Malaysian Journal of Computing, 5 (1): 403-413, 2020
Copyright © UiTM Press
eISSN: 2600-8238 online

A STEP-LENGTH FORMULA FOR CONJUGATE GRADIENT METHODS

Adam Ajimoti Ishaq[1], Tolulope Latunde[2] and Kazeem Babatunde Akande[3]

Department of Physical Sciences, Al-Hikmah University, Ilorin, Nigeria
Department of Mathematics, Federal University Oye-Ekiti, Oye-Ekiti, Nigeria
Department of Science and Research, Iqra College, Ilorin, Nigeria
[1]aaishaq@alhikmah.edu.ng, [2]tolulope.latunde@fuoye.edu.ng, [3]akande.kb22002@gmail.com

ABSTRACT

A newly step-length formula is proposed for implementing conjugate gradient methods' algorithm to solve unconstrained optimization problems. The unified formula for obtaining step-length does not involve any matrix operation. Numerical results obtained are graphically illustrated using performance profiling software. This showed that the new formula performs efficiently in terms of computational efforts and execution time compared with some existing formulae for obtaining the step-length without line search procedures.

Keywords: *Step-length, conjugate gradient method, unconstrained optimization problem.*

Received for review: 23-01-2020; Published: 30-04-2020

1. Introduction

We consider the conjugate gradient method (CGM) for the unconstrained optimization problem

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

where f is continuously differentiable and a real-valued function. The CGM constitutes an active choice for efficiently solving (1), particularly when the dimension n is very large, due to its simplicity analysis, very low memory requirement, fast convergence and its splendid numerical performance, for engineers and mathematicians (Jinhong & Genjiao, 2013). It uses the following iterative plot of the form:

$$x^{(k+1)}=x^{(k)}+\alpha_k d^{(k)}, \tag{2}$$

where, $x^{(k)}$ is the present iterative point, $\alpha_k>0$, the step-length which can be carried out by various step-length rules and $d^{(k)}$ is the search direction calculated by:

$$d^{(k)}=\begin{cases}-g^{(k)}+\beta_{(k)}d^{(k-1)}; k\geq 1\\ -g^{(k)}; \quad k=0\end{cases}, \tag{3}$$

where $g^{(k)}=\nabla f(x^{(k)})$ and β_k is the CG update parameter which decides the distinctive CGMs which in turn lead to very assorted computational efficiency and convergence results of the comparing methods (Zhang, 2010). The following are some well-known CG update parameters:

$$\beta_k^{FR}=\frac{\|g^{(k+1)}\|^2}{\|g^{(k)}\|^2}, \text{ Fletcher \& Reeves (1964)[FR]} \tag{4}$$

$$\beta_k^{PRP}=\frac{g^{(k+1)T}y^{(k)}}{\|g^{(k)}\|^2}, \text{ Polak \& Ribiere (1969)[PRP]} \tag{5}$$

$$\beta_k^{HS}=\frac{g^{(k+1)T}Y^{(k)}}{g^{(k)T}g^{(k)}}, \text{ Hestenes \& Stiefel (1952)[HS]} \tag{6}$$

$$\beta_k^{CD}=-\frac{\|g^{(k+1)}\|^2}{g^{(k)T}d^{(k)}}, \text{ Fletcher (1987)[CD]} \tag{7}$$

$$\beta_k^{DY}=\frac{\|g^{(k+1)}\|^2}{y^{(k)T}d^{(k)}}, \text{ Dai \& Yuan (2000)[DY]} \tag{8}$$

$$\beta_k^{LS}=-\frac{g^{(k+1)T}y^{(k)}}{g^{(k)T}d^{(k)}}, \text{ Liu \& Storey (1992)[LS]} \tag{9}$$

$$\beta_k^{BAN}=\frac{g^{(k+1)T}y^{(k)}}{g^{(k)T}y^{(k)}}, \text{ Bamigbola \textit{et al.} (2010)[BAN]} \tag{10}$$

$$\beta_k^{HZ}=\left(y^{(k)}-2d^{(k)}\frac{\|y^{(k)}\|^2}{y^{(k)T}d^{(k)}}\right)^{(T)}\frac{g^{(k+1)T}d^{(k)}}{y^{(k)T}d^{(k)}}, \text{ Hager \& Zhang (2005)[HZ],} \tag{11}$$

where $y^{(k)}=(g^{(k+1)}-g^{(k)})$.

