ZANCO Journal of Pure and Applied Sciences

TYPE Original article
DOI http://dx.doi.org/10.21271/ZJPAS.37.3.7

OPEN ACCESS

*Corresponding author
Basim A. Hassan
basimah@uomosul.edu.iq

RECEIVED :15 /11 /2024
ACCEPTED :08/03/ 2025
PUBLISHED :30/ 06/ 2025

KEYWORDS:
Optimization; conjugate gradient Method; Global Convergence; COVID-19.

# A Novel Conjugate Gradient Algorithm for Unconstrained Optimization and Its Application in COVID-19 Data Parameterization

Basim A. Hassan[1*], Ibrahim Sulaiman Mohammed[2,3] , Alaa Luqman Ibrahim[4], Faisal Falah Aiwa[1]

1Department of Mathematics, College of Computers Sciences and Mathematics University of Mosul, Mosul 41002, Iraq
2Institute of Strategic Industrial Decision Modelling, School of Quantitative Sciences, Universiti Utara Malaysia, Sintok 06010, Malaysia
3Faculty of Education and Arts, Sohar University, Sohar 311, Oman.
4Department of Mathematics, College of Science, University of Zakho, Zakho, Kurdistan Region Iraq.

## ABSTRACT

To solve unconstrained optimization problems, this study proposes a conjugate gradient (CG) algorithm that satisfies both the convergence and descent conditions. The technique improves on conventional CG techniques through guaranteeing quick convergence and increased solution accuracy across a range of test functions. The study also applies this CG method to model COVID-19 transmission dynamics within a parameterized optimization scope. Analyzing reported cases from January 28, 2024, to September 29, 2024, demonstrates the model's effectiveness in capturing nonlinear trends, with a resurgence of COVID-19 cases in the latter half of the study period. This emphasizes the need for adaptive public health strategies in response to fluctuating infection rates, highlighting the significance of advanced mathematical modeling in infectious disease management.

## 1.Introduction

In this study, the focus is on CG) methods applied to the nonlinear unconstrained optimization problem:

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

where $f: \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function bounded from below. Nonlinear CG methods generate a sequence $x_k$, for $k \geq 1$, beginning from an initial estimate $x_0 \in \mathbb{R}^n$, using the recurrence relation:

$$x_{k+1} = x_k + \alpha_k d_k \quad (2)$$

where the positive step size $\alpha_k$ is determined by a line search, and the search directions $d_k$ are generated by the following rule:

$$d_{k+1} = \begin{cases} -g_k, & k = 0 \\ -g_{k+1} + \beta_k d_k, & k \geq 1 \end{cases} \quad (3)$$

From (3), $\beta_k$ is the CG update parameter and $g_k = \nabla f(x_k)$, with $\nabla f(x_k)$ representing the gradient of $f$ at $x_k$ as a row vector, and $g_k$ as a column vector. Various CG methods differ primarily in their choices for the scalar $\beta_k$.

One of the efficient CG approaches was proposed by Polak, Ribière, and Polyak (PR) [1], [2], with the CG parameter defined as:

$$\beta_k^{PR} = \frac{g_{k+1}^T y_k}{\|g_k\|^2} \quad (4)$$

where $y_k = g_{k+1} - g_k$, and $||.||$ denotes the Euclidean norm. The PR method is considered one of the most effective CG methods for practical computation and has been widely studied. Polak and Ribière [1] demonstrated the global convergence of the PR method for strongly convex problems with exact line search. However, Powell [3] constructed an example where the PR method, even with exact line searches, produced a non-convergent sequence. To address this, Powell recommended restricting $\beta_k = \max\{\beta_k^{\mathrm{PR}}, 0\}$ to ensure convergence. Building on Powell's findings, Gilbert and Nocedal [4] proved that the PR method is globally convergent when $\beta_k^{\mathrm{PR}}$ is constrained to be nonnegative and the sufficient descent condition $g_k^T d_k \leq -c\|g_k\|^2$ holds during the Wolfe line search step.

In addition to their importance in optimization theory, CG methods, especially the PR variant, have important practical applications. They are a popular option for usage in machine learning, signal processing, and engineering design because of their effectiveness and scalability in large-scale challenges. In recent years, CG methods have expanded to various fields such as data estimation [5], image restoration [6], [7], and signal processing [8], [9]. As artificial intelligence grows in importance, CG methods have become essential for training neural networks, helping them learn effectively through gradient-based optimization [10]–[13].

The design of this paper is as follows: The developed CG method is presented in Section 2, along with information on its formulation, descent conditions, and global convergence proof. Numerical results showing the effectiveness of the method in solving unconstrained optimization problems are presented in Section 3. The CG method's practical applicability is demonstrated in Section 4 by applying it to a parameterized COVID-19 model. A summary and closing thoughts are provided in Section 5.

## 2.Modified conjugate gradient

The primary objective of this paper is to derive new formulae for a modified conjugate gradient method. By applying the second-order Taylor series expansion to the function $f$, we have:

$$f(x) = f(x_{k+1}) - g_{k+1}^T s_k + \frac{1}{2} s_k^T Q(u_{k+1}) s_k, \quad (5)$$

The gradient at $x_{k+1}$, becomes:

$$g_{k+1} = g_k + Q(u_{k+1}) s_k, \quad (6)$$

The second-order curvature is derived from (6) in (5), and $y_k$ is determined using algebraic manipulation as follows:

$$s_k^T Q(u_k) s_k = (g_k^T s_k)^2 / (2y_k^T s_k + 2(f_k - f_{k+1}) + 2g_k^T s_k), \quad (7)$$

