A new Spectral Gradient Coefficient is based on the Convex Combination of the two Different Conjugate Coefficients for Optimization

Dr. Basim A. Hassan[1], Sawsan S. Ismael[2], Osam M. Taher[3]

[1]Assist Professor / College of Computers Sciences and Math.\ University of Mosul / Iraq
[2]Teacher / College of Education \ University of Mosul / Iraq
[3]Assist Teacher / College of Basic Education \ University of Mosul / Iraq

ABSTRACT

This paper given a new spectral gradient coefficient is obtained using a convex combination of two different gradient coefficients for solving unconstrained optimization. The new method always generates a sufficient descent direction independent of the line search employed. We establish the global convergence of method .The numerical results show that the given method is competitive to the other conjugate gradient methods for the test problems.

Keyword: Unconstrained optimization, Spectral cnjugate gradient method, Global convergent property.

HOW TO CITE THIS ARTICLE

Dr. Basim A. Hassan, Sawsan S. Ismael, Osam M. Taher, "A new Spectral Gradient Coefficient is based on the Convex Combination of the two Different Conjugate Coefficients for Optimization", International Journal of Enhanced Research in Science, Technology & Engineering, ISSN: 2319-7463, Vol. 7 Issue 2, February-2018.

INTRODUCTION

The nonlinear conjugate gradient methods (CG) are useful in finding the minimum value of function for unconstrained optimization. When applied to solve the unconstrained optimization problem $\min_{x\in R^n} f(x)$, CG method usually generates a sequence $\{x_k\}$ by :

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

where x_k is the kth iterative point, α_k is called the step-size determined by some line search. In (2), d_k is the search direction defined by:

$$d_{k+1} = -g_{k+1} + \beta_k d_k \;, \; d_0 = -g_0 \qquad .... (2)$$

where β_k is a scalar and its gradient $\nabla f(x)$ is often denoted by $g(x)$. The best-known formulas for β_k are called the Fletcher-Reeves (FR) [5] and the Dai and Yuan (DY) [4], formulas and are given by:

$$\beta_k^{FR} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} \;, \; \beta_k^{DY} = \frac{g_{k+1}^T g_{k+1}}{y_k^T d_k} \qquad .... (3)$$

In the already-existing convergence analysis and implementations of the CG, the Wolfe conditions, namely:

$$f(x_k + \alpha_k d_k) \le f(x_k) + \delta_1 \alpha_k d_k^T g_k \qquad .... (4)$$

$$g(x_k + \alpha_k d_k)^T d_k \geq \delta_2 d_k^T g_k \qquad \text{..........} \quad (5)$$

where d_k is descent direction, i.e., $g_k^T d_k < 0$ and $0 < \delta_1 \leq \delta_2 < 1$. More performance profile, is given in [7]. Dai and Yuan [3] showed that the FR method is globally convergent if the strong Wolfe conditions are satisfied.

In this paper, we focus on the spectral conjugate gradient methods. We organized this paper as follows: In Section 2, a new spectral conjugate gradient formula and the corresponding algorithm are given. Section 3, establishes global convergent with the Wolfe line search for the new method. The results of some numerical experiments are presented in Section 4. Finally we present the discussion of result and conclusion in the last part.

NEW SPECTRAL CONJUGATE GRADIENT METHOD

In the past few years, to generate sufficient descent directions, some modified conjugate gradient methods are proposed and this is classified as class.

The approach is to modify search direction such that the generated direction satisfies $g_{k+1}^T d_{k+1} \leq -c\|g_{k+1}\|^2$. Zhang et al. [8] proposed a modified Fletcher–Reeves conjugate gradient method, in which the direction d_k is given by:

$$d_{k+1} = -\left(1 + \beta_k \frac{d_k^T g_{k+1}}{\|g_{k+1}\|}\right) g_{k+1} + \beta_k d_k . \qquad \text{..........} \quad (6)$$

In an effort to improve the CG method, Basim et al. [2], we combine the INQ method which has good computational properties with the BSQ method which has strong convergence properties.

