Earthline Journal of Mathematical Sciences
ISSN (Online): 2581-8147
Volume 6, Number 1, 2021, Pages 163-174
https://doi.org/10.34198/ejms.6121.163174

A New Conjugate Gradient Method with Sufficient Descent Property

O.B. Akinduko

Department of Mathematical Sciences, Adekunle Ajasin University, Akungba-Akoko, Nigeria
e-mail: akindukooluwaseun@gmail.com

Abstract

In this paper, by linearly combining the numerator and denominator terms of the Dai-Liao (DL) and Bamigbola-Ali-Nwaeze (BAN) conjugate gradient methods (CGMs), a general form of DL-BAN method has been proposed. From this general form, a new hybrid CGM, which was found to possess a sufficient descent property is generated. Numerical experiment was carried out on the new CGM in comparison with four existing CGMs, using some set of large scale unconstrained optimization problems. The result showed a superior performance of new method over majority of the existing methods.

1 Introduction

No investor wants to go for investment without returns or with high risk; hence the need for decision making. Optimization is central to any problem involving decision making which arises from the fields of Engineering, Economics, Science, etc. It entails choosing the best out of various alternatives (Chong and Zak [3]). A way to handle this kind of problem involves solving an unconstrained optimization problem of the form:

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

Problems of the form (1) arise in many theoretical fields because most of the optimization problems can be reduced to an unconstrained optimization problem

Received: December 16, 2020; Accepted: February 3, 2021

2010 Mathematics Subject Classification: 90C30.

Keywords and phrases: conjugate gradient method, line search, sufficient descent property, unconstrained optimization, numerical experiment.

(Mangasarian [11]). The CGM is a computational scheme for solving the unconstrained minimization problem (1). The first CGM was developed in 1952 by Hestenes and Stiefel as an algorithm for solving algebraic equations, which was later applied to nonlinear unconstrained optimization problems as reported by Fletcher and Reeves in 1964. Since then, the CGM has been an area of active research. Out of the many iterative methods for solving (1), the CGM is very popular due to the simplicity of its analysis, its low memory requirements and its ease of implementation. The iterative scheme is given by:

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

where x_k is the kth solution iterate to (1), $\alpha_k > 0$ denotes the step size, usually obtained by a line search and d_k, given by

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

is the search direction, $g_k = \nabla f(x_k)$ is the gradient and β_k is a scalar known as the conjugate update parameter. Different choices of β_k has resulted in different CGMs. Some well known classical CGMs, developed by Hestenes and Stiefel [9], Fletcher and Reeves [7], Fletcher [8], Dai and Liao [4] and Bamigbola *et al.* [2] are:

$$\beta_k^{HS} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}, \ \beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2}, \ \beta_k^{CD} = \frac{\|g_k\|^2}{-d_{k-1}^T g_{k-1}}, \tag{4}$$

$$\beta_k^{DL} = \frac{g_k^T (y_{k-1} - t s_{k-1})}{d_{k-1}^T y_{k-1}}, \text{ and } \beta_k^{BAN} = \frac{-g_k^T y_{k-1}}{g_{k-1}^T y_{k-1}}. \tag{5}$$

In the classical methods (4 - 5), we have $y_{k-1} = g_k - g_{k-1}$, $t > 0$, $s_{k-1} = x_k - x_{k-1}$ and $||\ .\ ||$ stands for the Euclidean norm.

