SEMARAK ILMU PUBLISHING
202103268166(003316878-P)

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Pre-Numerical Tests for a New Conjugate Gradient Method

Rahma Mohamed[1], Shalela Mohd Mahali[1,2], Noor Ilyani Izzati[1], Rakhmatillo Djuraevich Aloev[3], Nur Aidya Hanum Aizam[1,2,*]

[1] Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu, 21030 Kuala Terengganu, Malaysia
[2] Marine Management Sciences Research Group, School of Informatics and Applied Mathematics, Universiti Malaysia Terengganu, 21030 Kuala Terengganu, Malaysia
[3] National University of Uzbekistan, University Street 4, Tashkent City, 100174, Uzbekistan

ARTICLE INFO

Article history:
Received 7 June 2023
Received in revised form 12 December 2023
Accepted 4 January 2024
Available online 7 February 2024

Keywords:
Conjugate Gradient Method; New Beta; Numerical Test

ABSTRACT

The performance of a new beta in conjugate gradient method is generally measured on its CPU time and the number of iterations using large scale problems without concerning the accuracy of the solution attained. The method is claimed to be the best when less CPU time is used in comparison with the other classical beta. However, the accuracy of the solution gained compared to the exact solution of the function has never been tested as part of the performance measurement. Our previous research has already proven that our new beta (β_k^{RAS}) perform well in both speed and number of iterations used. Therefore, in this research, we will accommodate this with alternative numerical test on the new beta in assessing the accuracy of the gained solution. This is to further ensure the efficiency of the new beta. Assessment was done with four other classical conjugate gradient methods on fourteen small scale problems. Results attained were promising to be tested for more tests in real functional problems.

1. Introduction

The nonlinear conjugate gradient methods (CG) are mainly used to find the minimum value of a function for unconstrained optimization problems that is either small or large. The method is largely used due to the simplicity and minimal storage requirements [1,2]. The best CG methods proposed is the ones that can obtain the smallest error value within small number of iterations and quickest CPU time. As for the Galerkin method, discretization is not needed to solve numerical approximation. However, it is not a suitable method to solve complicated problem [3]. Generally, the method has the following form

$$min\ f\ (x), x \in R^n \quad (1)$$

* *Corresponding author.*
E-mail address: aidya@umt.edu.my

https://doi.org/10.37934/araset.39.1.1525

where is continuously differentiable nonlinear function and whose gradient denoted by $g(x) = \nabla f(x)$. This type of problem appears in a broad spectrum of engineering applications, including but not limited to motion control in robotics [4,5], parameter estimation [6], energy minimization [7], image processing [8] and numerous other domains.

The CG methods are given by an iterative method of the form

$$x_{k+1} = x_k + \alpha_k d_k \qquad k = 0,1,2,\ldots \tag{2}$$

where $x_k$is the current iterative point. The ∝_> 0 is a step size and $d_k$is the search direction. The step-size is obtained by carrying out a one-dimensional search, known as the 'line search'. Where

$$\alpha_k = \underset{\alpha \geq 0}{argmin} f(x_k + \alpha d_k) \tag{3}$$

The search direction, d_k is defined by,

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

where β_k is a scalar and $g_k$is the gradient of the nonlinear function. Six classical formulas for β_k that can be found in the literature that are often used as comparison indicator: Hestenes-Stiefel [9], Fletcher – Reeves [10], Polak-Ribiere-Polyak [11,12], Conjugate Descent [13], Liu – Storey [14] and Dai – Yuan [15]. There are given respectively in the form below:

$$\beta_k^{HS} = \frac{g_k^T(g_k - g_{k-1})}{(g_k - g_{k-1})^T d_{k-1}} \tag{1}$$

$$\beta_k^{FR} = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}} \tag{2}$$

$$\beta_k^{PRP} = \frac{g_k^T(g_k - g_{k-1})}{g_{k-1}^T g_{k-1}} \tag{3}$$

$$\beta_k^{CD} = \frac{g_k^T g_k}{d_{k-1}^T g_{k-1}} \tag{4}$$

$$\beta_k^{LS} = \frac{g_k^T(g_k - g_{k-1})}{d_{k-1}^T g_{k-1}} \tag{5}$$

$$\beta_k^{DY} = \frac{g_k^T g_k}{(g_k - g_{k-1})^T d_{k-1}} \tag{6}$$

In our recent research [16], we proposed a new class of nonlinear conjugate gradient parameter which is a modification of the Wei-Yao-Liu method [17]. It was proven to be globally convergent and satisfies the sufficient descent condition. The parameter in our method is computed as follows

$$\beta_k^{RAS} = \frac{\|g_k\|^2 - \frac{\|g_k\|}{\|d_{k-1}\|}|g_k^T d_{k-1}|}{max(\|g_{k-1}\|^2, \|d_{k-1}\|^2)} \quad (11)$$

