New conjugacy condition with pair-conjugate gradient methods for unconstrained optimization

Abbas Y. Al-Bayati **Huda I. Ahmed**
profabbasalbayati@yahoo.com
College of Computer Sciences and Mathematics
University of Mosul/Iraq

Received on:28/8/2005 **Accepted on:26/12/2005**

ABSTRACT

Conjugate gradient methods are wildly used for unconstrained optimization especially when the dimension is large. In this paper we propose a new kind of nonlinear conjugate gradient methods which on the study of Dai and Liao (2001), the new idea is how to use the pair conjugate gradient method with this study (new cojugacy condition) which consider an inexact line search scheme but reduce to the old one if the line search is exact. Convergence analysis for this new method is provided. Our numerical results show that this new methods is very efficient for the given ten test function compared with other methods.

Keywords: Unconstrained optimization, conjugate gradient methods.

طريقة تدرج مترافق جديدة معتمدة على تقنية التدرج المترافق المزدوج للامثلية غير الخطية

هدى عصام احمد **عباس يونس البياتي**
كلية علوم الحاسوب والرياضيات/جامعة الموصل/العراق

تاريخ استلام البحث : 2005/8/28 **تاريخ قبول البحث : 2005/12/26**

الملخص

طرق التدرج المترافق تستعمل بكثرة في الامثلية اللاخطية وخصوصا للمسائل ذات الأبعاد الكبيرة. في هذا البحث افترضنا طريقة تدرج مترافق جديدة للدوال غير الخطية معتمدة على فكرة Dai and Liao (2001) وهذه الطريقة تستخدم تقنية التدرج المترافق المزدوج مع البحث الخطي غير المضبوط والتي تتحول الى طريقة التدرج المترافق القياسية باستعمال البحث الخطي المضبوط. تم التطرق كذلك الى تحليل تقارب هذه الطريقة ثم استخدام هذه الطريقة عدديا مع استعمال عشرة دوال غيرخطية ومن ثم الحصول على نتائج كفوءة جدا.

الكلمات المفتاحية: الامثلية اللاخطية، طرائق التدرج المترافق.

1.Introduction

We are concerned with the following unconstrained minimization problem:

$$\text{minimize} \quad f(x) \qquad \ldots(1)$$

where $f: R^n \to R$ is smooth and its gradient $g(x) = \nabla f(x)$ is exist. There are several kinds of numerical methods for solving (1), which include the steepest descent method, the Newton method and quasi-Newton methods, for example. Among them the conjugate gradient method is one choice for solving large-scale problems, because it does not need any matrices. Conjugate gradient methods are iterative methods of the form

$$x_k = x_{k-1} + \alpha_k d_{k-1} \qquad \ldots(2)$$

$$d_k = \begin{cases} -g_k & \text{for k} = 1 \\ -g_k + \beta_k d_{k-1} & \text{for k} \geq 2 \end{cases} \qquad \ldots(3)$$

where g_k denotes $\nabla f(x_k)$ and β_k is a scalar.

If $f(x)$ is a strictly convex quadratic function:

$$f(x) = \frac{1}{2} x^T G x + b^T x + c \qquad \ldots(4)$$

where $G \in R^{nXn}$ is asymmetric positive definite matrix, and α_k is given by:

$$\alpha_k = \frac{\|g_k\|^2}{d_k^T A d_k} \qquad \ldots(5)$$

then the method (2)-(3) is called the linear conjugate gradient method, where $\|.\|$ denotes the Euclidean norm. The linear conjugate gradient method was originally proposed by Hestenes and Stiefel (1952) for solving linear system of equations

$$Gx = b \qquad \ldots(6)$$

within the framework of linear conjugate gradient methods, the conjugacy condition is defined by

$$d_i^T G d_j = 0, \qquad \text{for i} \neq \text{j} \qquad \ldots(7)$$

for search directions, and this condition guarantees the finite termination of the linear conjugate gradient methods.

