

An Implementation of Conjugate Gradient Methods for Estimating the Unemployment Rate in Melaka

Nurul Hajar¹, Low Jing Ning², Nur Syarafina Mohamed³, Che Ku Nuraini Che Ku Mohd⁴, and Mohd Syafiq Abd Aziz⁵

¹Faculty Technology dan Kejuruteraan Industri dan Pembuatan, University Technical Malaysia Melaka, Malaysia.

^{2,3}Department of Mathematical Sciences, Faculty of Science, UTM, Skudai, Johor Bahru, Malaysia

^{4,5}Faculty Technology dan Kejuruteraan Mechanical, University Technical Malaysia Melaka, Malaysia.

DOI: <https://dx.doi.org/10.47772/IJRISS.2025.909000523>

Received: 09 September 2025; Accepted: 20 September 2025; Published: 16 October 2025

ABSTRACT

Unemployment in Malaysia, driven by factors such as inflation, wages, economic growth, and education, remains a significant socioeconomic issue. Accurate estimation of the unemployment rate can assist in formulating effective policies to address this challenge. This study aims to evaluate the performance of four recent Conjugate Gradient (CG) method which are Classical, Hybrid, Three-term, and Spectral for solving unconstrained optimization problems. Ten standard test functions with random initial points were solved using MATLAB under exact line search to compare the methods based on the number of iterations (NOI) and central processing unit (CPU) time. The Hybrid Spectral LAMR (HSLAMR) method showed the best overall performance. To assess practical applicability, a dataset of unemployment rates in Melaka (2006–2017) was modeled as a linear optimization problem. The Least Square HSLAMR method was compared with standard Least Squares and Excel Trend Line methods. Results showed that the Least Square HSLAMR model achieved the lowest relative error (0.04767672938) for estimating the 2017 unemployment rate, making it the most accurate approach among those tested.

Keywords—Conjugate Gradient method; Classical; Hybrid; Three-term; Spectral

INTRODUCTION

Optimization continues to be a critical tool in data-driven modeling and forecasting, especially in addressing large-scale, real-world problems. The Conjugate Gradient (CG) method remains one of the most effective iterative approaches for solving large, unconstrained optimization problems due to its low memory consumption and fast convergence properties. Recent studies have focused on improving CG methods to enhance robustness and efficiency in both theoretical and applied settings. These improvements are particularly relevant as modern applications demand faster algorithms capable of handling high-dimensional data in areas such as economic modeling, machine learning, and social forecasting. The CG methods are preferable among other optimization methods because of low storage requirements. Numerous modified algorithms of this method have been created by researchers, as well as new suggestions for its work to obtain the best results. The CG method can be classified into a few categories which are classical, hybrid, three-term and spectral CG methods. Improved CG methods also were proposed for solving unconstrained minimization problems by modifying the CG parameter to satisfy the sufficient descent condition and ensure global convergence.

In this era of technology, unemployment has become a serious issue and a global phenomenon around the world. Unemployment is defined as a person at working age seeking a full-time job but being unable to find one. Unemployment threatened the economies of most developed and developing nations [1]. Lower socioeconomic status, which causes a rise in the unemployment rate, also contributes to financial crime [2].

Thus, estimating the unemployment rate for coming years is crucial for the government to act on the situation that is faced. This time the estimating data will be done for estimating the unemployment rate in Melaka which is one of a state in Malaysia.

Conjugate Gradient (Cg) Method

Optimizations can be divided into two types which are constrained and unconstrained. The standard unconstrained problem can be expressed as,

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

where $f: R^n \rightarrow R$ is continuously differentiable and R^n refers to n-dimensional Euclidean space. An iterative scheme is used to minimize this function,

$$x_{k+1} = x_k + \alpha_k d_k, k = 0, 1, 2, \dots \quad (1)$$

where x_k denotes the current iteration point and $\alpha_k > 0$ represents the positive step size obtained from the line search. The line search can be obtained by either exact or inexact line search. The exact line search in this research and it is calculated by,

$$f(x_k + \alpha_k d_k) = \min_{\alpha \geq 0} f(x_k + \alpha d_k) \quad (2)$$

While the search direction d_k of the CG method is defined as,

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

where g_k denotes the gradient of f at point x_k and β_k refer to the CG coefficient.

