also satisfies the Wolfe conditions in (5) and (7), then*

$$\sum_{k=0}^{\infty} \frac{||g_k||^4}{||d_k||^2} < \infty. \tag{38}$$

Proof. The proof of Lemma 3.3 is in [34]. □

Theorem 3.2. *Let Assumption 3.1 hold. Suppose that α_k is computed by (5) and (7) for the CG method AS. Then,*

$$\lim_{k\to\infty} \inf ||g_k|| = 0. \tag{39}$$

Proof. The result is shown by contradiction. This implies that

$$||g_k|| \geq c_1 \ \ \forall\ k. \tag{40}$$

See [30].

$$\beta_k^{AS} = \frac{||g_k||^2 - tg_k^T s_{k-1} - tg_k^T g_{k-1}}{d_{k-1}^T y_{k-1}}. \tag{41}$$

$$\beta_k^{AS} \leq \frac{||g_k||^2 + tg_k^T s_{k-1} - tg_k^T g_{k-1}}{d_{k-1}^T y_{k-1}}. \tag{42}$$

By (18), we have

$$\beta_k^{AS} \leq \frac{||g_k||^2 + t||g_k||||s_{k-1}|| - t||g_k||||g_{k-1}||}{\alpha_{k-1}||d_{k-1}||^2}. \tag{43}$$

Then,

$$\beta_k^{AS} \leq \frac{||g_k||(||g_k - tg_{k-1}||) + t||g_k||||s_{k-1}||}{\alpha_{k-1}||d_{k-1}||^2}. \tag{44}$$

From Equations (16) and (21), we have

$$||g_k - g_{k-1}|| \leq L||s_{k-1}||,$$

which implies

$$||g_k - tg_{k-1}|| \leq \frac{L}{t}||s_{k-1}||, \tag{45}$$

with the value of t as stated above. Hence, (44) becomes

$$\beta_k^{AS} \leq \frac{||g_k||(\frac{L}{t}||s_{k-1}||) + t||g_k||||s_{k-1}||}{\alpha_{k-1}||d_{k-1}||^2}. \tag{46}$$

By (17),

$$\beta_k^{AS} \leq \frac{||g_k||(\frac{L}{t} + \alpha_{k-1}t)||d_{k-1}||}{\alpha_{k-1}||d_{k-1}||^2}, \tag{47}$$

$$= \frac{||g_k||(\frac{L}{t} + \alpha_{k-1}t)}{\alpha_{k-1}||d_{k-1}||}. \tag{48}$$

Hence,

$$||d_k|| \leq ||g_k|| + \frac{||g_k||(\frac{L}{t} + \alpha_{k-1}t)}{\alpha_{k-1}||d_{k-1}||}||d_{k-1}||. \tag{49}$$

That is,

$$||d_k|| \leq (1 + \frac{(\frac{L}{t} + \alpha_{k-1}t)}{\alpha_{k-1}})||g_k||, \tag{50}$$

$$||d_k||^2 \leq (\frac{\alpha_{k-1}^2 + 2\alpha_{k-1}(\frac{L}{t} + \alpha_{k-1}t) + (\frac{L}{t} + \alpha_{k-1}t)^2}{\alpha_{k-1}^2})||g_k||^2, \tag{51}$$

$$\frac{||d_k||^2}{||g_k||^4} \leq \frac{(\frac{\alpha_{k-1}^2 + 2\alpha_{k-1}(\frac{L}{t} + \alpha_{k-1}t) + (\frac{L}{t} + \alpha_{k-1}t)^2}{\alpha_{k-1}^2})c_1^2}{c_1^4}, \tag{52}$$

$$= \frac{\alpha_{k-1}^2 + 2\alpha_{k-1}(\frac{L}{t} + \alpha_{k-1}t) + (\frac{L}{t} + \alpha_{k-1}t)^2}{c_1^2\alpha_{k-1}^2}. \tag{53}$$

Hence,

$$\frac{||g_k||^4}{||d_k||^2} \geq \frac{c_1^2\alpha_{k-1}^2}{\alpha_{k-1}^2 + 2\alpha_{k-1}(\frac{L}{t} + \alpha_{k-1}t) + (\frac{L}{t} + \alpha_{k-1}t)^2}. \tag{54}$$

(54) implies

$$\sum_{k=0}^{\infty} \frac{||g_k||^4}{||d_k||^2} \geq \frac{c_1^2\alpha_{k-1}^2}{(\alpha_{k-1}^2 + 2\alpha_{k-1}(\frac{L}{t} + \alpha_{k-1}t) + (\frac{L}{t} + \alpha_{k-1}t)^2} = \infty. \tag{55}$$

This is a contradiction as a result of $||g_k|| \geq c_1$. Hence, (39) is satisfied and the result follows. See Theorem 5.8 [30] □

4 Numerical Test and Analysis

In this section, we consider both the numerical results of our method against other methods in the literature and the explanation of the numerical tests conducted.