On the other hand, the method (2)-(3) is called nonlinear conjugate gradient method for general unconstrained optimization problem (general nonlinear function). The nonlinear conjugate gradient method was first proposed by Fletcher and Reeves (Fletcher and Reeves, 1964). Within the framework of nonlinear conjugate gradient methods, the conjugacy condition is replaced by

$$d_k^T y_{k-1} = 0 \qquad \ldots(8)$$

Where

$$y_{k-1} = g_k - g_{k-1} \qquad \ldots(9)$$

for search direction, because the relations.

$$d_k^T G d_{k-1} = \frac{1}{\alpha_{k-1}} d_k^T G(x_k - x_{k-1}) = \frac{1}{\alpha_{k-1}} d_k^T (g_k - g_{k-1}) = \frac{1}{\alpha_{k-1}} d_k^T y_{k-1}.$$

Hold for the strictly convex quadratic objective function. Multiplying y_{k-1} in (3) and using (8), we can deduce a formula for the scalar β_k, as:

$$\beta_k = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} \qquad \ldots(10)$$

This is the so-called HS formula, which was given by Hestenes and Stiefel (1952), also there is well-known formulae for β_k are the Fletcher-Reeves (FR), (Fletcher, 1964) and Polak Ribiere (PR), (Polak,1969) and (Polyak, 1969) they are given by

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \qquad \ldots(11)$$

$$\beta_k^{PR} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2} \qquad \ldots(12)$$

To establish the convergence results methods mentioned above, it is usually required that the step α_k should satisfy the following strong Wolfe conditions

$$f(x_k + \alpha_k d_k) - f(x_k) \le \delta \alpha_k g_k^T d_k \qquad \ldots(13)$$

$$\left| g(x_k + \alpha_k d_k)^T d_k \right| \le -\sigma \ g_k^T d_k \qquad \ldots(14)$$

where $0 < \delta < \sigma < 1$. On the other hand, many numerical methods (e.g. the steepest descent method and quasi-Newton methods) for unconstrained optimization are proved to be convergent under the Wolfe conditions:

$$f(x_k + \alpha_k d_k) - f(x_k) \le \delta \alpha_k g_k^T d_k \qquad \ldots(15)$$

$$g(x_k + \alpha_k d_k)^T d_k \ge \sigma \ g_k^T d_k \qquad \ldots(16)$$

Thus it is an important issue to study global convergence of conjugate gradient methods under the Wolfe conditions instead of the strong Wolfe conditions.

2. The Dai and Liao method

As stated in section 1, the conjugacy condition which may be represented by the
form:

$$d_k^T y_{k-1} = 0 \qquad \ldots(17)$$

for nonlinear conjugate gradient methods. The extension of the conjugacy condition was studied by Peery and also Shanno (Peery,1978) and (Shanno, 1978). However, both the conjugacy conditions (7) and (17) depend on the exact line searchs. In practical computation, one normally carries out inexact line search instead of exact line searches. In the case when $g_{k+1}^T d_k \ne 0$, the conjugacy conditions (7) and (17) may have some disadvantages, for this reason the extension of the conjugacy condition studied by Perry (1978), he tried to accelerate the conjugate gradient method by incorporating the secod-order information into it, specifically, he used the quasi-Newton condition

$$H_k y_{k-1} = s_{k-1} \qquad \ldots(18)$$

where H_k is nxn symmetric and positive definite matrix and $s_{k-1} = \alpha_{k-1} d_{k-1}$. For quasi-Newton methods, the search direction d_k can be calculated as:

$$d_k = -H_k g_k \qquad \ldots(19)$$

by (18) and (19), we have that

$$d_k^T y_{k-1} = -(H_k g_k)^T y_{k-1} = -g_k^T (H_k y_{k-1}) = -g_k^T s_{k-1} \qquad \ldots(20)$$

eq(20) is called Perry condition, which implies (17) holds if the line search is exact since, in this case $g_k^T s_{k-1} = 0$. However, practical algorithms normally adopt inexact line searches instead of exact line searches. For this reason Dai and Liao (2001) replaced the conjugacy condition (17) with the condition:

$$\mathrm{d}_k^T y_{k-1} = -t\mathrm{g}_k^T s_{k-1} \qquad \ldots(21)$$

where $t \geq 0$ is a scalar. In the case $t = 0$, (21) reduces to the usual conjugacy condition (17). On the other hand, in the case $t = 1$, (21) becomes Perry's condition (20). To ensure the search direction d_k satisfies condition (21), by substituting (3) in to (21), they had obtained

$$-g_k^T y_{k-1} + \beta_k d_{k-1}^T y_{k-1} = -tg_k^T s_{k-1} \qquad \ldots(22)$$

