Academic Science Journal (ASJ)
Vol. 4, No. 2, April 2026, pp. 94~103
DOI: 10.24237/04.02.871

Using MATLAB for Steepest Descent Algorithm (SD) with Conjugate Gradient Algorithm (CG) For Minimizing Unconstrained problems

Huda H Al-Zobiadi, Adawiya A. Mahmood Al -Nuaimi

Department of Mathematics, College of Science, University of Diyala, Diyala, Iraq.

Article Info

Article history:

Received 29,12, 2025
Revised 07, 01, 2026
Accepted 11,03, 2026
Published 30, 04,2026

Keywords:

Optimization,
Nonlinear,
Minimization,
Steepest Descent Algorithm,
Conjugate Gradient Algorithm,
Unconstrained Problems.

ABSTRACT

This paper considers the steepest descent algorithm (SD) and the conjugate gradient algorithm (CG) with a new parameter β_k to minimize nonlinear unconstrained problems. We improve a new parameter β_k for the conjugate gradient algorithm (CG). This new β_k makes the conjugate gradient algorithm (CG) more efficient. Algorithms are submitted and run in the MATLAB program for several examples of minimization problems. The nonlinear functions for unconstrained minimization problems that were used in this paper have not been studied in previous publications. The implementation of the MATLAB program shows that the conjugate gradient algorithm (CG) with new β_kIt is more effective than the steepest descent algorithm (SD) while the run time for the steepest descent algorithm (SD) is less than that of the conjugate gradient algorithm (CG). The contribution of this paper will be important to solving more complicated unconstrained nonlinear problems for more variables.

Corresponding Author:

Huda H Al-Zobiadi
Department of Mathematics, College of Science,
University of Diyala, Baqubah City, Diyala Governorate, Iraq
Email: scimathms242508@uodiyala.edu.iq

1. INTRODUCTION

Optimization denotes discovering of the optimal solution among many feasible solutions. Feasible solutions are those that satisfy all the constraints from the optimization problem [1]. Optimization problems draw on fundamental tools from decision theory and physical systems analysis. The process of determining the optimal solution to a problem in a particular sense and under certain circumstances is known as optimization [2]. Algorithms for minimizing continuously differentiable functions have been researched by researchers [3]–[7]. Following the development of computers in the 1980s, numerous complex issues were resolved. Today's optimization challenges are multidisciplinary and multi-objective in nature [1],[8]. Due to its ease of use and success in numerous implementations, the steepest descent method (SD) has drawn a lot of attention [9]-[11]. Fletcher developed a method for quadratic and nonquadratic goal functions that makes significant performance gains by utilizing the accessibility of more "long" storage vectors [12]. Hamoda M.A. et al. in [13] focus on a novel nonlinear CG technique for solving large-scale unconstrained minimization problems. Guo J.&et .al. In [14] provide a novel three-term CG approach to effectively solve unconstrained minimization issues. Because scientific computing and massive data mining frequently result in unconstrained optimization problems [15]-[18], Creating effective numerical algorithms to address these issues is beneficial. When compared to other comparable algorithms, it appears that no method of the literature is for a dominant situation when it comes for solving most unconstrained minimization issues [19]– [24]. Salih D. T. M.&et .al. in [25] introduce a comparison between the SD and the CG for minimizing non-linear functions. In [26], the standard three-term CG-algorithms are developed to find the minimum values of non-linear test functions.

The successful implementation of it has been seen that the nonlinear CG can be used to control three-degree-of-freedom robotic planar motion control systems [27]. Abbas I.T.et al.in [28] used the gravity search algorithm and combined it with nonlinear regression problems to estimate and interpret the parameters. A tabu search algorithm for quadratic assignment issues is suggested (TSQAP) in [29]. Jasim S.H. and Abbas I.T. in [30] presented a proposed effective model for the role of using waiting queue models in improving the performance of health institutions in Iraq, Baghdad.

2. METHOD

2.1 Formulation of unconstrained minimization problem

The unconstrained minimization problem is of the form [31].

$$\begin{matrix} Minimize f(x) \\ Subject\ to\ x \in \Omega \end{matrix} \cdots\cdots (1)$$

Such that f is a real _valued function. and Ω, the feasible set is a subset of E^n .
The question that occurs in the unconstrained(minimization) problem (1), is if there is a solution. A theorem, which asserts that the solution exists if f is continuous and Ω is compact, is the fundamental conclusion that can be applied to this problem. This result should be kept in mind throughout distinguishing solution points and discovering efficient algorithms to explore these points.

For a study of the issue (1), we present two types of solution points: local minimum points and global minimum points.

Definition(1) :[31] A point X*ϵ Ω is considered to be a relative minimum point or local minimum point of f over Ω if there is a $\gamma > 0$ such that f(X) ≥ f(X*) for all X ϵ Ω with in a distance γ of X* (i.e , X ϵ Ω and | X - X*| < γ) if f(X) > f(X*) for all X ϵ Ω , X ≠ X*, with in a distance γ of X* , then X* is said to be a strict relative minimum point of f over Ω .

Definition (2):[31] a point X* ϵ Ω is considered to be a global minimum point of f over Ω if f (X) ≥ f (X*) ∀ Xϵ Ω. If f(X) > f(X*) ∀ X ϵ Ω, X ≠ X*, then X* is said to be a strict global minimum point of f over Ω.

In formulating and solving problem (1), we are search for a relative minimum point. We introduce two algorithms, the steepest descent algorithm and conjugate gradient algorithm to obtain minimum points.