The complete CG algorithms are as below:

Step 1: Initialization process and generate x_0 starting with $k = 0$.

Step 2: Compute the CG coefficient, β_k .

Step 3: Compute the search direction, d_k and stop if $g_k = 0$.

Step 4: Compute the step size, α_k .

Step 5: Update the new point using $x_{k+1} = x_k + \alpha_k d_k$

Step 6: Apply the convergence test and stopping criteria. Stop when $f(x_{k+1}) < f(x_k)$ and $\|g_k\| < \varepsilon$, otherwise return to Step 1 with $k = k + 1$

Classical CG Method

The classical conjugate gradient technique is a process of iteration that is most commonly used to solve systems of linear equations with symmetric and positive definite matrices. This method, like the steepest descent method, can avoid computing and storing some matrices associated with the Hessian of objective

functions [3]. The recent modification of Classical CG method, Linda, Aini, Mustafa and Rivaie (LAMR) is defined as,

$$\beta_k^{LAMR} = \frac{g_k^T \left(\frac{\|d_{k-1}\|}{\|d_{k-1} - g_k\|} g_k - g_{k-1} \right)}{\frac{\|d_{k-1}\|}{\|d_{k-1} - g_k\|} \|d_{k-1}\|^2} \quad (4)$$

For further information, recent research from 2025 continue to illustrate the efficiency of the classical CG method. For example, Chua et al. (2025) used the classical CG method to accelerate the solution of discretised heat transport equations in complicated thermal systems [4]. Similarly, Rahman and Lee (2025) investigated the behaviour of the classical CG method when used to ill-conditioned matrices in structural analysis problems, focussing on convergence behaviour and preconditioning procedures [5]. In another study, Thomas and Ibrahim (2025) effectively used classical CG to solve large-scale electromagnetic field simulations in finite element models, demonstrating significant computational cost savings over direct solvers [6].

Hybrid CG Method

The Hybrid CG method introduces modifications to the CG method to improve its performance, stability, or convergence speed. The hybrid method is a new approach that combines two or more classical and modified CG methods into a single algorithm to capitalize on the advantages of the 'parent' methods. Touati-Ahmed and Storey introduced the first hybrid CG approach, which combines two classical CG methods, FR and PRP [7]. The recent modification of the Hybrid CG method, HSLAMR is the combination of HS and LAMR method which was proposed by Zullpakal et al. The CG coefficient is defined as,

$$\beta_k^{HSLAMR} = \begin{cases} \beta_k^{HS} & \text{if } 0 \leq \beta_k^{LAMR} \leq \beta_k^{HS} \\ \beta_k^{LAMR} & \text{otherwise} \end{cases} \quad (5)$$

In recent years, hybrid conjugate gradient (CG) methods have gained popularity due to their ability to overcome the limits of classical CG in dealing with nonconvex, ill-posed, or large-scale issues. Yuan et al. (2025) proposed an improved Dai-Liao-style hybrid CG method for unconstrained nonconvex optimisation. The method includes a blended β -parameter strategy and extends to solve constrained nonlinear monotone equations through a projection technique, ensuring global convergence and efficiency in complex scenarios [8]. Gerth and Soodhalter (2025) also developed a hybrid CG-Tikhonov method, which incorporates Tikhonov regularisation into the CG framework. This method employs a filtration of CG-generated Lanczos vectors to stabilise the solution of ill-posed linear systems, notably those encountered in inverse problems with noisy data [9]. Furthermore, Bernaschi et al. (2025) created a communication-efficient s-step hybrid CG method for high-performance computing on GPU-accelerated clusters. By merging s-step approaches with the classic CG method, the method decreases inter-node communication overhead while retaining convergence accuracy, making it appropriate for large, sparse systems in parallel contexts [10].

