African Journal of Advanced Pure and Applied Sciences (AJAPAS)
Online ISSN: 2957-644X
Volume 1, Issue 4, October-December 2022, Page No: 192-202
Website: https://aaasjournals.com/index.php/ajapas/index

Modification Of The Fletcher-Reeves Conjugate Gradient Method In Unconstrained Optimization

Khalil K. Abbo[1], Naba M. Hasan[2*]
[1]Mathematics department, University of Telafer, Telafer, Iraq
[2]Mathematics department, Mosul University, Mosul, Iraq

[*]*Corresponding author:* nabaa.20csp128@student.uomosul.edu.iq

Received: October 24, 2022	*Accepted: November 09, 2022*	*Published: November 11, 2022*

Abstract:
Conjugate gradient (CG) methods are a class of unconstrained optimization algorithms with strong local and global convergence qualities and low memory needs. The Hestenes-Stiefel and Polack-Ribier techniques are occasionally more efficient than the Fletcher-Reeves (FR) approach, although it is slower. The numerical performance of the Fletcher-Reeves method is sometimes inconsistent, since when the step (S_k) is small, then, $g_{k+1} \approx g_k$ hence d_{k+1} and d_k can be dependent. This paper introduces a modification to the FR method to overcome to this disadvantage. The algorithm uses the strong Wolfe line search conditions. The descent property and global convergence for the method is provided. Numerical results are also reported.

Keywords: Optimization, Unconstrained Optimization , Conjugate Gradient, Modified FR

Cite this article as: K. K. Abbo, N. M. Hasan, "Modification Of The Fletcher-Reeves Conjugate Gradient Method In Unconstrained Optimization," *African Journal of Advanced Pure and Applied Sciences (AJAPAS)*, vol. 1, no. 4, pp. 192-202, October-December2022.

1-Intrduction

The gradient of the nonlinear conjugate (CG) approach has several applications in a variety of areas and is a particularly effective method for addressing unconstrained minimization problems on a large scale [1].
The current gradient and prior direction are needed for each iteration of this approach, which is an iterative process and is distinguished by minimal memory needs and strong local and global convergence qualities [2,3] .In this article, we emphasize the use of conjugate gradient techniques to solve the non-linear unconstrained minimization issue [4]

$$\min \ f(x), \quad x \in R^n \ . \qquad (1.1)$$

Where $f : R^n \to R$ is Constantly differentiable function that is below bounded. A sequencer is created by using the conjugate gradient approach. $x_k, \ k \geq 1$ Starting with the first guess $x_1 \in R^n$, we use repetition

$$x_{k+1} = x_k + \alpha_k d_k \qquad (1.2)$$

When a line search is used to determine the positive step size and the rule is used to create the directions:

$$d_{k+1} = -g_{k+1} + \beta_k d_k , \quad d_1 = -g_1 \qquad (1.3)$$

Where $g_k = \nabla f(x_k)$, and let $y_k = g_{k+1} - g_k$ and $s_k = x_{k+1} - x_k$, here β_k is the CG update parameter. There are many conjugate gradient techniques available.[5,6,7,8,9,10,11,12] and an excellent survey of them, with special attention on their global convergence, is given by[13]

Different CG algorithms correspond to different choices for the scalar parameter β_k. Some of these methods, such as [14],[15]] and Conjugate descent proposed by [3]

$$\beta_k^{FR} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}. \qquad \beta_k^{DY} = \frac{g_{k+1}^T g_{k+1}}{y_k^T d_k}. \qquad \beta_k^{CD} = \frac{g_{k+1}^T g_{k+1}}{-d_k^T g_k}.$$

Have high convergence qualities, however jamming may cause them to function poorly in . Contrarily, the techniques of [16],[17,18]

$$\beta_k^{PR} = \frac{y_k^T g_{k+1}}{g_k^T g_k}. \qquad \beta_k^{HS} = \frac{y_k^T g_{k+1}}{y_k^T d_k}. \qquad \beta_k^{LS} = \frac{y_k^T g_{k+1}}{-d_k^T g_k}.$$

May not generally be convergent, but they often have better computational performance. Despite the fact that the aforementioned formulas are all identical for convex quadratic functions, however, they perform differently for non-quadratic functions, the performance

Coefficient has a significant impact on a non-linear CG method β_k.