To any CGM, the determination of the search direction d_k and the step size α_k is very important. A careful choice of the line search strategy is needed to obtain a descent direction (Nocedal [13]). Basically, two types of line search are used in computing α_k, namely the exact and inexact line search rules. By the exact line search, α_k is computed such that:

$$\alpha_k = argmin\{f(x_k + \alpha d_k); \alpha \geq 0\} \tag{6}$$

This approach is expensive in terms of evaluating the function and gradient. The limitation of the exact line search led researchers to the use of inexact line search, where α_k is computed numerically by ensuring a reasonable reduction in the value of the objective function at a minimal cost. One of the most popular inexact line search is the StrongWolfe line search given by:

$$f(x_k) - f(x_k + \alpha_k d_k) \geq -\delta \alpha_k g_k^T d_k \tag{7}$$

and

$$|g(x_k + \alpha_k d_k)^T d_k| \leq \sigma |g_k^T d_k| \tag{8}$$

with $0 \leq \delta \leq \sigma \leq 1$. It is required that the search direction, d_k satisfy:

$$g_k^T d_k < 0, \forall k \geq 0 \tag{9}$$

which guarantees a descent direction of $f(x)$ at x_k.

A class of CGMs known as hybrid CGMs, which are modifications of the classical CGMs have been proposed by various authors. This is due to the part it plays in achieving better computational performance as well as retaining the strong global convergence of the methods involved (Li and Zhao [10]). By taking into consideration the convex combination of the numerators and denominators of the update parameters of Fletcher-Reeves and Hestenes-Stiefel methods, Nazareth [12] proposed a two-parameter family of CGMs. Dai and Liao [4] extended this by adding one more parameter, where their three-parameter family included six standard CGMs. By forming a linear combination of the update parameters of the Dai-Yuan and Hestenes-Stiefel methods and that of Fletcher-Reeves and Polak-Ribiere-Polyak methods, Xu and Kong [15] proposed two new hybrid CGMs, with the aid of the generalized Wolfe line search. Recently, Osinuga and Olofin [14] presented an extended hybrid CGM which was proved to be globally convergent with Armijo-type line search, while Djordjevic [5] proposed another new CGM by a convex combination of the update parameters of Liu-Storey and Fletcher-Reeves methods.

Desirous to generate many methods by varying coefficients from a linear combination of the numerator and denominator terms of the update parameters of DL and BAN methods, this paper presents a new CGM.

2 The New Method

By linearly combining the update parameters of DL and BAN methods, the following general form of the DL-BAN method is proposed:

$$\beta_k^{DB} = \frac{\lambda_1 g_k^T y_{k-1} + \lambda_2 t g_k^T S_{k-1}}{\omega_1 d_{k-1}^T y_{k-1} + \omega_2 g_{k-1}^T y_{k-1}}, \tag{10}$$

where $\lambda_i, \omega_i \in \Re$ and $t > 0$. Taking $\lambda_1 = 1, \lambda_2 = -1, \omega_1 = 1$, and $\omega_2 = 0$, we have β_k^{DL} and by taking $\lambda_1 = -1, \lambda_2 = 0, \omega_1 = 0$, and $\omega_2 = 1$, we have β_k^{BAN}. Thus, several new methods can be generated from (10) by varying the values of λ_i and ω_i. Therefore, taking $\lambda_1 = 0, \lambda_2 = 1, \omega_1 = 0$, and $\omega_2 = 1$ and $t = 1$ from (10), a new hybrid of DL and BAN methods is proposed in this paper and it is given by:

$$\beta_k^{NM} = \frac{g_k^T s_{k-1}}{g_{k-1}^T y_{k-1}} \tag{11}$$

The following Algorithm is used to implement the NM CG method

Algorithm 2.1

Step 1: Choose $\epsilon = 10^{-6}$ and $x_0 \in \Re^n$, set $k := 0$
Step 2: Stop if $\|g_k\| \leq \epsilon$
Step 3: Compute β_k based on (11)
Step 4: Compute d_k by (3)
Step 5: Compute α_k by (7) and (8)
Step 6: Update a new point by (2)
Step 7: Set $k := k + 1$, and return to step 2.

3 Sufficient Descent Analysis of β_k^{NM} Method

The sufficient descent analysis of β_k^{NM} shall be carried out based on the following lemmas

Lemma 3.1. *In the Conjugate Gradient Method,*

$$g_i^T d_k = 0, \; k = i - 1 \tag{12}$$

Proof. See Chong and Zak ([3]; Lemma 10.2, pp. 174-176) for the proof.

