



Research Article

Integration of Modified Classical Conjugate Gradient Methods for Unconstrained Optimization

¹Oluwaseun B. Onuoha, ¹Yoyinade J. Aborisade, ²Grace Egenti and ³Rasheed O. Ayinla

¹Department of Mathematical Sciences, Adekunle Ajasin University, Akungba Akoko, Nigeria

²ACETEL, National Open University of Nigeria

³Department of Mathematics and Statistics, Kwara State University, Malet, Nigeria

*Corresponding author's Email: oluwaseun.akinduko@aau.edu.ng, doi.org/10.55639/607.474645

ARTICLE INFO:

Keywords:

Optimization problems,
Performance tests,
Convergence rates,
Computational
efficiency

ABSTRACT

The integration of modified classical conjugate gradient methods (CGMs) for unconstrained optimization represents a crucial and evolving area of research within the field of optimization algorithms. Over time, numerous studies have put forth diverse modifications and novel approaches to enhance the effectiveness of classical CGMs. These modifications aim to address specific challenges and improve the overall performance of optimization algorithms in unconstrained scenarios. In order to tackle unconstrained optimization challenges and improve our understanding of their synergies, this ongoing study aims to unify different modified classical CGMs. Conventional CGMs have proven effective for optimization tasks, and a range of different approaches have been produced by carefully modifying these techniques. The main goal of this paper is to combine these modified versions, with particular attention to those that have similar numerators. The integration process involves systematically merging the advantageous aspects of these modified methods to develop not only innovative but also more resilient approaches to unconstrained optimization problems. The ultimate goal of this unification effort is to capitalize on the strengths inherent in different approaches to create a cohesive framework that significantly improves overall optimization performance. To thoroughly assess the efficacy of the integrated methods, a series of comprehensive performance tests are conducted. These tests include a meticulous comparison of outcomes with those of classical CGMs, providing valuable insights into the relative strengths and weaknesses of the modified approaches across diverse optimization scenarios. The evaluation criteria encompass convergence rates, solution accuracy, and computational efficiency. The conclusive outcome demonstrates that the unified approaches consistently outperform individual methods across all three crucial evaluation criteria.

Corresponding author: Oluwaseun B. Onuoha, **Email:** oluwaseun.akinduko@aau.edu.ng
Department of Mathematical Sciences, Adekunle Ajasin University, Akungba Akoko, Nigeria

INTRODUCTION

The significance of unconstrained optimization problems lies in their applicability to a wide range of real-world situations. By allowing flexibility in variable values, these problems provide a natural representation for many optimization challenges. The role of CGMs in solving unconstrained problems is pivotal, as these methods offer efficient and robust approaches to iteratively approaching the optimal solution. Unconstrained optimization problems can arise directly in certain circumstances, but

$$\min f(x), x \in \mathbb{R}^n \tag{1.0}$$

whose gradient is denoted by $g(x)$. The iterative formula of the CGM is given by:

$$x_i = x_{i-1} + \varphi_{i-1}d_{i-1}, \tag{1.1}$$

where d_{i-1} is the search direction and φ_{i-1} is the step-length. For nonlinear CGMs, the current direction, d_i is a linear combination of the

$$d_i = -g_i + \beta_{i-1}d_{i-1}, \tag{1.2}$$

where β_i is a scalar and $g_i = \nabla f(x_i)$ denotes the gradient. Several well-known classical CGMs, proposed by pioneers such as Hestenes and Stiefel (1952), Fletcher and Reeves (1964), Polak

they can also arise indirectly by reformulating constrained optimization problems. By substituting the constraint with penalized terms in the objective function, it is generally possible to tackle an optimization problem as an unconstrained problem.

Consider the function f minimizing an objective function that depends on real variables with no restrictions on their values. Let $x \in \mathbb{R}^n$ be a real vector with $n \geq 1$ components, and let $f: \mathbb{R}^n \rightarrow \mathbb{R}$ be a smooth function. The unconstrained problem is, then:

steepest descent direction and the previous direction d_{i-1} , i.e.,

and Ribiere (1969), and Polyak (1969), Dai and Yuan (1999), Fletcher (1987), and Liu and Storey (1991), offer distinct ways to compute β_i . Their β_i formulae are given by:

$$\beta_i^{HS} = \frac{g_{i+1}^T y_i}{d_i^T y_i} \tag{1.3}$$

$$\beta_i^{FR} = \frac{\|g_{i+1}\|^2}{\|g_i\|^2} \tag{1.4}$$

$$\beta_i^{PRP} = \frac{g_{i+1}^T y_i}{\|g_i\|^2} \tag{1.5}$$

$$\beta_i^{DY} = \frac{\|g_{i+1}\|^2}{d_i^T y_i} \tag{1.6}$$

$$\beta_i^{CD} = \frac{\|g_{i+1}\|^2}{-d_i^T g_i} \tag{1.7}$$

$$\beta_i^{LS} = \frac{g_{i+1}^T y_i}{-d_i^T g_i} \tag{1.8}$$

where y_i represents difference in gradients given by $y_i = g_{i+1} - g_i$, $\|g_{i+1}\|^2 = g_{i+1}^T g_{i+1}$, $\|\cdot\|$ denotes the Euclidean norm and T denotes transpose.

