# COMPARATIVE STUDY OF CONJUGATE GRADIENT METHODS UNDER ARMIJO LINE SEARCH

**Nurfarahi Naabihah[1], Wan Athirah[2], Fariz Norsazali[3] and Norhaslinda Zullpakkal[4*]**

*Mathematical Sciences Studies, College of Computing, Informatics and Media, Universiti Teknologi MARA (UiTM) Terengganu Branch Kuala Terengganu Campus, 21080 Kuala Terengganu, Terengganu, Malaysia*

[1]2016289556@student.uitm.edu.my, [2]2020601836@student.uitm.edu.my,
[3]2020621596@student.uitm.my, [4*]lindazullpakkal@uitm.edu.my
*(* indicates the corresponding author)*

## ABSTRACT

*The conjugate gradient (CG) method is one of the optimization methods that is often used in practical applications. The continuous and numerous studies conducted on the CG method have led to vast improvements in its convergence properties and efficiency. In this project, a few CG methods are chosen to be tested under Armijo line search based on their efficiency and robustness. These methods are tested with a set of test functions with different variable. There are four random initial points used for each test function. The number of iteration (NOI) and CPU time are evaluated in order to find the best method. Based on the results, LAMR method is known to be the best method compared to AMRI, NRMI, AMRO and MRM as it has quite good performance as well as it can solve 100% of the test functions.*

**Keywords**: *Armijo, Conjugate Gradient Method, Convergence*

Received for review: 31-03-2023; Accepted: 09-04-2023; Published: 14-04-2023

## 1. Introduction

The study of optimization has grown immensely over the past few decades. The optimization is applicable in problems arising such as in supply chain management network design and operation as well as scheduling. In mathematics, optimization is used by a system designer, system operator or human decision maker in order to supervise the process, check the results, and modify the problem when necessary (Boyd and Vandenberghe, 2004). Basically, optimization is significant to decision science and also for the analysis of physical system.

Some objective is known as variables should be acknowledged. The process to identify the objective, variables and also constraints of any particular problem is called as modeling. After the model has been formulated, an optimization algorithm is used to find its solution (Nocedal and Wright, 1999). Then, the model can be recognized whether it has succeeded in finding the best solution amongst the set of all possible solutions. The unconstrained optimization problem is defined as

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

where $f : R^n \to R$ is a continuously differentiable and $R^n$ is an n-dimensional Euclidean space.

It is commonly solved by an iterative method,

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

where $x_k$ is the current iteration point and $\alpha_k$ is the step size which can be obtained by any line search. There are two types of line search which are exact and inexact line searches. Armijo is stated to be the simplest inexact line search since it is easy to be implemented in computation procedure. The Armijo line search is written as,

$$f(x_k + \alpha_k d_k) \le f(x) + \mu \alpha_k g_k^T d_k \tag{3}$$

The $d_k$ in the iterative method refers to the search direction. The standard search direction is described as

$$d_k = \begin{cases} -g_k & \text{if } k = 0 \\ -g_k + \beta_k d_{k-1} & \text{if } k \ge 1 \end{cases}. \tag{4}$$

The $\beta_k$ in the search direction refers to the conjugate gradient (CG) coefficient. CG method are classified into classical CG, modified CG, scaled CG, hybrid CG and three-term CG. The examples of CG coefficients are listed in Table 1.