The success of an optimization method in the iterative scheme given in (2) depends largely on the accuracy of computing $d^{(k)}$ and α_k.

In performing the iterative scheme in (2), various methods were applied to determine the step-length, one of such methods is the line search techniques. On the other hand, a fixed formula can also be used to obtain the step-length that is simply called step-length with no line search procedure (Sun & Zhang, 2001). But the line search schemes need many evaluations of some function and gradient expressions, thus reducing the numerical efficiency of CGMs in large-scale optimization problems (Zhang, 2010). Sun & Zhang (2001) developed a CGM where the step-length is computed by a formula in their method instead of a line search scheme. The following is their formula:

$$\alpha_k=\frac{-\delta g^{(k)}d^{(k)}}{\|d^{(k)}\|^2_{Q_{(k)}}}, \tag{12}$$

where $\|d^{(k)}\|_{Q_{(k)}}=\sqrt{d^{(k)T}Q_{(k)}d^{(k)}}$ $\delta\in\left(0,\frac{V_{min}}{\tau}\right)$, τ is a Lipschitz constant of the function f and $\{Q_{(k)}\}$ is a sequence of positive definite matrices satisfying the positive constant V_{min} and V_{max} such that

$$V_{min}d^Td\le d^TQ_{(k)}d\le V_{max}d^Td, d\in R^n. \tag{13}$$

But the formula presented in (13) involves $\left\{Q_{(k)}\right\}$, and this may take a toll extra memory space and execution time amid the numerical experiments for large-scale optimization issues. Wu (2011), later derived a formula for the step-length that is matrix-free by updating the formula in (13) using the quasi-Newton methods in Zhang *et al.*, (1999), and the formula is presented as

$$\alpha_k=\frac{-\delta g^{(k)T}d^{(k)}}{(\bar{g}^{(k+1)}+g^{(k)})^Td^{(k)}+\gamma\theta_k}, \tag{14}$$

where $\theta_k=6(f^{(k)}-\bar{f}^{(k+1)})+3(g^{(k)}-\bar{g}^{(k+1)})^Td^{(k)}$, δ and γ are parameter satisfying, $\delta\in(0,\frac{K}{\tau})$ and $\gamma\geq0$ if $\tau=K$.

Ajimoti & Bamigbola (2016) derived a step-size formula that uses only gradient information to obtain the step-length in the iterative scheme (2) and the formula is given as

$$\alpha_k=\frac{-\delta g^{(k)T}d^{(k)}}{(\bar{g}^{(k+1)}-g^{(k)})^Td^{(k)}},\delta\in\left(0,\frac{\tau}{\lambda}\right). \tag{15}$$

In this work, the idea of enhancing the line search procedure is undertaking by prescribing a new step-length rule without a line search for CGMs involving both function and gradient information. The rest of the paper is constituted as follows. In Section 2, the derivation of the new formula is presented. In Section 3, numerical results illustrating the efficiency of the proposed formula and comparing its performance with two existing formulae using some CGMs were reported, thereafter, the conclusion was drawn in Section 4.

2. The New Step-length Formula

In this section, the new step-length formula is presented. The following assumptions which are commonly used in the literature on function f are required.

Assumption 2.1:

i. The level set $L=\{x\in R^n|f(x)\leq f(x^{(1)})$, with $x^{(1)}$ to be the current or starting point of the iterative method (2) is bounded.

ii. The objective function f(x) is Lipschitz continuous and differentiable function and strongly convex in $\Re^n$, i.e., $\exists\ \lambda>0$ and $\tau\geq0$ such that,

$$\left\|g^{(k+1)}-g^{(k)}\right\|\leq\lambda\left\|x^{(k+1)}-x^{(k)}\right\| \tag{16}$$

and

$$(g^{(k+1)}-g^{(k)})^T(x^{(k+1)}-x^{(k)}|\geq\tau\left\|x^{(k+1)}-x^{(k)}\right\|^2. \tag{17}$$

Lemma 2.1:

Suppose that Assumption 2.1 hold. Then,

$$\tau\left\|d^{(k)}\right\|^2\leq\bar{f}^{(k+1)}-2f^{(k)}+\bar{f}^{(k-1)}\leq2\lambda\left\|d^{(k)}\right\|^2 \tag{18}$$

holds for all k.