To derive a new CG parameter, the conjugate condition is employed. This condition is given by:

$$d_{k+1}^T Q(u_k) s_k = 0, \quad (8)$$

By combining equations (3), (7), and equation (8), we derive the new parameter $\beta_k$ as follows:

$$\beta_k = \frac{\left[\frac{(g_k^T s_k)^2}{s_k^T y_k (2y_k^T s_k + 2(f_k - f_{k+1}) + 2g_k^T s_k)}\right] g_{k+1}^T y_k}{d_k^T y_k}, \quad (9)$$

which, under exact line search conditions, simplifies to:

$$\beta_k = \frac{\left[\frac{(g_k^T s_k)^2}{s_k^T y_k (2y_k^T s_k + 2(f_k - f_{k+1}) + 2g_k^T s_k)}\right] \|g_{k+1}\|^2}{d_k^T y_k}, \quad (10)$$

This new formula introduces an adjustment to the

traditional CG formula that potentially enhances convergence properties under specific conditions. Below is an outline of the new CG algorithm based on this parameter.

**Algorithm: Implementation of the New formula.**

**Step 1 :** (Initialization) Given an initial point $w_0 \in R^n$ , parameters $0 < \delta < \sigma < 1$, and $\varepsilon > 0$. Set $d_0 = -g_0$, and $k = 0$.

**Step 2 :** If $\|g_k\| \leq \varepsilon$ then stop.

**Step 3 :** Compute the step size $\alpha_k$ by the weak Wolfe line search,

**Step 4 :** Generate the next iteration by (2).

**Step 5 :** compute $d_{k+1}$ by (3) and choose an appropriate conjugate parameter $\beta_k$ by (10).

**Step 6 :** Set $k := k + 1$ and go to Step 1.

The following theorem establishes the descent condition for the algorithm, providing a foundational result for its convergence analysis.

**Theorem 1:** The New algorithm generates search directions $d_{k+1}$ that satisfy the following conditions:

$d_{k+1}^T g_{k+1} < 0$ and $d_{k+1}^T g_{k+1} = \beta_k d_k^T g_k$. (11)

**Proof:** Obviously, $d_k = -g_k$ is required for $d_1^T g_1 < 0$. For any $k$, the $d_k^T g_k < 0$ should be considered. One can easily acquire it from (3) and (10):

$d_{k+1}^T g_{k+1} = -g_{k+1}^T g_{k+1} + \beta_k d_k^T g_{k+1}$,

Substituting for $\beta_k$ from equation (10), we get:

$$d_{k+1}^T g_{k+1} = -\beta_k \frac{s_k^T y_k}{\left[\frac{(g_k^T s_k)^2}{s_k^T y_k (2y_k^T s_k + 2(f_k - f_{k+1}) + 2g_k^T s_k)}\right]} + \beta_k d_k^T g_{k+1}, \quad (12)$$

This yields the relation:

$$d_{k+1}^T g_{k+1} = \beta_k [d_k^T g_{k+1} - \frac{s_k^T y_k}{\left[\frac{(g_k^T s_k)^2}{s_k^T y_k (2y_k^T s_k + 2(f_k - f_{k+1}) + 2g_k^T s_k)}\right]}, \quad (13)$$

By using (6) and (7) in (13), we obtain:

$d_{k+1}^T g_{k+1} = \beta_k d_k^T g_k$, (14)

since $d_k^T g_k < 0$, it follows directly that:

$d_{k+1}^T g_{k+1} < 0$. (15)

Thus, the descent condition is satisfied, and the proof is complete.

To analyze the algorithm's global convergence properties. Initially, we establish the following:

1. On the given set $\Omega = \{u: u \in R^n, f(u) \leq f(u_1)\}$, $f(u)$ is bounded from below.
2. The following inequality is met by $L > 0$, $\tau, v \in R^n$ since the derivative $\nabla f(u)$ is Lipschitz continuous:
3. $\|g(\tau) - g(v)\| \leq L\|\tau - v\|, \forall \tau, v \in R^n$ , (16)

See, [14]. In order to analyze the overall convergence properties of the conjugate gradient method, a comprehension of the Zoutendijk condition [15] is necessary.

**Lemma1:** If we assume that both (1) and (2) are true, that $\alpha_k$ fulfill the Wolfe conditions, as well as $d_k$ being descent direction, then:

$\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty$. (17)

**Theorem 2:** Assuming the premises and lemma 1 are true and $\{u_k\}$ is a new sequence, then:

$\lim_{k\to\infty} (inf \|g_k\|) = 0$. (18)

**Proof:** Equation (18) is untrue by contradiction. We can identify a $r > 0$ such that, for each $k$:

$\|g_{k+1}\| > r$, (19)

One can get the following outcome by squaring the search duration as $d_{k+1} + g_{k+1} = \beta_k d_k$ on both sides:

$\|d_{k+1}\|^2 + \|g_{k+1}\|^2 + 2d_{k+1}^T g_{k+1} = (\beta_k)^2 \|d_k\|^2$. (20)

When (14) is applied to (20), the results are obtained:

$\|d_{k+1}\|^2 = \frac{(d_{k+1}^T g_{k+1})^2}{(d_k^T g_k)^2} \|d_k\|^2 - 2d_{k+1}^T g_{k+1} - \|g_{k+1}\|^2$. (21)

Divided (21) by $(d_{k+1}^T g_{k+1})^2$, then result is:

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

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

$$\leq \frac{\|d_k\|^2}{(d_k^T g_k)^2} + \frac{1}{\|g_{k+1}\|^2} \quad , \quad (22)$$

As a result, we found:

$\frac{\|d_{k+1}\|^2}{(d_{k+1}^T g_{k+1})^2} \leq \sum_{i=1}^{k+1} \frac{1}{\|g_i\|^2}$. (23)

Suppose $c_1 > 0$ has $\|g_k\| \geq c_1$ for every $k \in n$. Then:

$\frac{\|d_{k+1}\|^2}{(d_{k+1}^T g_{k+1})^2} < \frac{k+1}{c_1^2}$. (24)

Ultimately, we have:

$\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} = \infty.$ (25)

Similarly, $\lim_{k\to\infty} inf \|g_k\| = 0$ holds according to Lemma 1.