Table 1. Examples of CG Coefficients

| **Coefficients** | **Descriptions** |
|---|---|
| $\beta_k^{AMRI} = \dfrac{g_k^T(g_k) - \dfrac{\lVert g_k \rVert}{\lVert g_{k-1} \rVert}\left\lvert g_k^T g_{k-1} \right\rvert}{d_{k-1}^T(d_{k-1})}$ | Abashar et al. (2014) introduced AMRI (Abashar-Mustafa-Rivaie-Ismail). |
| $\beta_k^{NMRI} = \dfrac{g_k^T(g_k - g_{k-1})}{g_k^T(g_k - d_{k-1})}$ | Shapiee et al. (2014) introduced NMRI (Norrlaili-Rivaie-Mustafa-Ismail). |
| $\beta_k^{AMRO} = \dfrac{g_k^T\left(g_k - \dfrac{\lVert g_k \rVert}{\lVert g_{k-1} \rVert} g_{k-1}\right)}{d_{k-1}^T\left(d_{k-1} - \dfrac{\lVert g_k \rVert}{\lVert g_{k-1} \rVert} g_k\right)}$ | Abdalla et al. (2014) introduced AMRO (Abdelrahman-Mustafa-Rivaie-Omer). |
| $\beta_k^{MRM} = \dfrac{g_k^T\left(g_k - \dfrac{\lVert g_k \rVert}{\lVert g_{k-1} \rVert} g_{k-1}\right)}{g_{k-1}^T(g_{k-1}) + \left\lvert g_k^T d_{k-1} \right\rvert}$ | Hamoda et al. (2015) introduced MRM (Mohamed-Rivaie-Mustafa). |
| $\beta_k^{LAMR} = \dfrac{g_k^T\left(\dfrac{\lVert d_{k-1} \rVert}{\lVert d_{k-1} - g_k \rVert} g_k - g_{k-1}\right)}{\dfrac{\lVert d_{k-1} \rVert}{\lVert d_{k-1} - g_k \rVert}\lVert d_{k-1} \rVert^2}$ | Zull et al. (2017) introduced LAMR (Linda-Aini-Mustafa-RIvaie). |

Nowadays, the CG coefficients are widely implemented in real life applications such as regression analysis, portfolio selection, robotic motion and image restoration problems. All of these examples can be read through Zull et al. (2019), Awwal et al. (2021), Sulaiman et al. (2022), and Malik et al. (2023).

## 2. Methodology

The first stage of this study is conducting the literature study on the previous CG coefficients and line searches by referring to articles, journals, and books. Then, few CG coefficients are identified such as AMRI, NRMI, AMRO, MRM, and LAMR. These methods are tested numerically using AMD E2-1800 APU with 2GB RAM. Next, the programming code is constructed using MATLAB R2016a software. The codes produce the numerical data which measure the number of iteration and CPU time. The algorithm for MATLAB test run is illustrated as in Figure 1.

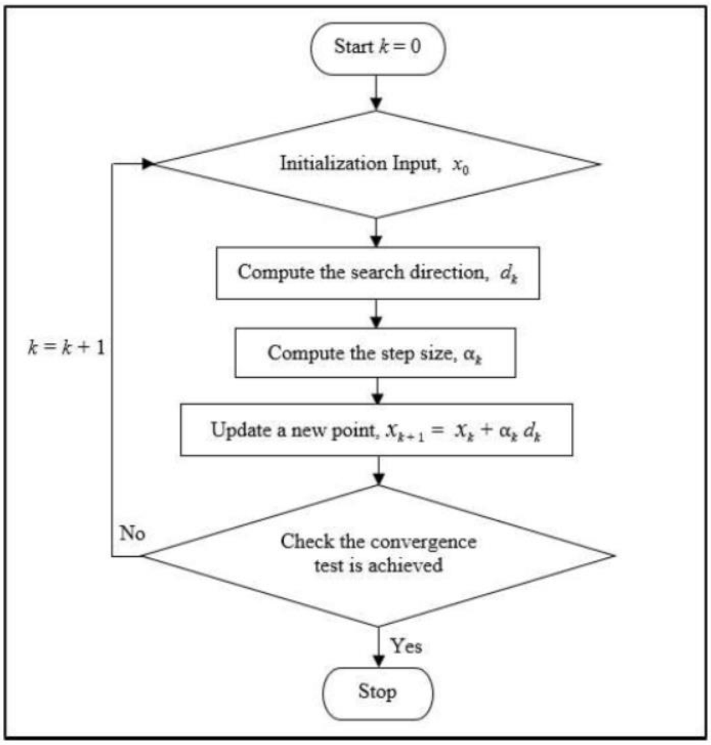


Figure 1. Algorithm

Based on Figure 1, the algorithm is stated as,

Step 1: Initialization. Given $x_0$, set $k = 0$.