this gives the Dai and Liao formula

$$\beta_k^{DL} = \frac{g_k^T (y_{k-1} - ts_{k-1})}{d_{k-1}^T y_{k-1}} \qquad \ldots(23)$$

we note that the case $t = 1$ reduces to Perry formula:

$$\beta_k^P = \frac{g_k^T (y_{k-1} - s_{k-1})}{d_{k-1}^T y_{k-1}} \qquad \ldots(24)$$

the equation (23) can be written by:

$$\beta_k^{DL} = \beta_k^{HS} - \mathrm{t}\,\frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}} \qquad \ldots(25)$$

for which we see that formula (23) with $t \in [0, \infty)$ really defines aclass of nonlinear conjugate gradient methods. Similarly, we call the method defined by (2)-(3) with β_k from (23), method (DL), the aim of Dai and Liao is how to fined the best value of t to give the best nonlinear conjugate gradient method. For any $t \geq 0$, denote d_k and $\bar{d}_k$ to be the search directions given by method (23) and the HS method, respectively, namely:

$$d_k = -g_k + \beta_k^{DL} d_{k-1} \qquad \ldots(26)$$

$$\bar{d}_k = -g_k + \beta_k^{HS} d_{k-1} \qquad \ldots(27)$$

Assume that $\mathrm{g}_k^T \bar{\mathrm{d}}_k < 0$. Then from (26), (27), (25) and $d_{k-1}^T y_{k-1} > 0$, we also have $g_k^T d_k < 0$. Thus if the direction generated by the HS method is descent, and if the line search provides the relation $d_{k-1}^T y_{k-1} > 0$, then the direction give by DL method (23) must also be a descent direction. Denote also α_k^* and $\bar{\alpha}_k$ to be one-dimensional minimize of f along the directions d_k and $\bar{d}_k$ respectively. Consider the following Lemma for quadratic functions (Dai and Liao, 2001).

2.1 Lemma

Suppose that f is quadratic function given in (4); then we have that:

$$f(x_k + \alpha_k^* d_k) - f(x_k + \bar{\alpha}_k \bar{d}_k) = \frac{(g_k^T d_{k-1})^2 t^2}{2(d_{k-1}^T G d_{k-1})(d_k^T G d_k)} \left[(\frac{2}{t} - \bar{\alpha}_k) g_k^T \bar{d}_k - \frac{(g_k^T s_{k-1})^2}{s_{k-1}^T y_{k-1}} \right] \quad \ldots(28)$$

The prove of this Lemma is defined in (Dai and Liao, 2001).
from Lemma Dai and Liao obtained the best value of t which defined by:

$$t = \frac{g_k^T \bar{d}_k}{\tau_k} \quad \ldots(29)$$

Where

$$\tau_k = \bar{\alpha}_k g_k^T \bar{d}_k + \frac{(g_k^T s_{k-1})^2}{s_{k-1}^T y_{k-1}} < 0 \quad \ldots(30)$$

3.New nonlinear conjugacy gradient method using pair direction

In this section we find the new value of t by using pair direction U and V, before that we give some definitions.

3.1 Definition

Vectors $p_1, p_2, ..., p_n \in R^n$ are called <u>left conjugate direction vectors</u> (LCD) of a nxn real nonsingular matrix G if

$$\begin{cases} p_i^T G p_j = 0 & \text{for i < j} \\ p_i^T G p_j \neq 0 & \text{for i = j} \end{cases} \quad \ldots(31)$$

that is $P^T G P = L = (\quad)$,
where $P = [p_1, p_2, ..., p_n]$. (Yuan and Golub, 2003).

3.2 Definition

Vectors $p_1, p_2, ..., p_n \in R^n$ are called <u>right conjugate direction vectors</u> (RCD) of a nxn real nonsingular matrix G if

$$\begin{cases} p_i^T G p_j = 0 & \text{for i > j} \\ p_i^T G p_j \neq 0 & \text{for i = j} \end{cases} \quad \ldots(32)$$

that is $P^T G P = U = (\quad)$,

where $P=[p_1, p_2,...,p_n]$. (Yuan and Golub, 2003).