4.1 Numerical results

This subsection examines the numerical experiment to test the proposed method with DY, FW [11], LS and DL methods. Test functions are taken from Andrei [1]. All algorithms are coded using MATLAB 2015a with compiler settings on windows 10 operating system and RAM 3GB.

Numerical results are determined under the algorithm code using the settings $\delta = 0.0001$ and $\sigma = 0.9$ for all the methods. The iteration terminates if $\|g_k\| \leq 10^{-6}$ or the number of iterations exceeds 4000. The following abbreviations are used on the Table 1 and Table 2: EU: Extended Hilmmelblau; DL4: Diagonal 4; RN1: Raydan 1; RN2: Raydan 2; DL1: Diagonal 1; DC: DQORTIC; RSE: RMODF SINE; EDB: Extended Booth; EBK: Extended Rosenbrock and ANB: ARGLINB. Moreover, ITN, GN, CPT and FV represent number of iterations, gradient norm, CPU time and Function evaluation respectively in the tables.

Numerical Tests are conducted based on the four metrics namely CPU time, function evaluation, number of iterations and gradient norm to establish robustness, effectiveness and strength of AS method with other known methods FW , DY, DL and LS. Table 1 and Table 2 represent the results generated from the experiments. On these tables, Problems, Dimensions and failure of a method to solve a particular problem are represented by $prob.$, $Dim.$ and F respectively. This work considered dimensions 2, 100, 1000, 10000 and 100000 for the experiments.
Dolan and More [9] tools were used to plot the graphs representing the four metrics used. The tools are stated as follow: Provided set S of p_s methods are to be tested depending on the given set P of n_s test functions. The following performance ratio

$$r_{p,s} = \frac{r_{p,s}}{min\{r_{p,s} : s \in S \; and \; p \in P\}} \tag{56}$$

gives the comparison between different methods where $r_{p,s}$ can either be any one of the four metrics to compare methods S for each problem P. Hence, the following probability function

$$\phi_s(\tau) = \frac{1}{n_s}|p \in P : logr_{p,s}| \leq \tau \tag{57}$$

is the overall evaluation of the performance of the methods where $\tau \geq 0$. The probability $\phi_s(\tau)$ is that $r_{p,s}$ is contained in the factor $\tau \geq 1$ in connection to the method s. Therefore, $\phi_s(\tau)$ is the probability that a particular solver will perform better than other solvers for the value $\tau = 1$. When $r_i = r_{p,s}$ for some value r_i, it implies the failure of a selected solver to solve a problem.

4.2 Numerical analysis

Graphical representations of the results recorded in Table 1 and Table 2 are presented in Figure 1-Figure 4 wherein the numerical strength of the five solvers were compared. This comparison was based on the four metrics highlighted above. In Figure 1 where the number of iterations was represented, AS method led other methods in term of performance. Figure 2 showcased function evaluation and AS method led other methods as well in term of performance. In Figure 3, the tools compared these methods in term of CPU time, AS method performed better than other methods which implies that it solves problems faster than others. Lastly, in the area of gradient norm given in Figure 4, our method competes favorably well with other methods compared.

Table 1: Numerical result of number of iterations and gradient norm.

s/n	Prob.	Dim.	DY ITN/GN	FW ITN/GN	AS ITN/GN	LS ITN/GN	DL ITN/GN
1	DL4	2	21/8.78E-07	83/9.96E-07	16/1.98E-08	25/8.07E-07	35/1.96E-07
2		100	24/7.01E-07	96/8.97E-07	16/1.40E-07	29/2.67E-07	21/8.42E-08
3		1000	25/7.46E-07	104/8.7E-07	16/4.43E-07	29/8.46E-07	32/2.80E-07
4		10000	27/4.60E-07	112/8.4E-07	18/1.59E-08	31/5.79E-07	26/1.36E-07
5		100000	28/8.41E-07	120/8.2E-07	18/5.04E-08	33/3.96E-07	55/9.06E-07