By using the following exact line search,

$$f(x_k + \alpha_k d_k) = \min_{\alpha \geq 0} f(x_k + \alpha d_k) \quad (12)$$

2. Numerical Test: Error Value

In our previous research [16], we have validated the CG method with a new β_k^{RAS} proposed through comparing the CPU time and number of iterations towards attaining a solution with three other betas; FR, PRP and RMIL. Favorable results were achieved where we were able to solve most of the 23 standard problems among the fastest CPU time with lesser number of iterations. Following the promising result, as to further verify the effectiveness of our CG method (β_k^{RAS}) and before claiming that an improvised beta is produced, we further our tests in a slightly different perspective. We determine the error value from the exact solution. The average error can be calculated to compare the accuracy of different methods [18]. This is an alternative test specifically to measure the accuracy of the resulted solution.

Due to the exact solution availability, small scale problems will be used for the accuracy test. The resulted solution from the tested CG methods (RAS, FR, PRP, WYL, RMIL) will be compared with the exact solutions. The error between these values will be calculated to measure the performance of the new CG method (β_k^{RAS}) in comparison with the other established CG methods (FR, PRP, WYL, RMIL). Smaller error value indicates a better method.

We will be using fourteen test problems from the 23 problems which were used by Mohamed *et al.*, [16]. These are the standard test problem functions see [19-22] ranging from two to four variables. The following formula is used to calculate the performances on the error value:

$$|f - f(z^*)| \quad (13)$$

where :-

z^* : Minimum point for function.
f : Value of the original function at solution point.
$f(z^*)$: denotes the function value at the point.

3. Numerical Results and Discussions

In this section, we report numerical results obtained for the CG methods FR, PRP, WYL, RMIL [23] and RAS. Fourteen test problems are selected randomly from Mohamed *et al.*, [16] and Andrei and Neculai [19]. We considered $\varepsilon = 10^{-6}$ the gradient value as the stopping criteria as Hillstrom, and Kenneth [24] suggested that $\|g_k\| \leq \varepsilon$. This stopping criterion has also been used in several current literatures such as by Ibrahim *et al.*, [25] and Sulaiman *et al.*, [26] and to the latest by Ishak *et al.*, [27]. For each test function, we used initial point that is a closer point to the solution for every problem. A list of problem functions and the initial points used are shown in Table 1, where all the problems are solved by Maple 14 subroutine programming. We used the exact line search to compute the step size. The *CPU* processor used was Intel(R) Core™ i3-4030U (1.9GHz), with RAM 2GB. In some cases, the computation stopped due to the failure of the line search to find the positive step size, and thus it was considered a failure.

Table 1 shows CG with β^{RAS} was able to solve accurately to more than 50% of the tested problems (9 out of 14). This result is similar when other classical betas were used. In fact, in the other 3 problems (Zettl, Six Hump Camel and Dixon-price), the result from β^{RAS} are the same as the other classical beta although they were not exact. In comparison to the accuracy with the other established beta, β^{RAS} has a good record in never fail to give a solution. This result shows that β^{RAS} is equivalent and could be better in certain cases compared to the other methods if we consider all types of tests.

Table 2 shows the comparison between the tested methods in terms of CPU time and number of iterations. RAS shows lowest computational cost compared to the other methods for 8 out of 14 problems. The other 3 problems are best solved using PRP and the rest are best solved using FR method. These result shows the RAS potential in saving computational cost.