Proof:

Note that $f^{(k)}$, $\bar{f}^{(k+1)}$ and $\bar{f}^{(k-1)}$ denote $f(x^{(k)})$, $f(x^{(k)}+d^{(k)})$ and $f(x^{(k)}-d^{(k)})$. Hence, the proof is straight forward by adopting (16) and (17).

Now we present the derivation of the step-length formula that uses available information on both function and gradient.

Iterative scheme (2) gives

$$\alpha_k d^{(k)}=x^{(k+1)}-x^{(k)} \tag{19}$$

put (19) in (16), then we have,

$$\|g^{(k+1)}-g^{(k)}\|\leq\lambda|\alpha_k|\|d^{(k)}\| \tag{20}$$

$$(g^{(k+1)}-g^{(k)})^T d^{(k)}\leq\|g^{(k+1)}-g^{(k)}\|\|d^{(k)}\|\leq\lambda\alpha_k\|d^{(k)}\|^2 \tag{21}$$

$$\Rightarrow\ (g^{(k+1)}-g^{(k)})^T d^{(k)}\leq\lambda\alpha_k\|d^{(k)}\|^2 \quad . \tag{22}$$

Using conjugacy property, i.e., $g^{(k+1)})^T d^{(k)}=0$, then (22) becomes

$$-g^{(k)T}d^{(k)}\leq\lambda\alpha_k\|d^{(k)}\|^2$$

$$\Rightarrow\ \frac{-g^{(k)T}d^{(k)}}{\lambda\alpha_k}\leq\|d^{(k)}\|^2 . \tag{23}$$

From Lemma 2.1, this results to

$$\|d^{(k)}\|^2\leq\frac{(\bar{f}^{(k+1)}-2f^{(k)}+\bar{f}^{(k-1)})}{\tau} \tag{24}$$

and combining (23) and (24), we have

$$-\tau g^{(k)T}d^{(k)}\leq\lambda\alpha_k(\bar{f}^{(k+1)}-2f^{(k)}+\bar{f}^{(k-1)})$$

$$\Rightarrow\ \lambda\alpha_k(\bar{f}^{(k+1)}-2f^{(k)}+\bar{f}^{(k-1)})\geq-\tau g^{(k)T}d^{(k)}$$

$$\Rightarrow\ \alpha_k\geq-\frac{\tau g^{(k)T}d^{(k)}}{\lambda(\bar{f}^{(k+1)}-2f^{(k)}+\bar{f}^{(k-1)})}.$$

Using the Sun & Zhang (2001) approach, we have

$$\alpha_k=-\frac{\mu g^{(k)T}d^{(k)}}{(\bar{f}^{(k+1)}-2f^{(k)}+\bar{f}^{(k-1)})} . \tag{25}$$

3. Computational Consideration

3.1 Algorithm: CGM Algorithm

Step 1: Select the initial point, $x^{(0)} \in R^n$, $\epsilon \geq 0$ (a small number called tolerance) and set $d^{(0)} = -g^{(0)} = -\nabla f(x^{(0)}), k=0$.

Step 2: Terminate process if $\|g^{(0)}\| \leq \epsilon$, else, go to the Next Step.

Step 3: Compute step length α_k by:

Step 3a: Using a fixed formula in (14).

Step 3b: Using a fixed formula in (15).

Step 3c: Using a fixed formula in (25).

Step 4: Set $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$; if $\|g^{(k+1)}\| \leq \epsilon$, then stop, else, go to the Next Step.

Step 5: Compute $d^{(k+1)} = -g^{(k+1)} + \beta_k d^{(k)}$, where β_k is given by equation (4-12).

Step 6: Set k=k+1, and go to Step 3.