$$\begin{aligned}\beta_k &= (1 - r_k)\beta_k^{INQ} + r_k \beta_k^{BSQ} \\ &= (1 - r_k)\frac{y_k^T g_{k+1}}{\xi_{k+1}} + r_k \frac{\|g_{k+1}\|^2}{\xi_{k+1}}\end{aligned} \qquad \text{..........} \quad (7)$$

where $\xi_{k+1} = \alpha_k (g_k^T d_k)^2 / 2\left(f_k - f_{k+1}\right)$. Given our new formula :

$$\beta_k = \begin{cases} (1 - r_k)\beta_k^{INQ} + r_k \beta_k^{BSQ} & 0 < \beta_k^{INQ} < \left|\beta_k^{BSQ}\right| \\ \beta_k^{BSQ} & other \quad wise \end{cases} \qquad \text{..........} \quad (8)$$

when $0 < \beta_k^{INQ} < \left|\beta_k^{BSQ}\right|$, we let if $r_k \leq 0$, then set $\beta_k^{SCB} = \beta_k^{INQ}$, if $r_k \geq 1$, then set $\beta_k^{SCB} = \beta_k^{BSQ}$.

Then when $0 < \beta_k^{INQ} \leq \left|\beta_k^{BSQ}\right|$ and $0 < r_k < 1$, we take the specific form of r_k, from (8) and conjugacy condition $y_k^T d_{k+1} = 0$, we obtain :

$$\begin{aligned}& y_k^T \left(-\vartheta_k g_{k+1} + \beta_k d_k\right) = 0 , \\ & -\left(1 + \beta_k \frac{d_k^T g_{k+1}}{\|g_{k+1}\|^2}\right) y_k^T g_{k+1} + \beta_k y_k^T d_k = 0 , \\ & -\left(1 + [(1 - r_k)\beta_k^{INQ} + r_k \beta_k^{BSQ}]\frac{d_k^T g_{k+1}}{\|g_{k+1}\|^2}\right) y_k^T g_{k+1} + \\ & [(1 - r_k)\beta_k^{INQ} + r_k \beta_k^{BSQ}] y_k^T d_k = 0 ,\end{aligned} \qquad \text{..........} \quad (9)$$

From (9) we get :

$$\begin{aligned} r_k &\left[\beta_k^{BSQ} \frac{d_k^T g_{k+1}}{\|g_{k+1}\|^2} y_k^T g_{k+1} - \beta_k^{INQ} \frac{d_k^T g_{k+1}}{\|g_{k+1}\|^2} y_k^T g_{k+1} - \beta_k^{BSQ} y_k^T d_k + \beta_k^{INQ} y_k^T d_k\right] \\ &= -\beta_k^{INQ} \frac{d_k^T g_{k+1}}{\|g_{k+1}\|^2} y_k^T g_{k+1} + \beta_k^{INQ} y_k^T d_k - y_k^T g_{k+1}\end{aligned} \qquad \text{..........} \quad (10)$$

Therefore :

$$r_k = \frac{\beta_k^{INQ}\left(y_k^T d_k - \frac{d_k^T g_{k+1}}{\|g_{k+1}\|^2} y_k^T g_{k+1} \right) - y_k^T g_{k+1}}{\left(\beta_k^{BSQ} - \beta_k^{INQ}\right)\left(\frac{d_k^T g_{k+1}}{\|g_{k+1}\|^2} y_k^T g_{k+1} - y_k^T d_k \right)} \quad .... \quad (11)$$

That the form of r_k was determined by (11).
In this section, we will give our new algorithm and the assumptions.

Algorithm BSS

Step 1: Given $x_1 \in R^n$, $\varepsilon > 0$, $d_1 = -g_1$, if $\|g_1\| \leq \varepsilon$, then stop.

Step 2 : Find α_k satisfying the Wolfe conditions (4-5), let $x_{k+1} = x_k + \alpha_k d_k$.

Step 3 : If $\|g_{k+1}\| \leq \varepsilon$, then stop, otherwise go to Step 4.

Step 4 : Compute β_k by the formula (8), then generate d_{k+1} by (6).

Step 5 : Let $k = k + 1$, and go to Step 2.

GLOBAL CONVERGENCE

To analyze the global convergence of Algorithm A, the following assumption is necessary.
i. The level set $\wedge = \{x \in R^n \mid f(x) \leq f(x_1)\}$ is bounded.

ii. In a neighborhood U of $\wedge$, $f(x)$ is continuously differentiable and its gradient is Lipschitz continuous, namely, there exists a constant $L > 0$ such that $\|g(x) - g(y)\| \leq \|x - y\|$, $\forall\, x, y \in U$. More details can be found in [6].
The following lemma gives the well-known Zoutendijk conditions [9].