Table 1
Performances comparison of *RAS* with other CG methods based on the function values, *f*(*z**)

Function	Beta	Initial point	*z**	*f*(*z**)
1- Three hump camel = 0	RAS	[0.2,0.2]	[E-6* 0.1965,E-6* 0.1177]	1.14E-13
at $-5 \leq x_i \leq 5$	RM	[0.2,0.2]	[E-6* -0.0576,E-6* -0.1458]	3.63E-14
	FR	[0.2,0.2]	F	F
	PRP	**[0.2,0.2]**	**[E-7*-0.2249,E-6*-0.0557]**	**1.17E-15**
	WYL	[0.2,0.2]	[E-6* 0.1143,E-6*-0.1658]	3.47E-14
2- Zettl = -0.00379	RAS	[0,0]	[-0.0299,0]	-3.80E-03
$x_i \in [-5,5]$	RM	[0,0]	[-0.0299,0]	-3.80E-03
	FR	[0,0]	[-0.0299,0]	-3.80E-03
	PRP	[0,0]	[-0.0299,0]	-3.80E-03
	WYL	[0,0]	[-0.0299,0]	-3.80E-03
3- Six hump camel = -1.0316	RAS	[2,2]	[1.7036,-0.7961]	-2.16E-01
$-3 \leq x_1 \leq 3, -2 \leq x_2 \leq 2$	RM	[2,2]	[1.7036,-0.7961]	-2.16E-01
	FR	[2,2]	[1.7036,-0.7961]	-2.16E-01
	PRP	[2,2]	[1.7036,-0.7961]	-2.16E-01
	WYL	[2,2]	[1.7036,-0.7961]	-2.16E-01
4- Leon function = 0	RAS	[0.9,0.9]	[1,1]	0.00E+00
	RM	[0.9,0.9]	[1,1]	0.00E+00
$x_i \in [-1.2,1.2]$	FR	[0.9,0.9]	[1,1]	0.00E+00

	PRP	[0.9,0.9]	[1,1]	0.00E+00
	WYL	[0.9,0.9]	[1,1]	0.00E+00
5- Trecanni = 0	RAS	[1,1]	[-2,0]	0.00E+00
$-5 \leq x_i \leq 5$	RM	[1,1]	[-2,0]	0.00E+00
	FR	[1,1]	[-2,0]	0.00E+00
	PRP	[1,1]	[-2,0]	0.00E+00
	WYL	[1,1]	[-2,0]	0.00E+00
6- Booth function = 0	RAS	[3,3]	[1,3]	0.00E+00
$-10 \leq x_i \leq 10$	RM	[3,3]	[1,3]	0.00E+00
	FR	[3,3]	[1,3]	0.00E+00
	PRP	[3,3]	[1,3]	0.00E+00
	WYL	[3,3]	[1,3]	0.00E+00
7-Sum squares function = 0	RAS	[5,5,5,5]	[E-6*0.1846,E-6*-0.0054,E-6*0.0368,E-6*-0.0600]	1.35E-14
$-10 \leq x_i \leq 10$	RM	[5,5,5,5]	[E-7*0.5594,E-7*-0.2689,E-7*0.3666,E-7*0.1359]	3.97E-15
	FR	[5,5,5,5]	[E-6*0.1178,E-6*-0.0071,E-6*0.0649,E-6*0.1097]	4.46E-14
	PRP	**[5,5,5,5]**	**[E-6*0.1291,E-6*-0.0083,E-6*0.025,E-6*0.0045]**	**1.43E-15**

	WYL	[5,5,5,5]	[E-6*0.0593,E-6*-0.0056,E-6*0.1007,E-6*0.0100]	2.06E-14
8- Colville function = 0	RAS	[6,6,6,6]	[1,1,1,1]	0.00E+00
	RM	[6,6,6,6]	[1,1,1,1]	0.00E+00
	FR	[6,6,6,6]	[1,1,1,1]	0.00E+00
$-10 \leq x_i \leq 10$	PRP	[6,6,6,6]	[1,1,1,1]	0.00E+00
	WYL	[6,6,6,6]	[1,1,1,1]	0.00E+00
9- Sphere function = 0	RAS	[100,100,100,100]	[0,0,0,0]	0.00E+00
$\lvert x \rvert = 100$	RM	[100,100,100,100]	[0,0,0,0]	0.00E+00
	FR	[100,100,100,100]	[0,0,0,0]	0.00E+00
	PRP	[100,100,100,100]	[0,0,0,0]	0.00E+00
	WYL	[100,100,100,100]	[0,0,0,0]	0.00E+00
10- Beale function = 0	RAS	[4,4]	[3,0.5]	0.00E+00
	RM	[4,4]	[3,0.5]	0.00E+00
$-4.5 \leq x_i \leq 4.5$	FR	[4,4]	[3,0.5]	0.00E+00
	PRP	[4,4]	[3,0.5]	0.00E+00
	WYL	[4,4]	[3,0.5]	0.00E+00
11- Himmelblau function = 0	RAS	[2,2]	[3,2]	0
$x_i \in [-6,6]$	RM	[2,2]	[3,2]	0
	FR	[2,2]	[3,2]	0
	PRP	[2,2]	[3,2]	0