Many writers have examined the convergence behavior of the aforementioned formulas under some line search conditions for a very long time. [2,3]

The weak Wolfe (WWF) line search conditions in the CG method's already-existing convergence analysis and implementation are as follows
[19]:

$$f(x_k + \alpha_k d_k) \le f(x_k) + \rho \alpha_k g_k^T d_k \qquad (1.4)$$

$$g(x_k + \alpha_k d_k)^T d_k \ge \sigma g_k^T d_k \qquad (1.5)$$

Where $0 < \rho < \sigma < 1$. and d_k is a descent direction. The strong Wolfe (SWF) conditions consist of [20] (1.6) and

$$\left| g(x_k + \alpha_k d_k)^T d_k \right| \le \sigma \left| g_k^T d_k \right| \qquad (1.6)$$

The sufficient descent property is another, namely

$$g_k^T d_k < -c \|g_k\|^2 \qquad (1.7)$$

The nonlinear conjugate gradient methods with the approximate line search techniques must globally converge, where c is a positive constant. [13,3]

This paper is organized as follows in Section 2 we modified the FR method and we introduce the modified algorithm. The descent property and global convergence for convex functions of our modification method is presented in Section 3. By contrasting our method with a few CG methods, some numerical findings are shown in Section 4.

2. Modified FR method

The last ten years have seen significant effort made into creating new conjugate gradient method modifications that are not only more computationally efficient than the traditional methods, but also have convergence features. These procedures are available in [6].

As a general remark, the convergence theory for the methods with numerator $g_{k+1}^T g_{k+1}$ is better developed than the theory for methods with numerator the $y_k^T g_{k+1}$ of β_k . However, the methods with $y_k^T g_{k+1}$ in numerator of β_k perform better in practice than the methods with $g_{k+1}^T g_{k+1}$ in numerator of β_k [21]In the FR method if a bad direction and a tiny step from x_k to

x_{k+1} are generated, the next d_{k+1} and the next step α_{k+1} are also likely to be

Poor unless a restart along the gradient direction is performed. This paper is informed by the above idea, a modified FR conjugate gradient method was proposed as follows:

$$d_{k+1} = -g_{k+1} + \beta_k^{KN1} d_k \tag{2.1}$$

$$\beta_k^{KN1} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} - \left| \frac{y_k^T g_{k+1}}{y_k^T d_k} \right| \frac{g_{k+1}^T d_k}{g_k^T g_k} \tag{2.2}$$

Where, KN1denotes Khalil and Naba. Any conjugate gradient algorithm has a very simple general structure as illustrated below.

The KN1 Algorithm.

Step 1: Set an initial point x_0 and $\varepsilon > 0$ sufficiently small. Set $d_0 = -g_0$; K=0

Step 2: Test a threshold for ceasing iterations. If this test is successful, quit; otherwise, move on to step 3

Step 3: Using the strong Wolfe line search conditions, determine α_k

Compute $x_{k+1} = x_k + \alpha_k d_k$, f_{k+1}, g_{k+1} and $y_k = g_{k+1} - g_k$, $s_k = x_{k+1} - x_k$

Step 4: Compute the conjugate parameter β_k^{KN1} from (2.2).

Step 5: Compute the search direction $d_{k+1} = -g_{k+1} + \beta_k^{KN1} d_k$ from (2.1)

Step 6: Start over criteria. If $\left| g_{k+1}^T g_k \right| > 0.2 \|g_{k+1}\|$ then set $d_{k+1} = -g_{k+1}$

Step 7: Set k=k+1 and continue with step 2.

In [22] showed that the FR method generates sufficient descent direction and it's global convergence for nonlinear objective functions according to the following theorem.

Theorem (2.1)

Assume that FR method is implemented with strong Wolfe line search (1.4) and (1.6), with $0 < \sigma < 1/2$. Then, The following inequalities are satisfied by the decent directions produced by the FR method.

$$g_{k+1}^T d_{k+1} \le \frac{2\sigma - 1}{1 - \sigma} \|g_{k+1}\|^2 \quad \text{for k=0, 1, …}$$

Based on the above theorem we can prove the descent property of our algorithm KN1with SWF line search in the following theorem

Theorem (2.2)

Consider a CG method with the search direction (2.1) and (2.2) which employs a strong Wolfe line search (1.4) and (16) with $0 < \sigma < 1/2$ then the search, directions are descent.