2.1.1 Mathematical Symbols

X : (x_1, x_2) Vector or point

f: Function

$$\text{m: gradient vector} = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}$$

$$\text{H: Hessian matrix} \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix}$$

2.2 Steepest Descent Algorithm (SD).

This algorithm can be constructed as in the following: [25,31]

Step (1): The initial point X_0 is given with
m_0 =grad f(X_0) and tolerance of convergence tol.

Step (2): put H_k is Hessian matrix of $f(x_k)$, $\alpha_k = \frac{m_k^T m_k}{m_k^T H_k m_k}$ $\forall k \epsilon N$

Step (3): put $x_{k+1} = x_k - \alpha_k m_k$ and $m_{k+1} = grad(f(x_{k+1}))$.

Step (4): if $||m_{k+1}|| < tol$, then stop, x_{k+1} is minimum point and $f(x_{k+1})$ is the minimum value, else, go to step (2).

Global convergence Analysis for the steepest descent algorithm (SD) is stated in [31]

2.3 Conjugate Gradient Algorithm (CG).

This algorithm can be constructed as in the following: [25],[31]

Step (1): The initial point X_0 is given, m_0= grad (f(X_0)) with tolerance of convergence tol and
d_0= -m_0

Step (2) : put H_k is Hessian matrix of $f(X_k)$, $\alpha_k = -\frac{m_k^T d_k}{d_k^T H_k d_k}$, ∀kϵ N

Step (3): put $x_{k+1} = x_k + \alpha_k d_k$, $m_{k+1} = grad(f(x_{k+1}))$.

Step (4): if $||m_{k+1}|| < tol$,then stop, x_{k+1} the minimum point and $f(x_{k+1})$ is the minimum value.

Step (5): find $\beta_k = \frac{m_{k+1}^T H_k d_k}{d_k^T H_k d_k}$, $d_{k+1} = -m_{k+1} + \beta_k d_k$ and go to step (2).
We improve a new β_k for step (5) to develop the efficiency of the conjugate gradient algorithm (CG) as follows: $\beta_k = \frac{m_{k+1}^T\, H_k\, m_{k+1}}{d_k^T\, H_k\, m_k}$,
Where m_{k+1}^T is the transpose of the $grad\ (f(x_{k+1}))$,
d_k^T is the transpose of $-m_k$

2.4 Global convergence Analysis for the conjugate gradient algorithm (CG) with new β_k:

Theorem 1. Let $\{x_k\}$ be a sequence obtained by algorithm (CG) with new $\beta_k = \frac{m_{k+1}^T\, H_k\, m_{k+1}}{d_k^T\, H_k\, m_k}$

$$\text{then} \quad m_k^T d_k \le -c\left||m_k|\right|^2\ \forall k \ge 0, c > \ 0 \qquad (2)$$

holds for all $k \ge 0$.
Proof. We prove by induction, that if k=0 then $m_0^T d_0 = -c\ \|m_0\|^2$
Hence, the condition holds; now we need to prove that:

$$m_k^T d_k \le -c\ \|m_k\|^2 for\ k \ge 1$$

From the search direction $d_k$where $d_k\ is\ defined\ as\ follows$:

$$d_k = \begin{cases} -m_k & if\ k = 0 \\ -m_k + \beta_{k-1} d_{(k-1)} & if\ k \ge 1 \end{cases} \quad \dots \quad (3)$$

We have $d_{k+1} = -m_{k+1} + \beta_k d_k$
Multiply both sides by m_{k+1}^T

$$m_{k+1}^T d_{k+1} = m_{k+1}^T(-m_{k+1} + \beta_k d_k) = -||m_{k+1}||^2 + \beta_k m_{k+1}^T d_k \qquad (4)$$

Using an exact line search, it can be shown that $m_{k+1}^T d_k$=0. Thus $m_{k+1}^T d_{(k+1)} = -\|m_{k+1}\|^2$
Thus, this condition satisfies for k+1. Therefore (2) holds. The global convergence for the conjugate gradient algorithm with new β_k is derived directly from the global convergence properties. These properties are described in [31].
Example 1. Suppose the nonlinear function $f(x_1, x_2) = x_1^2 + x_2^2 + e^{-x1}$ such that $x_0 = (0,0)^T$, tol =0.1.
Solution: The application of steepest descent algorithm (SD) is as in the following:
The gradient of f at X_0 is m_0=grad(f(X_0)) = $\begin{bmatrix} 2x_1 - e^{-x1} \\ 2x_2 \end{bmatrix} = \begin{bmatrix} 2*0 - e^0 \\ 2*0 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$
The Hessian matrix of f(X_0) is H=$\begin{bmatrix} 2+e^{-x1} & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 2+e^0 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}$
We get $\alpha_0 = \frac{m_0^T m_0}{m_0^T H_0 m_0} = \frac{[-1\quad 0]\begin{bmatrix} -1 \\ 0 \end{bmatrix}}{[-1\quad 0]\begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} -1 \\ 0 \end{bmatrix}} = \frac{1}{[-3\quad 0]\begin{bmatrix} -1 \\ 0 \end{bmatrix}} = \frac{1}{3} = 0.333333$
put $x_1 = x_0 - \alpha_0 m_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - 0.333333 \begin{bmatrix} -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} -0.333333 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.333333 \\ 0 \end{bmatrix}$
Then $m_1 = grad(f(X_1)) = \begin{bmatrix} 2*0.333333 - e^{(-0.333333)} \\ 2*0 \end{bmatrix} = \begin{bmatrix} -0.049865 \\ 0 \end{bmatrix}$
Thus $\|m_1\| = \sqrt{(-0.049865)^2 + (0)^2} = 0.049865 < tol$,stop
Hence, $x_1 = (0.333333{,}0)^T$is the minimum point and $f(\, x_1) = 0.827642$ is the minimum value.
Now, we will apply the conjugate gradient algorithm (CG) for example 1 as in the following:
m_0=grad(f(X_0)) = $\begin{bmatrix} 2x_1 - e^{-x1} \\ 2x_2 \end{bmatrix} = \begin{bmatrix} 2*0 - e^0 \\ 2*0 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$
The Hessian matrix of f(X_0) is H=$\begin{bmatrix} 2+e^{-x1} & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 2+e^0 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}$
thus $d_0 = -m_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\alpha_0 = -\frac{m_0^T d_0}{d_0^T H_0 d_0} = -\frac{[-1\quad 0]\begin{bmatrix} 1 \\ 0 \end{bmatrix}}{[1\quad 0]\begin{bmatrix} 3 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix}} = -\frac{-1}{3} = 0.333333$
and $X_1 = X_0 + \alpha_0 d_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + 0.33333 \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0.33333 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.33333 \\ 0 \end{bmatrix}$
$m_1 = grad(f(X_1)) = \begin{bmatrix} (2*0.33333) - e^{(-0.33333)} \\ 2*0 \end{bmatrix} = \begin{bmatrix} -0.04987 \\ 0 \end{bmatrix}$