3.2 Numerical Comparison

The numerical experiments carried out in this work incorporated with fixed formulae in (14), (15) and (25) into the CGM Algorithm 3.1. We aim to perform an experiment that aid in measuring the effectiveness of obtaining the step-length α_k using a fixed formula rather than any line search procedures. The newly derived formula for finding step-length was compared with some existing fixed formulae by using nine kinds of conjugate gradient methods to solve thirty unconstrained optimization test functions obtained from the CUTE collection made available by Andrei (2008) and Jamil & Yang (2013) with standard starting points and each test function is given with two different dimensions (n=5,000 and n=10,000). The CGM Algorithm 3.1 was implemented via Matlab 8.0 version and run on a PC HP EliteBook 6930p with 2.00GB RAM, 2.20GHZ processor and 3.4 windows experience index operating system. In Figures 1-6, we adopt the performance profiles introduced by Dolan & More (2002) to evaluate and compare the performance of different kinds of CGM against various rules of obtaining the step size α_k, to test the efficiency of these methods using optimization software based on the CPU time and the number of iteration where NSF (formula presented in (25)) represents new step-length formula, ESF1 (formula presented in (14)) represents existing step-length formula one and ESF2 (formula presented in (15)) represents existing step-length formula two. A table containing the test functions and their sources is presented in Table 1 and their numerical results are graphically illustrated in Figures 1-6.

Table 1. A list of Test Problems

S/N	PROBLEM	SOURCE
1	Extended Rosenbrock Function	(Andrei, 2008)
2	Linear Function - rank 1	(Andrei, 2008)
3	Quadratic Diagonal Perturbed Function	(Andrei, 2008)
4	Perturbed Quadratic Function	(Andrei, 2008)

5	Quadratic QF1 Function	(Andrei, 2008)
6	ARGLINB(m=20) Function	(Andrei, 2008)
7	Almost Perturbed Quadratic Function	(Andrei, 2008)
8	Extended White & Holst Function	(Andrei, 2008)
9	Raydan 1 Function	(Andrei, 2008)
10	Raydan 2 Function	(Andrei, 2008)
11	Extended Three Exponential Terms Function	(Andrei, 2008)
12	Generalized Rosenbrock Function	(Andrei, 2008)
13	Generalized White & Holst Function	(Andrei, 2008)
14	Extended Block Diagonal BD1 Function	(Andrei, 2008)
15	HimmelBG Function	(Andrei, 2008)
16	Power Function	(Andrei, 2008)
17	Extended Dixon and Price	(Andrei, 2008)
18	Extended Booth Function	(Andrei, 2008)
19	Extended Boh2 Function	(Andrei, 2008)
20	Diagonal 3 Function	(Jamil & Yang, 2013)
21	Hager Function	(Jamil & Yang, 2013)
22	Extended Penalty Function	(Andrei, 2008)
23	Extended Cliff & Roth Function	(Andrei, 2008)
24	Extended Quadratic Penalty QP1 Function	(Andrei, 2008)
25	Extended EP1 Function	(Andrei, 2008)
26	ARWHEAD Function	(Andrei, 2008)
27	Extended Freudenstein & Roth Function	(Andrei, 2008)
28	Cube Function	(Andrei, 2008)
29	Extended Goldstein & Price Function	(Andrei, 2008)
30	Chebyquad Function	(Andrei, 2008)