Lemma 3.2. *The new CG method β_k^{NM} satisfy the sufficient descent condition, i.e.,*

$$g_k^T d_k \leq -c \left\| g_k \right\|^2, \qquad 0 < c \leq 1. \tag{13}$$

Proof. By (3) and (11),

$$g_k^T d_k = -g_k^T g_k + \beta_k^{NM} g_k^T d_{k-1} = -g_k^T g_k + \frac{g_k^T s_{k-1}}{g_{k-1}^T y_{k-1}} g_k^T d_{k-1}.$$

Let the second term be expressed in the form $u^T v \leq \frac{1}{2}(\|u\|^2 + \|v\|^2)$ by making $u = \frac{2}{\sqrt{5}} g_k$, $v = \frac{\sqrt{5}}{2} \frac{s_{k-1} g_k^T d_{k-1}}{g_{k-1}^T y_{k-1}}$ and applying Lemma 3.1, we have:

$$\begin{aligned}
g_k^T d_k &\leq -\left\| g_k \right\|^2 + \frac{1}{2}\left[\left\| \frac{2}{\sqrt{5}} g_k \right\|^2 + \left\| \frac{\sqrt{5}}{2} \frac{s_{k-1} g_k^T d_{k-1}}{g_{k-1}^T y_{k-1}} \right\|^2 \right], \\
&\leq \frac{1}{2}\left[\frac{4}{5}\left\| g_k \right\|^2 + \frac{5}{4} \frac{\left\| s_{k-1} \right\|^2 (g_k^T d_{k-1})^2}{\left(g_{k-1}^T y_{k-1} \right)^2} \right], \\
&\leq -\left\| g_k \right\|^2 + \frac{2}{5}\left\| g_k \right\|^2 \\
&\leq -\frac{3}{5}\left\| g_k \right\|^2
\end{aligned}$$

Therefore, β_k^{NM} method satisfies (13) with $c = \frac{3}{5}$.

4 Numerical Consideration

In this section, a report of the numerical experiment carried out on the new CGM, using a set of large-scale unconstrained minimization problems, taken from Andrei [1], is presented.

4.1 Computational details

A total of 27 unconstrained optimization problems, each of dimensions 5000 and 10000, were solved using the Strong Wolfe line search. The iterations were terminated when $\|g_k\| \leq 10^{-6}$, and a failure declared if this condition was not satisfied after 2000 iterations. The nonlinear conjugate gradient algorithm (CGA) was written in Matlab codes and run on a PC with 2.16 GHz processor, 4GB Ram and Windows 10 operating system.

4.2 Presentation of numerical results

The numerical results obtained for the new method in comparison with four existing CGMs are presented in Tables 1-2. In tabulating the numerical results, the following notations were used:

> **Dim** - Dimension; **NM**- new method; **CD** - Conjugate Descent method; **F** - Failed; **Itr** - Iteration; **HS** - Hestenes-Stiefel method; **DL** - Dai-Liao method; **BAN** - Bamigbola-Ali-Nwaeze method; $\|g(x^*)\|$ - norm of gradient of the objective function at iteration x^*; **Cpu** - computational time.

The performance profile of Dolan and More [6] was adopted to compare the new method with the four existing CGMs. For each method, a fraction $P(\tau)$ of the problems for which the method is within a factor τ of the best time is plotted as shown in Figures 1 and 2. The vertical axis to the left hand side of the curves gives the percentage of the test problems for which a method is the fastest. The percentage of the problems solved successfully for each method, based on the number of iterations and the CPU time, are as follows: 56.6% for NM, 49.1% for BAN, 50.9% for HS, 41.5% for CD, and 83.0% for DL methods.

The figures indicate that, based on the number of iterations and the CPU time, the NM method is the next in performance after the DL method, with the CD method as the least performer. This shows that the new method is ranked second out of five methods, thus competing favourably with the existing methods.