Lemma 3.1.
Suppose that Assumption (i) and (ii) hold, and consider any iteration of the form (2), where d_k is a descent direction and α_k satisfies the weak Wolfe conditions (4) and (5). Then the Zoutendijk condition :

$$\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty \quad .... \quad (12)$$

holds.

Lemma 3.2.
Consider the Algorithm 2.1, where α_k satisfies the Wolfe conditions (4) and (5). Then the direction d_{k+1} given by (8) satisfies the sufficient descent condition :

$$d_{k+1}^T g_{k+1} = -\|g_{k+1}\|^2, \quad \forall\ k . \quad .... \quad (13)$$

Proof :
From (8), the result clearly holds for $k = 0$, we get $d_1^T g_1 = -\|g_1\|^2$. When $k > 0$, we get :

$$\begin{aligned} g_{k+1}^T d_{k+1} &= -\left(1 + \beta_k \frac{g_{k+1}^T d_k}{\|g_{k+1}\|^2}\right)\|g_{k+1}\|^2 + \beta_k g_{k+1}^T d_k \\ &= -\|g_{k+1}\|^2 - \beta_k \frac{g_{k+1}^T d_k}{\|g_{k+1}\|^2}\|g_{k+1}\|^2 + \beta_k\, g_{k+1}^T d_k \quad .... \quad (14) \\ &= -\|g_{k+1}\|^2 \end{aligned}$$

We see that (13) holds for all $k > 0$, which concludes the proof.

Lemma 3.3.

Suppose that Assumption (i) and (ii) hold. Consider the Algorithm 2.1, where α_k satisfies the Wolfe conditions (4) and (5). Then β_k determined by (8) satisfies :

$$0 < \beta_k < \beta_k^{BSQ} . \quad \text{..........} \quad (15)$$

Proof :

From the line search conditions (4), (5) and the sufficient descent condition (13), we have :

$$\frac{\alpha_k (g_k^T d_k)^2}{2(f_k - f_{k+1})} \le \frac{\alpha_k (g_k^T d_k)^2}{-2\delta g_k^T v_k} \le \frac{1}{-2\delta} g_k^T d_k \ge \frac{1}{2\delta} \|g_k\|^2 \quad \text{..........} \quad (16)$$

In this case, it is easy to show that :

$$0 < \beta_k^{BSQ} = \frac{\|g_{k+1}\|^2}{\alpha_k (g_k^T d_k)^2 / 2(f_k - f_{k+1})} \le \frac{\|g_{k+1}\|^2}{(1/2\delta)\|g_k\|^2} . \quad \text{..........} \quad (17)$$

Therefore :

$$\left| \beta_k^{BSQ} \right| < \beta_k^{BSQ} . \quad \text{..........} \quad (18)$$

When $0 < \beta_k^{INQ} < \left| \beta_k^{BSQ} \right|$, if $0 < r_k < 1$, by (8), we have :

$$\begin{aligned} \beta_k &= (1 - r_k)\beta_k^{INQ} + r_k \beta_k^{BSQ} \\ &< (1 - r_k)\beta_k^{BSQ} + r_k \beta_k^{BSQ} \\ &< \beta_k^{BSQ} , \end{aligned} \quad \text{..........} \quad (19)$$

else if $r_k \ge 1$,

$$\beta_k = \beta_k^{BSQ} > 0 , \quad \text{..........} \quad (20)$$

else

$$0 < \beta_k = \beta_k^{INI} \le \beta_k^{BSQ} . \quad \text{..........} \quad (21)$$

Other with we have :

$$\beta_k = \beta_k^{BSQ} > 0 . \quad \text{..........} \quad (22)$$

Theorem 3.4.

Suppose that Assumption (i) and (ii) hold. $\{x_k\}$ is a sequence generated by Algorithm 2.1, α_k satisfies the Wolfe conditions (4) and (5). Then either $g_{k+1} = 0$ for some k or

$$\lim_{k \to \infty} \inf \|g_{k+1}\| . \quad \text{..........} \quad (23)$$

Proof :

If the theorem is not true, there exists a constant $\varepsilon > 0$, such that :

$$\|g_{k+1}\| > \varepsilon , \quad \forall k . \quad \text{..........} \quad (24)$$

We now rewrite (6) as :

$$d_{k+1} + \vartheta_k g_{k+1} = \beta_k d_k . \quad \text{..........} \quad (25)$$