A line search method, which may be exact or inexact, is required to calculate the CGM's step

length. The precise value of φ_i is computed using an exact line search. This is typically tedious and susceptible to errors. Researchers therefore opted to use the inexact line search, which computes the value of φ_i numerically, to navigate around this restriction. The strong Wolfe

(SW) line search, which is outlined below, is among the most widely used inexact line

$$\begin{cases} f(x_i + \varphi_i d_i) \leq f(x_i) + \gamma x_i g_i^T d_i, \\ |g(x_i + \varphi_i d_i)^T d_i| \leq \vartheta |g_i^T d_i|, \end{cases} \quad (1.9)$$

where $0 < \gamma < \vartheta < 1$.

REVIEW OF RELATED LITERATURES

The CGM was invented by Hestenes and Stiefel (1952) in their seminal publication, which provided an algorithm for solving symmetric, positive-definite linear algebraic equations. Reid's 1970 article introduced the conjugate gradient (CG) method as a primary area of active research in unconstrained optimization, following a brief period of stagnation. Fletcher and Reeves (1964) developed the FR method, which is widely regarded as the first nonlinear CG algorithm. Since then, several other CG algorithm versions have been proposed. These versions have made significant contributions to the field of unconstrained optimization.

Based on similarities in their numerators and denominators, Hager and Zhang (2006) revealed different β_i possibilities for the early CGMs. They are methods with numerators of the form $\|g_{i+1}\|^2$ or $g_{i+1}^T y_i$ and denominators of the form $\|g_i\|^2$, $d_i^T y_i$ or $-d_i^T g_i$. They demonstrated that while methods with $\|g_{i+1}\|^2$ in their numerators have a strong convergent property, they are prone to jamming. In contrast, methods with a common numerator of $g_{i+1}^T y_i$ have a built-in restart feature that solves the jamming issue, but they might not be generally convergent. To enhance the functionality of these traditional CGMs, researchers have thus made a variety of modifications utilizing various inexact line searches. Andrei (2008a), for example, developed new accelerated CG algorithms that are essentially modifications of the CG algorithm of Dai and Yuan. The accelerated technique

searches.

proposed in the study was able to increase the algorithms' efficiency. In 2001, Dai and Liao introduced a novel conjugacy condition that gives the old conjugacy condition when the line search is exact but takes into account an inexact scheme. Two novel CGMs were proposed based on the new conjugacy requirement, one of which was found to be particularly efficient. A modified PRP CG approach was introduced by Mohammed *et al.* (2015) to solve large-scale unconstrained optimization problems where the parameter $\sigma < \frac{1}{4}$ is restricted and the sufficient descent property is satisfied under the strong Wolfe-Powell line search.

Zhang (2009) introduced two new variants of Hestenes-Stiefel nonlinear CGM, based on the secant condition commonly met by quasi-Newton methods, which are descent methods even with inexact line searches. A modified hybrid CGM was introduced by Fang *et al.* (2021) and has proven effective in handling large-scale, unconstrained optimization problems. Regardless of the line search, this approach consistently produces accurate directions at each iteration, demonstrating its effectiveness under the Wolfe line search. Based on a modified secant equation, Nezhadhossein (2020) suggested a modified descent-spectral CG approach. This method combines a suitable descent requirement with higher-order precision in estimating second-order curvature information.

Dai and Wen (2012) presented two modified PRP and HS methods, with the following β_i formulae:

$$\beta_i^{DPRP} = \frac{\|g_{i+1}\|^2 - \frac{\|g_{i+1}\|}{\|g_i\|} |g_{i+1}^T g_i|}{\|g_i\|^2 + \mu |d_i^T g_{i+1}|}, \quad \mu > 1 \quad (2.1)$$

and

$$\beta_i^{DHS} = \frac{\|g_{i+1}\|^2 - \frac{\|g_{i+1}\|}{\|g_i\|} |g_{i+1}^T g_i|}{y_i^T d_i + \mu |d_i^T g_{i+1}|}, \quad \mu > 1. \quad (2.2)$$

Again, Du *et al.* (2016) suggested two modified PRP and HS methods, where the resulting CG coefficients are defined as follows:

$$\beta_i^{NVPRP*} = \frac{g_{i+1}^T \left(g_{i+1} - \frac{|g_{i+1}^T g_i|}{\|g_i\|^2} g_i \right)}{\|g_i\|^2}, \quad (2.3)$$

and

$$\beta_i^{NVHS*} = \frac{g_{i+1}^T \left(g_{i+1} - \frac{|g_{i+1}^T g_i|}{\|g_i\|^2} g_i \right)}{y_i^T d_i}. \quad (2.4)$$

Recently, Mehamdia *et al.* (2023) proposed two modified CG methods in the form:

$$\beta_i^{MCB1} = \frac{\|g_{i+1}\|^2 - \rho_1 |g_{i+1}^T g_i| \omega_i}{d_i^T y_i + \mu_1 |d_i^T g_{i+1}|}, \quad (2.5)$$

where $\omega_i = \frac{|g_{i+1}^T d_i| |g_{i+1}^T d_i|}{\|g_i\| \|g_{i+1}\| \|d_i\|^2}$, $\rho_1 \in [0, 1]$ and $\mu_1 > 1 + \rho_1$.

and

$$\beta_i^{MCB2} = \frac{\|g_{i+1}\|^2 - \rho_2 |g_{i+1}^T g_i| \omega_i}{\|g_i\|^2 + \mu_2 |d_i^T g_{i+1}|}, \quad (2.6)$$

where $\rho_2 \in [0, 1]$ and $\mu_2 > 1 + \rho_2$.

All the aforementioned authors have contributed in one way or another to the field of unconstrained optimization by improving the traditional CG methods, though there were no special focus on specific sets of the traditional methods. To bridge this gap, this work is therefore motivated to give further improvements on the traditional methods by focusing

specifically on those ones with similar numerators. By the unification of the two sets of methods with similar numerators, this study aims to address the jamming and convergence problems of these classical methods.