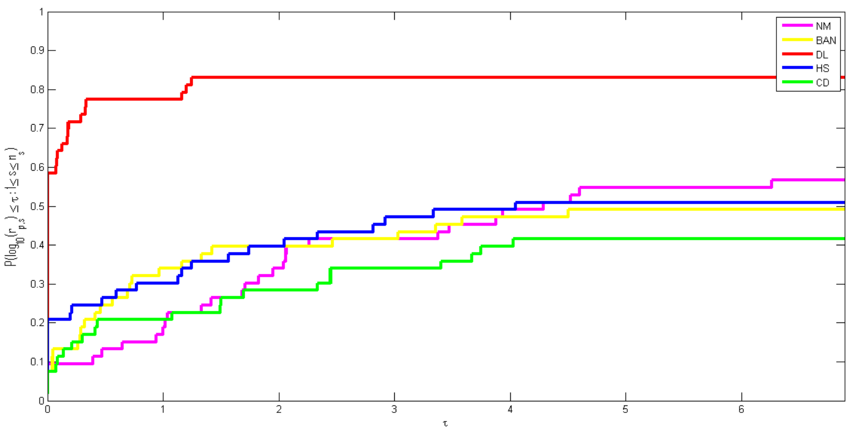

Figure 1: Performance profile for CPU time.

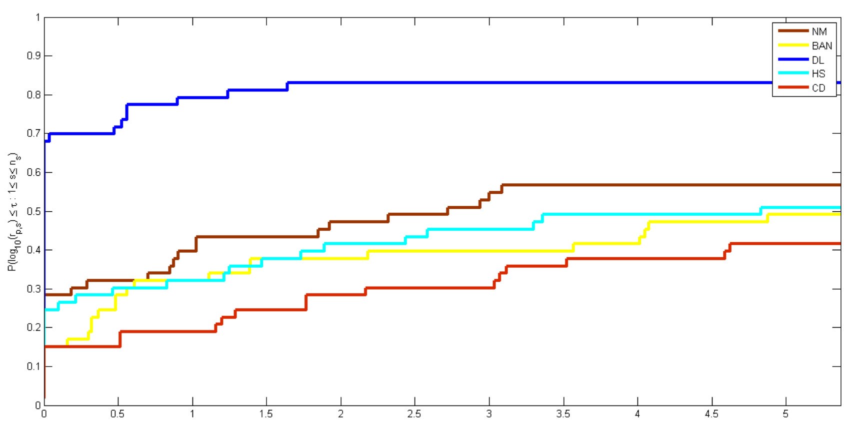

Figure 2: Performance profile for number of iterations.