Squaring both sides of the above equation, we get :

$$\|d_{k+1}\|^2 + 2\vartheta_k g_{k+1}^T d_{k+1} + (\vartheta_k)^2 \|g_{k+1}\|^2 = (\beta_k)^2 \|d_k\|^2 . \quad \text{..........} \quad (26)$$

Since the sufficient descent condition (13) is hold, we obtain :

$$\begin{aligned}\|d_{k+1}\|^2 &= (\beta_k)^2\|d_k\|^2 - (\vartheta_k)^2\|g_{k+1}\|^2 - 2\vartheta_k g_{k+1}^T d_{k+1} \\ &= (\beta_k)^2\|d_k\|^2 - (\vartheta_k)^2\|g_{k+1}\|^2 + 2\vartheta_k\|g_{k+1}\|^2 \qquad \text{..........} \quad (27) \\ &= (\beta_k)^2\|d_k\|^2 - (\vartheta_k - 1)^2\|g_{k+1}\|^2 + \|g_{k+1}\|^2\end{aligned}$$

Dividing both sides by $(g_{k+1}^T d_{k+1})^2$ and applying (27), we have :

$$\begin{aligned}\frac{\|d_{k+1}\|^2}{(g_{k+1}^T d_{k+1})^2} &= (\beta_k)^2\frac{\|d_k\|^2}{(g_{k+1}^T d_{k+1})^2} - (\vartheta_k - 1)^2\frac{\|g_{k+1}\|^2}{(g_{k+1}^T d_{k+1})^2} + \frac{\|g_{k+1}\|^2}{(g_{k+1}^T d_{k+1})^2} \\ &\le (\beta_k)^2\frac{\|d_k\|^2}{(g_{k+1}^T d_{k+1})^2} + \frac{\|g_{k+1}\|^2}{(g_{k+1}^T d_{k+1})^2} \qquad \text{..........} \quad (28) \\ &\le (\beta_k^{BSQ})^2\frac{\|d_k\|^2}{\|g_{k+1}\|^4} + \frac{1}{\|g_{k+1}\|^2}\end{aligned}$$

By (17), we obtain :

$$0 < \beta_k^{BSQ} \le \frac{\|g_{k+1}\|^2}{(1/2\delta)\|g_k\|^2}. \qquad \text{..........} \quad (29)$$

Then we get :

$$\begin{aligned}\frac{\|d_{k+1}\|^2}{(g_{k+1}^T d_{k+1})^2} &\le \left(\frac{\|g_{k+1}\|^2}{(1/2\delta)\|g_k\|^2}\right)^2\frac{\|d_k\|^2}{\|g_{k+1}\|^4} + \frac{1}{(g_{k+1}^T d_{k+1})^2} \\ &= \frac{\|d_k\|^2}{(1/2\delta)\|g_k\|^4} + \frac{1}{\|g_{k+1}\|^2} \qquad \text{..........} \quad (30) \\ &= \frac{\|d_k\|^2}{(1/2\delta)(g_k^T d_k)^2} + \frac{1}{\|g_{k+1}\|^2}\end{aligned}$$

Noting that :

$$\frac{\|d_1\|^2}{(g_1^T d_1)^2} = \frac{1}{\|g_1\|^2} \qquad \text{..........} \quad (31)$$

With this, from (24) we have :

$$\frac{\|d_{k+1}\|^2}{(g_{k+1}^T d_{k+1})^2} \le \sum_{i=1}^{k+1}\frac{1}{(1/2\delta)\|g_i\|^2} \le \frac{k}{(1/2\delta)\ \varepsilon^2}. \qquad \text{..........} \quad (32)$$

Then we get from (19) that :

$$\frac{(g_{k+1}^T d_{k+1})^2}{\|d_{k+1}\|^2} \ge \frac{(1/2\delta)\ \varepsilon^2}{k}. \qquad \text{..........} \quad (33)$$

which indicates :

$$\sum_{k=1}^{\infty}\frac{(g_{k+1}^T d_{k+1})^2}{\|d_{k+1}\|^2} \ge \sum_{k=1}^{\infty}\frac{(1/2\delta)\ \varepsilon^2}{k} = \infty. \qquad \text{..........} \quad (34)$$

This contradicts the Zoutendijk condition (12), concluding the proof.

NUMERICAL EXPERIMENTS

In this section, we give some numerical results of new Algorithm to show that the method is efficient for unconstrained optimization problems. The problems that we tested are from [1].

Table 1 show the computation results, where the columns have the following meanings : NI, NR and NF stand for the total number of all iterations, the number of restart calls and the total number of function evaluations, respectively; $\|g_{k+1}\|$ is the norm of the residual at the stopping point. The code is written in Fortran 90.