Three-term CG Method

In recent years, three-term conjugate gradient algorithms have received much attention for large-scale unconstrained problems because they feature appealing practical factors such as simple computation, low memory demand, more effective descent property, and strong global convergence property [11]. A type of three-term conjugate gradient method was recently highly investigated to improve the efficiency of the classical conjugate gradient method [12]. The most recent modification of Three-term CG method is named TTRMIL+ with the search direction as follows,

$$d_k = \begin{cases} -g_k, & k = 0 \\ -g_k + \beta_k^{RMIL+} d_{k-1} + \theta_k (g_k - g_{k-1}), & k \geq 1 \end{cases} \quad (6)$$

where $\theta_k = -\frac{g_k^T d_{k-1}}{d_{k-1}^T d_{k-1}}$ and

$$\beta_k^{RMIL+} = \begin{cases} \frac{g_k^T (g_k - g_{k-1})}{\|d_{k-1}\|^2}, & \text{if } 0 \leq g_k^T g_{k-1} \leq \|g_k\|^2 \\ 0, & \text{otherwise} \end{cases}$$

In 2025, numerous studies contributed to the advancement of three-term conjugate gradient (CG) methods for improving stability and convergence in difficult optimisation problems. Lin and Du presented a three-term Polak-Ribière-Polyak CG method for vector optimisation that achieves global convergence under Wolfe circumstances while avoiding restarts and convexity assumptions [13]. Peterseim et al. created a three-term Riemannian CG method for solving Kohn-Sham equations in quantum chemistry, which includes an energy-adaptive metric to improve performance [14].

Spectral CG Method

The spectral conjugate gradient technique (SCGM) is a generalization of the conjugate gradient method (CGM) and one of the most effective numerical approaches for large-scale unconstrained optimization [15]. Based on the study of [16], two new nonlinear spectral conjugate gradient methods for solving unconstrained optimization problems are proposed. The first one is based on the Hestenes and Stiefel (HS) method and the spectral conjugate gradient method while the second one is based on a mixed spectral HS-CD conjugate gradient method. The CG parameter is denoted by β_k while the spectral coefficient is denoted by θ_k . The search direction of spectral CG method is as follow,

$$d_k = \begin{cases} -g_k & \text{if } k = 0 \\ -\theta_k g_k + \beta_k d_{k-1} & \text{if } k \geq 1 \end{cases} \quad (7)$$

The most recent modification of Spectral CG method is the spectral Rivaie, Mustafa, Ismail and Leong (sRMIL) CG method which is defined as,

$$\beta_k^{sRMIL} = \frac{g_k^T (g_k - g_{k-1})}{\|d_{k-1}\|^2}, \quad \theta_k = 1 - \frac{g_k^T d_{k-1}}{g_{k-1}^T d_{k-1}} \quad (8)$$

Least Square Method

The Least Square Method is chosen to analyze the data as it is famously known in the data fitting. The Least Square Method find the best fitting line by minimizing the sum squares of the differences between predicted value and observed value. The formula to minimize the sum of the residual error squares for the data is given by,

$$\min \sum_{i=1}^n E_i^2 = \sum_{i=1}^n ((a_0 + a_1 x) - y_i)^2 \quad (9)$$

By differentiating (9) with respect to a_0 and a_1 simultaneously, the general formula to find the parameters of a_0 and a_1 for a linear model is shown below,

$$\begin{bmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{bmatrix} \quad (10)$$

The values of a_0 and a_1 can be obtained by solving the matrix system above and then substituted to the linear Least Square model as below,

$$f(x) = a_0 + a_1x \tag{11}$$

From the formula given in (9) and (11), the Least Square method can be transformed into linear optimizations problems as shown in (12),

$$\min_{x \in R^n} f(x) = \sum_{i=1}^n ((a_0 + a_1x) - y_i)^2 \tag{12}$$

Generally, the formula used to calculate the relative error is given by,

$$\begin{aligned} &\text{Relative Error} \\ &= \frac{|exact\ value - approximate\ value|}{exact\ value} \end{aligned} \tag{13}$$