## 3.Numerical Results

This part assesses the performance of the suggested algorithm using a variety of functions from the CUTE library [16] as well as other sets of unconstrained optimization problems [17], [18] to cover a range of situations and dimensions. We conducted the experiments on an HP laptop using MATLAB 2013b.

To verify the reliability of the proposed algorithm, it was compared with the established PR conjugate gradient method, using the same test functions for consistency. Both algorithms applied Wolfe line search conditions, with parameters set to $\delta = 0.01$ and $\sigma = 0.3$ . The stopping criteria included: $\| g_k \| < 10^{-6}$, iteration exceeding 2000, or an algorithm attaining a computation time limit of 500 seconds.

This configuration was selected to clearly reveal performance differences between the proposed algorithm and the PR conjugate gradient method. Table 1 provides a comparison in terms of the number of iterations (NI), function evaluations (NF), and CPU time **(CPUT).** Furthermore, performance profiles, based on the Dolan and Moré method [19], are shown in Figures 1, 2, and 3, offering a visual assessment of computational efficiency and demonstrating the proposed approach's advantages across varied optimization contexts.

**Table 1:** Numerical Result.

| Function | N | PR | | | New | | |
|---|---|---|---|---|---|---|---|
| | | NI | NF | TCPU | NI | NF | TCPU |
| cosine | 500 | 109 | 265 | 9.65E-07 | 20 | 75 | 9.62E-08 |
| cosine | 1000 | NaN | NaN | NaN | 18 | 83 | 2.84E-07 |
| cosine | 5000 | NaN | NaN | NaN | 28 | 91 | 9.56E-07 |
| cosine | 10000 | NaN | NaN | NaN | 16 | 84 | 9.06E-07 |
| dixmaana | 1500 | 14 | 75 | 4.96E-07 | 15 | 72 | 6.76E-07 |
| dixmaana | 3000 | 15 | 79 | 1.93E-07 | 12 | 75 | 6.68E-07 |
| dixmaana | 15000 | 18 | 83 | 6.99E-07 | 16 | 81 | 5.13E-07 |
| dixmaana | 30000 | 17 | 86 | 6.28E-07 | 19 | 89 | 4.62E-07 |
| dixmaanb | 1500 | 18 | 81 | 6.33E-07 | 17 | 82 | 7.48E-07 |
| dixmaanb | 3000 | 18 | 79 | 5.19E-07 | 18 | 80 | 3.86E-07 |
| dixmaanb | 15000 | 24 | 89 | 2.14E-07 | 18 | 86 | 5.32E-08 |
| dixmaanb | 30000 | 17 | 85 | 4.70E-07 | 18 | 92 | 4.50E-08 |
| dixmaanc | 1500 | 21 | 85 | 5.94E-07 | 15 | 77 | 5.98E-07 |
| dixmaanc | 3000 | 22 | 86 | 9.04E-07 | 18 | 80 | 3.76E-07 |
| dixmaanc | 15000 | 23 | 91 | 4.97E-07 | 17 | 83 | 4.94E-07 |
| dixmaand | 30000 | 31 | 105 | 4.56E-07 | 21 | 102 | 1.01E-07 |
| dixmaand | 1500 | 15 | 81 | 2.49E-07 | 17 | 81 | 9.85E-07 |
| dixmaand | 3000 | 61 | 134 | 7.90E-07 | 20 | 85 | 9.44E-07 |
| dqrtic | 500 | 38 | 125 | 9.08E-08 | 37 | 118 | 3.30E-07 |
| dqrtic | 1000 | 39 | 128 | 7.32E-07 | 34 | 124 | 1.04E-07 |
| dqrtic | 5000 | 61 | 173 | 2.43E-07 | 40 | 144 | 9.58E-08 |
| dqrtic | 10000 | 93 | 216 | 5.49E-07 | 68 | 205 | 3.48E-07 |
| edensch | 500 | 58 | 216 | 8.20E-07 | 77 | 433 | 7.26E-07 |
| edensch | 1000 | 722 | 3522 | 9.71E-07 | 57 | 206 | 7.45E-07 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| edensch | 5000 | 183 | 968 | 8.73E-07 | 49 | 150 | 9.68E-07 |
| edensch | 10000 | 75 | 407 | 9.41E-07 | 92 | 453 | 6.81E-07 |
| fletchcr | 1000 | 84 | 450 | 5.86E-07 | 137 | 229 | 7.86E-07 |
| fletchcr | 5000 | 218 | 1919 | 7.24E-07 | 114 | 213 | 8.10E-07 |
| fletchcr | 10000 | 268 | 2220 | 8.90E-07 | 190 | 336 | 2.75E-07 |
| fletchcr | 12000 | 1666 | 16274 | 8.99E-07 | 188 | 308 | 2.30E-07 |
| himmelbg | 500 | 2 | 9 | 3.12E-31 | 2 | 9 | 3.09E-31 |
| himmelbg | 1000 | 2 | 13 | 7.57E-30 | 2 | 13 | 7.57E-30 |