Step 2: Compute each $\beta_k$ as in Table 1.

Step 3: Compute $d_k$, based on (4).

Step 4: Compute $\alpha_k$, based on (3).

Step 5: Updating a new point based on (2).

Step 6: Convergence test and stopping criteria.

If the conditions are satisfied, then stop. Otherwise go to Step 1 with $k = k+1$.

Next, the numerical test is conducted by solving ten test functions with different dimensions and initial points as in Table 2.

Table 2. List of Test Functions

| **Test Functions** | **Dimensions** | **Initial Points** |
|---|---|---|
| Six Hump | 2 | (4,4), (8,8), (20,20) |
| Dixon and Price | 2 | (8,8), (16,16), (20,20) |
| Generalized Quartic | 2 | (4,4), (8,8), (16,16) |
| Zettl | 2 | (4,4), (12,12), (20,20) |
| Extended DENSCHNB | 2,4,10 | (12,12), (16,16), (20,20) |
| FLETCHCR | 2,4,10 | (4,4), (8,8), (20,20) |
| Perturbed | 2,4,10 | (4,4), (8,8), (12,12) |
| Extended Himmelblau | 2,4,10 | (4,4), (12,12), (16,16) |
| Diagonal 4 | 2,4,10 | (4,4), (8,8), (12,12) |
| Shallow | 2,4,10 | (4,4), (8,8), (12,12) |

Lastly, the results are discussed in terms of the efficiency and robustness of CG algorithms. The performance of the CG algorithms is illustrated in performance profile. The performance profile by Dolan and More (2002) is extensively used in the field of optimization due to its ability to provide detailed results. This method defines the performance profile of a solver as a cumulative distribution function for a performance metric where it uses the ratio of the certain solver versus the best time of all solvers. The performance profile is generated by evaluating and comparing the performance of the set of solvers *S* on a test set *P* of problems according to information data as NOI and CPU times. Then, define that $S_p = \{s \in S\}$, where s successfully solves *p*. Let $n_s$ be the number of solvers while $n_p$ is the number of each problem. For each problem *p* and solver *s*,

$$t_{p,s} = \text{computing time required by solver s to solve problem p.}$$

The performance ratio is used as the baseline for the comparisons by evaluating the performance of solvers *s* in order to solve problem *p* with the best performance of any solver on a similar problem. Then, the performance ratio is specified by

$$\tau_{p,s} = \frac{t_{p,s}}{\min\left\{t_{p,s} : s \in S_p\right\}}$$

where the parameter $r_m \geq r_{p,s}$ for all *p*, *s* and $r_m = r_{p,s}$ if and only if solver *p* cannot be solved by solver *s*. Then it is conveyed as:

$$p_s(\tau) = \frac{size\left\{p \in P : r_{p,s} \leq \tau\right\}}{n_p}$$

which $p_s(\tau)$ is the probability for solver $s \in S$ of the performance ratio, $r_{p,s}$ is within a factor $\tau \in \mathbb{R}$ of the best possible ratio. It is clearly shown that the function $p_s$ is the cumulative distribution function for the performance ratio.

In general, the solvers with large probability $p_s(\tau)$ are preferable if the set of problems $p$ is suitably large. The performance profile $p_s : \mathbb{R} \rightarrow [0,1]$ for a solver is a piecewise constant function, continuous and non-decreasing from the right at each breakpoint. In any case, the best solver will be acknowledged by winning over the rest of the solvers with the highest value of $p_s(\tau)$ or reaching the value of $p_s(1)$.

## 3. Results and Discussion

The results for each algorithm, AMRI, NRMI, AMRO, MRM and LAMR are analyzed based on the NOI and CPU times. All these CG algorithms are tested under Armijo line search. Four random initial points are applied for each test function. The numerical results are interpreted into performance profile as in Figures 2 and 3.