UNIFICATION OF THE CLASSICAL GRADIENT METHODS

To generate new methods that could be applied to the solution of nonlinear, unconstrained

optimization problems, we combine the classical gradient methods listed in Section 1 that have similar numerators. The following Lemmas are applied in the derivation of the classical CGMs.

LEMMA 1

The search direction, d_i , $i = 0,1, \dots$ are conjugate i.e.,

$$d_i^T A d_j = 0 \quad j = 0,1, \dots i - 1. \tag{3.1}$$

LEMMA 2

In a conjugate direction algorithm,

$$g_i^T g_j = 0 \quad j = 0,1, \dots i - 1. \tag{3.2}$$

LEMMA 3

The gradient and the search direction are conjugate i.e.,

$$g_k^T d_i \quad \forall k \quad 0 \leq i \leq k. \tag{3.3}$$

CLASSICAL CGMs WITH SAME NUMERATORS

From the conventional CGMs given by (1.3) - (1.8), there are two sets of CGMs with similar numerators. The first set has the same numerator $\|g_{i+1}\|^2$, while the second set has a similar numerator $g_{i+1}^T y_i$. In this subsection, we show

derivation of two unified CGMs, based on Lemmas 1, 2 and 3.

Unified CGM with Numerator $\|g_{i+1}\|^2$

The conventional CGMs with same numerator $\|g_{i+1}\|^2$ are β_i^{DY} , β_i^{CD} , and β_i^{FR} . Let β_i^{UN1} represents a unified CGM that combines these three classical CGMs. Then:

$$\begin{aligned} \beta_i^{UN1} &= \frac{\|g_{i+1}\|^2}{\|g_i\|^2} + \frac{\|g_{i+1}\|^2}{-d_i^T g_i} + \frac{\|g_{i+1}\|^2}{d_i^T y_i}, \\ &= \frac{\|g_{i+1}\|^2}{\|g_i\|^2} - \frac{\|g_{i+1}\|^2}{d_i^T g_i} + \frac{\|g_{i+1}\|^2}{d_i^T y_i}, \\ &= \|g_{i+1}\|^2 \left(\frac{1}{\|g_i\|^2} - \frac{1}{d_i^T g_i} + \frac{1}{d_i^T y_i} \right), \\ &= \|g_{i+1}\|^2 \left(\frac{d_i^T y_i - g_i^T y_i + \|g_i\|^2}{\|g_i\|^2 d_i^T y_i} \right). \end{aligned}$$

By further simplification of the numerator, we have:

$$\begin{aligned} d_i^T y_i - g_i^T y_i + \|g_i\|^2 &= d_i^T (g_{i+1} - g_i) - g_i^T (g_{i+1} - g_i) + \|g_i\|^2, \\ &= d_i^T g_{i+1} - d_i^T g_i - g_i^T g_{i+1} + g_i^T g_i + \|g_i\|^2. \end{aligned}$$

By using (1.2) together with Lemmas 2 and 3, the above becomes:

$$\begin{aligned} &-d_i^T g_i + \|g_i\|^2 + \|g_i\|^2 \\ &= -(-g_i + \beta_{i-1} d_{i-1})^T g_i + 2\|g_i\|^2, \\ &= g_i^T g_i - \beta_{i-1} d_{i-1}^T g_i + 2\|g_i\|^2, \\ &= \|g_i\|^2 + 2\|g_i\|^2, \\ &= 3\|g_i\|^2. \end{aligned}$$

Therefore.

$$\beta_i^{UN1} = \frac{3\|g_i\|^2 \|g_{i+1}\|^2}{\|g_i\|^2 d_i^T y_i}. \quad (3.4)$$

Unified CGM with Numerator $g_{i+1}^T y_i$

As listed in Section 1, the CGMs β_i^{HS} , β_i^{PRP} , and β_i^{LS} have a common numerator $g_{i+1}^T y_i$. By

denoting a CGM combining these three classical CGMs by β_i^{UN2} , we have:

$$\begin{aligned} \beta_i^{UN2} &= \frac{g_{i+1}^T y_i}{d_i^T y_i} + \frac{g_{i+1}^T y_i}{\|g_i\|^2} + \frac{g_{i+1}^T y_i}{-d_i^T g_i}, \\ &= \frac{g_{i+1}^T y_i}{d_i^T y_i} + \frac{g_{i+1}^T y_i}{\|g_i\|^2} - \frac{g_{i+1}^T y_i}{d_i^T g_i}, \\ &= g_{i+1}^T y_i \left(\frac{\|g_i\|^2 + d_i^T y_i - g_i^T y_i}{\|g_i\|^2 d_i^T y_i} \right). \end{aligned}$$

Factorizing the numerator where $y_i = g_{i+1} - g_i$, we have:

$$\begin{aligned} &d_i^T (g_{i+1} - g_i) - g_i^T (g_{i+1} - g_i) + \|g_i\|^2 \\ &= d_i^T g_{i+1} - d_i^T g_i - g_i^T g_{i+1} + g_i^T g_i + \|g_i\|^2, \\ &= -d_i^T g_i + g_i^T g_i + \|g_i\|^2 \text{ (By Lemmas 2 and 3)}. \end{aligned}$$