	WYL	[2,2]	[3,2]	0
12- Extended Rosenbrock = 0	RAS	[3,3]	[1,1]	0
	RM	[3,3]	[1,1]	0
$-5 \leq x_i \leq 5$	FR	[3,3]	[1,1]	0
	PRP	[3,3]	[1,1]	0
	WYL	[3,3]	[1,1]	0
13- Dixon-price function = 0	RAS	[8,8]	[1, 0.7071]	7.36E-10
	RM	[8,8]	[1, 0.7071]	7.36E-10
$-10 \leq x_i \leq 10$	FR	[8,8]	[1, -0.7071]	7.36E-10
	PRP	[8,8]	[1, 0.7071]	7.36E-10
	WYL	[8,8]	[1, 0.7071]	7.36E-10
14- Powell = 0	RAS	[0,0,0,0]	[0,0,0,0]	0
	RM	[0,0,0,0]	[0,0,0,0]	0
$-4 \leq x_i \leq 5$	FR	[0,0,0,0]	[0,0,0,0]	0
	PRP	[0,0,0,0]	[0,0,0,0]	0
	WYL	[0,0,0,0]	[0,0,0,0]	0

Table 2
Performances comparison of *RAS* with other CG methods based on number of iteration (NOI) and CPU time

No	**Function**	**Dimension**	**Initial points**	**RAS**		**RMIL**		**FR**		**PRP**		**WYL**	
				NOI	**CPU**	**NOI**	**CPU**	**NOI**	**CPU**	**NOI**	**CPU**	**NOI**	**CPU**
1	Three hump camel	2	[0.2, 0.2]	**8**	**0.0784**	10	0.1663	F	F	12	0.1167	17	0.1965
2	Zettl	2	[0, 0]	**1**	**0.011**	1	0.011	1	0.016	1	0.0111	1	0.0116
3	Six hump camel	2	[2, 2]	**4**	**0.0409**	4	0.0449	4	0.0961	4	0.0416	4	0.0454
4	Leon function	2	[0.9, 0.9]	13	0.1074	13	0.1205	20	0.2174	**8**	**0.088**	24	0.2566
5	Trecanni	2	[1, 1]	**4**	**0.0334**	4	0.054	4	0.0349	4	0.0351	4	0.0342
6	Booth function	2	[3, 3]	**3**	**0.0716**	3	0.1066	3	0.0948	3	0.0895	3	0.0921
7	Sum squares function	4	[5, 5, 5, 5]	15	0.1752	15	0.1987	**4**	**0.0564**	5	0.0604	4	0.1235
8	Colville function	4	[6, 6, 6, 6]	148	1.2026	223	1.8112	**32**	**0.2624**	47	0.3875	216	1.8017
9	Sphere function	4	[100, 100, 100, 100]	**1**	**0.0134**	1	0.0139	1	0.0737	1	0.0715	1	0.0136
10	Beale function	2	[4, 4]	11	0.0892	22	0.172	16	0.1351	**7**	**0.0608**	10	0.1075
11	Himmelblau function	2	[2, 2]	7	0.0609	7	0.062	9	0.0727	**6**	**0.0543**	10	0.0861
12	Extended Freudenstein & Roth fun	2	[7, 7]	7	0.0558	**6**	**0.0473**	7	0.0792	9	0.0746	7	0.0538
13	Extended Rosenbrock	2	[3, 3]	**20**	**0.1351**	24	0.1576	148	0.9931	21	0.1453	20	0.1482
14	Dixon-price function	2	[8, 8]	**11**	**0.1378**	20	0.2459	25	0.3009	10	0.1725	11	0.1391