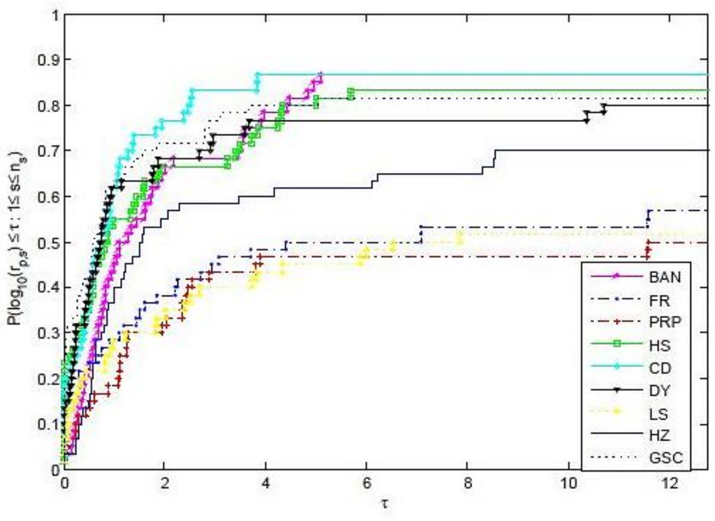

Figure 1. Performance profile for various CGMs by ESF1 based on CPU time

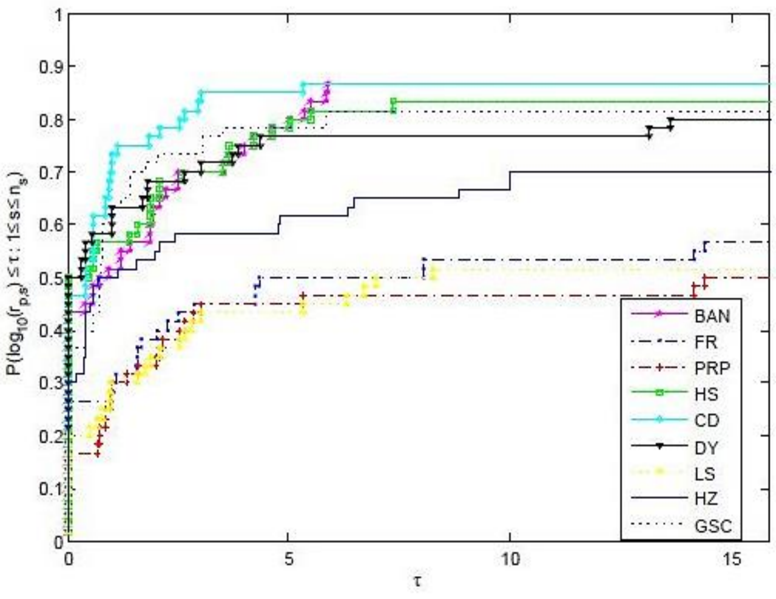

Figure 2. Performance profile for various CGMs by ESF1 based on number of iterations

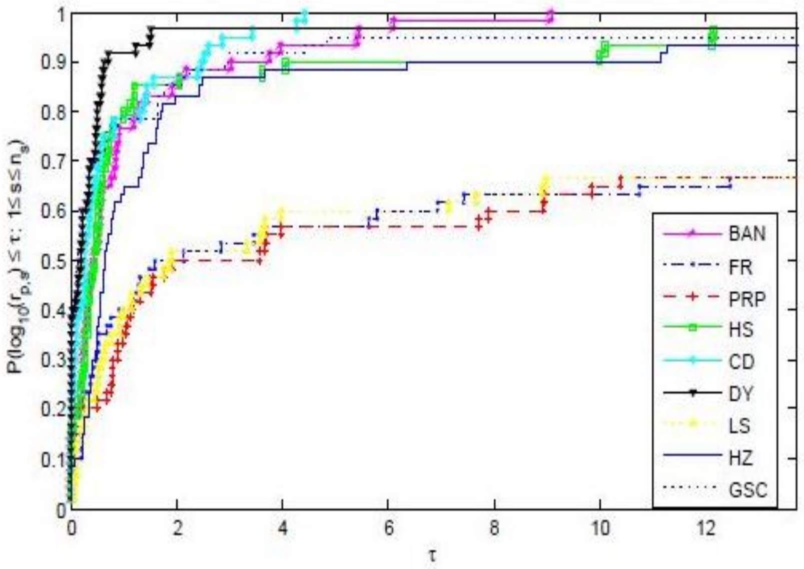

Figure 3. Performance profile for various CGMs by ESF2 based by CPU time

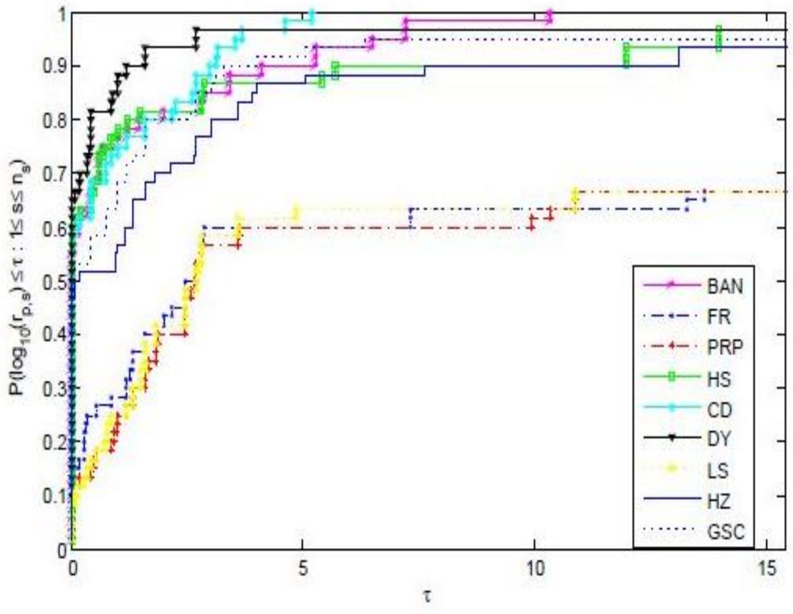

Figure 4. Performance profile for various CGMs by ESF2 by the number of iterations