6	EU	2	123/8.44E-07	45/7.34E-07	61/3.32E-07	54/7.88E-07	F/F
7		100	133/9.83E-07	49/8.51E-07	61/8.45E-07	58/3.72E-07	F/F
8		1000	142/4.44E-07	52/3.95E-07	61/9.69E-07	60/3.05E-07	F/F
9		10000	145/6.67E-07	55/1.83E-07	62/3.26E-07	60/9.69E-07	50/1.98E-07
10		100000	155/8.51E-07	57/5.64E-07	63/3.24E-07	64/8.03E-07	70/F
11	RN1	2	6/0.00E+00	4000/0.139E	31/1.39E-01	4000/0.149E	19/7.93E-07
12		100	1/4.74E-120	1/4.74E-120	1/4.74E-120	1/4.74E-120	1/4.74E-120
13		1000	1/0.00E+00	1/0.00E+00	1/0.00E+00	1/0.00E+00	1/0.00E+00
14		10000	1/0.00E+00	1/0.00E+00	1/0.00E+00	1/0.00E+00	1/0.00E+00
15		100000	1/0.00E+00	1/0.00E+00	1/0.00E+00	1/0.00E+00	1/0.00E+00
16	RN2	2	24/6.52E-07	6/8.51E-08	13/8.44E-07	9/4.65E-08	6/5.27E-07
17		100	27/8.80E-07	6/6.02E-07	14/9.26E-07	9/3.29E-07	7/4.16E-10
18		1000	29/8.35E-07	7/6.47E-09	15/4.46E-07	10/1.92E-08	7/1.31E-09
19		10000	31/7.93E-07	7/2.04E-08	16/1.79E-07	10/6.06E-08	7/4.16E-09
20		100000	33/7.53E-07	7/6.47E-08	16/5.66E-07	10/1.92E-07	11/7.51E-08
21	DL1	2	29/9.58E-07	5/1.16E-10	5/1.16E-10	21/4.78E-07	7/4.75E-09
22		100	29/9.58E-07	5/1.16E-10	5/1.16E-10	21/4.78E-07	7/4.75E-09
23		1000	29/9.58E-07	5/1.16E-10	5/1.16E-10	21/4.78E-07	7/4.75E-09
24		10000	29/9.58E-07	5/1.16E-10	5/1.16E-10	21/4.78E-07	7/4.75E-09
25		100000	29/9.58E-07	5/1.16E-10	5/1.16E-10	21/4.78E-07	123/9.3E-07
26	DC	2	50/6.65E-07	27/6.47E-07	24/6.76E-07	11/2.40E+08	45/7.59E-07
27		100	143/8.15E-07	64/8.80E-07	124/6.6E-07	128/7.17E-07	135/8.1E-07
28		1000	213/9.96E-07	79/9.20E-07	110/5.9E-07	85/8.39E-07	141/9.1E-07
29		10000	227/6.72E-07	87/6.59E-07	86/8.70E-07	133/6.17E-07	107/9.9E-07
30		100000	114/8.29E-07	93/8.39E-07	73/9.25E-07	149/6.7E-07	118/6.9E-07
31	RSE	2	20/5.28E-07	47/8.62E-07	18/9.33E-07	59/8.09E-07	7/9.08E-10
32		100	22/8.70E-07	54/8.14E-07	21/6.23E-07	67/9.60E-07	7/6.42E-09
33		1000	24/6.41E-07	58/8.14E-07	22/8.93E-07	72/9.95E-07	7/2.03E-08
34		10000	26/4.73E-07	62/8.15E-07	24/5.84E-07	78/8.24E-07	7/6.42E-08
35		100000	27/7.97E-07	66/8.15E-07	25/8.41E-07	83/8.54E-07	141/9.6E-07
36	EDB	2	27/8.62E-07	27/8.62E-07	22/8.28E-07	28/4.85E-07	28/4.73E-07
37		100	30/9.74E-07	30/9.74E-07	23/3.86E-07	29/9.45E-07	F/F
38		1000	32/6.55E-07	32/6.55E-07	25/6.90E-07	35/3.95E-07	F/F
39		10000	34/9.83E-07	34/9.83E-07	27/5.80E-07	36/4.29E-07	32/9.81E-07
40		100000	37/9.09E-07	36/9.09E-07	30/9.59E-07	36/8.63E-07	23/7.45E-07
41	EBK	2	596/F	272/5.4E-07	192/7.8E-07	1327/9.5E-07	F/F
42		100	633/F	132/6.5E-07	103/4.0E-07	929/7.63E+09	F/F
43		1000	F/F	93/6.53E-07	115/3.5E-07	231/2.68E+10	F/F
44		10000	445/1.18E+10	347/6.8E-07	168/4.56E-07	4000/1.31E+06	F/F
45		100000	413/2.50E+08	394/2.1E-07	132/4.74E-07	866/3.10E+09	11/F
46	ANB	2	2/2.98E-15	2/2.98E-15	2/6.91E-07	53/8.04E-07	2/2.98E-15
47		100	1/1.18E+11	1/1.18E+11	1/1.18E+11	1/1.18E+11	1/1.18E+11
48		1000	1/3.30E+17	1/3.30E+17	1/3.30E+17	1/3.30E+17	1/3.30E+17
49		10000	1/9.06E+23	1/9.06E+23	1/9.06E+23	1/9.06E+23	1/9.06E+23
50		100000	1/2.45E+30	1/2.45E+30	1/2.45E+30	1/2.45E+30	1/2.45E+30