Table 1: Numerical Results for NM, BAN and DL Methods

		NM			BAN			DL		
Test problems	**Dim**	$\|g(x^*)\|$	**Iter**	**cpu**	$\|g(x^*)\|$	**Iter**	**cpu**	$\|g(x^*)\|$	**Iter**	**cpu**
Arwhead	5000	F	F	F	6.12E-07	85	1.996	4.74E-07	163	3.565
	10000	F	F	F	F	F	F	9.01E-07	193	7.976
Diagonal 4	5000	7.30E-07	44	2.208	4.90E-07	30	0.551	F	F	F
	10000	7.11E-07	45	2.642	6.93E-07	30	1.306	F	F	F
Diagonal 5	5000	2.46E-07	19	0.867	2.64E-19	7	0.095	1.97E-09	5	0.078
	10000	8.09E-07	33	2.049	1.24E-19	7	0.167	2.79E-09	5	0.139
Extended Beale	5000	9.01E-07	1223	137.086	F	F	F	F	F	F
	10000	9.94E-07	1243	218.033	F	F	F	F	F	F
Extended Block Diagonal	5000	9.70E-07	31	1.86	F	F	F	9.37E-07	43	0.969
	10000	5.57E-07	41	2.913	F	F	F	9.03E-07	36	1.155
Extended Powell	5000	F	F	F	F	F	F	8.42E-07	177	10.937
	10000	F	F	F	4.46E-07	329	22.263	F	F	F
Extended Rosenbrock	5000	9.84E-07	260	9.78	F	F	F	F	F	F
	10000	7.91E-07	239	12.755	F	F	F	F	F	F
Extended Tridiagonal-1	5000	F	F	F	F	F	F	9.96E-07	48	1.042
	10000	F	F	F	F	F	F	9.30E-07	50	1.244
Generalized Tridiagonal-1	5000	8.62E-07	155	9.917	6.17E-07	407	8.29	8.32E-07	159	3.087
	10000	7.35E-07	155	12.967	6.11E-07	406	16.111	5.36E-07	289	14.705
Generalized White and Holst	5000	7.51E-07	163	10.892	F	F	F	9.97E-07	80	2.566
	10000	7.51E-07	163	18.046	F	F	F	9.97E-07	80	5.861
Hager	5000	9.58E-07	11	1.281	F	F	F	1.54E-09	9	0.307
	10000	1.15E-14	13	1.533	F	F	F	1.46E-09	8	0.367
Modified Extended Beale	5000	F	F	F	7.23E-07	367	17.898	1.17E-07	31	3.236
	10000	F	F	F	7.70E-07	1067	101.431	8.24E-07	95	15.635
RMODF COSINE	5000	F	F	F	F	F	F	5.04E-07	17	0.324
	10000	F	F	F	9.88E-07	19	0.718	7.12E-07	17	0.485
Staircase1	5000	0.00E+00	1	0.039	0.00E+00	1	0.018	0.00E+00	1	0.012
	10000	0.00E+00	1	0.032	0.00E+00	1	0.022	0.00E+00	1	0.018
Staircase2	5000	7.29E-08	23	0.922	F	F	F	0.00E+00	3	0.038
Diagonal 9	5000	F	F	F	F	F	F	9.87E-07	194	2.961
	10000	F	F	F	F	F	F	3.06E-07	80	2.712
Extended MCCORMCK	5000	F	F	F	9.33E-07	121	1.219	3.14E-07	56	0.732
	10000	F	F	F	7.34E-07	195	2.783	5.60E-07	43	1.245
Extended DENSCHNB	5000	4.87E-07	19	0.46	8.74E-07	28	0.405	3.14E-07	28	0.332
	10000	6.88E-07	19	0.558	4.69E-07	29	0.744	4.44E-07	28	0.59
Full Hessian FH3	5000	3.51E-07	20	1.7	F	F	F	1.34E-11	4	0.111
	10000	5.92E-07	34	4.262	F	F	F	1.39E-11	4	0.218
Generalized PSC1	5000	F	F	F	F	F	F	7.17E-07	793	12.608
	10000	F	F	F	F	F	F	8.59E-07	744	18.092
MDF EXPLIN 1	5000	3.68E-07	18	1.267	8.90E-07	147	1.246	1.13E-09	5	0.055
	10000	9.54E-07	40	6.691	F	F	F	1.59E-09	5	0.087
MODF COSINE	5000	F	F	F	F	F	F	7.98E-09	3	0.042
	10000	F	F	F	F	F	F	1.41E-09	3	0.066
MODF SINE	5000	1.76E-07	1	0.048	1.76E-07	1	0.018	1.76E-07	1	0.022
	10000	6.23E-08	1	0.047	6.23E-08	1	0.031	6.23E-08	1	0.034
NONSCOMP	5000	F	F	F	9.63E-07	53	0.917	1.32E-05	F	F
	10000	F	F	F	2.58E-07	76	3.684	5.82E-06	F	F
QUARTC	5000	0.00E+00	1	0.037	0.00E+00	1	0.018	0.00E+00	1	0.019
	10000	0.00E+00	1	0.042	0.00E+00	1	0.033	0.00E+00	1	0.034
RMDF GENHUMPS	5000	1.98E-07	18	1.445	F	F	F	9.35E-09	10	0.139
	10000	F	F	F	F	F	F	1.32E-08	10	0.236
RMDF SINE	5000	F	F	F	7.76E-07	116	0.82	1.55E-08	7	0.08
	10000	F	F	F	8.20E-07	118	1.648	2.19E-08	7	0.137