Proof:

As an example of induction, consider the following. Consider the search direction

$$d_{k+1} = -g_{k+1} + \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} - \left|\frac{y_k^T g_{k+1}}{y_k^T d_k}\right| \frac{g_{k+1}^T d_k}{g_k^T g_k} d_k \tag{2.3}$$

For k=0 then $g_1^T d_1 = -\|g_1\|$. Now assume that $g_k^T d_k \le -c\|g_k\|$. Multiply both sides of (2.3) by g_k^T then

V

$$g_{k+1}^T d_{k+1} = -\|g_{k+1}\| + \left(\frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} - \left|\frac{y_k^T g_{k+1}}{y_k^T d_k}\right| \frac{g_{k+1}^T d_k}{g_k^T g_k}\right) g_{k+1}^T d_k$$
$$= -\|g_{k+1}\| + \left(\frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} g_{k+1}^T d - \left|\frac{y_k^T g_{k+1}}{y_k^T d_k}\right| \frac{(g_{k+1}^T d_k)^2}{g_k^T g_k}\right)$$

Since the last term is positive, therefor

$$g_{k+1}^T d_{k+1} < -\|g_{k+1}\| + \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} g_{k+1}^T d_k$$

Hence by theorem (2.1) it is descent method the prove is complete.

Either of the following assumptions is often utilized in convergence analysis for CG algorithms

Assumptions (A)

1- The level set $S = \{x \in R^n : f(x) \le f(x_0)\}$ is bonded, there exists a constant $B > 0$ so that $\|x\| \le B \ \forall \ x \in S$

2- In some neighborhood $\aleph$ of S, f is continuously differentiable and it's gradient is Lipschitz continuous, i.e there exists a constant $L > 0$ so that $\|g(x) - g(y)\| \le L\|x - y\|$ for all $x, y \in \aleph$

By assumption (A) there exists Γ that $\|g(x)\| \le \Gamma$ for all $x, y \in S$

[2] show that any method with $|\beta| \le \beta_k^{FR}$ is convergent. Based on the following theorem our method is convergent

Theorem (2.3)

Suppose that the assumptions (A) holds. Consider any conjugate gradient Method (1.2)-(1.3) where $|\beta| \le \beta_k^{FR}$ and where the step-size is determined by the strong Wolfe line search (1.4) and (1.6) with $0 < \rho < \sigma < 1/2$. Then, $\lim_{k\to\infty} \inf \|g_k\| = 0.$

For prove see [2]

Since $\left|\beta_k^{KN1}\right| \le \left|\frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}\right| - \left|\frac{y_k^T g_{k+1}}{y_k^T s_k}\right| \left|\frac{s_k^T g_{k+1}}{g_k^T g_k}\right| \le \beta_k^{FR}$

4. Comparative data and results

In this section we present the computation performance of a

MATLAB using a set of unrestricted optimization test problems to implement the KN1and FR algorithms. Extensive or generalized versions of (49) large-scale unconstrained optimization test problems were chosen from [8].
Different dimensions have been taken into account for each function (where n is the number of variables). All algorithms implement the strong Wolfe line search conditions with $\rho = 0.0001$ and $\sigma = 0.9$ and same stopping criterion $\|g_k\|_2 < 10^{-6}$, where $\| \, . \, \|_2$ is the maximum absolute component of a vector.

The comparison of algorithms are given in the following context. We say that, in the particular problem i the performance of Algorithm (Alg1)

Performance of Alg1 was superior to that of Alg2 if the number of restarts (irs), iterations (iter), or function-gradient evolutions (fg) of Alg1 was smaller than the equivalent number of each for Alg2 in each case.

Details of the numerical outcomes for the Fletcher-Revees (FR), Hestenes-Stiefel (HS), and our approach are presented in Table (1). (SPDY).

