In Accurate CG-Algorithm for Unconstrained Optimization Problems

Nidhal Al-Assady **Maysoon M. Aziz** **Ban Ahmed Mitras**

dr.banah.mitras@gmail.com

College of Computer Sciences and Mathematics
University of Mosul, Iraq

Received on: 13/05/2002 **Accepted on: 01/09/2002**

Abstract

An algorithm for unconstrained minimization is proposed which is invariant to a non-linear scaling of a strictly convex quadratic function and which generates mutually conjugate directions for extended quadratic function. It is derived for inexact line searches and is designed for general use, it compares favorably numerical tests [over eight test functions and dimensionally up to (2-100) with the H/S, DX, F/R, P/R, and A/B algorithms on which this new algorithm is based.

Keyword: CG algorithm, in accurate CG method.

خوارزمية CG المقيسة لمسائل الأمثلية غير المقيدة

نضال الاسدي **ميسون مال الله عزيز** **بان احمد متراس**

كلية علوم الحاسوب والرياضيات، جامعة الموصل

تاريخ الاستلام: 2002/05/13 **تاريخ القبول: 2002/09/01**

الملخـص

في هذا البحث تم تقديم خوارزمية للامثلية غير المقيدة و التي اختصت بالتقييس غير الخطي للدوال التربيعية المحدبة بشدة وعممت لاتجاهات مترافقة موزعة

لتوسيع الدالـة التربيعيـة. واشتقت صيغة لخط البحث غير التام وصممت للاستخدام العام. كما تمت مقارنة هذه الطريقة مع الطرائق السابقة لثماني دوال اختباريه وبأبعاد (2−100) مع صيغ H/S, DX, F/R, P/R. A/B. وقد كانت نتائج الخوارزمية الجديدة اكثر كفاءة" من الخوارزميات الأصلية.

الكلمات المفتاحية: خوارزمية التدرج المترافق، خوارزميات التقييس.

1. Introduction:

The searches related with Conjugate Gradient method begin to apply inexact line searches; i.e not to iterate until the “line minima” is found to some predetermined (small) tolerance in order to reduce the number of function evaluations (NOF).

In order to show improvement of the local rate of convergence and the efficiency of the traditional CG-method several well-known methods are discussed later, namely Dixon (1975) and Nazareth (1977).

The type of these algorithms has quadratic termination property by using an error vector even if inexact searches are used, while Sloboda (1982) presented an algorithm which retains the quadratic termination property without using an error vectors.

In this paper we develop a new general way of the CG algorithm with inexact line searches. This new algorithm is similar to that derived by Dixon (1975) CG method with inexact line searches but does not require the correction term as extra vector storage and different formula is obtained.

2. General Background:

In this section we shall present a brief description of the Dixon Sloboda, Nazareth and Nocedal algorithms.We shall then discuss properties of these algorithms.

2.1 The Dixon's Method (1975):

In Dixon's method the idea is to determine directions parallel to the CG directions (for a quadratic) without line searches, and so he develops a conjugate set. Explicitly, the search directions d_k, k= 1, 2, …, are given by

$d_{k+1} = -\bar{g}_{k+1} + \bar{\beta}_k d_k$,

where $d_1 = -g_1$,

and $\bar{g}_k, \bar{\beta}_k$ are estimation values defined below. The new iterate is defined as:

$$x_{k+1} = x_k + \lambda_k d_k , k \geq 1$$

where λ_k is chosen simply to satisfy conditions defined in the line search subprogram. The gradient $\bar{g}_k$ is estimate of the gradients at the points x_k that we would have reached, if we had performed exact line searches along the d_k. for a quadratic, these gradients can be evaluated exactly, and defined from:

$$\bar{g}_{k+1} = \bar{g}_k + \left(1 - \frac{g_{k+1}^T d_k}{y_k^T d_k}\right) y_k \qquad , k 1 \geq$$

$$\bar{g}_{k+1} = \bar{g}_k + \frac{g_k^T d_k}{y_k^T d_k} y_k$$

and the approximated expressions for the coefficient $\bar{\beta}_k$ (rather than the "true" β_k which occurs in classical CG methods) are defined:

$$\bar{\beta}_k = \frac{\bar{g}_{k+1}^T(\bar{g}_{k+1} - \bar{g}_k)}{d_k^T(\bar{g}_{k+1} - \bar{g}_k)} \quad , \quad \text{(Hestenes and Stiefel , 1952)}$$

$$\bar{\beta}_k = \frac{-\bar{g}_{k+1}^T \bar{g}}{d_k^T \bar{g}_i}$$

(Dixon , 1975)

we have used these formula in the computations which are reported later.