Thus $\|m_1\| = \sqrt{(-0.04987)^2 + (0)^2} = 0.04987$<tol, stop.
Hence $x_1 = (0.33333\ {,}0)^T$is the minimum point and $f(x_1) = 0.827642$ is the minimum value.

3. RESULTS AND DISCUSSION

3.1 MATLAB RUN

MATLAB code is run to minimize three functions. The steepest descent algorithm (SD) and the conjugate gradient algorithm (CG) with new β_k are run in MATLAB 2024. The input parameter settings for the two algorithms found in [25].

The outcomes for the steepest descent (SD) algorithm in Example 1 are presented in Table 1.

Table 1. Outcomes of example 1 for the steepest descent algorithm(SD)

Iteration (k)	x_1	x_2	α_k	$\|m\|$	$f(x_k)$	m_1	m_2
0	0	0	0.333333	1	1	-1	0
1	0.3333333	0	0.368304	0.049865	0.827642	-0.049865	0

The function for example 1 is minimized after two iterations for the steepest descent algorithm (SD). Hence, $x_1 = (0.333333,0)^T$ is the minimum point and $f(x_1) = 0.827642$ is the minimum value and m =(-0.049865,0)^T $\|m\|$=0.049865 <tol. The execution time for the steepest descent algorithm (SD) with plots is 0.630929 seconds. The plots of the function and the decrease of $f(x_k)$ per iteration for (SD) are illustrated in Figure 1 and Figure 2 consequently for example 1.

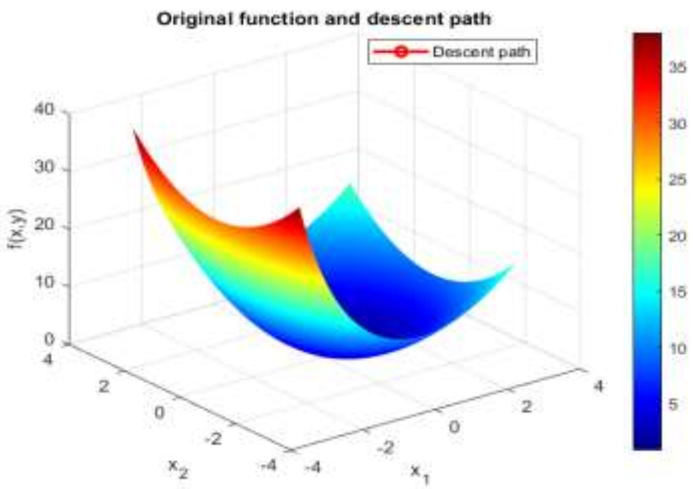

Figure 1. The function $f(x_1, x_2) = x_1^2 + x_2^2 + e^{-x1}$

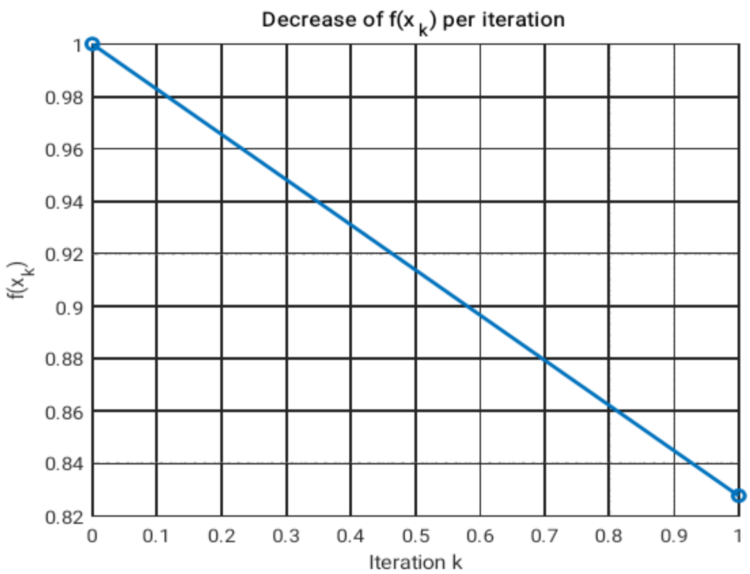

Figure 2. The decrease of $f(x_k)$ per iteration

The outcomes for the conjugate gradient algorithm (CG) with new β_k of example 1 is obtained in Table 2. For the conjugate gradient (CG) algorithm with the new β_k, the function for example 1 is minimized after two iterations. Hence, $x_1 = (0.33333,0)^T$ is the minimum point and $f(x_1) = 0.827642$ is the minimum value and m =(- 0.049865,0)^T $\|m\|$= 0.049865 <tol. The execution time for the conjugate gradient algorithm with new β_k, with plots is 0.700708 seconds. The plot of the decrease of $f(x_k)$ per iteration for (CG) is illustrated in Figure 3, for example 1