Compare the results of the method KN with those of other methods (FR&HS) in Table-1

No.	Problems Name/n	KN1 Itr /Tcpu/$\|g_{k+1}\|$	FR Itr /Tcpu/$\|g_{k+1}\|$	S Itr /Tcpu/$\|g_{k+1}\|$
1.	cosine/100	32/0.184/9.18e-007	51/0.008/8.63e-007	348/0.041/9.54e-007
	cosine/1000	21/0.011/4.91e-007	Maximum iteration	Maximum iteration
2.	dixmaana/90	15/0.019/4.88e-007	21/0.011/4.04e-007	33/0.023/9.41e-007
	dixmaana/900	17/0.045/1.32e-007	21/0.047/4.21e-007	33/0.023/9.41e-007
3.	dixmaanb/90	15/0.020/7.16e-007	13/0.010/1.57e-007	31/0.021/9.37e-007
	dixmaanb/900	13/0.062/7.36e-008	20/0.056/7.40e-007	20/0.058/6.90e-007
4.	dixmaanc/90	16/0.021/3.91e-007	23/0.012/4.11e-007	26/0.016/7.02e-007
	dixmaanc/900	16/0.046/4.38e-007	25/0.051/7.44e-007	32/0.110/7.84e-008
5.	dixmaand/90	18/0.019/4.18e-007	27/0.012/2.72e-007	27/0.017/4.76e-007
	dixmaand/900	20/0.049/8.77e-007	26/0.055/9.73e-007	39/0.151/7.67e-007
6.	dixmaane/90	85/0.034/6.24e-007	216/0.065/9.95e-007	90/0.027/9.77e-007
	dixmaane/900	268/0.329/8.57e-007	579/0.734/9.67e-007	220/0.225/9.05e-007
7.	dixmaanf/90	74/0.032/6.19e-007	230/0.068/9.86e-007	77/0.024/9.51e-007
	dixmaanf/900	171/0.205/7.44e-007	419/0.576/9.90e-007	177/0.185/8.15e-007
8.	dixmaang/90	77/0.034/6.75e-007	230/0.066/9.19e-007	79/0.024/5.16e-007
	dixmaang/900	229/0.263/7.85e-007	Maximum iteration	188/0.184/9.29e-007
9.	dixmaanh/90	71/0.031/4.05e-007	150/0.046/9.65e-007	81/0.022/5.69e-007
	dixmaanh/900	230/0.267/7.81e-007	491/0.620/9.91e-007	189/0.205/9.42e-007
10.	dixmaani/90	827/0.237/9.27e-007	1615/0.512/9.88e-00	722/0.168/7.21e-007
	dixmaani/900	"Maximum iteration	Maximum iteration	Maximum iteration
11.	dixmaanj/90	675/0.203/5.88e-007	Maximum iteration	582/0.128/9.55e-007
	dixmaanj/900	"Maximum iteration	Maximum iteration	Maximum iteration
12.	dixmaank/90	1106/0.340/7.09e-00	Maximum iteration	744/0.173/9.99e-007
	dixmaank/900	Maximum iteration	Maximum iteration	Maximum iteration
13.	dixmaanl/90	443/0.144/7.39e-007	Maximum iteration	421/0.123/9.62e-007
	dixmaanl/900	982/1.189/7.37e-007	Maximum iteration	Maximum iteration
14.	dixon3dq/30	388/0.022/8.40e-007	1641/0.071/9.27e-00	307/0.011/9.34e-007
	dixon3dq/300	Maximum iteration	Maximum iteration	Maximum iteration
15.	dqdrtic/30	75/0.008/4.19e-007	184/0.009/7.14e-007	75/0.005/7.23e-007
	dqdrtic/300	113/0.008/8.43e-007	157/0.011/5.31e-007	74/0.007/6.49e-007
16.	edensch/100	31/0.010/8.59e-007	138/0.035/3.72e-007	45/0.010/9.92e-007
	edensch/1000	42/0.062/6.61e-007	732/1.863/7.33e-007	215/0.748/4.98e-007
17.	eg2/100	Maximum iteration	633/0.129/5.30e-007	297/0.039/5.03e-007
	eg2/1000	Maximum iteration	Maximum iteration	Maximum iteration
18.	fletchcr/100	71/0.010/1.63e-007	356/0.055/9.49e-007	140/0.022/5.61e-007
	fletchcr/1000	Maximum iteration	Maximum iteration	Maximum iteration
19.	freuroth/100	Maximum iteration	Maximum iteration	Maximum iteration
	freuroth/1000	Maximum iteration	Maximum iteration	Maximum iteration
20.	genrose/100	Maximum iteration	Maximum iteration	917/0.040/6.24e-007
	genrose/1000	Maximum iteration	Maximum iteration	Maximum iteration
21.	himmelbg/100	2/0.005/3.28e-028	2/0.000/3.36e-028	2/0.000/3.42e-026