Table 2: Numerical Results for HS and CD Methods

		HS			CD		
Test problems	**Dim**	$\|g(x^*)\|$	**Iter**	**cpu**	$\|g(x^*)\|$	**Iter**	**cpu**
Arwhead	5000	2.97E-07	69	1.595	F	F	F
	10000	8.10E-07	62	3.357	F	F	F
Diagonal 4	5000	F	F	F	1.95E-07	24	0.535
	10000	F	F	F	2.78E-07	24	0.806
Diagonal 5	5000	F	F	F	5.20E-07	17	0.22
	10000	F	F	F	7.36E-07	17	0.394
Extended Beale	5000	F	F	F	F	F	F
	10000	F	F	F	F	F	F
Extended Block Diagonal	5000	9.99E-07	882	16	F	F	F
	10000	F	F	F	F	F	F
Extended Powell	5000	F	F	F	F	F	F
	10000	6.07E-07	382	25.599	F	F	F
Extended Rosenbrock	5000	F	F	F	F	F	F
	10000	F	F	F	F	F	F
Extended Tridiagonal-1	5000	F	F	F	F	F	F
	10000	F	F	F	F	F	F
Generalized Tridiagonal-1	5000	5.98E-07	214	4.661	3.59	F	F
	10000	6.51E-07	166	6.394	3.599	F	F
Generalized White and Holst	5000	8.43E-07	80	2.268	195.87	F	F
	10000	8.43E-07	80	4.668	195.87	F	F
Hager	5000	F	F	F	8.33E-109	22	0.647
	10000	F	F	F	F	F	F
Modified Extended Beale	5000	6.85E-07	115	7.095	5.56E+01	F	F
	10000	8.40E-07	66	12.387	7.87E+01	F	F
RMODF COSINE	5000	8.60E-07	92	0.96	6.76E-07	39	1.051
	10000	3.73E-07	47	2.002	7.98E-07	38	0.562
Staircase1	5000	0.00E+00	1	0.011	0.00E+00	1	0.011
	10000	0.00E+00	1	0.016	0.00E+00	1	0.017
Staircase2	5000	F	F	F	7.57E-07	26	0.402
Diagonal 9	5000	F	F	F	F	F	F
	10000	F	F	F	F	F	F
Extended MCCORMCK	5000	4.39E-07	575	5.162	F	F	F
	10000	F	F	F	F	F	F
Extended DENSCHNB	5000	5.72E-07	19	0.568	F	F	F
	10000	9.11E-07	63	1.323	F	F	F
Full Hessian FH3	5000	F	F	F	7.04E-07	18	0.605
	10000	F	F	F	1.84E-07	46	2.774
Generalized PSC1	5000	1.00E-06	1408	42.258	2.9158	F	F
	10000	9.96E-07	1724	91.176	6.3358	F	F
MDF EXPLIN 1	5000	F	F	F	9.40E-07	41	0.3
	10000	F	F	F	8.87E-07	42	0.44
MODF COSINE	5000	4.41E-03	F	F	8.26E-07	72	0.685
	10000	3.12E-03	F	F	9.94E-07	74	0.886
MODF SINE	5000	1.76E-07	1	0.018	1.76E-07	1	0.024
	10000	6.23E-08	1	0.03	6.23E-08	1	0.033
NONSCOMP	5000	8.84E-07	126	2.051	2.45E+02	F	F
	10000	7.36E-07	59	1.885	4.66E+02	F	F
QUARTC	5000	0.00E+00	1	0.025	0.00E+00	1	0.019
	10000	0.00E+00	1	0.037	0.00E+00	1	0.032
RMDF GENHUMPS	5000	F	F	F	F	F	F
	10000	F	F	F	F	F	F
RMDF SINE	5000	4.22E-07	69	0.81	2.64E-07	10	0.108
	10000	4.57E-07	42	1.039	3.74E-07	10	0.169