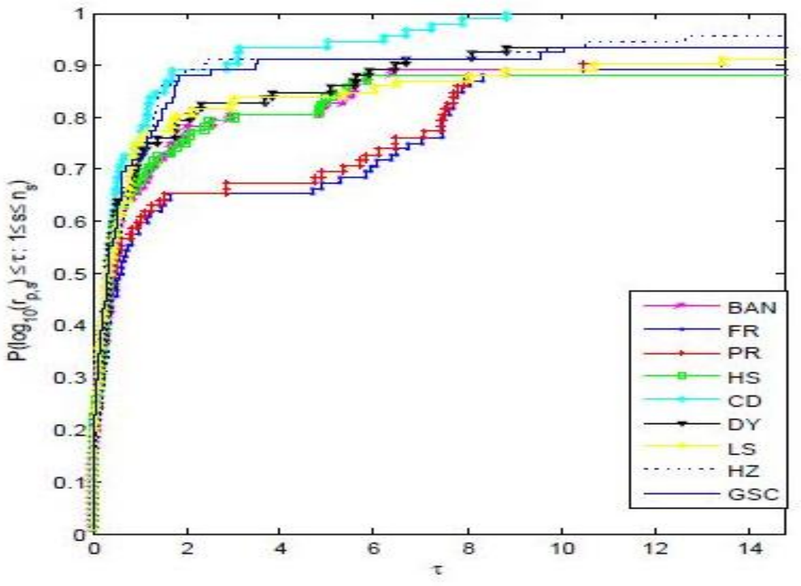

Figure 5. Performance profile various CGMs by NSF based by CPU time

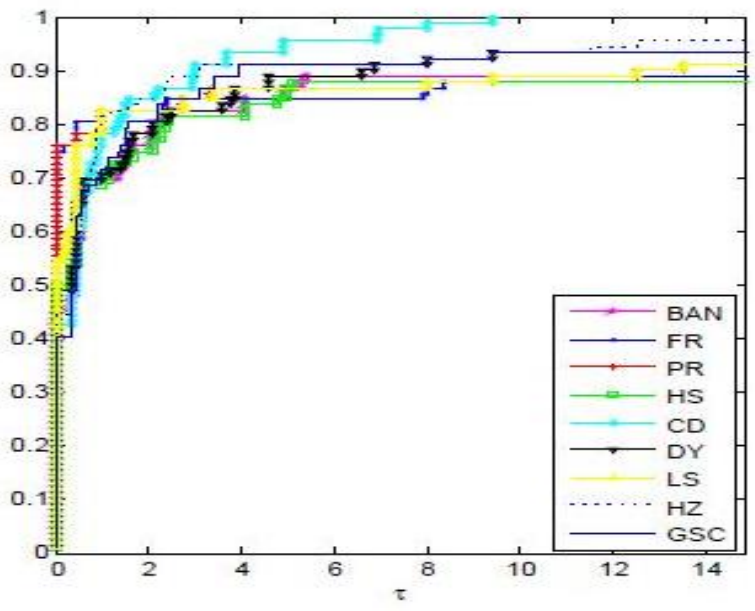

Figure 6. Performance profile for various CGMs by NSF by the number of iterations

3.3 Discussion on Numerical Results

For better description and understanding of the Figures 1-6 above, the solvability measure considering the number of iteration and the execution time for the step-length formulae (SLF) used in this work are presented below in Table 2 based on the number of successes and failures in percentage recorded by each CGM:

Table 2. Percentage Solvability Index for Different SLFs by CGMs

SSR	**Solvability Index**	**CG Methods**									**Average Success/Failure**
		BAN	**FR**	**PRP**	**HS**	**CD**	**DY**	**LS**	**HZ**	**GSC**	
ESF1	Success	88	58	50	84	88	80	52	70	82	72.4
	Failure	12	42	50	16	12	20	48	30	28	27.6
ESF2	Success	100	67	67	96	100	96	67	93	94	86.7
	Failure	00	33	33	04	00	04	33	07	06	13.3
NSF	Success	94	89	82	89	100	94	92	96	94	92.2
	Failure	06	11	18	11	00	06	08	04	06	7.8

Table 2 which measures the effectiveness of each CGM and the various SLFs presented in our work. The newly derived formula NSF displaced efficient performance over ESF1and ESF2 considering the number of iteration and the execution time taken in solving each of the thirty test functions using nine distinctive CGMs. It is observed that almost all the CG methods considered attained the highest level of efficiency using the newly derived formula, and it is evident to finalize that NSF showed better efficient performance displaced over ESF2 and ESF1 judged by the calculated average as presented in Table 2 above.

4. Conclusion

In this work, nine kinds of CG methods namely BAN, FR, PRP, HS, CD, DY, LS, HZ and GSC CG methods were employed in solving thirty unconstrained optimization test functions with two different dimensions (5000 and 10000) using the newly derived formula to obtain the step-length. To test the efficiency of the new step-length formula, it was assessed and compared with already existing formulae numerically. To better investigate our observations, performance profiling software is employed to plot the performance profiles of the different methods and afterward presented our report. It is also established that the BAN CGM and CD method exhibit better efficiency computationally when compared to the other CGMs mentioned in this research work.

References

Ajimoti, A. & Bamigbola, O. M. (2016). A gradient based step-size rule for conjugate gradient methods. *Proceeding of 35th Annual Conference of the Nigerian Mathematical Society,* (pp 150-194) NMS.

Andrei, N. (2008). Unconstrained optimization test functions unpublished manuscript. Advanced Modeling and Optimization, 10, 147-161.

Bamigbola, O. M., Ali, M. M., & Nwaeze, E. (2010). An efficient and convergent method for unconstrained nonlinear optimization. A paper presented at the International Congress of Mathematicians, Hyderabad, India.

Dai, Y. & Yuan, Y. (2000). A nonlinear conjugate gradient with strong global convergence properties. *SIAM Journal on optimization,* 10, 177-182.

Dolan, E. O. & More, J. J. (2002). Benchmarking optimization software with performance profiles. Mathematical *programming*, 19, 201-213.

Fletcher, R. (1987). *Practical method of Optimization*. John Wiley, New York., 2nd edition.

Fletcher, R. & Reeves, C. M. (1964). Function minimization by conjugate gradients. *Computer Journal*, 7, 149-154.

Hager, W. W. & Zhang, H. (2005). A new conjugate gradient method with guaranteed descent and an efficient line search. *SIAM Journal on Optimization,* 16, 170-172.

Hestenes, M. R. & Stiefel, E. (1952). Methods of conjugate gradient for solving linear equations. *J. Res. Nat. Bur. Stand.,* 49, 409-436.

Jamil, M. & Yang, X. (2013). A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modeling and Numerical Optimisation,* 4(2), 150-194.

Jinhong, H. & Genjiao, Z. (2013). A conjugate gradient method without line search and the convergence analysis. *Fourth International Conference on Emerging Intelligent Data and Web Technologies* (pp 37-86).

Liu, Y. & Storey, C. (1992). Efficient generalized conjugate gradient algorithms. *Journal of Optimization Theory and Application*, 69, 129-137.

Polak, E. & Ribiere, G. (1969). Note sur la convergence de directions conjugees. *Rev. Francaise Informat Recherche Operationelle,* 16,35-43.

Sun, J. & Zhang, J. (2001). Global convergence of conjugate gradient methods without line search. *Journal of Annals of Operation Research*, 103, 161-173.

Wu, A. (2011). A nonlinear conjugate gradient method without line search. International *Conference on Computation and Information Sciences*, 4(2)150-194.

Zhang, B. (2010). Convergence of modified conjugate gradient methods without line search. *Second International Conference on Computational Intelligence and Natural Computing (CINC).*

Zhang, J. Z., Deng, N. Y., & Chen, L. H. (1999). A new quasi-newton equation and related methods for unconstrained optimization. *Journal of Optimization and Applications*, 102, 147-167.