| himmelbg | 5000 | 2 | 13 | 3.40E-28 | 2 | 13 | 3.39E-28 |
| himmelbg | 10000 | 2 | 17 | 5.67E-47 | 2 | 17 | 5.67E-47 |
| penalty1 | 500 | 18 | 87 | 5.84E-07 | 18 | 88 | 8.93E-07 |
| penalty1 | 10000 | 21 | 96 | 1.82E-07 | 18 | 92 | 7.13E-08 |
| quartc | 500 | 38 | 125 | 9.08E-08 | 37 | 118 | 3.30E-07 |
| quartc | 1000 | 39 | 128 | 7.32E-07 | 34 | 124 | 1.04E-07 |
| quartc | 5000 | 61 | 173 | 2.43E-07 | 40 | 144 | 9.58E-08 |
| quartc | 10000 | 93 | 216 | 5.49E-07 | 68 | 205 | 3.48E-07 |
| bdexp | 500 | 2 | 7 | 2.12E-108 | 2 | 7 | 1.96E-108 |
| bdexp | 1000 | 2 | 11 | 3.24E-133 | 2 | 11 | 3.21E-133 |
| bdexp | 5000 | 2 | 11 | 4.40E-111 | 2 | 11 | 4.39E-111 |
| bdexp | 10000 | 2 | 17 | 2.63E-61 | 2 | 17 | 2.63E-61 |
| exdenschnf | 500 | 66 | 137 | 8.50E-07 | 26 | 95 | 4.24E-07 |
| exdenschnf | 1000 | 70 | 143 | 7.48E-07 | 20 | 90 | 6.55E-07 |
| exdenschnf | 5000 | 50 | 127 | 5.42E-07 | 24 | 99 | 1.43E-07 |
| exdenschnf | 10000 | 40 | 114 | 1.54E-07 | 21 | 96 | 8.39E-07 |
| exdenschnb | 500 | 26 | 85 | 4.26E-07 | 15 | 74 | 5.27E-07 |
| exdenschnb | 1000 | 15 | 77 | 9.83E-08 | 18 | 80 | 1.93E-08 |
| exdenschnb | 5000 | 32 | 97 | 7.20E-07 | 20 | 84 | 3.79E-08 |
| exdenschnb | 10000 | 38 | 106 | 7.58E-07 | 17 | 82 | 8.68E-07 |
| genquartic | 500 | 114 | 182 | 9.55E-07 | 23 | 83 | 6.85E-07 |
| genquartic | 1000 | 580 | 697 | 9.56E-07 | 21 | 89 | 7.91E-07 |
| genquartic | 5000 | 20 | 83 | 3.62E-07 | 15 | 77 | 1.99E-08 |
| genquartic | 10000 | 54 | 125 | 7.23E-07 | 20 | 83 | 9.65E-07 |
| biggsb1 | 4 | 34 | 89 | 9.37E-07 | 30 | 71 | 7.30E-07 |
| biggsb1 | 10 | 59 | 109 | 8.82E-07 | 133 | 222 | 7.97E-07 |
| sine | 1000 | 61 | 124 | 1.00E-06 | 19 | 75 | 1.90E-07 |
| sine | 3000 | NaN | NaN | NaN | 693 | 1004 | 2.73E-07 |
| sine | 4000 | NaN | NaN | NaN | 29 | 96 | 4.64E-07 |
| fletcbv3 | 10 | 476 | 598 | 9.99E-07 | 54 | 95 | 5.10E-07 |
| fletcbv3 | 50 | NaN | NaN | NaN | 1956 | 3449 | 1.58E-07 |
| fletcbv3 | 100 | NaN | NaN | NaN | 1357 | 2580 | 5.41E-07 |
| nonscomp | 500 | NaN | NaN | NaN | NaN | NaN | NaN |
| nonscomp | 1000 | NaN | NaN | NaN | 83 | 170 | 8.00E-07 |
| nonscomp | 5000 | NaN | NaN | NaN | 64 | 138 | 8.24E-07 |
| nonscomp | 8000 | NaN | NaN | NaN | NaN | NaN | NaN |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| power1 | 4 | 93 | 190 | 8.87E-07 | 88 | 142 | 8.04E-07 |
| power1 | 10 | 212 | 306 | 9.90E-07 | 367 | 550 | 8.42E-07 |
| raydan2 | 500 | 13 | 68 | 2.72E-08 | 13 | 66 | 8.85E-07 |
| raydan2 | 1000 | 17 | 81 | 4.90E-07 | 17 | 81 | 4.90E-07 |
| raydan2 | 5000 | 12 | 81 | 9.73E-07 | 14 | 103 | 6.81E-07 |
| raydan2 | 10000 | 17 | 101 | 3.82E-07 | 12 | 86 | 4.41E-07 |
| diagonal1 | 4 | 29 | 73 | 9.90E-07 | 24 | 69 | 5.66E-07 |
| diagonal1 | 10 | 51 | 105 | 9.93E-07 | 51 | 102 | 9.36E-07 |
| diagonal1 | 50 | 336 | 882 | 9.46E-07 | 117 | 232 | 8.77E-07 |
| diagonal1 | 100 | 454 | 2281 | 9.56E-07 | 283 | 849 | 9.88E-07 |
| diagonal3 | 4 | 36 | 80 | 8.76E-07 | 26 | 71 | 5.80E-07 |
| diagonal3 | 10 | 98 | 149 | 9.60E-07 | 50 | 93 | 5.80E-07 |
| diagonal3 | 50 | 150 | 380 | 8.73E-07 | 188 | 298 | 9.11E-07 |
| diagonal3 | 100 | 378 | 2057 | 9.83E-07 | 372 | 642 | 9.43E-07 |
| bv | 1000 | NaN | NaN | NaN | 320 | 480 | 8.99E-07 |
| ie | 10 | 13 | 52 | 4.33E-07 | 16 | 72 | 7.67E-07 |
| ie | 100 | 17 | 60 | 2.39E-07 | 15 | 75 | 3.36E-07 |
| ie | 500 | 15 | 60 | 7.16E-07 | 17 | 61 | 9.34E-07 |
| lin | 10 | 12 | 71 | 6.56E-07 | 12 | 71 | 6.56E-07 |
| lin | 100 | 13 | 63 | 8.84E-07 | 13 | 63 | 8.84E-07 |
| lin | 500 | 18 | 82 | 4.63E-07 | 18 | 82 | 4.63E-07 |