Table 2: Numerical result of CPU time and function evaluation.

s/n	Prob.	Dim.	DY CPT/FV	FW CPT/FV	AS CPT/FV	LS CPT/FV	DL CPT/FV
1	DL4	2	0.189/124	0.817/673	0.131/152	0.292/723	0.595E/847
2		100	0.212/139	0.903/777	0.142/154	0.309/819	0.266/450
3		1000	0.238/144	0.982/841	0.242/154	0.39/818	0.518/633
4		10000	0.453/154	1.967/905	0.481/174	1.23/863	0.85/411
5		100000	2.037/159	12.579/969	2.09/174	10.943/906	9.546/698
6	EU	2	1.108/1029	0.384/326	0.524/625	0.617/1180	F/F
7		100	1.151/1112	0.496/355	0.571/625	0.738/1256	F/F
8		1000	1.359/1186	0.569/378	0.711/625	0.828/1290	F/F
9		10000	3.121/1210	1.272/400	1.875/729	2.702/1278	1.208/410
10		100000	23.621/1293	7.191/416	13.876/839	15.018/836	12.549/686
11	RN1	2	0.061/61	42.993/219613	0.28/111	44.59/215681	0.188/39
12		100	0.01/3	0.013/3	0.009/3	0.01/3	0.011/3
13		1000	0.013/3	0.012/3	0.009/3	0.011/3	0.011/3
14		10000	0.013/3	0.016/3	0.013/3	0.015/3	0.017/3
15		100000	0.038/3	0.046/3	0.039/3	0.044/3	0.056/3
16	RN2	2	0.203/49	0.056/13	0.108/27	0.096/19	0.064/13
17		100	0.224/55	0.055/13	0.116/29	0.092/19	0.07/15
18		1000	0.253/59	0.067/15	0.133/31	0.105/21	0.073/15
19		10000	0.372/63	0.101/15	0.197/33	0.147/21	0.113/15
20		100000	1.18/67	0.273/15	0.583/33	0.429/25	0.483/23
21	DL1	2	0.416/59	0.078/11	0.072/11	0.281/43	0.094/15
22		100	0.455/59	0.073/11	0.071/11	0.279/43	0.082/15
23		1000	0.396/59	0.075/11	0.074/11	0.358/43	0.092/15
24		10000	0.444/59	0.076/11	0.073/11	0.308/43	0.093/15
25		100000	0.779/59	0.162/11	0.13/11	0.55/43	3.145/247
26	DC	2	0.434/515	0.254/261	0.233/233	0.119/109	0.451/451
27		100	1.257/1424	0.605/578	1.145/1120	1.405/2815	1.413/1317
28		1000	2.116/2118	0.809/699	1.059/1020	0.955/2165	1.54/1342
29		10000	4.319/2179	1.64/774	1.554/865	4.79/3362	2.233/1030
30		100000	13.406/1124	10.577/830	7.858/654	43.145/3932	14.003/1167
31	RSE	2	0.176/41	0.492/95	0.147/37	0.481/119	0.078/15
32		100	0.18/45	0.544/109	0.178/43	0.591/135	0.076/15
33		1000	0.202/49	0.55/117	0.236/45	0.602/145	0.077/15
34		10000	0.295/53	0.881/125	0.312/49	1.109/157	0.108/15
35		100000	1.339/55	2.678/133	1.01/51	3.361/167	5.897/283
36	EDB	2	0.245/131	0.246/131	0.207/105	0.332/406	0.292/151
37		100	0.365/146	0.297/146	0.209/111	0.362/412	F/F
38		1000	0.386/155	0.309/155	0.238/120	0.469/490	F/F
39		10000	0.87/164	0.524/164	0.477/130	1.041/496	0.566/164
40		100000	2.952/174	2.9/174	2.324/143	7.982/494	2.324/130
41	EBK	2	5.807/15642	2.46/2636	1.815/2885	14.834/28707	F/F
42		100	6.429/16701	1.166/1295	1.026/1607	10.505/18135	F/F
43		1000	F/F	0.903/919	1.182/1763	3.004/4464	F/F
44		10000	15.642/11057	7.191/3543	4.477/2711	141.314/83276	F/F