	himmelbg/1000	2/0.002/7.51e-030	2/0.001/7.57e-030	2/0.001/1.22e-027
22.	liarwhd/100	89/0.014/9.60e-007	72/0.009/7.93e-007	37/0.008/2.19e-008
	liarwhd/1000	222/0.045/2.64e-007	81/0.020/6.25e-007	40/0.015/9.21e-007
23.	penalty1/100	18/0.033/2.48e-007	20/0.026/7.49e-008	Maximum iteration
	penalty1/1000	30/0.761/4.66e-007	Maximum iteration	Maximum iteration
24.	quartc/100	16/0.008/9.85e-007	33/0.007/6.47e-007	29/0.005/8.17e-007
	quartc/1000	41/0.063/3.05e-007	32/0.078/3.64e-007	52/0.067/6.06e-007
25.	tridia/100	455/0.029/9.51e-007	1275/0.063/9.77e-00	345/0.015/9.59e-007
	tridia/1000	1902/0.246/4.59e-00	Maximum iteration	1504/0.150/9.68e-00
26.	woods/100	Maximum iteration	Maximum iteration	Maximum iteration
	woods/1000	Maximum iteration	Maximum iteration	Maximum iteration
27.	bdexp/100	2/0.006/3.13e-083	2/0.000/1.44e-082	Maximum iteration
	bdexp/1000	2/0.003/2.42e-133	2/0.003/3.25e-133	Maximum iteration
28.	exdenschnf/100	22/0.007/5.65e-007	41/0.003/9.72e-007	27/0.004/4.89e-007
	exdenschnf/1000	24/0.008/6.16e-007	36/0.009/3.36e-007	31/0.015/4.15e-007
29.	exdenschnb/100	16/0.005/3.67e-007	25/0.002/6.58e-007	32/0.004/5.48e-007
	exdenschnb/1000	24/0.004/7.22e-008	45/0.006/3.24e-007	27/0.007/3.62e-007
30.	genquartic/100	18/0.006/5.88e-007	36/0.002/9.98e-007	25/0.002/4.41e-007
	genquartic/1000	20/0.006/3.70e-007	216/0.034/9.41e-007	35/0.011/9.55e-007
31.	biggsb1/100	661/0.039/7.27e-007	1362/0.069/1.00e-00	585/0.025/8.96e-007
	biggsb1/1000	Maximum iteration	Maximum iteration	Maximum iteration
32.	sine/100	25/0.007/2.74e-007	NaN/NaN/NaN	Maximum iteration
	sine/1000	24/0.012/5.99e-007	145/0.073/9.94e-007	Maximum iteration
33.	fletcbv3/100	Maximum iteration	Maximum iteration	Maximum iteration
	fletcbv3/1000	Maximum iteration	Maximum iteration	Maximum iteration
34.	nonscomp/100	48/0.007/8.65e-007	Maximum iteration	52/0.005/9.97e-007
	nonscomp/1000	50/0.014/4.15e-007	Maximum iteration	160/0.029/4.70e-007
35.	power1/100	1769/0.090/8.99e-00	Maximum iteration	1550/0.057/9.52e-00
	power1/1000	Maximum iteration	Maximum iteration	Maximum iteration
36.	raydan1/100	91/0.008/4.96e-007	248/0.015/9.28e-007	84/0.004/9.71e-007
	raydan1/1000	344/0.053/7.66e-007	Maximum iteration	Maximum iteration
37.	raydan2/100	14/0.004/4.84e-007	14/0.001/4.84e-007	Maximum iteration
	raydan2/1000	18/0.007/4.11e-007	29/0.007/8.18e-008	20/0.011/7.87e-008
38.	diagonal1/100	99/0.011/8.38e-007	Maximum iteration	290/0.046/8.61e-007
	diagonal1/1000	Maximum iteration	Maximum iteration	Maximum iteration
39.	diagonal2/100	80/0.009/7.57e-007	158/0.010/7.02e-007	79/0.005/8.68e-007
	diagonal2/1000	313/0.103/7.31e-007	413/0.106/9.99e-007	246/0.057/9.60e-007
40.	diagonal3/100	119/0.013/5.03e-007	Maximum iteration	373/0.064/6.81e-007
	diagonal3/1000	Maximum iteration	Maximum iteration	Maximum iteration
41.	bv/100	Maximum iteration	Maximum iteration	Maximum iteration
	bv/1000	98/0.944/9.10e-007	Maximum iteration	89/1.577/9.27e-007
42.	ie/100	17/0.258/1.90e-009	33/0.358/7.37e-007	19/0.362/9.77e-007
	ie/1000	15/21.924/1.57e-007	34/33.037/7.63e-007	21/30.012/8.91e-007