Table 2. Outcomes of example1 for (CG)algorithm with new β_k

Iteration (k)	x_1	x_2	m_1	m_2	$\|m\|$	α_k	New β_k	$f(x_k)$
0	0	0	-1	0	1	0.33333	-0.0025	1
1	0.33333	0	-0.049865	0	0.049865	0.33333	-0.002486	0.827642

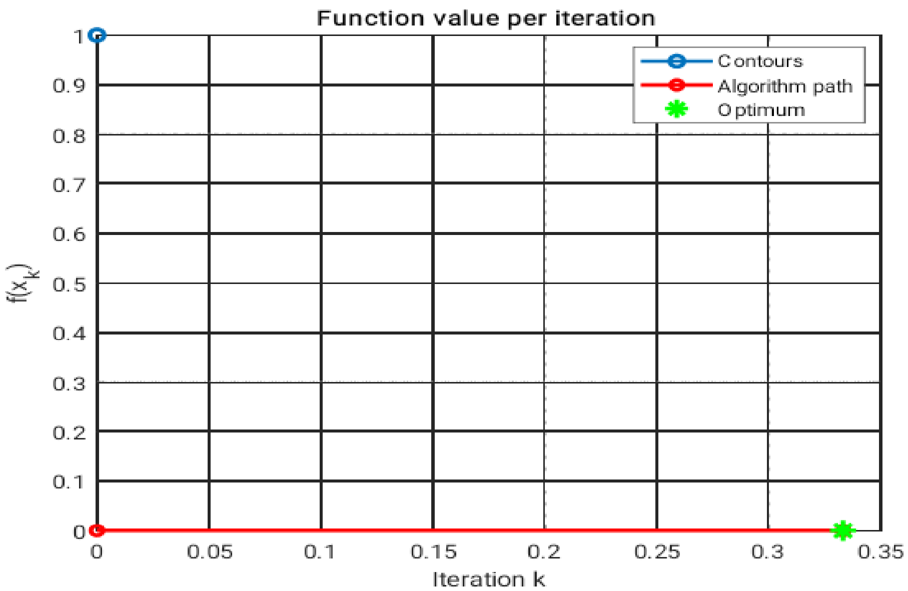

Figure 3. The decrease of $f(x_k)$ per iteration

Example 2. Suppose the nonlinear function $f\,(x_1, x_2) = x_1^2 + x_2^2 + x_1x_2 + 4x_2$ such that $x_0 = (0,0)^T, tol = 0.01$. The outcomes for the steepest descent algorithm (SD) of example 2 are obtained in Table 3 by using MATLAB programming.

Table 3. outcomes of example 2 for the steepest descent algorithm (SD)

Iteration (k)	x_1	x_2	α_k	m_1	m_2	$\|m\|$	$f(x_k)$
0	0	0	0.5	0	4	4	0
1	0	-2	0.5	-2	0	2	-4
2	1	-2	0.5	0	1	1	-5
3	1	-2.5	0.5	-0.5	0	0.5	-5.25
4	1.25	-2.5	0.5	0	0.25	0.25	-5.3125
5	1.25	-2.625	0.5	-0.125	0	0.125	-5.328125
6	1.3125	-2.625	0.5	0	0.0625	0.0625	-5.332031
7	1.3125	-2.65625	0.5	-0.03125	0	0.03125	-5.333008
8	1.328125	-2.65625	0.5	0	0.015625	0.015625	-5.333252
9	1.328125	-2.66406	0.5	-0.007812	0	0.007812	-5.333313

The function for example 2 is minimized after 10 iterations for the steepest descent algorithm (SD). Hence $x_9 = (1.328125, -2.66406)$ is the minimum point and $f(x_9) =$ -5.333313 is the minimum value and $m = (-0.007812,0)^T$ ||m||= 0.007812<tol. The execution time for the steepest descent algorithm (SD) with plots is 0.633336 seconds. The plots of the function and the decrease of $f(x_k)$ per iteration for (SD) are illustrated in Figure 4 and Figure 5 consequently for example 2.

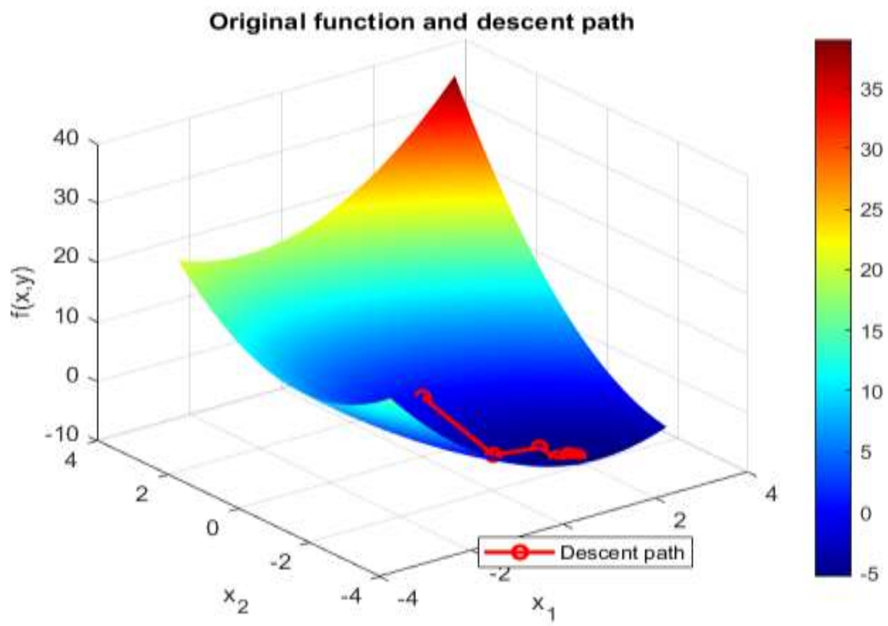

Figure 4. The function f $(x_1, x_2) = x_1^2 + x_2^2 + x_1x_2 + 4x_2$)