After *n* steps the error vector used to find the minimum of quadratic.

Dixon's method moves along a direction parallel to the CG direction, and so retains the property of the quadratic convergence. (Dixon, 1975).

2.2 The Sloboda Method (1982):

Sloboda developed an algorithm which generates conjugate direction with imperfect searches and has the quadratic termination property without using an error vector. The algorithm for general function is as follows:

The Outline of the SLO-Algorithm:

Step (1) : Set x_0 , $\hat{g}_0 = g_0, \hat{d}_0 = -g_0$

Step (2) : For k = 1,2,…, n

Compute $x_{k+1} = x_k + \lambda_k \hat{d}_k$ where λ_k chosen to satisfy the condition of the line search

Step (3) : If $\|g_{k+1}\| < \varepsilon$ then stop , else go to step (5)

Step (4) : If $k=n+1$ then set $k=0$ and go to step (1), else compute

$$\hat{g}_{k+1} = (g_{k+1} - g_k) - \left[\frac{(g_{k+1} - g_k)^T \hat{g}_k}{\hat{g}_k^T g_k}\right]\hat{g}_k \ ,$$

Step(5) : If $\left\|\hat{g}_{k+1}\right\| < \varepsilon$ go to step (1) , else set $k=k+1$, compute

$$\hat{d}_{k+1} = -\hat{g}_{k+1} + \left[\frac{(g_{k+1} - g_k)^T \hat{g}_{k+1}}{(g_{k+1} - g_k)^T \hat{d}_k} \right] \hat{d}_k \quad ,$$

set $k=k+1$, and go to step (3)

2.3 The Multi-Step Method, Nazareth and Nocedal (1978):

Nazareth and Nocedal show that with inexact line searches a natural extension of conjugate gradient method, the algorithm can be obtained of this method is called NAZ- NOC.

This algorithm is considered as a modification of Gram – Schmidt orthogonalization process, where Nazareth and Nocedal show that not all the coefficients of the Gram-Schmidt process must be computed at every iteration , then suggested the following algorithm :

Step (1) : Set x_1 , $d_1 = -g_1$, $e_1 = 0$

Step (2) : For k= 1,2,……..,n compute

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

Step (3) : Check for convergence .

If $\|g_{k+1}\| < \varepsilon$ then stop, else go to step (4)

Step (4) : if k<n , set k=k+1 , compute

$$e_k = e_{k-1} + \frac{y_{k-1}^T g_{k+1}}{y_{k-1}^T d_{k-1}} d_{k-1} \quad ,$$

go to step (2) , else set k=1 and go to step (1)

The algorithm will have quadratic termination property when using the error term e_{k+1} in the extra step, even if implemented without line search.

3. The New CG –Method with Inexact Line Searches:

In this section a new general way for the conjugate gradient type methods is presented. This new approach has the property of the quadratic termination even if the line search is not exact and the extra correction term is not essential.

Now let the g_k be the gradient of the quadratic function and let

$$g^*_{k+1} = g_{k+1} - \frac{g^T_{k+1} d_k}{g^T_k y_k} y_k \qquad (1)$$

Where $y_k = g_{k+1} - g_k$, then the following lemma is hold.

Lemma(1):

For the quadratic function, the term g^*_{k+1} which is defined in eq.(1) is equivalent to that the gradient g_{k+1} which is obtained by Hestenes and Stiefel method in (1952).

Proof:

Let We Let's define $g_k = Gx_k - b$, and $d_0 = -g_0$ where g_k is the gradient of quadratic function. The biothogonlization process of Hestenes and Stiefel is as follows:

$x_{k+1} = x_k + \lambda_k d_k$, where

$\lambda_{k=}$ - $\frac{g^T{}_k d_k}{g^T{}_k G d_k}$ for exact line searches and G is a symmetric positive define matrix . Thus

$g_{k+1}=g_k + \lambda_k G\, d_k$, (2)

and

$d_{k+1=}$ $-g_{k+1} + \frac{y^T{}_k g_{k+1}}{d^T{}_k y_k} d_k$

Now, in order to prove that g_{k+1} which is obtained by Hestenes and Stiefel exact line searches algorithm is identical to the term $g^*{}_{k+1}$ which is defined in eq.(1), we proceed as follows:

From the definition of $\lambda_{k=}$ $-\frac{g_k^T d_k}{g_k^T G d_k}$,then rewriting eq.(2), we get