43.	singx/100 singx/1000	Maximum iteration Maximum iteration	229/0.034/9.96e-007 170/4.583/5.90e-007	221/0.036/9.55e-007 115/3.627/9.87e-007
44.	lin/100 lin/1000	13/0.153/8.84e-007 20/256.591/4.00e-007	13/0.147/8.84e-007 23/282.201/3.07e-007	34/0.480/3.35e-007 23/491.338/7.57e-007
45.	osb2/11	1235/0.210/9.63e-007	Maximum iteration	719/0.115/7.64e-007
46.	pen1/100 pen1/1000	109/0.310/8.98e-007 Maximum iteration	Maximum iteration Maximum iteration	Maximum iteration 263/22.136/6.91e-007
47.	pen2/100 pen2/1000	163/0.069/8.46e-007 "Maximum iteration	"Maximum iteration Maximum iteration	542/0.379/9.37e-007 Maximum iteration
48.	rosex/100 rosex/1000	225/0.037/3.74e-007 188/4.089/5.42e-007	134/0.024/6.15e-007 113/3.144/2.24e-007	39/0.009/7.58e-007 53/1.854/3.56e-007
49.	trid/100 trid/1000	77/0.029/7.52e-007 39/0.667/6.83e-007	352/0.103/9.18e-007 366/5.045/9.85e-007	92/0.028/7.08e-007 49/0.951/8.30e-007