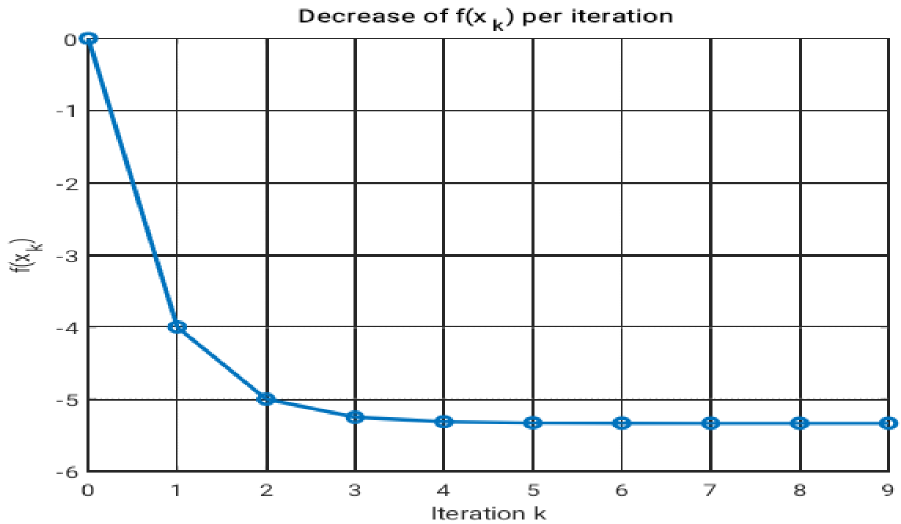

Figure 5. The decrease of $f(x_k)$ per iteration

The outcomes for the conjugate gradient algorithm (CG) with new β_k of example 2 are obtained in Table 4 by using MATLAB programming.

Table 4. Outcomes of example 2 for the conjugate gradient algorithm (CG) with new β_k

Iteration (k)	x_1	x_2	m_1	m_2	$\|\|m\|\|$	α_k	New β_k	$f(x_k)$
0	0	0	0	4	4	0.5000	-0.2500	0
1	0	-2	-2	0	2	0.2857	-0.1959	-4
2	0.5714	-1.7143	-0.5714	1.1429	1.2778	0.5153	-1.1629	-4.5714
3	0.6640	-2.4042	-1.0762	-0.1444	1.0859	0.1151	-0.1611	-4.9921
4	0.7638	-2.2084	-0.6808	0.3471	0.7641	0.8524	-0.7830	-5.0599
5	1.2249	-2.7379	-0.2880	-0.2508	0.3819	0.1579	-0.2382	-5.3088
6	1.2035	-2.6215	-0.2144	-0.0394	0.2180	0.5187	-0.1367	-5.3203
7	1.3315	-2.6921	-0.0291	-0.0528	0.0603	0.3793	-0.0072	-5.3326
8	1.3298	−2.6651	-0.0055	-0.0004	0.005556	0.3793	-0.00722	−5.333324

For the conjugate gradient algorithm (CG) with new β_k, the function of example 2 is minimized after 9 iterations. Hence $x_8 = (1.3298, -2.6651)^T$ is the minimum point and $f(x_8) = -5.333324$ is the minimum value and $m = (-0.0055, -0.0004)^T$ ||m||= 0.005556< tol.

The execution time for the conjugate gradient algorithm (CG) with new β_k, with plots is 0.956773 seconds. The plot of the function value per iteration for (CG) is illustrated in Figure 6 for example 2.

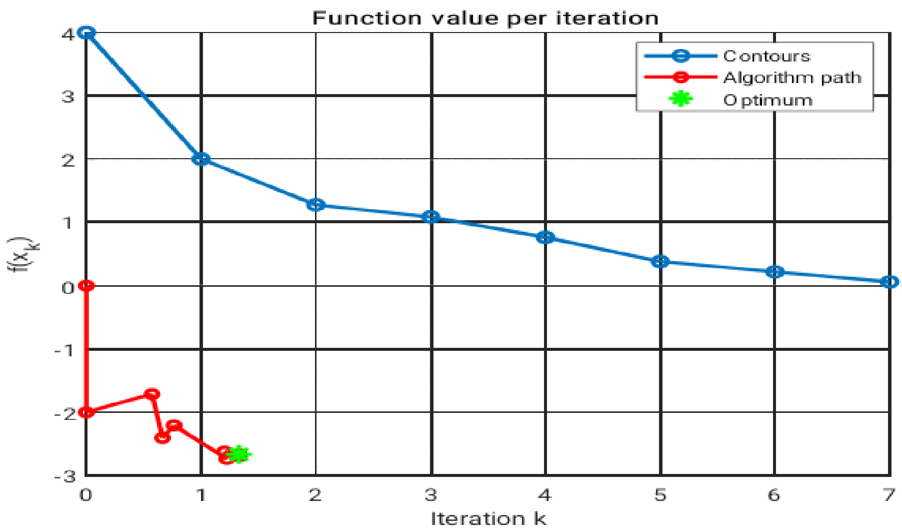

Figure 6. The decrease of $f(x_k)$ per iteration

Example 3. Suppose the nonlinear function $f\,(x_1, x_2) = 5x_1^2 + 5x_2^2 - 8\,x_2 - e^{x_1}$, such that $x_0 = (0{,}0)^T, tol = 0.01$. The outcomes for the steepest descent algorithm (SD) of example 3 is obtained in Table 5 by using MATLAB programming.