4. Conclusion

We have successfully proposed a new improvised conjugate gradient method with β_k^{RAS} to solve unconstrained problems by proving to its fast CPU time and lesser number of iterations. It has also proven that this CG method has excellently gained comparable solution with the other established method in evaluating the optimum function value. Based on the favorable achievements, we are currently working on employing this method to solve real case problems that could be beneficial towards saving computational time without compromising the accuracy. In addition to the numerical test, the performance of this β_k^{RAS} need to be further studied theoretically. The convergence in function spaces and the number of iteration steps for convergence with a certain accuracy need will be analysed in our future study.

Acknowledgement
The authors would like to thank the Ministry of Higher Education (MOHE) under the Grant No. (RAGS 57107) and the government of Libya for their financial support towards this research.

References

[1] Abubakar, Auwal Bala, and Poom Kumam. "A descent Dai-Liao conjugate gradient method for nonlinear equations." *Numerical Algorithms* 81 (2019): 197-210. https://doi.org/10.1007/s11075-018-0541-z
[2] Wang, Jian, and Xuebin Chi. "Cg global convergence properties with goldstein linesearch." *Bulletin of the Brazilian Mathematical Society* 36, no. 2 (2005): 197-204. https://doi.org/10.1007/s00574-005-0036-0
[3] Goh, Rui Zher. "Solution of Dirichlet Pure Diffusion Problem Using Galerkin Method." *Progress in Energy and Environment* (2017): 1-6. http://www.akademiabaru.com/progee.html
[4] Saleh Nazzal Alsuliman, Nur Fadhilah Ibrahim, & Nur Aidya Hanum Aizam. (2023). A New Conjugate Gradient Parameter via Modification of Liu-Storey Formula for Optimization Problem and Image Restoration. Journal of Advanced Research in Applied Sciences and Engineering Technology, 35(2), 175–186. https://doi.org/10.37934/araset.35.2.175186
[5] Saleh Nazzal Alsuliman, Nur Fadhilah Ibrahim, & Nur Aidya Hanum Aizam. (2023). A Modified Wei-Yao-Liu Conjugate Gradient Method for Unconstrained Optimization and Motion Control of Robotic Motion Manipulators. Journal of Advanced Research in Applied Sciences and Engineering Technology, 34(1), 314–327. https://doi.org/10.37934/araset.34.1.314327
[6] Simon Hubmer, Andreas Neubauer, Ronny Ramlau & Henning U. Voss (2020) A conjugate-gradient approach to the parameter estimation problem of magnetic resonance advection imaging, Inverse Problems in Science and Engineering, 28:8, 1154-1165, DOI: 10.1080/17415977.2019.1708911.
[7] Watowich, S. J., Meyer, E. S., Hagstrom, R., & Josephs, R. (1988). A stable, rapidly converging conjugate gradient method for energy minimization. Journal of Computational Chemistry, 9(6), 650–661. doi:10.1002/jcc.540090611
[8] Hassan, B. A., Moghrabi, I. A., & Sulaiman, I. M. (2023). New conjugate gradient image processing methods. Asian-European Journal of Mathematics, 16(06), 2350099.
[9] Hestenes, Magnus R., and Eduard Stiefel. "Methods of conjugate gradients for solving linear systems." *Journal of research of the National Bureau of Standards* 49, no. 6 (1952): 409-436. https://doi.org/10.6028/jres.049.044
[10] Fletcher, Reeves, and Colin M. Reeves. "Function minimization by conjugate gradients." *The computer journal* 7, no. 2 (1964): 149-154. https://doi.org/10.1093/comjnl/7.2.149
[11] Polyak, Boris Teodorovich. "The conjugate gradient method in extremal problems." *USSR Computational Mathematics and Mathematical Physics* 9, no. 4 (1969): 94-112. https://doi.org/10.1016/0041-5553(69)90035-4
[12] E. Polak and G. Ribiere, "Note on the convergence of the conjugate directions (in French)", *Rev. Francaise Informat Recherche Opertionelle,* no. 3, (1969): 35–43. https://doi.org/10.1051/m2an/196903R100351
[13] R. Fletcher. *Practiccorneal Method of Optimization* (New York: John Wiley & Sons, 2013).
[14] Liu, Y., and C. Storey. "Efficient generalized conjugate gradient algorithms, part 1: theory." *Journal of optimization theory and applications* 69 (1991): 129-137. https://doi.org/10.1007/BF00940464
[15] Dai, Yu-Hong, and Yaxiang Yuan. "A nonlinear conjugate gradient method with a strong global convergence property." *SIAM Journal on optimization* 10, no. 1 (1999): 177-182. https://doi.org/10.1137/S1052623497318992
[16] R. Mohamed, Nur Aidya Hanum Aizam, Shalela Mohd Mahali and A. Abashar, " A new conjugate gradient method for unconstrainted optimization with exact line search", in Proceeding of Postgraduate Colloquium on Informatics