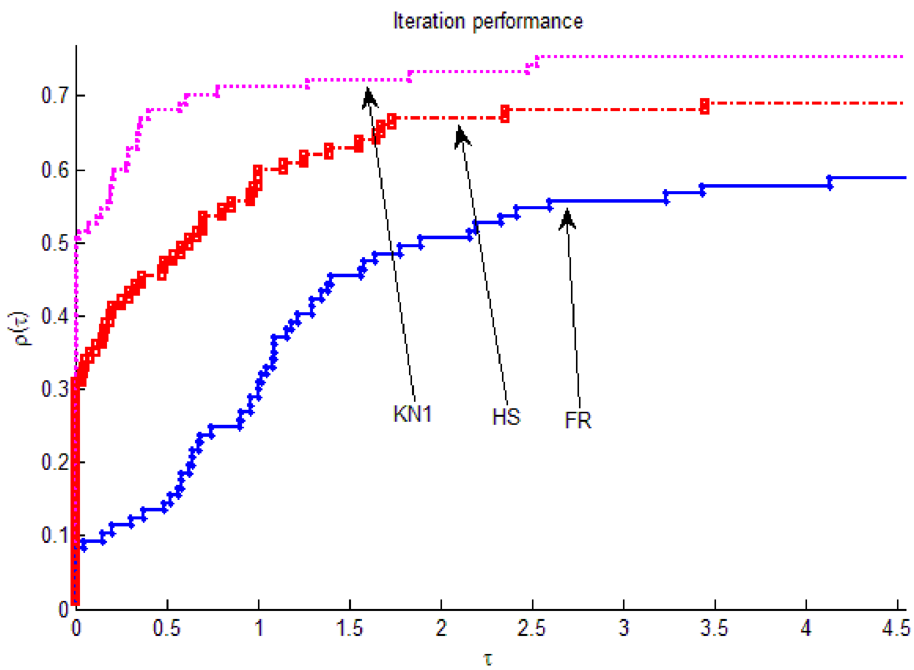

Figure (1) number of iteration

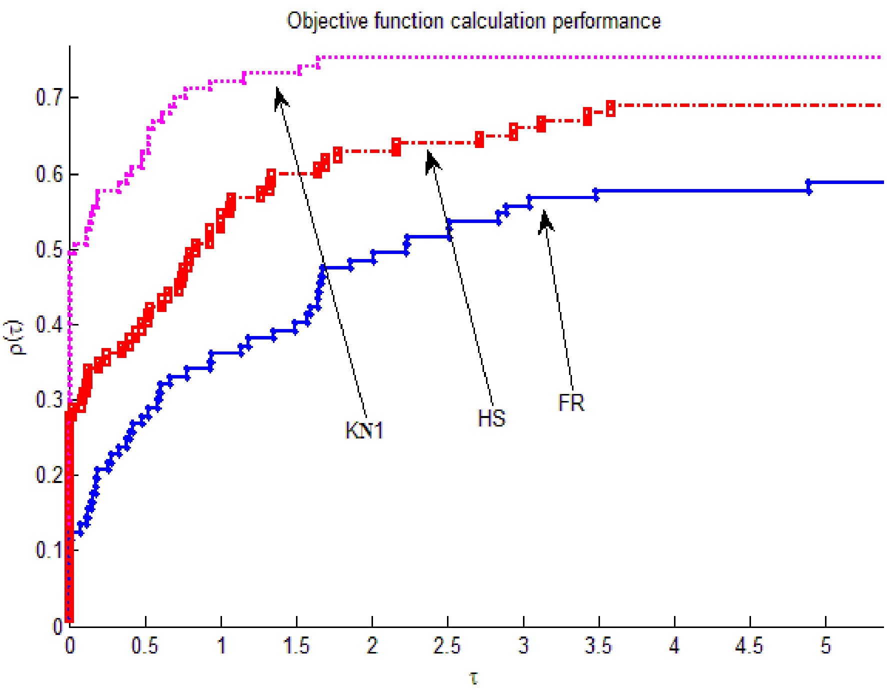

Figure (2) function calculation

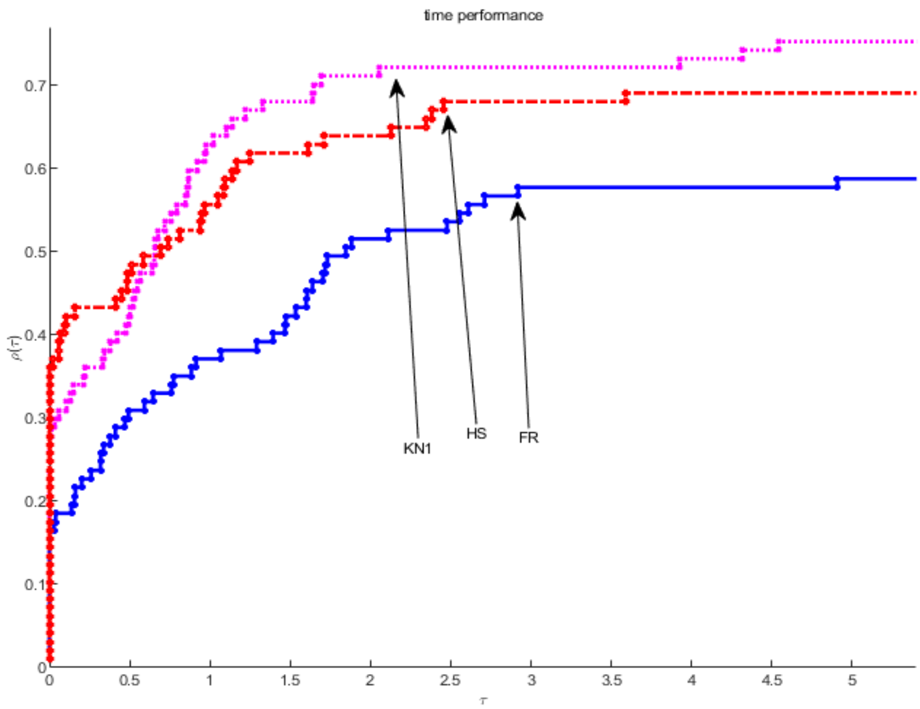

Figure (3) the time it takes to solve the function

Conclusion

The general objective of the presented study is to propose a conjugate gradient algorithm, which can be used to obtain the solution of nonlinear optimization functions. Since the two methods, (FR and DY) have modest practical performance, but they have strong affinity properties. In addition, the two methods (HS and PRP) often have better computational behavior and are not always convergent. Therefore, we proposed method of hybrid conjugate gradient, in order to obtain conjugate gradient methods with high computational efficiency and good convergence properties. That is, we suggested it to avoid the failure of the method and to improve the performance of classical conjugate gradient algorithms, "because the hybrid conjugate gradient algorithms are better than the classical conjugated gradient algorithms. Therefore, it was found that the KN1 algorithm proved its efficiency in all the problems that were examined and solved.