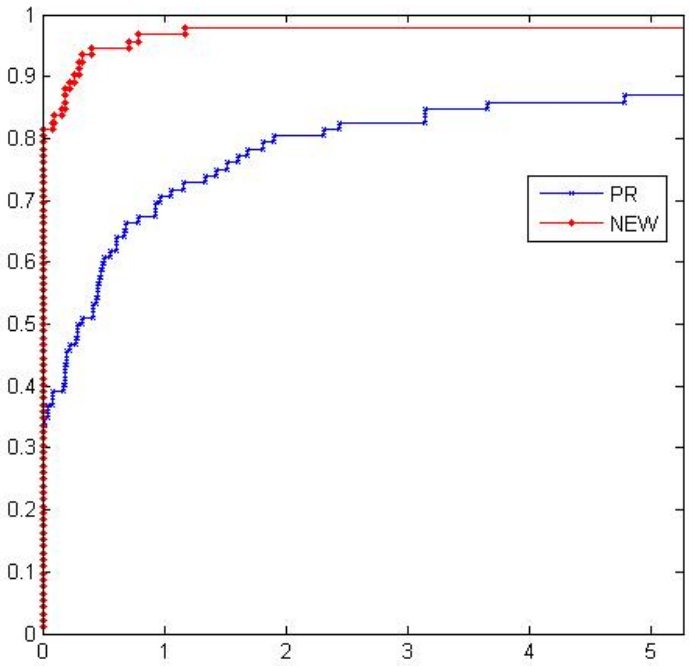


Figure 1: Iteration number.

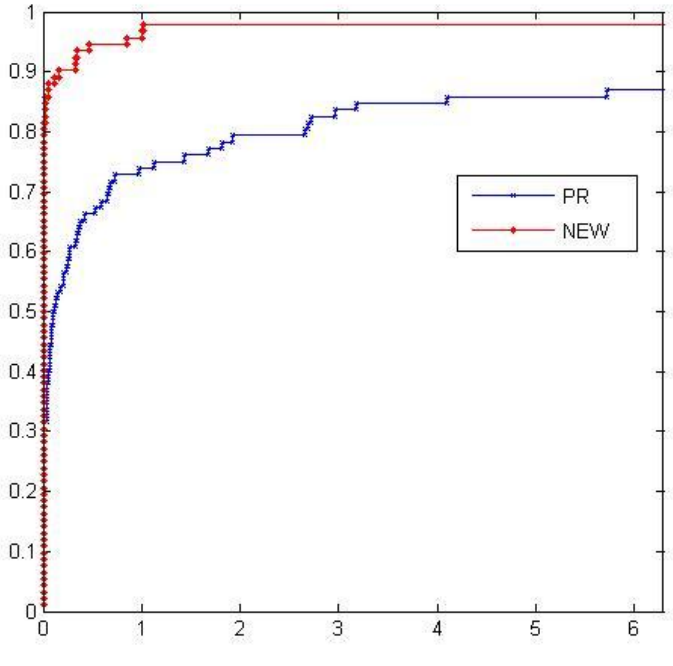


Figure 2: Function evaluation

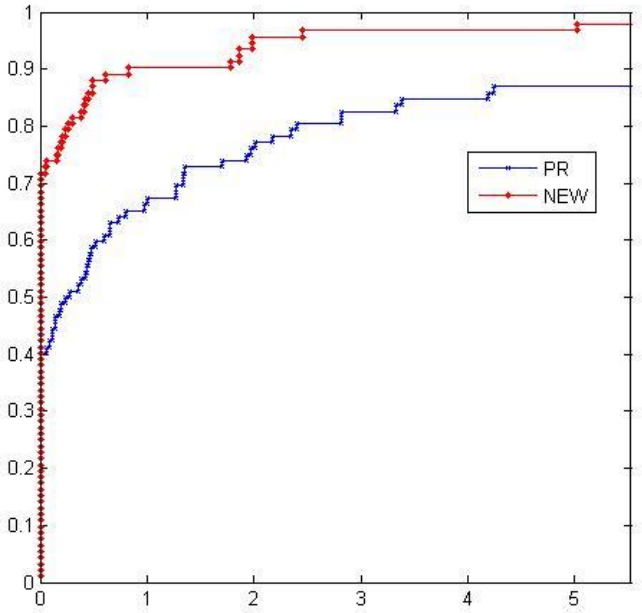


Figure 3: CPU Time.

## 4.Application of the Conjugate Gradient Method to a Parameterized COVID-19 Model

Coronavirus disease (COVID-19), whose first case was recorded in late 2019, has continued to influence society and global health despite the numerous waves of infection. This disease, caused by the SARS-CoV-2 virus, spreads mainly via respiratory droplets when an affected individual sneezes, coughs, or talks. Severe cases of this infection may lead to death, in addition to the mild to moderate symptoms, such as breathing difficulties, cough, and fever, experienced from infected individuals. Until date, COVID-19 remains a major health concern in several parts of the world despite widespread vaccination efforts. This might be due to the emergence of new variants and ongoing mutations.