Table 5. Outcomes of example 3 for the steepest descent algorithm (SD)

iteration	x_1	x_2	α_k	m_1	m_2	$\|m\|$	$f(x_k)$
0	0	0	0.1	-1	-8	8.062258	-1
1	0.1	0.8	0.1	-0.105171	0	0.105171	-4.255171
2	0.110517	0.8	0.1	-0.011685	0	0.011685	-4.255785
3	0.111686	0.8	0.1	-0.001306	0	0.001306	-4.255793

The function for example 3 is minimized after 4 iterations for the steepest descent algorithm (SD). Hence $x_3 = (0.111686, 0.8)$ is the minimum point and $f(x_3) = -4.255793$ is the minimum value and $m = (-0.011685\ ,0)^T$ ||m||= 0.001306 < tol. The execution time for the steepest descent algorithm (SD) with plots is 0.505912 seconds. The plots of the function and the decrease of $f(x_k)$ per iteration for (SD) are illustrated in Figure 7 and Figure 8 consequently for example 3.

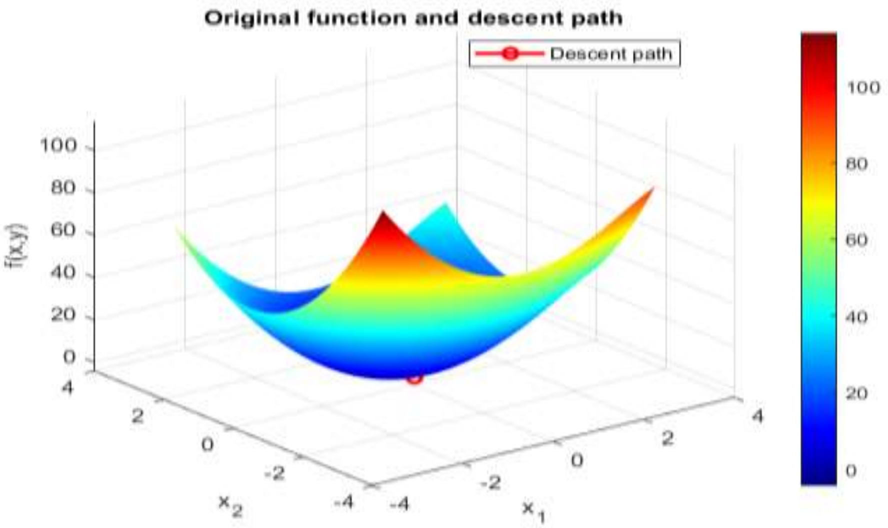

Figure 7. The function $f\,(x_1, x_2) = 5x_1^2 + 5x_2^2 - 8\,x_2 - e^{x_1}$)

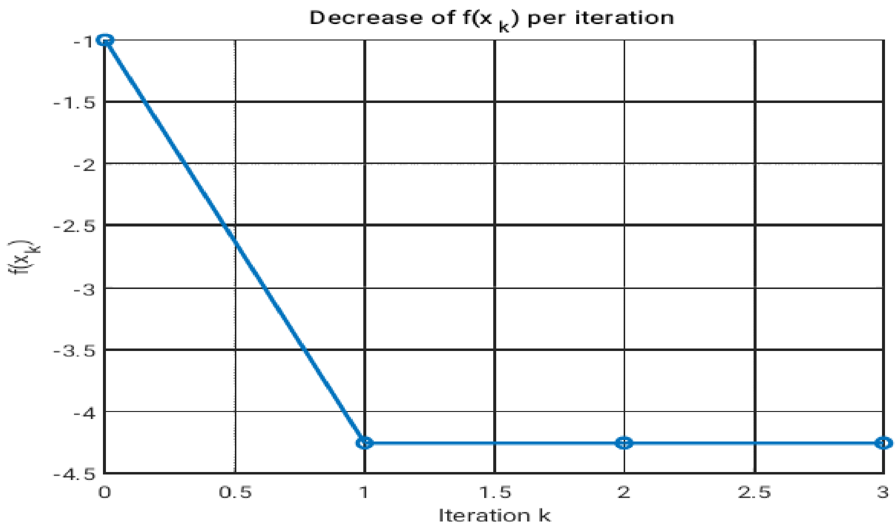

Figure 8. The decrease of $f(x_k)$ per iteration

The outcomes for the conjugate gradient algorithm (CG) with new β_k Examples 3 are obtained in Table 6 by using MATLAB programming.

Table 6. Outcomes of example 3 for the conjugate gradient algorithm (CG) with new β_k

Iteration (k)	x_1	x_2	m_1	m_2	$\|\|m\|\|$	α_k	New β_k	$f(x_k)$
0	0	0	-1	-8	8.0623	0.1002	-0.00015003	-1
1	0.1002	0.8012	-0.1038	0.0123	0.1045	0.1122	-0.00085291	-4.2552
2	0.1118	0.7997	-0.0004	-0.0028	0.002884	0.112187	-0.000853	-4.255793

For the conjugate gradient algorithm (CG) with new β_k, the function of example 3 is minimized after 3 iterations. Hence $x_2 = (0.1118,0.7997)^T$ is the minimum point and $f(x_2) =$ -4.255793 is the minimum value and $m = (-0.0004\,,-0.0028\,)^T$ ||m||= 0.002884 < tol. The execution time for the conjugate gradient algorithm (CG) with new β_k, with plots is 0.674644 seconds. The plot of the function value per iteration for (CG) is illustrated in Figure 9 for example 3.