By using (1.2) the above yields:

$$-(-g_i + \beta_{i-1} d_{i-1})^T g_i + \|g_i\|^2 + \|g_i\|^2$$

$$\begin{aligned}
 &= g_i^T g_i - \beta_{i-1} d_{i-1}^T g_i + 2\|g_i\|^2, \\
 &= \|g_i\|^2 - 0 + 2\|g_i\|^2, \text{ (By Lemma 3)} \\
 &= 3\|g_i\|^2.
 \end{aligned}$$

Therefore,

$$\begin{aligned}
 \beta_i^{UN2} &= g_{i+1}^T y_i \left(\frac{3\|g_i\|^2}{\|g_i\|^2 d_i^T y_i} \right), \\
 &= \frac{3\|g_i\|^2 g_{i+1}^T y_i}{\|g_i\|^2 d_i^T y_i}, \\
 &= \frac{2\|g_i\|^2 g_{i+1}^T y_i}{d_i^T y_i}.
 \end{aligned}$$

The unified classical conjugate gradient method with similar numerator $g_{i+1}^T y_i$ is thus given by:

$$\beta_i^{UN2} = \frac{2\|g_i\|^2 g_{i+1}^T y_i}{d_i^T y_i}. \tag{3.5}$$

Thus, Equations (3.4) and (3.5) depict unified versions of the classical CGMs, which we obtain by applying Lemmas 1, 2, and 3 to the simplification procedures of the traditional methods with the same numerators $\|g_{i+1}\|^2$ and $g_{i+1}^T y_i$. The UN1 and UN2 approaches' enhanced performance is made possible by the term

" $\|g_i\|^2$ " being included in the numerators. A numerical experiment is carried out to demonstrate the effectiveness of the new approaches in contrast to the conventional methods; the specifics are provided in the next section. The following algorithm is suggested in order to implement the two new techniques:

Unified CGMs Algorithm

- Step 1** Select the initial starting point $x_0 \in f$ and compute: $f_0 = f(x_0)$ and $g_0 = \nabla f(x_0)$. Set $d_0 = -g_0$ and $i = 0$
- Step 2** If $\|g_i\| \leq 10^{-6}$, then stop; otherwise continue with step 3.
- Step 3** Determine the step length φ_i by (1.9).
- Step 4** Update the variables as: $x_{i+1} = x_i + \varphi_i d_i$. Compute f_{i+1} and g_{i+1} .
- Step 5** Determine β_i^{UN1} and β_i^{UN2} by (3.4) and (3.5).
- Step 6** Compute the search direction by (1.2).
- Step 7** Set $i := i + 1$ and return to Step 2.

RESULTS AND DISCUSSION

In comparison to conventional methods, the numerical performance of the two proposed unified approaches is presented in this section. A numerical experiment refers to a simulation or computation conducted on a computer to analyze

and understand a phenomenon, system, or process. Instead of physical experimentation, numerical experiments use mathematical models and algorithms to explore and study various scenarios, helping researchers gain insights and make predictions. Thus, in this study, twenty (20)

large-scale, unconstrained minimization problems, each of dimension 5000, were subjected to a numerical experiment in order to test the efficacy of the two new methods in comparison to the conventional methods. To compute the step length, the SW line search is used for efficiency and cost-effectiveness. Andrei (2008) compiled the test problems, which were taken from the CUTE library. In an effort to provide an environment for testing both constrained and unconstrained nonlinear optimization methods, Bongartz et al. (1995) developed the CUTE package. The test problems' names, dimensions, initial points, and sources are listed in Table 1, while Tables 2 and 3 include the numerical results for the number of iterations and CPU time.

The computational algorithm was implemented using MATLAB R2007b (7.5.0.342). The numerical codes were run on a Dell PC with the specifications "Intel (R) CoreTM i3, 2.20GHz processor, 4GB RAM, and windows 10 pro-operating system of 64-bit". The iterations were terminated when $\|g_i\| \leq 10^{-6}$, and a failure declared if this condition was not satisfied after 2000 iterations. ITR indicates the number of iterations, while CPU indicates the processing time (in seconds), according to Tables 2 and 3.

Dolan and Moré (2002) presented their proposed graphical tool for benchmarking algorithms as "Performance profiles", which they described as the cumulative function for a performance metric, which could be the ratio of the computing time of the solver against the best time of all the solvers or the ratio of the number of iterations of the solver to the best minimum number of all the solvers. To compare the performance of the two novel methods to the existing methods, the performance profiles of Dolan and Moré were used. Figures 1 through 4 show the comparison based on CPU time and number of iterations.

The data displayed in Tables 2 and 3 were used in statistical analysis to more precisely quantify the variations in the performances of the new methods UN1 and UN2. Specifically, we performed an analysis of variance (ANOVA) test with a level of significance of $\alpha = 0.05$. The statistical results for the UN1 technique in regard to the FR, CD, and DY techniques are displayed in Tables 2A, 2B, 2C, and 2D, while the statistical results for the UN2 method in relation to the HS, LS, and PRP methods are displayed in Tables 3A, 3B, 3C, and 3D. The results of the statistical analysis indicate that the UN1 and UN2 approaches are better than the ones that are already in use because of their much shorter average computational times and iteration counts.