3.3 Definition

Vectors $p_1, p_2,...,p_n \in R^n$ are called <u>conjugate gradient vectors</u> (CG) of nxn real nonsingular matrix G if

$$\begin{cases} p_i^T G p_j = 0 & \text{for i} \neq \text{j} \\ p_i^T G p_j \neq 0 & \text{for i = j} \end{cases} \qquad \ldots(33)$$

that is $P^T GP = D = (0 \backslash 0)$,

where $P=[p_1, p_2,...,p_n]$. (Yuan and Golub, 2003).

3.4 Definition

Vectors $p_1, p_2,...,p_n \in R^n$ are called <u>semi-conjugate vectors</u> (SCD) of G if they are LCD vectors or RCD vectors of G. (Yuan and Golub, 2003).

3.5 Remark

If G is symmetric and nonsingular, then we observe that the left conjugate direction vectors of G are also right direction vectors of G. In this case, we call the vectors conjugate gradient vector of G. In terms of Stewart's definition (Stewart, 1973), U and V are G-conjugate if $V^T GU$ is Lower triangular. Of course Stewart's G-conjugate direction is the Left conjugate direction when U=V=P. (Yuan and Golub, 2003).

3.6 Definition

Let G, U and V be nonsingular nxn matrices. Then (U,V)is an G-conjugate pair if $L = V^T GU$ is Lower triangular (Wyk, 1977).

3.7 New delimitative for finding the value of t for pair conjugate gradient method.

Suppose that f is given in (4). Then we have that :

$$f(x_k+\alpha_k^* v_k) - f(x_k+\bar{\alpha}_k u_k) = [f(x_k+\alpha_k^* v_k) - f(x_{kv})] - [f(x_k+\bar{\alpha}_k u_k) - f(x_{ku})] + f(x_{kv}) - f(x_{ku}) \qquad \ldots(34)$$

by the definitions of α_k^* and α_k^- for the pair direction, it is easy to show that

$$\alpha_k^* = \frac{-g_k^T v_k}{v_k^T G v_k} \text{ and } \bar{\alpha}_k = \frac{-g_k^T v_k}{v_k^T G u_k} \qquad \ldots (35)$$

and by the strong Wolfe condition (13), we have

$[\ f(x_k+\alpha_k d_k) - f(x_k) \le \delta \alpha_k g_k^T d_k\ ]$

then (34) becomes

$$f(x_k+\alpha_k^* v_k)-f(x_k+\alpha_k u_k)\le \delta(\frac{-g_k^T v_k}{v_k^T\,G\,v_k})\,g_k^T v_k-\delta(\frac{-g_k^T v_k}{v_k^T\,G\,u_k})\,g_k^T u_k+f(x_{kv})-f(x_{ku}) \quad \ldots(36)$$

since $0<\delta<1$ then we can take $\delta=\frac{1}{2}$,

$$f(x_k+\alpha_k^* v_k)-f(x_k+\overline{\alpha}_k u_k)=\frac{1}{2}(\frac{-g_k^T v_k}{v_k^T\,G\,v_k})\,g_k^T v_k-\frac{1}{2}(\frac{-g_k^T v_k}{v_k^T\,G\,u_k})\,g_k^T u_k+f(x_{kv})-f(x_{ku}) \quad \ldots(37)$$

$$=\frac{1}{2}\left(\frac{g_k^T v_k}{v_k^T\,G\,u_k}\right)g_k^T u_k-\frac{1}{2}\left(\frac{(g_k^T v_k)^2}{v_k^T\,G\,v_k}\right)+f(x_{kv})-f(x_{ku}) \quad \ldots(38)$$

$$=\frac{\Gamma}{2(v_k^T\,G\,v_k)(v_k^T\,G\,u_k)}+f(x_{kv})-f(x_{ku}) \quad \ldots(39)$$

where

$$\Gamma=(g_k^T v_k)(g_k^T u_k)(v_k^T\,G\,v_k)-(g_k^T v_k)^2(v_k^T\,G\,u_k) \quad \ldots(40)$$

since

$$v_k=-g_k \quad \ldots(41)$$

$$u_k=v_k+\beta_k^{DL}u_{k-1} \quad \ldots(42)$$

Where β_k^{DL} is defined in (25) [$\beta_k^{DL}=\beta_k^{HS}-\mathrm{t}\,\frac{g_k^T s_{k-1}}{d_{k-1}^T y_{k-1}}$]

and define $\mu_k=\frac{-g_k^T s_{k-1}}{u_{k-1}^T y_{k-1}}$ …(43)