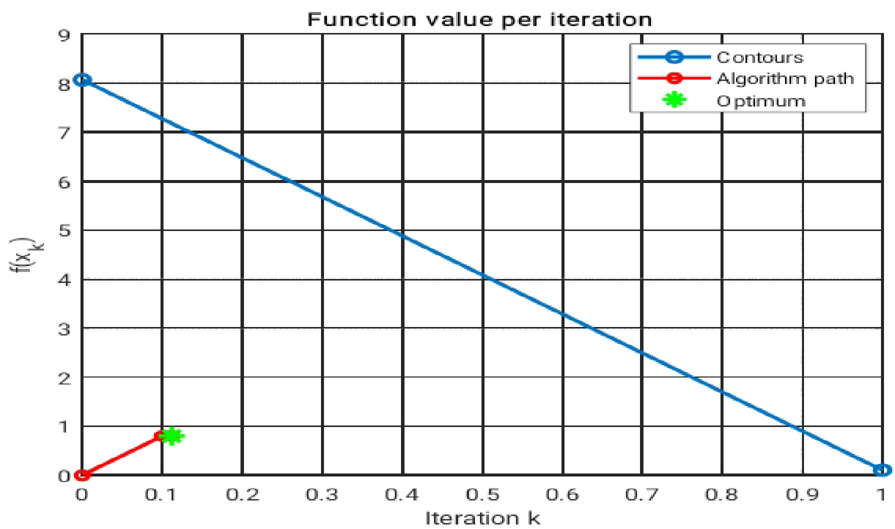

Figure 9. The decrease of $f(x_k)$ per iteration

4. CONCLUSION

The steepest descent algorithm (SD) and conjugate gradient algorithm (CG) with an improved β_k parameter are presented in this paper. Three challenging instances of nonlinear functions for the unconstrained problem are solved by applying the two algorithms to minimize the unconstrained problem. The steepest descent approach (SD) is applied using MATLAB programming and handwriting in Example 1. Additionally, MATLAB programming for the conjugate gradient algorithm (CG) with new β_k is used to solve example 1. For example2 and example 3, MATLAB programming is used to run the conjugate gradient algorithm (CG) with new β_k and the steepest descent algorithm (SD).

Based on the results of the two algorithms, we demonstrate that the conjugate gradient algorithm (CG) requires less iterations than the steepest descent technique (SD). The nonlinear functions' values for the examples of the conjugate gradient algorithm (CG) is more effective than the steepest descent algorithm (SD). the execution time for the steepest descent algorithm (SD) with plots is less than the conjugate gradient algorithm (CG). The contribution of this paper is in solving more complicated unconstrained problems with more variables.

ACKNOWLEDGEMENTS

The author gratefully acknowledges the support of department of mathematics, college of science, university of Diyala.

Conflict of interest

The authors declare that they have no conflicts of interest.

Source of funding

No funding.

Ethical Clearance

This research is theoretical in the field of mathematics, therefore, ethical approval is not required.

REFERENCES

[1] Arora, R. K., "Optimization: algorithms and applications", 1st ed., CRC press, New York, USA, 2015, pp. 303-310.

[2] Djordjevic, S. S., "Unconstrained Optimization Methods:Conjugate Gradient Methods and Trust-Region", Appl. Math., 2019, doi: 10.5772/intechopen.84374.

[3] E. Carrizosa, E. Conde, M. MuñozMárquez, J. Puerto, "Planarpoint objective location problems with nonconvex constraints: a geometrical construction," J. Global Optim., Vol. 6, no. 1, pp. 77–86, 1995, https://doi.org/10.1007/BF01106606.

[4] E. Carrizosa, J. B. G. Frenk, "Dominating sets for convex functions with some applications," J. Optim. Theo. Appl., Vol. 96, no. 2, pp. 281–29, 1998, https://doi.org/10.1023/A:1022614029984.

[5] Das, I. and Dennis, J. E., "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," SIAM j. on optim., Vol. 8, no. 3, pp. 631-657, 1998, https://doi.org/10.1137/S1052623496307510.

[6] H. Eschenauer, J. Koski, A. Osyczka, Multicriteria Design Optimization, 1st ed., Berlin, Germany, 1990, pp. 71-81, https://doi.org/10.1007/978-3-642-48697-5

[7] G. W. Evans, "An overview of techniques for solving multi-objective mathematical programs," Manag. Sci., Vol. 30, no. 11, pp. 1268–1282, 1984, doi: 10.1287/mnsc.30.11.1268.

[8] D. J. White, "Epsilon-dominating solutions in mean-variance portfolio analysis," European J. Oper. Res., Vol. 105, no. 3, pp. 457–466, 1998.

[9] Curtis, F. E. and Guo, W., "Handling nonpositive curvature in a limited memory steepest descent method," IMA J. of Nume. Anal., Vol. 36, no. 2, pp. 717-742, 2016, doi:10.1093/imanum/drv034.

[10] Drummond, L.G. and Svaiter, B. F., "A steepest descent method for vector optimization," J. of comp. and appl. Math., Vol. 175, no. 2, pp. 395-414, 2005, https://doi.org/10.1016/j.cam.2004.06.018.