Table 1: Details of the Test Problems (Dimension 5000)

TEST PROBLEMS	INITIAL POINTS	SOURCES
Extended Block Diagonal	[0.1, 0.1..., 0.1]	Bongartz et al. (1995)
Power function	[1, 1, ..., 1]	Andrei (2008)
Arwhead Function	[1, 1, ..., 1]	Andrei (2008)
Diagonal 5 Function	[1.1, 1.1, ..., 1.1]	Andrei (2008)
Qf1 Function	[1, 1, ..., 1]	Bongartz et al. (1995)
Qf2 Function	[0.5, 0.5, ..., 0.5]	Bongartz et al. (1995)
Chebyquad Function	[1, 1, ..., 1]	Bongartz et al. (1995)
Diagonal 4 function	[1, 1, ..., 1]	Andrei (2008)
Arglinc Function	[1, 1, ..., 1]	Andrei (2008)
Brownbs Function	[1, 1, ..., 1]	Bongartz et al. (1995)
Bdexp Function	[1, 1, ..., 1]	Bongartz et al. (1995)
Bdqrtic Function	[1, 1, ..., 1]	Andrei (2008)
Staircase 1 Function	[1, 1, ..., 1]	Andrei (2008)
Staircase 2 Function	[0, 0, ..., 0]	Andrei (2008)
Liarwhd Function	[4, 4, ..., 4]	Andrei (2008)
Extended Beale Function	[1,0.8, ..., 1, 0.8]	Andrei (2008)
Extended Wood Function	[-3,-1,-3,-1,..., -3,-1,-3,-1]	Andrei (2008)
Extended Tridigonal-1 Function	[2, 2, ..., 2]	Andrei (2008)
Hager Function	[1, 1, ..., 1]	Andrei (2008)
Generalised Rosenbrock	[-1.2,1, ..., -1.2,1]	Andrei (2008)

Table 2: Test results of the UN1, FR, CD, and DY methods

TEST PROBLEMS	UN1	FR	CD	DY
	ITR/CPU	ITR/CPU	ITR/CPU	ITR/CPU
Extended Block Diagonal	2/0.044	492/18.517	22/0.481	116/2.484
power function	1/0.036	1/0.037	1/0.047	1/0.041
Arwhead Function	6/0.554	103/2.627	2000/52.491	109/2.693
Diagonal 5 Function	5/0.091	13/0.227	7/0.217	23/0.404
Qf1 Function	4/0.211	2000/83.544	2000/50.072	233/6.334
Qf2 Function	10/0.205	10/0.196	39/0.755	2/0.048
Chebyquad Function	5/0.161	471/6.417	3/0.074	2/0.048
Diagonal 4 function	3/0.142	17/0.423	17/0.339	15/0.034
Arglinc Function	1/0.052	1/0.052	1/0.052	1/0.0313
Brownbs Function	8/1.657	6/0.49	22/1.034	9/0.052
Bdexp Function	3/0.119	3/0.078	2/0.05	3/0.443
Bdqrtic Function	4/1.494	2000/241.031	2000/193.278	2000/0.084
Staircase 1 Function	1/0.018	1/0.02	1/0.019	1/217.868
Staircase 2 Function	3/0.096	16/0.294	92/1.706	2/0.019
Liarwhd Function	8/1.065	2000/129.882	2000/58.106	2000/0.043
Extended Beale Function	4/2.162	2000/266.686	2000/145.709	2000/83.873
Extended Wood Function	8/2.507	1680/192.577	2000/110.738	121/317.994
Extended Tridigonal Function	4/0.495	2000/155.925	2000/56.291	2000/14.857
Hager Function	7/1.938	17/0.816	11/0.683	8/0.429
Generalised Rosenbrock Function	3/0.157	2000/207.261	134/3.86	2000/209.261

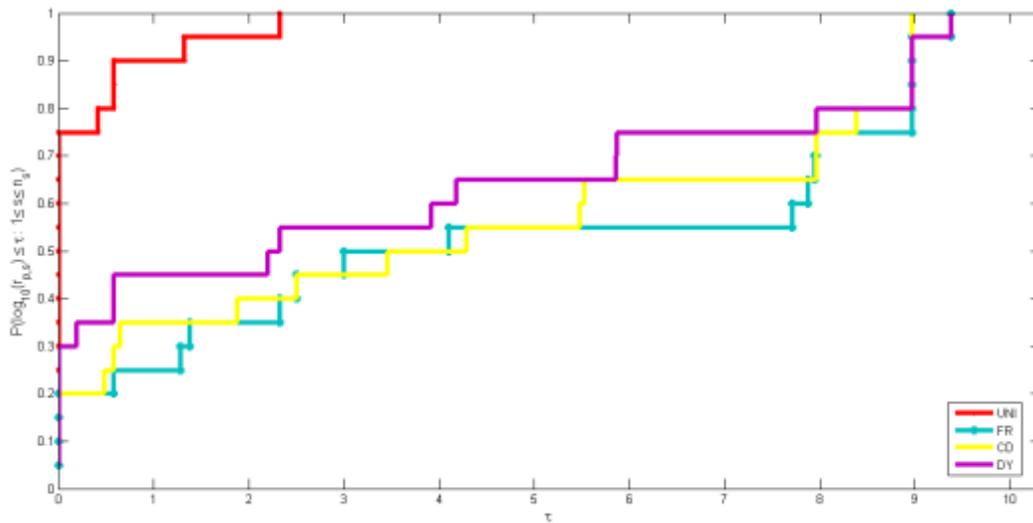


Figure 1: Performance Profile on the Number of Iterations (UN1 versus FR, CD, and DY)

Table 2A: Statistical Comparison of UN1 Method with FR, CD and DY Methods in Terms of Number of Iterations

<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>
UN1	20	90	4.5	6.789474
FR	20	14831	741.55	857901.9
CD	20	14352	717.6	933178.4
DY	20	10646	532.3	759385.5

Table 2A presents a comparative analysis of the efficiency of UN1, FR, CD, and DY methods based on the number of iterations across 20 test problems. The UN1 approach demonstrated the lowest average number of iterations (4.5), suggesting that it is an efficient way of reaching convergence. On the other hand, the DY technique showed a noticeably greater average number of iterations (532.3), with the greatest

average (741.55) being achieved by the FR approach, followed by the CD method (717.6). Variance values shed light on how many iterations are distributed among the methods in each group. These results indicate that, in terms of iteration count efficiency, the UN1 approach performs better than the DY, CD, and FR methods on average, making it the better option for reaching convergence in iterative processes.