References

[1] A.C.Bovik,(2010).Handbook of Image and Video Processing Academic Press.

[2] Gilbert,J.C.Nocedal,J.(1992): Global convergence properties of conjugate gradient methods for optimization.SIAM J.Optim.,2,pp.21-42

[3] R.Fletcher,Practical Methods of Optimization.John Wiley&Sons, NewYork,NY(2013.(

[4] Perry,J.M.(1977) A class of conjugate gradient algorithms with a two-step variable-metric memory, Discussion Paper 269, Center for Mathematical Studiesin Economic and Management Sciences,Northwestern University,Evanston,IIIionis.،

[5] Andrei,N.(2007),Scaled conjugate gradient algorithms for unconstrained ptimization.Computational Optimization Optimization and Applications,38,pp.401-416

[6] Andrei,N.(2008)`40 conjugate gradient algorithm for unconstrained Optimization` ICI.Technical Report No.13108

[7] Hisham M.Khudhur and Khalil K.Abbo(2021) A New Type of Conjugate Gradient Technique for Solving Fuzzy Nonlinear Algebraic Equations Ibn AI-Haitham Internation Conference for Pure and Applied Sciences(IHICPS) J.Of physics:Conference Series 1879 022111,IOP Publishing. doi:10.1088/1742-6596/1879/2/022111

[8] Hisham M.Khudur and Khalil K.Abbo(2022) Learning Fuzzy Neural Networks by Using Improved Conjugate Gradient Techniques.Sinkron:Jurnal dan Penelitian Teknik Informatika Volume7,Number 3.DOI:https://doi.org/10.33395/sinkron.v7i3.11442

[9] Kalil K.Abbo and Farah H.Mohammed (2011)Spectral Fletcher-Reeves Algorithm for Solving Non-Linear Unconstrained Optimization Problems. Iraqi Journal of Statisical Sciences (19),p.p.]21-38

[10] Nocedal,j.(1996),Conjugate gradient methods and nonlinear optimization,in L.Adams and J.L.Nazareth(Eds) Linear and Nonlinear Conjugate Gradient-Related Methods, SIAM, Philadelphia,,pp.9-33.

[11] Osama M.T.Wais,Khalil A.Abbo and Ibrahim A.Saleh,(2022)A New Spectral Conjugate Gradient method for solving unconstrained Optimization problems. College of Basic Education Researchers J.ISSN:7452-1992Vol.(18),No(3(

[12] Shanno,D.F.,Phua,V:Algorithm 500,Minimization of unconstrained multivariate function,ACM Trans. On Math.Soft.,2, pp.87-94,1976

[13] Hager,W.W.,Zhang,H.(2006):A survey of nonlinear conjugate gradient methods. Pacific journal of Optimization,2,pp.35-58.

[14] R.Fletchher and C.M.Reeves,Function minimization by conjugate gradients. Comput.J.7(1964) 149-154

[15] Y-H.Dai and Y.Yuan,(1999) A nonlinear conjugate gradient method with a strong global convergence Proprety.SIAM J.Optim.10,177-182.

[16] Polak,E.,Ribiere,G.:Note sur la convergence de methods de directions conjuguee,Revue Francaise Informat.Recherche Operationnelle,3e Annee 16(1969),pp.35-43.

[17] Hestenes,M.R,Stiefel,E.:Methods of conjugate gradients for solving liner systems,J.Research Nat.Bur.Standards Sec.B.48,pp.409-436,1952.

[18] Liu,Y.,Storey,C.:Efficient generalized conjugate gradient algorithms,Part 1:Theory.JOTA,69(1991),PP.129-137.

[19] Wolfe,P.,Convergence conditions for ascent methods,SIAM Rev.,Vol.11, pp.226-235,1968.

[20] Wolfe,P., Convergence conditions for ascent methods,(II):some corrections. SIAM Review 13.(1971) 185-188.

[21] Andrei,N.(2008) An unconstrained optimization test functions Collection,Advanced Modeling and Optimization,10.

[22] Al-Baali M.(1985) `Descent property and global convergence of the Fletcher and Revees Method with inexact line search`. SIAM J.Numer.Anal 5