$g_{k+1}= g_k- \frac{g_k^T d_k}{g_k^T G d_k}\; G\, d_k$ (3)

Now multiplying and dividing the second term of the eq. (3) by λ, it becomes as:

$g_{k+1}= g_k- \frac{g_k^T d_k}{g_k^T G\lambda d_k}$ □$G\, d_k$,

from the definition we have $y_k= g_{k+1}- g_k = \lambda_k G\, d_k$, then replacing y_k instead of $\lambda_k G\, d_k$ in the above equation then we get:

$g_{k+1}= g_k- \frac{g_k^T d_k}{g_k^T y_k}\; y_k$, thus this equation is identical to the equation which is defined in eq.(1) as:

$g^*{}_{k+1} = g_k - \frac{g^T{}_k d_k}{g^T{}_k y_k} y_k$, thus the is of vectors $\overset{*}{g}_0,\ldots\ldots\ldots,\overset{*}{g}_k$ is orthogonal as in Hestenes and Stiefel method in quadratic function . From the above argument we have the following two corollaries.

Corollary (1):

The term g^*_{k+1} which is defined in eq.(1) is used to be obtained by the following form:

$$\overset{*}{d}_{k+1} = -g^*_{k+1} + \frac{\left(\overset{*}{g}_{k+1} - \overset{*}{g}_k\right)^T g_{k+1}}{\overset{*}{d}{}_k^T\left(\overset{*}{g}_{k+1} - \overset{*}{g}_k\right)}\overset{*}{d}_k , \qquad (4)$$

is parallel to that search direction given by Hestenes and Stiefel algorithm for quadratic function.

Corollary (2):

The search direction which is defined in eq. (4) is a descent direction even if for non-quadratic functions i.e.

$$d^{*T}_{k+1}\, g_{k+1}^* < 0.$$

Proof:

Re-write the direction in eq. (4)

$$d^*_{k+1} = -g^*_{k+1} + \beta^* d_k ,$$

Multiplying this direction by g^*_{k+1}, then we have:

$$d^{*T}_{k+1} g^{*}_{k+1} = -g^{*T}_{k+1} g^{*}_{k+1} + \beta^{*} d^{T}_{k} g^{*}_{k+1}$$

and then $d^{*T}_{k+1} g^{*}_{k+1} = -\left\| g^{*}_{k+1} \right\|^{2}$

(which is always negative relation) .

This result is true, because we have $d^{T}_{k} g^{*}_{k+1} = 0$,it is easy to prove this result as follows:

We have

$$g^{*}_{k+1} = g_k - \frac{g_k^T g_k}{g_k^T y_k} y_k \qquad (5)$$

So $y_k = g_{k+1} - g_k$. Now multiply eq. (5) by d^{T}_{k} ,then we obtain the following result:

$$d_k^T \overset{*}{g}_{k+1} = d_k^T g_k - \frac{d_k^T g_k}{g_k^T y_k} g_k^T y_k \quad ,$$

thus , we get :

$$d_k^T \overset{*}{g}_{k+1} = 0$$

4. The Outline of the New Algorithm:

Step (1) : $\overset{*}{d}_0 \; - = g_0 = - \overset{*}{g}_0$

Step (2): For k=1,2… compute

$x_{k+1} = x_k + \square_k d_k$, for compute $\square$. where λ is the step size.

Step (3): Check for convergence $\|g_{k+1}\| > \square$, then stop. Otherwise go to step (4).

Step (4) : Compute g^{*}_{k+1} as defined in eq. (1)

Step (5) : Find the new search direction

$$\overset{*}{d}_{k+1} = -\overset{*}{g}_{k+1} + \beta^{*}{}_{k} d^{*}_{k}$$,where

$$\overset{*}{\beta}_{k} = - \frac{g^{*T}_{k+1} g^{*}{}_{k+1}}{d^{T}_{k} g^{*}_{k}} , \qquad 6$$

$$\overset{*}{\beta}_{k} = \frac{y^{T}_{k} g^{*}{}_{k+1}}{g^{*T}_{k} g^{*}{}_{k}} , \qquad 7$$

$$\overset{*}{\beta}_{k} = - \frac{g^{*T}_{k+1} g^{*}{}_{k+1}}{d^{T}_{k} g^{*}{}_{k}} , \qquad 8$$

$$\overset{*}{\beta}_{k} = \frac{y^{T}_{k} g^{*}{}_{k+1}}{y^{T}_{k} d_{k}} , \qquad 9$$

These definitions of $\overset{*}{\beta}_{k}$ in eq. (6) due to Dixon (1975);in $\overset{*}{\beta}_{k}$ eq.(7), due to Polok and Ribiere (1969) ; $\overset{*}{\beta}_{k}$ in eq. (8) due to Al-Bayati and Al- Assady (1996); $\overset{*}{\beta}_{k}$ in eq. (9) due to Hestenes and Stiefel (1952).