45		100000	124.289/10168	49.66/3822	27.064/2118	219.95/16748	2.153/159
46	ANB	2	0.027/15	0.03/15	0.092/71	0.608/1220	0.03/10
47		100	0.015/40	0.011/40	0.012/40	0.016/40	0.014/40
48		1000	0.018/60	0.019/60	0.014/60 0.017	60	0.016/60
49		10000	0.069/80	0.059/80	0.057/80	0.071/80	0.073/80
50		100000	0.504/100	0.62/100	0.507/100	0.619/100	0.639 100

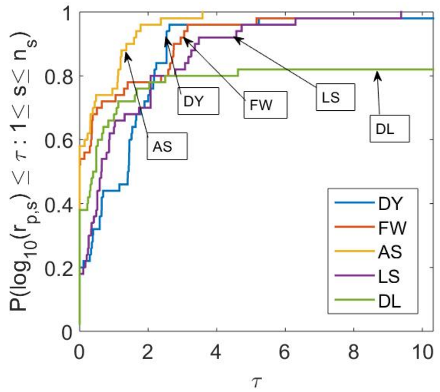

Figure 1: Number of iterations (ITR).

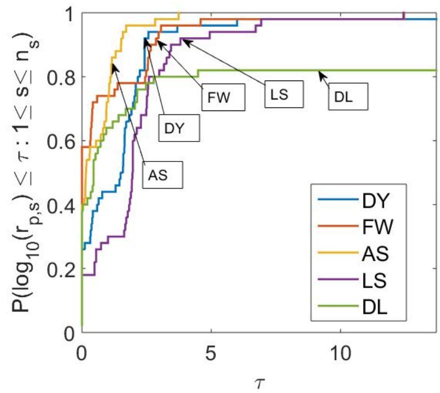

Figure 2: Value of function f (FV).

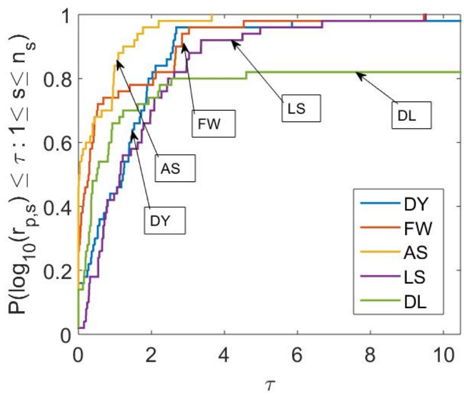

Figure 3: CPU time.

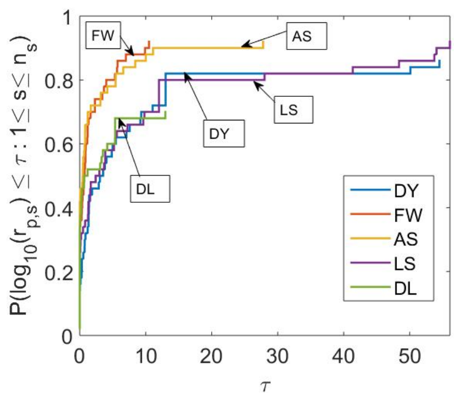

Figure 4: Gradient norm (GN).