then (42) becomes

$$u_k=v_k+[\beta_k^{HS}+t\mu_k]u_{k-1} \quad \ldots(44)$$

$$u_k=v_k+\beta_k^{HS}u_{k-1}+t\,\mu_k u_{k-1} \quad \ldots(45)$$

Now substitute (45) in (40) to get

$$\Gamma=(g_k^T v_k)(v_k^T\,G\,v_k)(g_k^T(v_k+\beta_k^{HS}u_{k-1}+t\,\mu_k u_{k-1}))-(g_k^T v_k)^2(v_k^T\,G\,(v_k+\beta_k^{HS}u_{k-1}+t\,\mu_k u_{k-1})) \quad \ldots(46)$$

since $(v_k^T\,G\,u_{k-1})=0$ (from the definition of the semi conjugate direction)
then (46) is becomes :

$$\begin{aligned}\Gamma&=(g_k^T v_k)(v_k^T\,G\,v_k)(g_k^T v_k+\beta_k^{HS}g_k^T u_{k-1}+t\,\mu_k g_k^T u_{k-1})-\\&\quad(g_k^T v_k)^2(v_k^T\,G\,v_k+\beta_k^{HS}v_k^T\,G\,u_{k-1}+t\,\mu_k v_k^T\,G\,u_{k-1})\\&=(g_k^T v_k)(v_k^T\,G\,v_k)(g_k^T v_k+\beta_k^{HS}g_k^T u_{k-1}+t\,\mu_k g_k^T u_{k-1})-(g_k^T v_k)^2(v_k^T\,G\,v_k)\\&=(g_k^T v_k)(v_k^T\,G\,v_k)[g_k^T v_k+\beta_k^{HS}g_k^T u_{k-1}+t\,\mu_k g_k^T u_{k-1}-g_k^T v_k]\\&=(g_k^T v_k)(v_k^T\,G\,v_k)[\beta_k^{HS}\,g_k^T u_{k-1}+t\,\,\mu_k\,g_k^T u_{k-1}]\end{aligned} \quad \ldots(47)$$

from (39) we have

$$= \frac{(g_k^T v_k)(v_k^T G v_k)[\beta_k^{HS}\ g_k^T u_{k-1} + t\ \mu_k\ g_k^T u_{k-1}]}{2(v_k^T G v_k)(v_k^T G u_k)} + f(x_{kv}) - f(x_{ku})$$

$$= \frac{(g_k^T v_k)[\beta_k^{HS}\ g_k^T u_{k-1} + t\ \mu_k\ g_k^T u_{k-1}]}{2(v_k^T G u_k)} + f(x_{kv}) - f(x_{ku})$$

$$\beta_k^{HS}\ g_k^T u_{k-1} + t\ \mu_k\ g_k^T u_{k-1} = f(x_{ku}) - f(x_{kv})\left(\frac{2(v_k^T G u_k)}{g_k^T v_k}\right)$$

$$(\beta_k^{HS} + t\ \mu_k)(g_k^T u_{k-1}) = f(x_{ku}) - f(x_{kv})\left(\frac{2}{-\bar{\alpha}_k}\right)$$

$$(\beta_k^{HS} + t\ \mu_k) = f(x_{ku}) - f(x_{kv})\left(\frac{2}{-\bar{\alpha}_k (g_k^T u_{k-1})}\right)$$

$$t\ \mu_k = f(x_{ku}) - f(x_{kv})\left(\frac{2}{-\bar{\alpha}_k (g_k^T u_{k-1})}\right) - \beta_k^{HS}$$

$$t = f(x_{ku}) - f(x_{kv})\left(\frac{2(u_{k-1}^T y_{k-1})}{\bar{\alpha}_k (g_k^T u_{k-1})(g_k^T s_{k-1})}\right) + \frac{(g_k^T y_{k-1})}{(g_k^T s_{k-1})} \qquad \ldots(48)$$

where $t > 0$ is a scalar. In practical if we have to take $\bar{t} = \frac{1}{t}$ which give the best result, then the new formula for the pair conjugate gradient method is defined by:

$$\beta_k^{New} = \beta_k^{HS} - \bar{t}\left(\frac{g_k^T s_{k-1}}{u_{k-1}^T y_{k-1}}\right) \qquad \ldots (49)$$

we call this new formula (49) with (2)-(3) by the new pair method.