Table 2B: ANOVA ($\alpha = 0.05$)

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	7045112	3	2348371	3.683036	0.015577	2.724944
Within Groups	48458979	76	637618.1			
Total	55504091	79				

Table 2B reveals that there is a significant difference in the average number of iterations recorded for each approach, as indicated by the P value (0.015577), which is less than 0.05.

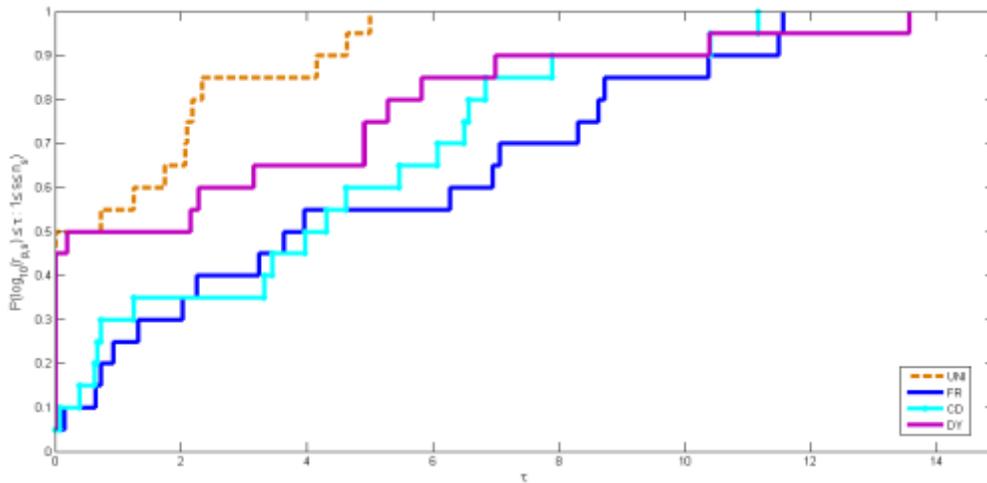


Figure 2: Performance Profile with respect to CPU time (UN1 versus FR, CD, and DY)

Table 2C: Statistical Results Comparing UN1 Method with FR, CD and DY Methods with Respect to CPU Time

Groups	Count	Sum	Average	Variance
UN1	20	13.204	0.6602	0.676679
FR	20	1307.1	65.355	9073.086
CD	20	676.002	33.8001	3148.91
DY	20	857.0403	42.85202	8576.865

The average computation times for the UN1, FR, CD, and DY approaches over 20 test problems are summarized in the above table. The UN1 approach showed the least time required for computation on average (0.6602s), highlighting its effectiveness. Following closely is the CD method with an average time of 33.8001s, while the DY method recorded an average time of

42.85202s. At 65.355s, the FR technique yielded a comparatively greater average time. The range of computation times for each method within its group is shown by the variance values. As a result, the UN1 approach is a better option for computational tasks because it is the fastest of the CD, DY, and FR approaches.

Table 2D: ANOVA ($\alpha = 0.05$)

Source of Variation	SS	Df	MS	F	P-value	F crit
Between Groups	43239.26	3	14413.09	2.771809	0.047218	2.724944
Within Groups	395191.2	76	5199.884			
Total	438430.5	79				

The P value (0.047218) is less than 0.05. This is an indication that there is a significant difference in the average computational time for UN1, FR, CD and DY methods, as demonstrated in Table 2C.

From Figures 1 and 2, it is obvious that amongst the four methods being compared, the UN1 method performs better than the DY and CD methods while the FR method has the least performance.

Table 3: Test results of the UN2, HS, LS, and PRP methods

PROBLEMS	UN2	HS	LS	PRP
	ITR/CPU	ITR/CPU	ITR/CPU	ITR/CPU
Extended Block Diagonal	2/0.044	882/20.586	102/3.841	107/4.066
power function	1/0.0372	1/0.037	1/0.043	1/0.037
Arwhead Function	2/0.444	30/0.727	94/2.61	216/5.891
Diagonal 5 Function	2/1.159	3/0.055	25/0.439	9/0.205
Qf1 Function	2/0.182	167/2.561	2000/63.103	492/13.803
Qf2 Function	2/0.244	3/0.053	19/0.473	19/0.437
Chebyquad Function	2/0.078	3/0.049	2000/27.755	2000/27.768
Diagonal 4 function	2/0.142	4/0.123	60/2.048	37/1.234
Arglinc Function	2/0.052	1/0.052	1/0.052	1/0.052
Brownbs Function	2/1.167	3/0.138	11/0.73	11/0.739
Bdexp Function	2/0.119	3/0.08	3/0.079	3/0.078
Bdqrtic Function	2/1.801	2000/152.458	2000/245.83	2000/213.384
Staircase 1 Function	2/0.018	1/0.019	1/0.019	1/0.028
Staircase 2 Function	2/0.101	7/0.154	45/1.076	45/1.043
Liarwhd Function	2/0.998	6/0.261	2000/89.95	45/17.907
Extended Beale Function	2/1.741	30/2.688	2000/216.515	446/100.48
Extended Wood Function	2/0.948	14/1.054	643/69.907	167/21.634
Extended Tridigonal FuInction	2/0.723	795/25.661	2000/118.307	2000/122.625
Hager Function	2/0.311	13/0.551	132/4.029	37/1.118
Generalised Rosenbrock Function	2/0.162	2000/209.374	2000/211.83	2000/204.364