5 Application to Mode Function

Conjugate gradient methods have proved to be efficient in solving large scale optimization problems arising from image restoration [20], signal processing [19], motion control [20], robotic [2], portfolio selection problems [24] and others in that direction. In statistics nonparametric problem, the mode of a probability density function (p.d.f) represents the distribution's peak and it is crucial for understanding the central tendency of any distribution. This finds real-world applications in decision-making problems. For instance in Transportation, Business etc. Estimating the mode of a p.d.f which can be determined for both independent and dependent data has taken a center stage over the years in theoretical and applied Statistics. A considerable number of authors have worked on determining this central tendency. Consult Sager [27] for a comprehensive historical survey. Parzen [25] estimated the unconditional mode in the independent and identically distributed case. Eddy [10] investigated the optimium kernel estimator of the mode for a distribution.

This section looks at the problem of finding the mode for a multivariate and unimodal probability density function f having its support in $\mathbb{R}^n$ taken into consideration the independent and identically distributed standard normal random variables $\{M_i\}_{i=1}^n$ with common probability density function f. Authors like Konakov [18] worked on the asymptotic normality of the mode of multidimensional distribution. Samanta [28] on the other hand, discussed the nonparametric estimation of the mode of a mutivariate density. Others have investigated problems of this kind.

Define density function f with unique mode t by

$$f(t) = max\{f(x) : x \in \mathbb{R}^n\}. \tag{58}$$

Let the random variable $\hat{t}$ be the kernel estimator of t and $f_n(x)$ the kernel estimator of $f(x)$, with $\hat{t}$ representing the maximizer of $f_n(x)$ such that

$$f(\hat{t}) = max\{f_n(x) : x \in \mathbb{R}^n\}, \tag{59}$$

where

$$f_n(x) = \frac{1}{nb_n^n}\sum_{j=1}^{n} P(\frac{x - M_j}{b_n}). \tag{60}$$

The sequence $\{b_n\}$ of positive real number which approaches zero as n approaches infinity is called the bandwidth while P is a p.d.f on $\mathbb{R}^n$. This simulation chooses between standard Guaussian kernel given by

$$P(x) = \frac{1}{(2\pi)^{\frac{n}{2}}}\exp(-\frac{1}{2}\sum_{i=1}^{n} x_i^2) \tag{61}$$

and Epanechnikov kernel defined by

$$P(x) = (\frac{3}{4})^n \prod_{i=1}^{n}(1 - x_i^2). \tag{62}$$

Meanwhile, the basic problem in kernel smoothing techniques is the selection of the bandwidth $\{b_n\}$. Hence, for this simulation, cross-validation technique method is used in choosing the optimal bandwidth. In this work, DY, FW, AS and DL algorithms are employed to solve problem (58) under standard Wolfe line search scheme. It is obvious from Table 3 that AS CG algorithm performed better than DY, FW and DL CG algorithms in term of number of iterations and CPU time in solving problem (58).

Table 3: Numerical result of Algorithms DL, FW, DY and AS for solving problem (50).

Dim.	DL	FW	DY	AS
Epanechnikov: Initial point $(0.05, \ldots, 0.05)$	ITN/CPT	ITN/CPT	ITN/CPT	ITN/CPT
8	18/1.535	20/2.328	22/0.844	15/1.200
10	15/0.239	14/0.390	16/1.728	12/0.210
12	6/0.388	8/0.235	10/0.203	5/0.350
14	5/0.187	6/0.256	9/0.384	4/0.170
16	9/0.231	7/0.453	8/0.391	7/0.200
20	3/0.525	4/0.656	6/0.782	2/0.480
40	1/0.390	2/0.782	5/0.656	1/ 0.360
113	4/1.812	3/0.391	7/0.453	3/ 1.500
Gaussian: Initial point $(0.1, \ldots, 0.1)$	ITN/CPT	ITN/CPT	ITN/CPT	ITN/CPT
160	6/16.466	7/87.000	10/228.180	5/14.500
170	5/12.066	9/33.000	8/99.080	4/11.000
190	14/45.551	6/20.000	7/69.830	12/40.000
180	14/46.782	4/19.000	6/60.440	11/38.000
270	4/30.020	1/28.000	5/204.830	3/28.000
290	24/162.810	3/13.000	4/109.750	8/150.000
320	12/127.570	2/15.000	3/155.180	10/120.000
340	9/107.190	5/10.000	2/111.310	7/100.000
Gaussian: Initial point $(-0.9, \ldots, -0.9)$	ITN/CPT	ITN/CPT	ITN/CPT	ITN/CPT
26	36/2.656	106/8.867	975/67.547	30/2.400
32	86/8.969	44/4.360	86/8.983	40/8.200
37	51/7.469	112/14.891	57/7.777	45/7.000
40	11/1.922	80/11.133	85/12.175	9/1.800
48	64/13.343	206/42.625	84/17.657	55/12.000
50	10/2.203	37/8.251	54/12.171	8/2.000
58	4/1.031	8/2.750	43/15.047	3/0.950
60	14/10.968	31/10.000	13/4.266	12/9.800