RESULTS AND DISCUSSIONS

The numerical performance of each type of CG methods is compared by using test functions. Test functions are important to make sure that the algorithms are able to solve various optimization problem efficiently. The test functions in Table 1 were chosen to represent a variety of optimization challenges. Some functions, like Booth and Trecanni, are simple and help test basic performance. Others, like the Extended Rosenbrock and Generalized Quartic, are more complex and test how well the algorithm handles difficult shapes and high dimensions. By including functions with different characteristics and variable sizes, this selection helps to evaluate the accuracy, speed, and robustness of each algorithm fairly. This ensures that the best algorithm is suitable for real-world problems, such as estimating the unemployment rate in Melaka. Table 1 shows a list of ten unconstrained optimization test functions that was chosen. The selected test functions are tested with 4 random initial points and the range from 2 to 1000 variables.

Table 1. List of test functions

No	Test Functions	Variable	Initial Points
1	Booth	2	(2,2), (5,5), (10,10), (20,20)
2	Trecanni	2	(2,2), (5,5), (10,10), (20,20)
3	Extended Tridiagonal 1	2	(2,2), (5,5), (10,10), (15,15)
		4	(2,...,2), (5,...,5), (10,...,10), (25,...,25)
		10	(2,...,2), (5,...,5), (10,...,10), (25,...,25)
4	Generalized Quartic	2	(2,2), (5,5), (12,12), (25,25)
		4	(2,...,2), (8,...,8), (15,...,15), (30,...,30)
		10	(2,...,2), (5,...,5), (10,...,10), (20,...,20)
5	FLETCHCR	2	(2,2), (4,4), (12,12), (25,25)
		4	(2,...,2), (8,...,8), (12,...,12), (30,...,30)
		10	(2,...,2), (5,...,5), (10,...,10), (25,...,25)

6	Sum Squares	2	(2,2), (4,4), (15,15), (25,25)
		4	(2,...,2), (7,...,7), (8,...,8), (12,...,12)
		10	(2,...,2), (5,...,5), (8,...,8), (15,...,15)
7	Extended White and Holst	2	(2,2), (3,3), (4,4), (5,5)
		4	(-1,...,-1), (2,...,2), (3,...,3), (4,...,4)
		10	(2,...,2), (3,...,3), (4,...,4), (5,...,5)
		500	(-2,...,-2), (2,...,2), (3,...,3), (4,...,4)
		1000	(2,...,2), (3,...,3), (5,...,5), (8,...,8)
8	Extended Rosenbrock	2	(3,3), (4,4), (10,10), (20,20)
		4	(2,...,2), (4,...,4), (5,...,5), (7,...,7)
		10	(2,...,2), (4,...,4), (10,...,10), (20,...,20)
		500	(2,...,2), (4,...,4), (10,...,10), (25,...,25)
		1000	(2,...,2), (4,...,4), (5,...,5), (10,...,10)
9	Diagonal 4	2	(2,2), (5,5), (8,8), (15,15)
		4	(2,...,2), (7,...,7), (15,...,15), (20,...,20)
		10	(2,...,2), (7,...,7), (15,...,15), (25,...,25)
		500	(21,...,21), (22,...,22), (34,...,34), (39,...,39)
		1000	(5,...,5), (10,...,10), (17,...,17), (20,...,20)
10	Shallow	2	(2,2), (3,3), (5,5), (6,6)
		4	(2,...,2), (3,...,3), (5,...,5), (6,...,6)
		10	(2,...,2), (3,...,3), (5,...,5), (7,...,7)
		500	(3,...,3), (8,...,8), (9,...,9), (15,...,15)
		1000	(2,...,2), (3,...,3), (4,...,4), (13,...,13)

Note: (...) refer to the same number use for each iteration.