5 Conclusion

In this paper, a general form of the DL-BAN CGM was proposed by forming a linear combination of numerator and denominator terms in the two existing classical CGMs. This approach is capable of producing many new methods, which can be obtained by the different arrangements of the coefficients in the DL-BAN update parameter. A new hybrid CGM has been generated from this general form which has been shown to possess a desirable feature, such as the satisfaction of the sufficient descent condition, which is very vital to the global convergence of the method.

A numerical test of the new method in comparison with four existing classical CGMs, confirmed that the new CGM is capable of superior computational performance over a larger number of the existing methods, with respect to the number of iterations and the CPU time. This result is indicative that it is very probable to generate other CGMs from (10) that are computationally optimal in efficiency. Hence, there is a need to explore the DL-BAN CGM for the best possible classical CGM.

References

[1] N. Andrei, An unconstrained optimization test functions collection, *Adv. Model. Optim.* 10(1) (2008), 147-161.

[2] O.M. Bamigbola, M. Ali and E. Nwaeze, *An efficient and convergent method for unconstrained nonlinear optimization*, Proceedings of International Congress of Mathematicians, Hyderabad, India, 2010.

[3] Edwin K.P. Chong and Stanislaw H. Zak, *An Introduction to Optimization*, 2nd ed., New York: John Wiley Sons, Inc., 2001.

[4] Y.H. Dai and L.Z. Liao, New conjugacy conditions and related nonlinear conjugate gradient methods, *Appl. Math. Optim.* 43 (2001), 87-101. https://doi.org/10.1007/s002450010019

[5] S.S. Djordjevic, New hybrid conjugate gradient method as a convex combination of Ls and Fr methods, *Acta Math. Sci.* 39(1) (2019), 214-228. `https://doi.org/10.1007/s10473-019-0117-6`

[6] E.D. Dolan and J.J. More, Benchmarking optimization software with performance profiles, *Math. Program.* 91 (2002), 201-213. `https://doi.org/10.1007/s101070100263`

[7] R. Fletcher and C.M. Reeves, Function minimization by conjugate gradients, *The Computer Journal* 7 (1964), 149-154. `https://doi.org/10.1093/comjnl/7.2.149`

[8] R. Fletcher, *Practical Methods of Optimization*, 2nd ed., Chichester: John Wiley, 1987.

[9] M.R. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, *Journal of Research of the National Bureau of Standards* 49 (1952), 409-436. `https://doi.org/10.6028/jres.049.044`

[10] X. Li and X. Zhao, A hybrid conjugate gradient method for optimization problems, *Natural Science* 3 (2011), 85-90. `https://doi.org/10.4236/ns.2011.31012`

[11] O.L. Mangasarian, *Nonlinear Programming*, New York: McGraw-Hill, 1969.

[12] J.L. Nazareth, Conjugate Gradient Methods, in: *Encyclopedia of Optimization* (C.A. Floudas and P.M. Pardalos, Editors), Kluwer Academic Publishers, Boston, 1999.

[13] J. Nocedal, Theory of algorithms for unconstrained optimization, *Acta Numerica* 1 (1992), 199-242. `https://doi.org/10.1017/S0962492900002270`

[14] I.A. Osinuga and I.O. Olofin, Extended hybrid conjugate gradient method for unconstrained optimization, *Journal of Computer Science and its Application* 25 (2018), 25-33.

[15] X. Xu and Fy. Kong, New hybrid conjugate gradient methods with the generalized Wolfe line search, *SpringerPlus* 5 (2016), Article number: 881. `https://doi.org/10.1186/s40064-016-2522-9`