6 Conclusions

This work constructed a new CG update parameter based on the conjugacy condition presented in the work of Dai-Liao [7]. The resulting update parameter embeds DY-update parameter. Our method is better than the existing methods compared because the global convergence property of DY method was enhanced. It also has some Hessian matrix information inherited from DL method. Consequently, the numerical ability of our method compared with other methods in the literature showed that it is promising. Furthermore, the practical application of our method was showcased in the area of nonparametric estimation of mode function.

Conflicts of Interest There is no conflict of interest among the authors.

References

[1] N. Andrei (2004). Test functions for unconstrained optimization. *Research Institute for informatics. Center for Advanced Modeling and Optimization*, pp. 8–10.

[2] A. M. Awwal & N. Pakkaranang (2025). New structured spectral gradient methods for nonlinear least squares with application in robotic motion control problems. *Journal of Computational and Applied Mathematics*, *469*, Article ID: 116671. https://doi.org/10.1016/j.cam.2025.116671.

[3] S. A. Ayinde, I. A. Osinuga & A. K. Adio (2025). A new descent extension of DY conjugate gradient method based on DL approach. *Filomat*, *39*(15), 5071–5084. https://doi.org/10.2298/FIL2515071A.

[4] S. Babaie-Kafaki (2023). A survey on the Dai-Liao family of nonlinear conjugate gradient methods. *RAIRO-Operations Research*, *57*, 43–58. https://doi.org/10.1051/ro/2022213.

[5] S. Babaie-Kafaki & R. Ghanbari (2014). The Dai–Liao nonlinear conjugate gradient method with optimal parameter choices. *European Journal of Operational Research*, *234*(3), 625–630. https://doi.org/10.1016/j.ejor.2013.11.012.

[6] Y. Cheng, Q. Mou, X. Pan & S. Yao (2016). A sufficient descent conjugate gradient method and its global convergence. *Optimization Methods and Software*, *31*(3), 577–590. https://doi.org/10.1080/10556788.2015.1124431.

[7] Y. Dai & L. Liao (2001). New conjugacy conditions and related nonlinear conjugate gradient methods. *Applied Mathematics and Optimization*, *43*, 87–101. https://doi.org/10.1007/s002450010019.

[8] Y.-H. Dai & Y. Yuan (1999). A nonlinear conjugate gradient method with a strong global convergence property. *SIAM Journal on optimization*, *10*(1), 177–182. https://doi.org/10.1137/S1052623497318992.

[9] E. D. Dolan & J. J. More (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, *91*, 201–213. https://doi.org/10.1007/s101070100263.

[10] W. F. Eddy (1980). Optimum kernel estimators of the mode. *The Annals of Statistics*, *8*(4), 870–882. https://doi.org/10.1214/aos/1176345080.

[11] M. Fang, M. Wang, M. Sun & R. Chen (2021). A modified hybrid conjugate gradient method for unconstrained optimization. *Journal of Mathematics*, *2021*(1), Article ID: 5597863. https://doi.org/10.1155/2021/5597863.

[12] R. Fletcher & C. M. Reeves (1964). Function minimization by conjugate gradients. *The computer journal*, *7*(2), 149–154. https://doi.org/10.1093/comjnl/7.2.149.

[13] H. Guebbai & M. Ghiat (2023). A new hybrid conjugate gradient algorithm for unconstrained optimization. *Vestnik Udmurtskogo Universiteta. Matematika. Mekhanika. Komp'yuternye Nauki*, *33*(2), 348–364. https://doi.org/10.35634/vm230211.

[14] W. W. Hager & H. Zhang (2005). A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on optimization*, *16*(1), 170–192. https://doi.org/10.1137/030601880.

[15] W. W. Hager & H. Zhang (2006). A survey of nonlinear conjugate gradient methods. *Pacific journal of Optimization*, *2*(1), 35–58.

[16] M. R. Hestenes, E. Stiefel et al. (1952). Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, *49*(6), 409–436. https://doi.org/10.6028/JRES.049.044.