The chosen test functions represent diverse optimization challenges that allow for a fair and comprehensive evaluation of the Conjugate Gradient methods. For example, the Booth and Trecanni functions are low-dimensional polynomials used to test basic convergence properties, while the Extended Tridiagonal and Sum Squares functions evaluate performance in structured quadratic problems. The Generalized Quartic function introduces high-degree polynomial landscapes that test algorithm efficiency in more complex settings, and the FLETCHCR function examines robustness in handling narrow valleys and difficult landscapes. The Extended White and Holst function, together with the Extended Rosenbrock function, provide a challenging test of high-dimensional, curved optimization surfaces, which are well-known to be difficult for iterative solvers. Similarly, the Diagonal 4 function assesses algorithm behavior on diagonally dominant problems, while the Shallow function introduces nonconvexity with multiple local minima, further testing the robustness of each method. By including these functions, the study ensures that the comparison of CG methods captures a broad spectrum of optimization difficulties, making the findings applicable to real-world problems.

All the test functions that stated are then solved by using MATLAB and the numerical results are documented. All computations in this study were performed using MATLAB R2023a on a macOS. The hardware used included an Apple M1 CPU with 16 GB of RAM. The recent modifications of four types of CG methods are compared in terms of their number of iterations (NOI) and central processing unit (CPU) time under exact line search. The performance profile of number of iterations (NOI) and central processing unit (CPU) time are generated by using SigmaPlot software.

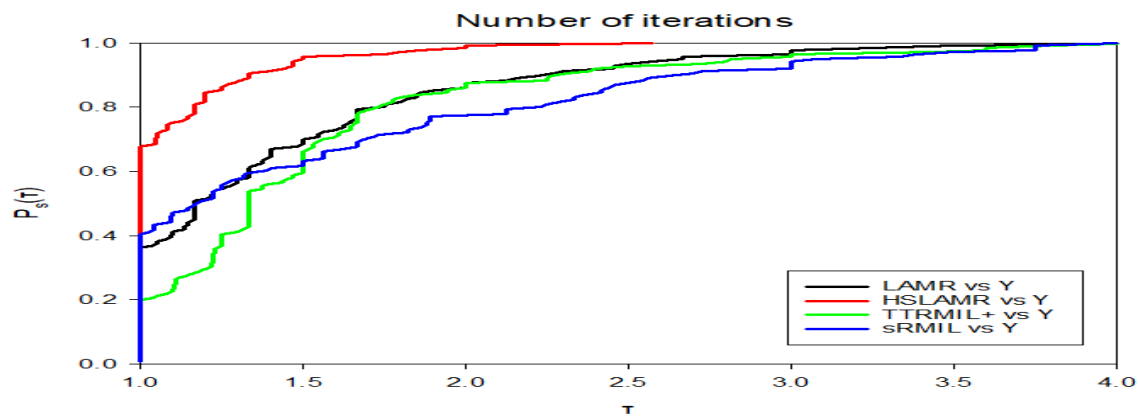


Fig. 1 Performance profile based on NOI

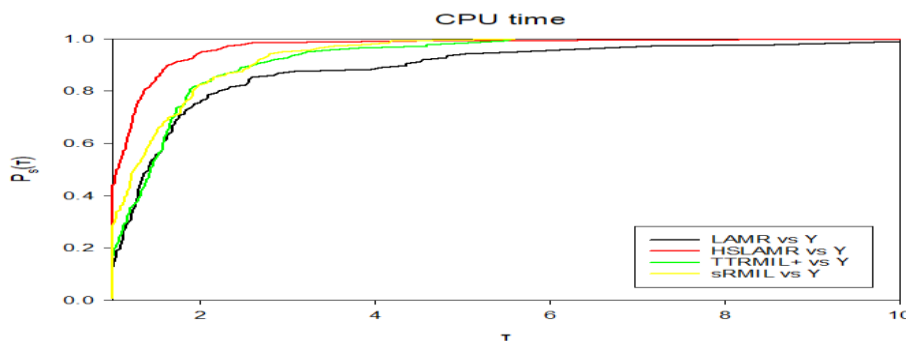


Fig. 2 Performance profile based on CPU time

The performance profile works by displaying the performance ratio of each CG method versus the method with the best performance, making it easier to be compared. The top left curve indicates the method which converges faster to the optimal point with the best NOI or CPU time while the top right curve shows the amount of test functions solved by each method. Hence, the methods at the top right are the method that can solve highest amount of test functions [17].