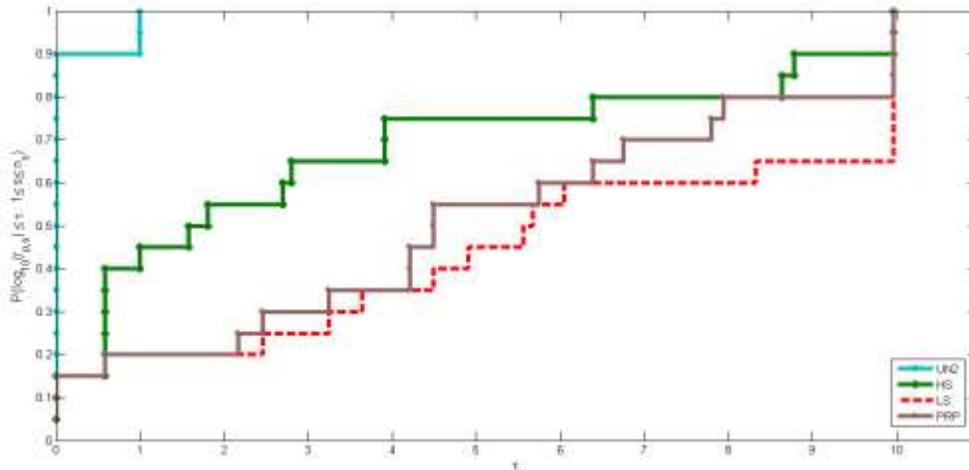


Figure 3: Performance Profile on the Number of Iterations (UN2 versus HS, LS, and PRP)

Table 3A: Statistical Results for UN2 VS HS, LS and PRP Based on Number of Iterations

Groups	Count	Sum	Average	Variance
UN2	20	39	1.95	0.05
HS	20	5966	298.3	403183.9
LS	20	15137	756.85	894758.9
PRP	20	9637	481.85	625828.9

The above table provides statistical results comparing the efficiency of the UN2, HS, LS, and PRP methods based on the number of iterations across 20 test problems. On average, the UN2 method demonstrated the lowest number of iterations (1.95), indicating its efficiency in converging to solutions. In contrast, the HS method exhibited a significantly higher average

number of iterations (298.3), followed by the PRP method (481.85), and the LS method with the highest average (756.85). Variance values reveal the dispersion of iteration counts within each method's group. These results suggest that, on average, the UN2 method is more efficient in terms of the number of iterations compared to HS, PRP, and LS methods.

Table 3B: ANOVA ($\alpha = 0.05$)

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	6037925	3	2012642	4.184783	0.008508	2.724944
Within Groups	36551662	76	480942.9			
Total	42589587	79				

The P value (0.008508) is less than 0.05, showing a significant difference in the average number of iterations needed by UN2, HS, LS, and PRP methods to achieve convergence.

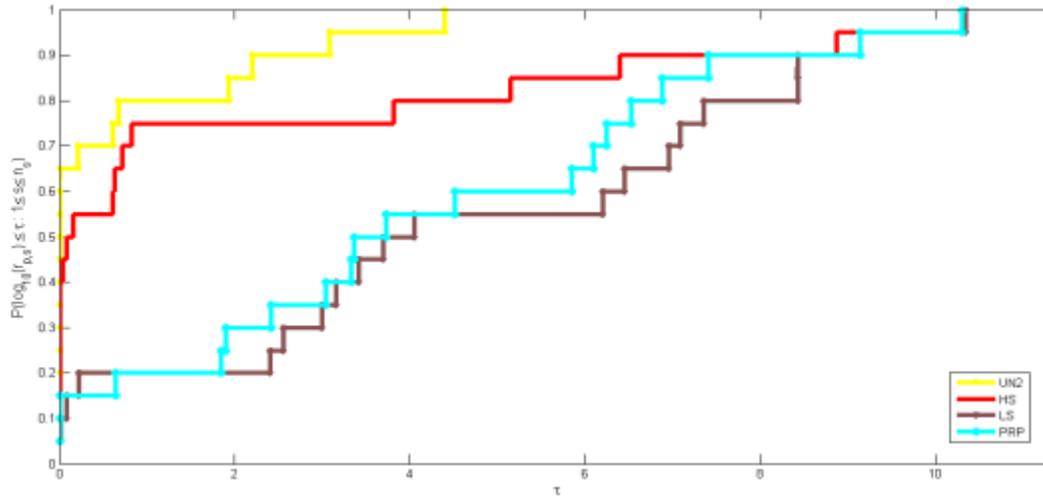


Figure 4: Performance Profile with respect to CPU time (UN2 versus HS, LS, and PRP)

Table 3C: Statistical Analysis of UN2 Method versus HS, LS and PRP Based on Computational Time

<i>Groups</i>	<i>Count</i>	<i>Sum</i>	<i>Average</i>	<i>Variance</i>
UN2	20	10.4712	0.52356	0.336037
HS	20	416.681	20.83405	3131.469
LS	20	1058.636	52.9318	6732.555
PRP	20	736.893	36.84465	4583.479

An overview of the average computing time for the UN2, HS, LS, and PRP algorithms over 20 test problems is shown in Table 3C. The outcomes illustrate how effective the UN2 approach is, with an average computational time of 0.52356s, the lowest of all methods. The HS approach had an average duration of 20.83405s,

while the PRP method came in close behind with an average time of 36.84465s. At 52.9318s, the LS method's average time was significantly higher. Within each method's group, variance values show how different the computing times are. According to the results, the UN2 approach beats the HS, LS, and PRP methods and is the most time-efficient.