Step (6): Check for restarting criterion if $k = n$ then set $k = 0$ and go to step (1) else go to step 2.

Because of the orthogonality property and the lemma (1) are held, this algorithm is identical to the original (H/S, Hestenes And Stiefel), (P/R; Polok and Ribiere), (DX; Dixon), and (AB; Al-Assady and Al-Bayati) algorithms in quadratic function.

Generally function is reduced P/R and A/B algorithms even it Inexact searches can be used as it well be shown in Corollary (2).

5. Numerical Results:

Several standard test functions were minimized to compare the new algorithm with standard CG algorithms. The same line search was employed in each of the algorithms. The cubic interpolation, and the algorithms were terminated if the norm of the gradient was reduced below 1×10^{-5} .We tabulate for all the algorithms the number of calls of the function evaluation (NOF), and the number of iterations (NOI). Overall totals are also given for NOF and NOI with each algorithm.

Table (1) contains the numerical results for the new algorithm with (DX) formula and the standard CG algorithm with the same formula. In this table we see that the new algorithm is more efficient than the standard CG algorithm, this is obtained from the NOF and the NOI of both algorithms.

Table (2) includes the results of the standard CG (H/S) formula and the new with (H/S) formula, this table indicates that the new algorithm is better than the standard CG (H/S) in (4) out of (12) cases and in (5) cases they are comparable.

Table (3) gives the comparison between the results of the new (A/B) algorithm which is presented in eq. (1) with classical CG (A/B), the results in this table indicate that the new method is better than the classical CG (A/B) algorithm in (6) out of (12) cases.

In table (4) we have compared the new algorithm with (P/R) formula with standard CG (P/R). It is obvious that the new algorithm improves the standard (P/R) algorithm in about (80.2%) NOI and (79.6%) NOF.

In table (5) we represent a numerical example to show that the performance of suggested algorithm is quick and has better performance since it requires less time to execute.

Table (1)

Comparative performance of the two algorithms for a group of test functions, by using **β :(DX).**

Test function	N	Standard CG	New method
		NOI(NOF)	NOI(NOF)
Powell	4	76(197)	40(105)
	40	104(219)	85(189)
	100	114(241)	107(223)
Wood	4	37(85)	22(47)
	20	64(176)	54(127)
Rosen	2	37(88)	32(76)
Cubic	2	17(48)	17(48)
	100	64(138)	40(96)
Dixon	10	27(56)	24(50)
Beale	2	10(26)	10(25)
Reciep	30	8(22)	7(20)
3-Powell	3	15(34)	13(29)
Total		573(1330)	451(1035)

Tools	Standard CG	NEW
NOI	100	78.7
NOF	100	77.8

Table (2)

Comparative performance of the two algorithms for a group of test functions, by using **β :(H/S)**.

Test function	**N**	**Standard CG**	**New method**
		NOI(NOF)	**NOI(NOF)**
Powell	4	50(114)	54(128)
	20	34(78)	31 (75)
	100	105(244)	105(230)
Wood	4	21(46)	23(49)
	40	48(101)	45(93)
Rosen	2	31(73)	31(73)
Cubic	2	18(50)	18(50)
	00	13(35)	13(35)
Dixon	10	22(46)	21(44)
Beale	2	10(26)	10(26)
Reciep	30	9(24)	9(24)
3-Powell	3	18(40)	19(41)
Total		379(877)	379(868)

Tools	Standard CG	NEW
NOI	100	100
NOF	100	98.9

Table (3)

Comparative performance of the two algorithms for a group of test functions, by using **β :(A/β).**

Test function	N	Standard CG	New method
		NOI(NOF)	NOI(NOF)
Powell	4	65(144)	40(105)
	20	68(155)	65(146)
	100	105(250)	107(223)
Wood	4	25(58)	22(47)
	20	44(95)	54(127)
Rosen	2	31(73)	32(76)
Cubic	2	19(52)	17(48)
	100	14(36)	40(96)
Dixon	10	23(48)	24(50)
Beale	2	10(26)	10(25)
Reciep	30	9(24)	7(20)
3-Powell	3	19(41)	13(29)
Total		432(1002)	431(992)