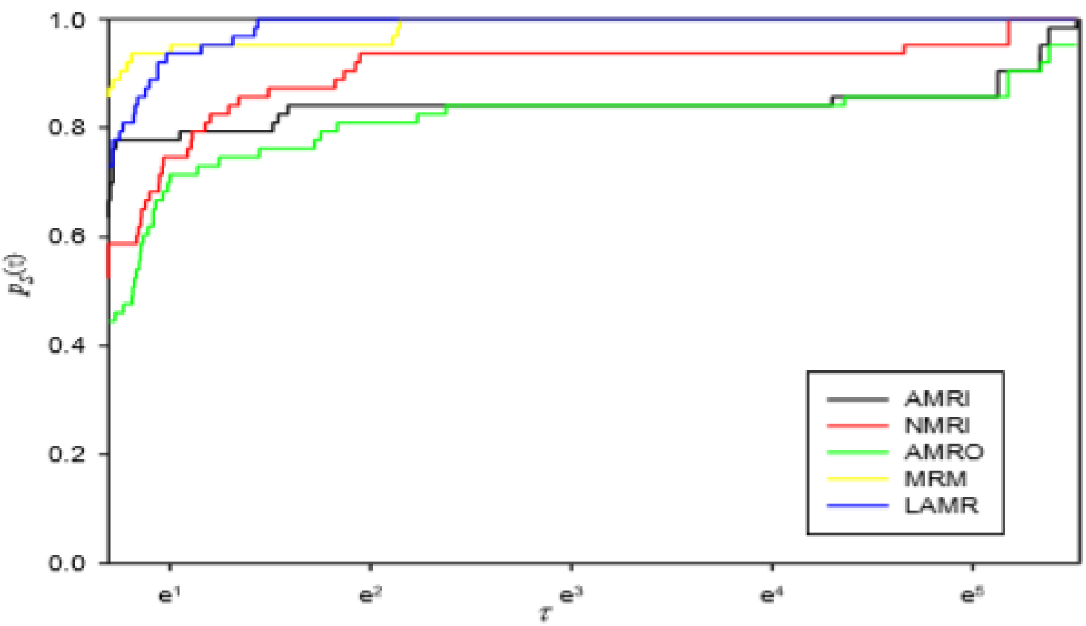


Figure 2. Performance Profile based on Number of Iterations.

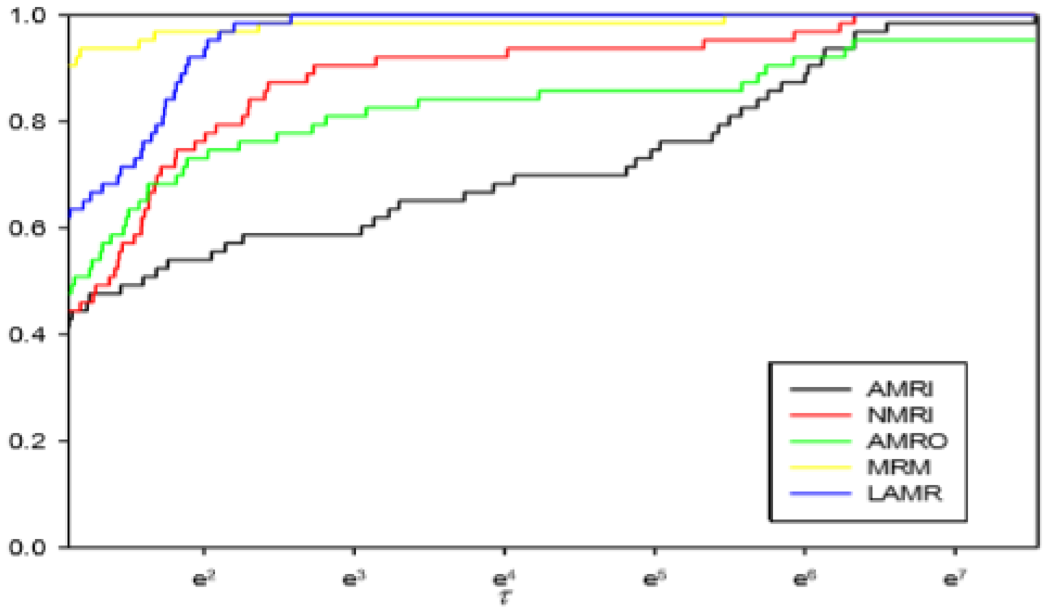


Figure 3. Performance Profile based on CPU time.