Table 3D: ANOVA ($\alpha = 0.05$)

<i>Source of Variation</i>	<i>SS</i>	<i>df</i>	<i>MS</i>	<i>F</i>	<i>P-value</i>	<i>F crit</i>
Between Groups	30118.81	3	10039.6	2.779545	0.046774	2.724944
Within Groups	274508.9	76	3611.96			
Total	304627.7	79				

In Table 3D, P is less than 0.05, with a value of 0.046774. This suggests that the average computing time for the UN2, HS, LS, and PRP

algorithms differs significantly among the 20 test problems.

From Figures 3 and 4, the new method UN2 performed better than the HS method and the PRP method while the LS method had the least performance for both the CPU time and number of iterations.

CONCLUSION

The goal of this study is to unite a variety of modified classical CGMs for unconstrained optimization. Thus, by fusing traditional CGMs with comparable numerators, two unique modified CGMs have been created. The integrated approaches' effectiveness is evaluated

through performance testing, wherein they are contrasted with traditional CGMs. In terms of convergence rates, accuracy of the solutions, and computing speed, the results demonstrate that the unified approaches are more effective and reliable than the conventional methods. Future studies will take into account the new approaches' global convergence properties.

CONFLICTS OF INTEREST

The authors declare that they have no conflict of interest

REFERENCES

- Andrei, N. (2008). An unconstrained optimization test functions. *Advanced Modeling and Optimization*. 10 (1), 147-161.
- Andrei, N. (2008a). *New accelerated conjugate gradient algorithms for unconstrained Optimization*. Research Institute for Informatics, center for Modelling and Optimization.
- Bongartz, I., Conn, A. R., Gould, N. I. M. and Toint, P. L. (1995). CUTE: Constrained and unconstrained testing environments. *ACM Transactions on Mathematical Software*. 21, 123-160.
- Dai, Y. H. and Yuan, Y. (1999). A nonlinear conjugate gradient method with a strong global convergence property. *SIAM Journal on optimization*. 10, 177-182.
- Dai, Y. H. and Liao, L. Z. (2001). New conjugacy conditions and related nonlinear conjugate gradient method. *Applied Mathematics and Optimization*. 43, 87-101.
- Dai, Z. and Wen, F. (2012). Another improved Wei-Yao-Liu nonlinear conjugate gradient method with sufficient descent property. *Applied Mathematics and Computation*. 218, 7421-7430.
- Dolan, E. D. and Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*. 91, 201-213.
- Du, X. W., Zhang, P. and Ma, W. (2016). Some modified conjugate gradient methods for unconstrained optimization. *Journal of Computational and Applied Mathematics*. 305, 92-114.
- Fang, M., Wang, M., Sun, M., and Chen, R. (2021). A modified hybrid conjugate gradient method for unconstrained optimization. *Journal of Mathematics*, Article ID 5597863, 9 pages. <https://doi.org/10.1155/2021/5597863>.
- Fletcher, R. (1987). *Practical Method of Optimization*, 2nd Edn. New York: John Wiley and Sons.
- Fletcher, R. and Reeves, C. M. (1964). Function minimization by conjugate gradients. *The Computer Journal*. 7, 149-154.
- Hager, W. W. and Zhang, H. (2006). A survey of nonlinear conjugate gradient methods. *Pacific Journal of Optimization*. 2, 35-58.
- Hestenes, M. R. and Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*. 9, 409-436.
- Liu, Y. and Storey, C. (1991). Efficient generalized conjugate gradient algorithms part 1, theory. *Journal of Optimization theory and Application*. 69, 322-340.
- Mehamdia, A. E., Chaib, Y. and Bechouat, T. (2023). Two modified conjugate gradient methods for solving unconstrained optimization and application. *Rairo Operations Research*. 57, 333-350.

- Mohammed, H., Mohd, R., Mustafa, M. and Zabidin, S. (2015). A conjugate gradient method with inexact line search for unconstrained optimization. *Applied Mathematical Sciences*. 9 (37), 1823-1832.
- Nezhad Hosein, S. (2020). A Modified Descent Spectral Conjugate Gradient Method for Unconstrained Optimization. *Iranian Journal of Science and Technology, Transactions A: Science*. 45, 209-220.
- Polak, E. and Ribiere, G. (1969). Note Sur la Convergence de directions conjugees. *ESAIM: Mathematical Modelling and Numerical Analysis*. 3, 35-43.
- Polyak, B. T. (1969). The conjugate gradient method in extreme problems. *USSR Computer and Mathematical Physics*. 9, 94-112.
- Reid, J. K. (1970). On the method of conjugate gradients for the solution of large sparse systems of linear equations. In Catherine, St. (Ed.). *Large Sparse Sets of Linear Equations* (pp. 231-254). London: Oxford Academic Press.
- Zhang, L. (2009). New versions of the Hestenes-Stiefel nonlinear conjugate gradient method based on the secant condition for optimization. *Journal of Computational and Applied Mathematics*. 28, 111-133.p.1-6.