With nearly 7 million death rates from over 770 million confirmed cases as at mid-2024, based on World Health Organization (WHO) [20], this virus continues to pose concerns for public health systems globally. The developing nature of the deadly disease has inspired extensive study into modelling the transmission dynamics of the disease, which has been very imperative for the prediction of future occurrences and supporting public health strategies.

This paper considers the global trends of COVID-19 from 28 of January 2024 through 29 September 2024, constructing a parameterized model using the number of cases reported to WHO during this period. This data represents the latest epidemiological data and trends. This model is formulated as an unconstrained optimization function, where the proposed CG formula will be employed for solve the parameterized model.

Consider the following regression analysis function:

$$y = f(x_1, x_2, \dots, x_p) + \varepsilon \quad (26)$$

where $y$ is the response variable, $x_i, i = 1,2, \dots, p, p > 0$ represents the predictors, and $\varepsilon$ is the error term. Problems of the form (26) frequently emerge in fields like physics, accounting, economics, finance, management, and other domains. Regression analysis serves as a powerful statistical method used to model and examine the relationships between a dependent variable and one or more independent variables [21]. To derive a linear regression model, we calculate $y$ as:

$$y = b_0 + b_1 x_1 + b_2 x_2 + \cdots + b_p x_p + \varepsilon \quad (27)$$

with the regression parameters coefficients given

as $b_0, \ldots, b_p$ . Regression analysis aims to estimate these parameters, $b_0, \ldots, b_p$ such that the error term $\varepsilon$ is minimized. In some cases, the relationship between the predictors "$x$" and dependent variable "$y$" can be approximated by a straight line, which is the foundation of linear regression. However, in practice, linear models may not always precisely attain the underlying relationships. Therefore, nonlinear regression schemes are often utilized to address these more complex relationships. This study will focus on applying nonlinear regression techniques to attain a better model fit.

To construct the approximate function, the study utilizes data on global COVID-19 reported cases from 28 January – 29 September 2024. A detailed explanation of the procedure is presented, drawing from the statistical information outlined in Table 2, which is based on data obtained from the WHO [1]. The dataset spans a period of nine months, with the months denoted by $x$ -variable, and the corresponding reported cases represented by the $y$-variable.

Table 2: Statistics of monthly COVID-19 reported cases (Jan 28–Sept 29, 2024)

| Months (Jan–Sept) $(x)$ | Date of recording data | Data of reported cases in hundreds $(y)$ | Statistics of reported cases in $(\%)$ |
|---|---|---|---|
| 1 | Jan 28 | 116,000 | 18.4 |
| 2 | Feb 25 | 76,600 | 12.1 |
| 3 | March 31 | 114,000 | 18.1 |
| 4 | April 28 | 34,800 | 5.5 |
| 5 | May 26 | 32,600 | 5.2 |
| 6 | June 30 | 48,300 | 7.7 |
| 7 | July 28 | 59,400 | 9.4 |
| 8 | August 25 | 63,600 | 10.1 |
| 9 | September 29 | 85,600 | 13.6 |

Using data presented in Table 2, the study derives the following approximate function for the nonlinear least square method:

$$y = 3435.39x^2 - 38625.56x + 154440.48 \quad (27)$$

Next, the study utilized function (27) to approximate the values of $y$ based on $x$ from the given data. Let the number of months be denoted by $x_j$ and the corresponding reported cases by $y_j$ . Then the least squares problem given in function (27) is reformulated into the following unconstrained minimization model:

$$\min_{x \in R^n} f(x) = \sum_{j=1}^{n} \left( \left( w_0 + w_1 x_j + w_2 x_j^2 \right) - y_j \right)^2 \quad (28)$$

where $w_0, w_1,$ and $w_2$ defined the regression parameter. Obviously, the above formulated problems exhibit the same characteristics of the typical unconstrained optimization model where the data $x_j$ and the value of $y_j$ possess some parabolic relations with $w_0, w_1,$ and $w_2$ and the regression function (28) and thus, can be re-written as:

$$\min_{x \in R^n} \sum_{j=1}^{n} E_j^2 = \sum_{j=1}^{n} \left( \left( w_0 + w_1 x_j + w_2 x_j^2 \right) - y_j \right)^2 \quad (29)$$

Now, based on the data from Table 2, the study can transform (29) into the following nonlinear quadratic unconstrained minimization model:

$$\begin{aligned} & 9w_0^2 + 90w_0w_1 + 570w_0w_2 \\ & - 1261800w_0 + 285w_1^2 \\ & + 4050w_1w_2 - 5796400w_1 \\ & + 15333w_2^2 - 36947200w_2 \\ & + 51826930000 \end{aligned} \quad (30)$$

Table 3: Performance of the algorithms on the constructed nonlinear least square problem (27)