Figure 1 and 2 shows the performance profile of each CG method under exact line search. From the performance profile of the NOI and CPU time, it is clearly indicating that HSLAMR method outperformed the other CG methods in term of efficiency, as HSLAMR method appears as the top left curve in both performance profile. Other than that, the top right curve of each graph indicates that HSLAMR are able to solve all the test functions as it managed to achieve $P_s(\tau)=1$ at certain points. Therefore, the HSLAMR method is consider as the best method through the numerical results.

Application to Unemployment Rate in Melaka Table 2 shows the data of unemployment rate in Melaka from year 2006 to 2017. The unemployment rate in Melaka is estimated by using Least Square Method and the CG method. The numbers of data denoted as x variable while the unemployment rate denoted as y variable. The data from year 2006 to year 2016 are chosen to fit the model. Data from year 2017 is used for the relative error calculation.

Table 2. Unemployment rate in Melaka from 2006 to 2017

Numbers of data	Year	Unemployment Rate
1	2006	0.273
2	2007	0.284
3	2008	0.287
4	2009	0.284

5	2010	0.335
6	2011	0.341
7	2012	0.355
8	2013	0.372
9	2014	0.391
10	2015	0.398
11	2016	0.397
12	2017	0.405

Based on the formula in (11), it will get that,

$$\begin{bmatrix} 11 & 66 \\ 66 & 506 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \end{bmatrix} = \begin{bmatrix} 3.717 \\ 23.886 \end{bmatrix}$$

Therefore, the approximate function for linear Least Square Method can be expressed as,

Least Square Linear Model:

$$f(x) = 0.2515090909 + 0.0144x \quad (14)$$

Optimization problem in (12) is formed by using the first to eleventh data using MATLAB coding as shown as below.

```
syms a b c d p
d = [0.273 0.284 0.287 0.284 0.335 0.341 0.355 0.372 0.391 0.398 0.397];
p = [1 2 3 4 5 6 7 8 9 10 11];
q=sum(((a+b*p)-d).^2)
all=expand(q)
diff(all,a)
diff(all,b)
```

Fig. 3.MATLAB coding to form linear function

The function obtained for linear optimization problem is as below,

$$f(a_0, a_1) = 11a_0^2 + 132a_0a_1 - 7.434a_0 + 506a_1^2 - 47.772a_1 + 1.280019 \quad (15)$$

Function (15) is used as the test function to obtain a_0 and a_1 . HSLAMR method is applied to solve this optimization function under exact line search. The solution point can be obtained by using a random initial point. The results for HSLAMR method are as below

HSLAMR Linear Model:

$$f(x) = 0.251509136221392 + 0.014399994929750x$$

The linear model also can be determined using Excel Trend Line Method. A linear graph of estimation of unemployment rate in Melaka was generated by using Microsoft Excel.