We used 15 test problems with dimension 100 and 1000 to test the performance of the proposed methods. We define a termination criterion for the methods as $\|g_{k+1}\| \le 10^{-6}$.

DISCUSSION OF RESULT AND CONCLUSION

From the above table, it is clear that our modification of the spectral Gradient coefficient is still very much in place. Though we obtained good results and rate of convergence was moderate improved in the new algorithm.

We conclude by affirming that the new algorithm in this work was effective in the computational treatment of unconstrained optimization problems.

Table 1: Comparison of different CG-algorithms with different test functions and different dimensions

		FR algorithm			BSS algorithm		
P. No.	**n**	**NI**	**NR**	**NF**	**NI**	**NR**	**NF**
1	100	47	18	93	38	20	83
	1000	78	45	131	37	16	83
2	100	43	18	88	37	21	85
	1000	46	19	92	37	19	78
3	100	32	15	52	13	7	26
	1000	22	10	42	15	10	28
4	100	32	13	64	10	5	20
	1000	77	46	129	14	7	28
5	100	37	8	67	41	16	68
	1000	73	27	115	55	22	93
6	100	12	5	25	10	6	19
	1000	14	6	29	11	6	21
7	100	15	9	31	8	6	17
	1000	8	6	17	7	5	15
8	100	180	60	313	85	23	155
	1000	F	F	F	96	31	182
9	100	124	41	231	53	9	93
	1000	445	196	711	166	30	299
10	100	71	35	110	36	14	66
	1000	47	15	84	22	8	44
11	100	13	7	25	12	8	23
	1000	15	7	29	12	7	25
12	100	121	65	218	69	24	114
	1000	345	169	634	262	79	462
13	100	47	21	123	97	31	150
	1000	370	88	616	245	66	406
14	100	23	11	45	21	13	41
	1000	27	11	55	21	10	44
15	100	49	22	80	18	12	31
	1000	129	67	166	14	10	27
Total		**2569**	**1060**	**4415**	**1466**	**510**	**2544**

Fail : The algorithm fail to converge.

Problems numbers indicant for : 1. is the Extended Rosenbrock, 2. is the Extended White & Holst, 3. is the Extended Beale, 4. is the Extended Tridiagonal 1, 5. is the Generalized Tridiagonal 2, 6. is the Extended Himmelblau, 7. is the Extended PSC1, 8. is the Extended Powell, 9. is the Quadratic Diagonal Perturbed, 10. is the Extended Wood, 11. is the NONDIA (CUTE), 12. is the DIXMAANE (CUTE), 13. is the Partial Perturbed Quadratic, 14. is the LIARWHD (CUTE), 15. is the DENSCHNC (CUTE).

REFERENCES

[1]. Andrie N., (2008), ' An Unconstrained Optimization Test functions collection ' Advanced Modeling and optimization. 10, pp.147-161.
[2]. Basim A. H. and Haneen A. A., (2016), New Nonlinear Conjugate Gradient Formulas for Solving Unconstrained Optimization Problems, Al-Mustansiriyah Journal of Science, 3, pp. 82-88.
[3]. Dai, Y. H., Yuan, Y., (1996),' Convergence properties of the Fletcher- Reeves method, IMA J. Numer. Anal. 2, pp.155-164.
[4]. Dai Y. H. and Yuan Y., (1999). A nonlinear conjugate gradient method with a strong global convergence property, SIAM J. optimization, pp. 177-182.
[5]. Fletcher, R. and Reeves C. (1964) 'Function minimization by conjugate gradients ' Computer J, 7, pp. 149-154.
[6]. Junli D. and Baocong J., (2010),' A new hybrid PRP-DY conjugate gradient method, Third International Joint Conference on Computational Science and Optimization, pp. 70-74.
[7]. Jian W. and Xuebin C., (2004),' CG Global Convergence Properties with Goldstein Linesearch, Bull Braz Math Soc, New Series 36(2), pp.197-204.
[8]. Zhang, L.W. Zhou, D. Li, (2006), 'Global convergence of a modified Fletcher–Reeves conjugate gradient method with Armijo-type line search', Numer. Math. 104, PP. 561–572.
[9]. Zoutendijk G., (1970),' Nonlinear Porgramming, computational methods, in : Integr and Nonlinear Programming, North-Holland, Amsterdam, pp. 37-86.