[17] M. A. I. Ishak & S. M. Marjugi (2024). A new hybrid three-term LS-CD conjugate gradient in solving unconstrained optimization problems. *Malaysian Journal of Mathematical Sciences*, *18*(1), 167–184. https://doi.org/10.47836/mjms.18.1.10.

[18] V. Konakov (1974). On the asymptotic normality of the mode of multidimensional distributions. *Theory of Probability & Its Applications*, *18*(4), 794–799. https://doi.org/10.1137/1118104.

[19] P. Kumam, A. B. Abubakar, A. H. Ibrahim, H. U. Kura, B. Panyanak & N. Pakkaranang (2022). Another hybrid approach for solving monotone operator equations and application to signal processing. *Mathematical Methods in the Applied Sciences*, *45*(12), 7897–7922. https://doi.org/10.1002/mma.8285.

[20] P. Kumam, A. B. Abubakar, M. Malik, A. H. Ibrahim, N. Pakkaranang & B. Panyanak (2023). A hybrid HS-LS conjugate gradient algorithm for unconstrained optimization with applications in motion control and image recovery. *Journal of Computational and Applied Mathematics*, *433*, Article ID: 115304. https://doi.org/10.1016/j.cam.2023.115304.

[21] G. Li, C. Tang & Z. Wei (2007). New conjugacy condition and related new conjugate gradient methods for unconstrained optimization. *Journal of Computational and Applied Mathematics*, *202*(2), 523–539. https://doi.org/10.1016/j.cam.2006.03.005.

[22] Y. Liu & C. Storey (1991). Efficient generalized conjugate gradient algorithms, part 1: Theory. *Journal of Optimization Theory and Applications*, *69*, 129–137. https://doi.org/10.1007/BF00940464.

[23] M. Lotfi & S. M. Hosseini (2020). An efficient Dai–Liao type conjugate gradient method by reformulating the cg parameter in the search direction equation. *Journal of Computational and Applied Mathematics*, *371*, Article ID: 112708. https://doi.org/10.1016/j.cam.2019.112708.

[24] M. Malik, I. M. Sulaiman, A. B. Abubakar, G. Ardaneswari et al. (2023). A new family of hybrid three-term conjugate gradient method for unconstrained optimization with application to image restoration and portfolio selection. *AIMS Mathematics*, *8*(1), 1. https://doi.org/10.3934/math.2023001.

[25] E. Parzen (1962). On estimating probability density function and mode. *The Annals of Mathematical Statistics*, *33*(3), Article ID: 10651076.

[26] E. Polak & G. Ribiere (1969). Note sur la convergence de directions conjugate. *Analysis- Mode 'elisation Math 'ematique et Analyse Num 'erique*, *3*(R1), 35–43.

[27] T. W. Sager (1979). An iterative method for estimating a multivariate mode and isopleth. *Journal of the American Statistical Association*, *74*(366a), 329–339. https://doi.org/10.1080/01621459.1979.10482514.

[28] M. Samanta (1973). Nonparametric estimation of the mode of a multivariate density. *South African Statistical Journal*, *7*(2), 109–117. https://journals.co.za/doi/pdf/10.10520/AJA0038271X_165.

[29] A. Semiu, O. Idowu, A. Adesina, A. Sunday, A. Joseph, U. Uchenna & A. Olufisayo (2024). A descent conjugate gradient method for optimization problems. *IAENG International Journal of Applied Mathematics*, *54*(9).

[30] I. B. Stanimirovic P. S. & M. D. (2020). A survey of gradient methods for solving nonlinear optimization. *Electronic Research Archive*, *28*(4), 1573–1624. https://doi.org/10.3934/era.2020115.

[31] D. Touati-Ahmed & C. Storey (1990). Efficient hybrid conjugate gradient techniques. *Journal of optimization theory and applications*, *64*(2), 379–397. https://doi.org/10.1007/BF00939455.

[32] M. Yabe H.and Takano (2004). Global convergence properties of nonlinear conjugate gradient methods with modified secant condition. *Computational Optimization and Applications*, *28*(6), 203–225. https://doi.org/10.1023/B:COAP.0000026885.81997.88.

[33] M. M. Zabidin S. Ahmad A. & M. R. (2016). An efficient modification of nonlinear conjugate gradient method. *Malaysian Journal of Mathematical Sciences*, *10*(S), 167– 128.

[34] G. Zoutendijk (1970). Nonlinear programming, computational methods. *Integer and nonlinear programming*, pp. 37–86.