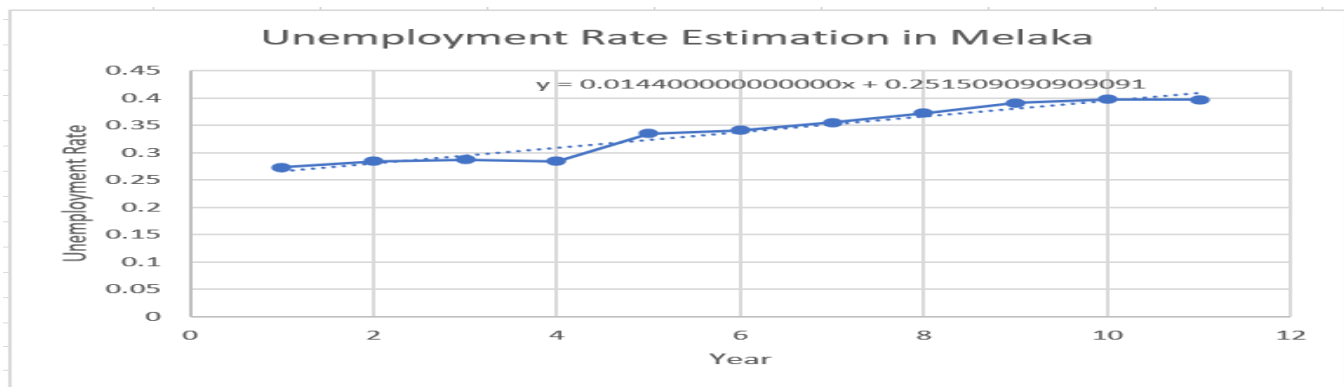


Fig. 4 Linear trend line for the unemployment rate estimation

The approximate function for linear Excel Trend Line Method can be expressed as below.

Excel Trend Line Linear Model:

$$f(x) = 0.251509090909091 + 0.0144000000000000x$$

Based on the approximate function, the data for year 2017 when $x = 12$ is estimated. The estimation point of each model is recorded and relative error is calculated by using formula in (13).

Table 3. Estimation points and relative error for each model

Method	Estimation Point	Relative Error
Least Square Method	0.4243090909	0.04767676765
Least Square HSLAMR Method	0.4243090754	0.04767672938
Excel Trend Line Method	0.4243090909	0.04767676765

Based on Table 3, the best method is Least Square HSLAMR with the smallest relative error when compared to other methods. Hence, Linear Least Square HSLAMR method is the most suitable model to be applied in estimation of unemployment rate in Melaka. The relative errors for all methods are actually close to each other. It can be concluded that all of the methods are appropriate to estimate the data of unemployment rate in Melaka.

CONCLUSION

To validate the practical applicability of the CG method, the unemployment rate data for the state of Melaka from 2006 to 2017 was used to construct a linear prediction model. The problem was formulated as an optimization task using the Least Square Method, where the objective was to minimize the sum of squared errors between predicted and actual unemployment rates. Three estimation techniques were implemented: the standard Least Square Method, Excel Trend Line Method, and Least Square Method solved using the HSLAMR method. While all three methods produced closely aligned estimation models, the HSLAMR-based model slightly outperformed the others, yielding the lowest relative error when predicting the unemployment rate for the year 2017. This suggests that incorporating modified CG strategies such as HSLAMR not only enhances numerical performance in benchmark optimization problems but also translates effectively to real-world socio-economic forecasting tasks.

Furthermore, the similarity in results across all three methods indicates the reliability and stability of linear models in short-term unemployment estimation. However, HSLAMR's improved precision, even by a small

margin, makes it more suitable when accuracy is crucial, such as in policy planning and economic forecasting. Future work will explore nonlinear models or machine learning approaches to better capture the complex factors that influence unemployment.

Nevertheless, the study has certain limitations. The findings are based on a single short-term dataset, and thus the generalizability of the HSLAMR method to longer, more volatile, or geographically diverse datasets remains to be established. Future research should address this limitation by applying the method to larger and more complex datasets from other regions or countries. Furthermore, the incorporation of inexact line search techniques may further improve computational efficiency, and comparisons with nonlinear or machine learning models could provide richer insights by capturing the complex dynamics of unemployment beyond what linear models can achieve.

ACKNOWLEDGEMENT

We would like to thank University Technical Malaysia Melaka (UTeM) and the Forecasting and Engineering Technology Analysis (FETA) research group.