| | | NEW | | | PR | | |
|---|---|---|---|---|---|---|---|
| Initial Points | Dim | NOI | CPU | NOF | NOI | CPU | NOF |
| (0.5, …, 0.5) | 2 | 25 | 0.001591 | 77 | 23 | 0.001278 | 74 |
| | 40 | 50 | 0.002100 | 106 | 136 | 0.002983 | 186 |
| | 10 | 38 | 0.001644 | 92 | 24 | 0.001143 | 64 |
| | 100 | 136 | 0.003654 | 194 | 61 | 0.001958 | 113 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| (2, …, 2) | 2 | 21 | 0.001373 | 73 | 21 | 0.001235 | 73 |
| | 4 | 28 | 0.002439 | 78 | 28 | 0.002469 | 78 |
| | 10 | 42 | 0.001726 | 94 | 28 | 0.002196 | 74 |
| | 100 | 173 | 0.005012 | 226 | 220 | 0.005019 | 276 |
| (3, …, 3) | 2 | 26 | 0.001235 | 76 | 26 | 0.001289 | 77 |
| | 4 | 24 | 0.001619 | 76 | 27 | 0.001345 | 78 |
| | 10 | 37 | 0.002645 | 91 | 42 | 0.003272 | 95 |
| | 100 | 308 | 0.007524 | 386 | 184 | 0.004523 | 240 |
| (11, …, 11) | 2 | 32 | 0.002222 | 87 | 32 | 0.002466 | 83 |
| | 4 | 36 | 0.002845 | 93 | 36 | 0.002132 | 87 |
| | 10 | 50 | 0.002139 | 102 | 50 | 0.003143 | 105 |
| | 100 | 214 | 0.007724 | 273 | 229 | 0.005184 | 281 |
| (5, …, 5) | 2 | 26 | 0.001254 | 74 | 25 | 0.001263 | 70 |
| | 4 | 32 | 0.001744 | 84 | 34 | 0.001534 | 86 |
| | 10 | 48 | 0.001640 | 104 | 47 | 0.001658 | 99 |
| | 100 | 234 | 0.005361 | 289 | 84 | 0.002189 | 131 |

The results in Table 3 compare the performance of the NEW algorithm and classical PR formula, on a nonlinear least squares problem (27) using different dimensions and initial points. The result shows the competitive performance of the proposed CG algorithm majority configurations, particularly higher dimensions and with smaller initial points. This demonstrates the efficiency and better scalability of the proposed algorithm in most cases.

***Trend line method***

A trend line is a line drawn beneath pivot lows or above pivot highs to indicate the general direction of price movement. In this section, the COVID-19 reported data was analyzed over a nine-month period using the proposed CG formula. The trend line is generated with Microsoft Excel, based on the data from Table 2. The equation of the trend line takes the form of a nonlinear quadratic equation. Representing the $y$ -axis by $y$ and the $x$-axis by $x$, the plot in Fig. 4 is produced using the actual data from Table 2.

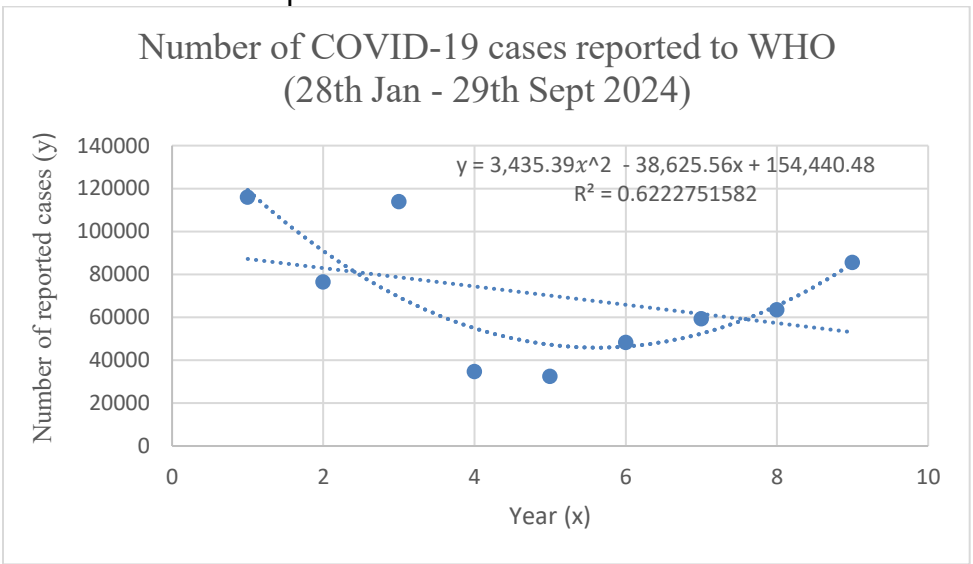


Figure 4: Nonlinear quadratic trend line for COVID-19 reported cases

The analysis of COVID-19 reported cases from 28 Jan to 29 Sept 2024, as illustrated in Table 2 and portrayed in Figure 4, shows a nonlinear quadratic trend. In the beginning, the cases dropped from 116,000 by 28$^{th}$ Jan to 76,600 as recorded in Feb, reflecting a significant drop.

However, by 31st of March, the trend reverses with a resurgence to 114,000 reported cases. Following this spike, the cases reduced again, attaining a low by 26th of May with only 32,600 reported cases. The quadratic trend line in Figure 4 captures these instabilities.

However, the trend line curves upward from June onwards, which reflects a steady increase in the reported cases, from 48,300 by 30th of June to 85,600 by 29th of September 2024. This upward curve demonstrates a resurgence in reported cases of COVID-19 virus during the second half of the considered period. The trend line in Figure 4 clearly shows these cyclical patterns, demonstrating that virus case numbers are not following a linear path but instead experience waves of fall and rise over time.

The COVID-19 model provides valuable insights for public health policy by identifying trends, forecasting outbreaks, and evaluating intervention effectiveness. By analyzing reported cases over time, policymakers can assess the impact of preventive measures such as lockdowns, vaccination campaigns, and social distancing policies. The proposed model aids in resource allocation, helping governments optimize healthcare capacity, medical supplies, and workforce distribution. Additionally, it supports early warning systems, enabling timely responses to emerging threats. However, effective strategy formulation requires continuous model refinement to account for evolving virus variants, public compliance, and socioeconomic factors influencing disease spread.