[11] Shewchuk, J. R., An introduction to the conjugate gradient method without the agonizing pain, 1994

[12] Fletcher, R., "A limited memory steepest descent method," Math. Progr., Vol. 135, no. 1, pp. 413-436, 2012, https://doi.org/10.1007/s10107-011-0479-6

[13] Hamoda, M. A., Rivaie, M., and Mamat, M., "A new nonlinear conjugate gradient method with exact line search for unconstrained optimization," J. of Bas. Scie., Vol. 30, pp. 1-16 , 2017, https://doi.org/10.59743/jbs.v30i.74

[14] Guo, J., and Wan, Z., "A new three-term conjugate gradient algorithm with modified gradient-differences for solving unconstrained optimization problems," AIMS Math., Vol. 8, no. 2, pp. 2473-2488, 2023, doi:10.3934/math.2023128.

[15] Tang, J., and Wan, Z., "Orthogonal dual graph-regularized nonnegative matrix factorization for co-clustering," J. of Sci. Comput., Vol. 87, no. 3, pp. 66, 2021, https://doi.org/10.1007/s10915-021-01489-w

[16] Li, T., and Wan, Z., "New adaptive barzilai–borwein step size and its application in solving large-scale optimization problems," ANZIAM Journal, Vol. 61, no. 1, pp. 76-98, 2019, https://doi.org/10.1017/S1446181118000263 .

[17] Hu, C., Wan, Z. M., Zhu, S., and Wan, Z., "An integrated stochastic model and algorithm for constrained multi-item newsvendor problems by two-stage decision-making approach," Math. and Comp. in Simul., Vol. 193, pp. 280-300, 2022, https://doi.org/10.1016/j.matcom.2021.10.018.

[18] Wan, Z., and Yu, C., "Reutilization of the existent sales network for recycling unwanted smartphones by nonlinear optimization," J. of Clean. Prod., Vol. 350, 131349, 2022, https://doi.org/10.1016/j.jclepro.2022.131349.

[19] Huang, S., & Wan, Z., "A new nonmonotone spectral residual method for nonsmooth nonlinear equations," J. of

Comput. and Appl. Maths., Vol. 313, pp. 82-101, 2017, https://doi.org/10.1016/j.cam.2016.09.014.

[20] Deng, S., Wan, Z., and Chen, X., "An improved spectral conjugate gradient algorithm for nonconvex unconstrained optimization problems," J. of Optimization Theory and appl., Vol. 157, no. 3, pp. 820-842, 2013, http://doi.org/10.1007/s10957-012-0239-7.

[21] Hager, W. W., and Zhang, H., "A new conjugate gradient method with guaranteed descent and an efficient line search," SIAM J. on Optimi., Vol. 16, no. 1, pp. 170-192, 2005, http://doi.org/10.1137/030601880.

[22] Andrei, N., "A simple three-term conjugate gradient algorithm for unconstrained optimization", J. of Comp. and Appl. Math., Vol. 241, pp. 19-29, 2013, http://doi.org/10.1016/j.cam.2012.10.002.

[23] Andrei, N., "On three-term conjugate gradient algorithms for unconstrained optimization," Appl. Math. and Comp., Vol. 219, no. 11, pp. 6316-6327, 2013, http://doi.org/10.1016/j.amc.2012.11.097.

[24] Zhang, L., Zhou, W., and Li, D. H., "A descent modified Polak–Ribière–Polyak conjugate gradient method and its global convergence," IMA J. of Num. Analy., Vol. 26, no. 4, pp. 629-640, 2006, http://doi.org/10.1093/imanum/drl016.

[25] Salih, DTM, and Faraj, BM, "Comparison between steepest descent method and conjugate gradient method by using MATLAB," J. of Stud in Scie. and Engin., Vol. 1, no. 1, pp. 20-31, 2021. https://doi.org/10.53898/josse2021113.

[26] Mahdi, F. Z. A. A., "Investigation of three-term CG-algorithm for non-linear optimization," M.Sc. Thesis, University of Tikrit, Tikrit, Iraq, 2008.

[27] Muhammad, L., Waziri, M. Y., Sulaiman, I. M., Moghrabi, I. A., & Sambas, A., " improved preconditioned conjugate gradient method for unconstrained optimization problem with application in Robot arm control," Eng. Repo., Vol. 6, no. 2, 2024, https://doi.org/10.1002/eng2.12968.

[28] Abbas, I. T., Abd, E. M., & Mohsen, M. J. A., "Using gravitational search algorithm for solving nonlinear regression analysis," Irq. J. of Sci., Vol. 66, no. 3, pp. 1217-1231, 2025, https://doi.org/10.24996/ijs.2025.66.3.20.

[29] Abdulsattar, R., and Abass, I. T., "Tabu search algorithm for solving quadratic assignment problem," In AIP Conference Proceedings, 7 May 2024, https://doi.org/10.1063/5.0209862.

[30] Jasaim, S. H., and Abbas, I. T., "Using waiting queue models to study reducing the phenomenon of delay in one of the health institutions in Baghdad," Irq. J. of Sci., 2025, https://doi.org/10.24996/ijs.2025.66.8.27.

[31] David G. Luenberger Yinyu Ye, Llinear and nonlinear programming, 4th ed., Springer, 2016, pp. 1-11, https://doi.org/10.1007/978-3-030-85450-8

BIOGRAPHIES OF AUTHORS

	**Prof. Dr. Adawiya Ali Mahmood Al-Nuaimi** is Professor of Applied Mathematics at Department of Mathematics College of Science, University of Diyala, Iraq. She can be contacted at Dr.adawiya@uodiyala.edu.iq
	**Huda Habeeb Al-Zobiadi** is Master's student in Department of Mathematics, College of Science, University of Diyala. She received her B. Sc degree in Mathematical Sciences from the College of Science, University of Diyala, Iraq. She can be contacted at Scimathms242508@uodiyala.edu.iq