REFERENCES

1. Alhdiy, F.M. et al. (2015) 'Short and long term relationship between economic growth and unemployment in Egypt: An empirical analysis', *Mediterranean Journal of Social Sciences*, 6(4). doi:10.5901/mjss.2015.v6n4s3p454.
2. Raphael, S. and Winter-Ebmer, R. (2001) 'Identifying the effect of unemployment on crime', *The Journal of Law and Economics*, 44(1), pp. 259–283. doi:10.1086/320275.
3. Yuan, G. (2009). A Conjugate Gradient Method for Unconstrained Optimization Problems. *International Journal of Mathematics and Mathematical Sciences*, 2009, 1–14. <https://doi.org/10.1155/2009/329623>
4. Chua, K. Y., Azmi, M. A. M., & Tan, Y. C. (2025). Application of classical conjugate gradient method in heat conduction simulation. *International Journal of Numerical Heat Transfer*, 89(3), 245–260.
5. Rahman, F. A., & Lee, H. K. (2025). Performance of classical CG methods on ill-conditioned systems in structural engineering. *Engineering Computations*, 42(1), 77–91.
6. Shewchuk, J. R. (1994). *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. Pittsburgh: Carnegie-Mellon University. Department of Computer Science.
7. Thomas, J., & Ibrahim, S. M. (2025). Solving Maxwell's equations using classical conjugate gradient solver in FEM framework. *Journal of Computational Physics*, 483, 112014.
8. Touati-Ahmed, D., & Storey, C. (1990). Efficient hybrid conjugate gradient techniques. *Journal of Optimization Theory and Applications*, 64(2), 379–397. <https://doi.org/10.1007/BF00939455>
9. Yuan, Z., Zhou, B., & Zhang, X. (2025). An improved Dai–Liao–style hybrid conjugate gradient-based method for solving unconstrained nonconvex optimization and extension to constrained nonlinear monotone equations. *Mathematical Methods in the Applied Sciences*, 48(2), 201–218.
10. Gerth, D., & Soodhalter, K. (2025). Hybrid CG–Tikhonov as a filtration of the CG Lanczos vectors. arXiv preprint arXiv:2505.14862.
11. Bernaschi, M., Bisson, M., & Fatica, M. (2025). Communication-reduced Conjugate Gradient Variants for GPU-accelerated Clusters. arXiv preprint arXiv:2501.01235.
12. Tian, Q., Wang, X., Pang, L., Zhang, M., & Meng, F. (2021). A New Hybrid Three-Term Conjugate Gradient Algorithm for Large-Scale Unconstrained Problems. *Mathematics* 2021, Vol. 9, Page 1353, 9(12), 1353. <https://doi.org/10.3390/MATH9121353>
13. Deng, S., & Wan, Z. (2015). A three-term conjugate gradient algorithm for large-scale unconstrained optimization problems. *Applied Numerical Mathematics*, 92, 70–81. <https://doi.org/10.1016/J.APNUM.2015.01.008>
14. Lin, G., & Du, S. (2025). A three-term Polak–Ribière–Polyak conjugate gradient method for vector optimization. arXiv preprint arXiv:2505.08408.
15. Peterseim, D., Püschel, M., & Stykel, T. (2025). A Riemannian three-term conjugate gradient method for Kohn–Sham DFT. arXiv preprint arXiv:2503.06794.

16. Jian, J., Yang, L., Jiang, X., Liu, P., & Liu, M. (2020). A Spectral Conjugate Gradient Method with Descent Property. *Mathematics* 2020, Vol. 8, Page 280, 8(2), 280. <https://doi.org/10.3390/MATH8020280>
17. Ghanbari, M., Ahmad, T., Alias, N., & Askaripour, M. (2013). Global convergence of two spectral conjugate gradient methods. *ScienceAsia*, 39(3), 306-311. <https://doi.org/10.2306/scienceasia1513-1874.2013.39.306>
18. Zulpakkal, N., Aini, N., Ghani, N. H. A., Mohamed, N. S., Idalisa, N., & Rivaie, M. (2022). Covid-19 data modelling using hybrid conjugate gradient method. *Journal of Information and Optimization Sciences*, 43(4), 837–853. <https://doi.org/10.1080/02522667.2022.2060610>