## 5.Conclusion

In conclusion, this study has reviewed a range of methods in the field of optimization, focusing on the derived a new CG algorithm with applications. The new algorithm satisfies both the convergence and descent conditions. Performance comparisons with conventional techniques were conducted to evaluate the effectiveness of the proposed methods. The experimental results indicated that the new techniques not only enhance the accuracy of the solutions obtained but also accelerate convergence. Additionally, this study employed a parameterized optimization framework to model the transmission dynamics of COVID-19 using the new CG Method. The findings underscore the significance of mathematical modeling in understanding the dynamics of infectious diseases and highlight the necessity for flexible public health measures in response to fluctuating infection rates.

## References


[1] E. Polak and G. Ribiere, “Note sur la convergence de méthodes de directions conjuguées,” *Revue française d’informatique et de recherche opérationnelle. Série rouge*, vol. 3, no. R1, pp. 35–43, 1969.

[2] Basim A. Hassan and Haneen A. Alashoor, (2023), On image restoration problems using new conjugate gradient methods, Indonesian Journal of Electrical Engineering and Computer Science Vol. 29, No. 3, March 2023, pp. 1438-1445.

[3] M. J. D. Powell, “Nonconvex minimization calculations and the conjugate gradient method,” in *Numerical Analysis: Proceedings of the 10th Biennial Conference held at Dundee, Scotland, June 28–July 1, 1983*, Springer, 1984, pp. 122–141.

[4] J. C. Gilbert and J. Nocedal, “Global Convergence Properties of Conjugate Gradient Methods for Optimization,” *SIAM Journal on Optimization*, vol. 2, no. 1, pp. 21–42, 1992, doi: 10.1137/0802003.

[5] S. Shoid *et al.*, “The application of new conjugate gradient methods in estimating data,” *International Journal of Engineering and Technology(UAE)*, vol. 7, no. 2, pp. 25–27, 2018, doi: 10.14419/ijet.v7i2.14.11147.

[6] A. Alhawarat, Z. Salleh, H. Alolaiyan, H. El Hor, and S. Ismail, “A three-term conjugate gradient descent method with some applications,” *Journal of Inequalities and Applications*, vol. 2024, no. 1, 2024, doi: 10.1186/s13660-024-03142-0.

[7] Z. Chen, H. Shao, P. Liu, G. Li, and X. Rong, “An efficient hybrid conjugate gradient method with an adaptive strategy and applications in image restoration problems,” *Applied Numerical Mathematics*, vol. 204, pp. 362–379, 2024, doi: https://doi.org/10.1016/j.apnum.2024.06.020.

[8] Basim A. Hassan and Hameed M. Sadiq, (2022), A new formula on the conjugate gradient method for removing impulse noise images, Bulletin of the South Ural State University. Ser. Mathematical Modelling, Programming & Computer Software (Bulletin SUSU MMCS), 2022, vol. 15, no. 4, pp. 123-130.

[9] G. Abbass, H. Chen, M. Abdullahi, and A. B. Muhammad, “An efficient projection algorithm for large-scale system of monotone nonlinear equations with applications in signal recovery,” *Journal of Industrial and Management Optimization*, vol. 0, no. 0, pp. 0–0, 2024, doi: 10.3934/jimo.2024066.

[10] W. Zhao and H. Huang, “Adaptive stepsize estimation based accelerated gradient descent

algorithm for fully complex-valued neural networks," *Expert Systems with Applications*, vol. 236, p. 121166, 2024, doi: https://doi.org/10.1016/j.eswa.2023.121166.

[11] H. Iiduka and Y. Kobayashi, "Training deep neural networks using conjugate gradient-like methods," *Electronics (Switzerland)*, vol. 9, no. 11, pp. 1–25, 2020, doi: 10.3390/electronics9111809.

[12] Basim A. Hassan and Hameed M. Sadiq, (2022), Efficient New Conjugate Gradient Methods for Removing Impulse Noise Images, Europear Journal of Pure and Applied Mathematics, Vol. 15, No. 4, 2011-2021.

[13] A. L. Ibrahim and M. G. Mohammed, "A new conjugate gradient for unconstrained optimization problems and its applications in neural networks," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 33, no. 1, pp. 93–100, 2024, doi: 10.11591/ijeecs.v33.i1.pp93-100.

[14] B. A. Hassan and A. Ahmed A. Abdulla, "Improvement of conjugate gradient methods for removing impulse noise images," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 29, p. 245, 2022, doi: 10.11591/ijeecs.v29.i1.pp245-251.

[15] G. Zoutendijk, "Nonlinear Programming Computational Methods," *In: Abadie, J. Ed., Integer and Nonlinear Programming, NorthHolllad, Amsterdam,* pp. 37-86., 1970.

[16] N. I. M. Gould, D. Orban, and P. L. Toint, "CUTEr and sifdec: A constrained and unconstrained testing environment, revisited," *ACM Transactions on Mathematical Software*, vol. 29, no. 4, pp. 373–394, 2003, doi: 10.1145/962437.962439.

[17] J. J. Moré, B. S. Garbow, and K. E. Hillstrom, "Testing Unconstrained Optimization Software," *ACM Transactions on Mathematical Software (TOMS)*, vol. 7, no. 1, pp. 17–41, 1981, doi: 10.1145/355934.355936.

[18] N. Andrei, "An unconstrained optimization test functions collection," *Advanced Modelling and Optimization*, vol. 10, no. 1, pp. 147–161, 2008.

[19] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002, doi: 10.1007/s101070100263.

[20] World Health Organization: https://data.who.int/dashboards/covid19/cases.

[21] I. M. Sulaiman, M. Malik, A. M. Awwal, P. Kumam, M. Mamat, and S. Al-Ahmad, "On three-term conjugate gradient method for optimization problems with applications on COVID-19 model and robotic motion control.," *Advances in continuous and discrete models*, vol. 2022, no. 1, p. 1, 2022, doi: 10.1186/s13662-021-03638-9.