The most robust method is determined from the curve that is located to the top right of the figure while the method with the fastest convergence is determined from the top curve at the left side of the figure. The robustness of the method is validated by the number of test functions solved. Figure 2 and 3 shows similar curves. Based on both graphs, MRM method is the most efficient as it has the highest convergence rate followed by LAMR, AMRI, NRMI and AMRO method respectively. However, at certain point, LAMR method shows a significant improvement, and it shows that LAMR method is better than MRM method. From the figures, AMRI, NRMI, MRM and LAMR moves to line 1 at the top right of the figure which shows that these methods are able to solve 100% of the test functions while AMRO shows no improvement and did not move to line 1 alongside with other methods as it only solve 95.2% of the test functions.

## 4. Conclusion and Recommendations

It can be concluded that LAMR is the best method based on NOI and CPU time since it overtakes other CG coefficients. Besides, this method is able to solve all of the test functions. This LAMR coefficient can be suggested to hybrid with other CG coefficients for a better result. This method can be applied in image restoration or robotic motion problems.

**Acknowledgement**

The authors would like to thank the editors and the referees for the suggestions and comments. The authors acknowledge Mathematical Sciences Studies, College of Computing, Informatics and Media, Universiti Teknologi MARA for supporting this research.

## References

Abashar, A., Mamat, M., Rivaie, M., Mohd, I., & Omer, O. (2014, June). The proof of sufficient descent condition for a new type of conjugate gradient methods. In *AIP Conference Proceedings* (Vol. 1602, No. 1, pp. 296-303). American Institute of Physics.

Abdalla, A. A., Mamat, M., Rivaie, M., & Omer, O. (2014). Global convergence analysis of a new nonlinear conjugate gradient coefficient with strong Wolfe line search. *Journal of Quality Measurement and Analysis*, 10(1), 75–85.

Awwal, A. M., Sulaiman, I. M., Malik, M., Mamat, M., Kumam, P., & Sitthithakerngkiet, K. (2021). A spectral RMIL+ conjugate gradient method for unconstrained optimization with applications in portfolio selection and motion control. *IEEE Access*, 9, 75398-75414.

Boyd, S., Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91, 201-213.

Hamoda, M., Rivaie, M., Mamat, M., & Salleh, Z. (2015). A conjugate gradient method with inexact line search for unconstrained optimization. *Applied Mathematical Sciences*, 9, 1823–1832.

Malik, M., Sulaiman, I. M., Abubakar, A. B., & Ardaneswari, G. (2023). A new family of hybrid three-term conjugate gradient method for unconstrained optimization with application to image restoration and portfolio selection. *AIMS Mathematics*, 8(1), 1-28.

Nocedal, J., & Wright, S. J. (Eds.). (1999). *Numerical optimization*. New York, NY: Springer New York.

Shapiee, N., Rivaie, M., Mamat, M., & Mohd, I. (2014, June). A new modification of Hestenes-Stiefel method with descent properties. In *AIP Conference Proceedings* (Vol. 1602, No. 1, pp. 520-526). American Institute of Physics.

Sulaiman, I. M., Malik, M., Awwal, A. M., Kumam, P., Mamat, M., & Al-Ahmad, S. (2022). 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*, 2022(1), 1-22.

Zull, N., Aini, N., Rivaie, M., & Mamat, M. (2019). A new gradient method for solving linear regression model. *International Journal of Recent Technology and Engineering*, 7(5), 624-630.

Zull, N., Shoid, S., Ghani, N. H. A., Mohamed, N. S., Rivaie, M., & Mamat, M. (2017, September). A conjugate gradient method with descent properties under strong Wolfe line search. In *Journal of Physics: Conference Series* (Vol. 890, No. 1, p. 012105). IOP Publishing.