Tools	Standard CG	NEW
NOI	100	99.7
NOF	100	99

Table (4)

Comparative performance of the two algorithms for group of test functions, by using **β :(P/R).**

Test function	N	Standard CG	New method
		NOI(NOF)	NOI(NOF)
Powell	4	77(183)	57(150)
	20	76(173)	41 (87)
	100	105(227)	93(188)
Wood	4	33(75)	26(56)
	20	44(93)	33(70)
Rosen	2	31(73)	31(73)
Cubic	2	19(52)	19(52)
	100	16(38)	12(32)
Dixon	10	22(46)	21(44)
Beale	2	10(26)	10(26)
Reciep	30	9(24)	9(24)
3-Powell	3	18(43)	17 (37)
Total		460(1053)	369(839)

Tools	Standard CG	NEW
NOI	100	80.2
NOF	100	79.6

Table (5)

The time of the execute, (sec) of the powell function at Dimension 100 by using β:H/S .

Standard CG Method	New Method
3.2517	1.9284

6. Appendix:

1- Generalized Powell Function:

$$f(x)=\sum_{i=1}^{n/4}[(x_{4i-3}+10x_{4i-2})^2+5(x_{4i-1}-x_{4i})+(x_{4i-2}-2x_{4i-1})^4+10(x_{4i-3}-x_{4i})^4]$$

$x_0=(3,-1,0,1,\ldots)^T$.

2- Generalized Wood Function:

$$f(x)=\sum_{i=2}^{n/4}100[(x_{4i-2}-x_{4i-3}^2)^2]+(1-x_{4i-3})^2+9(x_{4i}-x_{4i-1}^2)^2+(1-x_{4i-1}^2)^2$$

$$+10.1[(x_{4i-2}-1)^2+(x_{4i}-1)^2]+19.8(x_{4i-2}-1)^2(x_{4i}-1)\,,$$

$x_0=(-3,-1,-3,-1,\ldots)^T$.

3- Generalized Rosenbrock Function:

$$f(x)=\sum_{i=1}^{n/2}\left[100(x_{2i}-x_{2i-1}^2)^2+(1-x_{2i-1})^2\right],\ x_0=(-1.2,1;\ldots)^T.$$

4- Cubic Function:

$$f(x)=\sum_{i=1}^{n/2}[100(x_{2i}-x_{2i-1}^3)^2+(1-x_{2i-1})^2]\,,x_0=(-1.2,1;\ldots)^T.$$

5- Generalized Dixon Function:

$$f(x)=\sum_{i=1}^{n}[(1-x_1)^2+(1-x_n)^2+\sum_{i=1}^{n-1}(x_i^2-x_{i-1})^3]\,,$$

$x_0=(-1;\ldots)^T$.

6- Generalized Beale Function:

$$f(x)=\sum_{i=1}^{n/2}\begin{Bmatrix}\left[1.5-x_{2i-1}(1-x_{2i})\right]^2+\left[2.25-x_{2i-1}(1-x_{2i}^2)\right]^2\\+\left[2.635-x_{2i-1}(1-x_{2i}^2)\right]^2\end{Bmatrix},$$

$x_0=(1,1;\ldots)^T$.

7-Generalized Reciep Function:

$$f(x)=\sum_{i=1}^{n/3}\left\{(x_{3i-1}-5)^2+x_{9i-1}^2+\frac{x_{3i}^2}{(x_{3i-1}-x_{3i-2})^2}\right\},$$

$x_0=(2,5,1;\ldots)^T$.

8- Generalized 3-Powell Function:

$x_0=(0,1,2;\ldots)^T$.

REFERENCES

1- Al-Bayati, A.Y. and Al-Assady, N.H., (1996), "An Investigation in the Restarting Techniques", Al-Rafidin Journal for Science, Vol.6, pp. 22-27.

2- Dixon, L.C.W., (1975), "Conjugate Gradient Algorithms: Quadratic Termination without Linear Searches", Journal of the Institute of Mathematics and its Applications, Vol. 15.

3- Hestenes, M. and Steifel, S., (1952), "Solving Linear System", Journal of Research of National Burin of Standard, Vol. 49, pp. 409-436.

4- Polak, E. and Ribiere, R., (1969), "Computational Methods in Optimization a Unified Approach", Academic Press, New York.

5- Nazareth, L., (1977), "A Conjugate Direction Algorithm without Line Searches", JOTA. Vol. 23.

6- Sloboda, F., (1982), " An Imperfect Conjugate Gradient Algorithm", Aplikace Mathematically. Vol. 27.