and Mathematics (KOSIM 2015), 27-28 October, School of Informatics and Applied Mathematics, Universiti Malaysia Terengganu, Malaysia (2015).

[17] Wei, Zengxin, Shengwei Yao, and Liying Liu. "The convergence properties of some new conjugate gradient methods." *Applied Mathematics and computation* 183, no. 2 (2006): 1341-1350. https://doi.org/10.1016/j.amc.2006.05.150

[18] Eshan, Abdul Rauf, and Wah Yen Tey. "Investigation of Shape Parameter for Exponential Weight Function in Moving Least Squares Method." *Progress in Energy and Environment* (2017): 17-24. http://www.akademiabaru.com/progee.html

[19] Andrei, Neculai. "An unconstrained optimization test functions collection." *Adv. Model. Optim* 10, no. 1 (2008): 147-161. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.665.3152&rep=rep1&type=pdf

[20] Molga, Marcin, and Czesław Smutnicki. "Test functions for optimization needs." *Test functions for optimization needs* 101 (2005): 48. www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf.

[21] Rivaie, Mohd, Mustafa Mamat, and Abdelrhaman Abashar. "A new class of nonlinear conjugate gradient coefficients with exact and inexact line searches." *Applied Mathematics and Computation* 268 (2015): 1152-1163. https://doi.org/10.1016/j.amc.2015.07.019

[22] Yuan, Gonglin, Junyu Lu, and Zhan Wang. "The PRP conjugate gradient algorithm with a modified WWP line search and its application in the image restoration problems." *Applied Numerical Mathematics* 152 (2020): 1-11. https://doi.org/10.1016/j.apnum.2020.01.019

[23] Rivaie, Mohd, Mustafa Mamat, Leong Wah June, and Ismail Mohd. "A new class of nonlinear conjugate gradient coefficients with global convergence properties." *Applied Mathematics and Computation* 218, no. 22 (2012): 11323-11332. https://doi.org/10.1016/j.amc.2012.05.030

[24] Hillstrom, Kenneth E. "A simulation test approach to the evaluation of nonlinear optimization algorithms." *ACM Transactions on Mathematical Software (TOMS)* 3, no. 4 (1977): 305-315. https://doi.org/10.1145/355759.355760

[25] Ibrahim, Sulaiman Mohammed, Usman Abbas Yakubu, and Mustafa Mamat. "Application of spectral conjugate gradient methods for solving unconstrained optimization problems." *An International Journal of Optimization and Control: Theories & Applications (IJOCTA)* 10, no. 2 (2020): 198-205. https://doi.org/10.11121/ijocta.01.2020.00859

[26] Sulaiman, I. Mohammed, N. Abu Bakar, Mustafa Mamat, Basim A. Hassan, Maulana Malik, and Alomari Mohammad Ahmed. "A new hybrid conjugate gradient algorithm for optimization models and its application to regression analysis." *Indones. J. Electr. Eng. Comput. Sci* 23 (2021): 1100-1109. https://doi.org/10.11591/ijeecs.v23.i2.pp1100-1109

[27] Ishak, M. I., S. M. Marjugi, and L. June. "A new modified conjugate gradient method under the strong wolfe line search for solving unconstrained optimization problems." *Mathematical Modeling and Computing* 9, no. 1 (2022): 111-118. https://doi.org/10.23939/mmc2022.01.111