4.The algorithm of the new pair conjugate gradient method

We list bellow the out lines of the new method

For an initial point x_0:

Step (1): set k=1, $v_{k-1} = -g_{k-1}$.

Step (2): set $x_k = x_{k-1} + \alpha_k v_{k-1}$, where α_k is a scalar chosen in such a way such that $f_k < f_{k-1}$.

Step (3): check for convergence, i.e. if $|f_k| \le \in$, where $\in$ is small positive tolerance, stop; otherwise continue.

Step (4): if k$\ge$2 go to step (5), else go to step (8).

Step (5): compute $x_k = x_{k-1} + \bar{\alpha}_k u_k$, where $\bar{\alpha}_k = -\alpha_k\left(\frac{g_k^T v_k}{u_k^T (g_{k+1}^* - g_k)}\right)$.

Step (6): check for convergence, i.e. if $|g_{k+1}| \le \in$, where $\in$ is small positive tolerance, stop; otherwise continue.
Step (7): compute the value of $\bar{t}$ where $\bar{t}$ becomes

$$\bar{t} = 1/\left[f(x_{ku}) - f(x_{kv}) \left(\frac{2(u_{k-1}^T y_{k-1})}{\overline{\alpha}_k (g_k^T u_{k-1})(g_k^T s_{k-1})} \right) + \frac{(g_k^T y_{k-1})}{(g_k^T s_{k-1})} \right].$$

Step (8): Compute the new search direction $u_k = -g_k + \beta_k^{New} u_{k-1}$, where β_k is computed by the following formula $\beta_k^{New} = \beta_k^{HS} - \bar{t}\,\dfrac{g_k^T s_{k-1}}{u_{k-1}^T y_{k-1}}$.

Step (9): if k=n or if $\|g_k^T g_{k-1}\| > 0.2\|g_k\|^2$ is satisfied go to step (1), else, set k=k+1, and go to step (2).

5. Generalized conjugate directions

We will now formulate the analogous generalized conjugate direction method for the minimization of function f(x) . Suppose that U and V form a conjugate pair. Set $x_0 =$ arbitrary, $g_0 = g(x_0)$,
for I=0,1,---,compute:

$$x_{k+1} = x_k + \alpha_k v_k \qquad \text{…(50.a)}$$

where α_k minimizes $f(x_k + \alpha_k v_k)$ as a function of α, and let

$$g_k = g(x_k),\ \ g^*_{k+1} = g(x^*_{k+1}), \qquad \text{…(50.b)}$$

$$\alpha_k^{**} = -\alpha_k \left[\frac{g_k^T v_k}{u_k^T (g_{k+1}^* - g_k)} \right] \qquad \text{…(50.c)}$$

$$x_{k+1} = x_k + \alpha_k^{**} u_k \qquad \text{…(50.d)}$$

Before we prove that this algorithm will find the minimum of quadratic function in n steps, then we show that if f is quadratic then α_k^{**} in (35) are the same as the $\overline{\alpha}_k = \dfrac{-g_k^T v_k}{v_k^T G u_k}$, in fact

$$\alpha_k^{**} = -\alpha_k \left[\frac{g_k^T v_k}{u_k^T (g_{k+1}^* - g_k)} \right] = \alpha_k \left[\frac{v_k^T (g_{k+1}^* - g_k)}{u_k^T (g_{k+1}^* - g_k)} \right]$$

$$= \alpha_k \left[\frac{v_k^T\, G\,(x_{k+1}^* - x_k)}{u_k^T\, G\,(x_{k+1}^* - x_k)} \right] = \alpha_k \left[\frac{v_k^T\, \frac{1}{\alpha_k} G\,(x_{k+1}^* - x_k)}{u_k^T\, \frac{1}{\alpha_k} G\,(x_{k+1}^* - x_k)} \right]$$

$$= \alpha_k \left[\frac{v_k^T\, G\, v_k}{u_k^T\, G\, u_k} \right] = -\left[\frac{v_k^T g_k}{v_k^T\, G\, v_k} \right]\left[\frac{v_k^T\, G\, v_k}{u_k^T\, G\, v_k} \right] = \overline{\alpha}_k$$

(see, Wyk, 1977).

6. Theorem

If the iteration (50) is applied to the quadratic function where (U,V) form a G-conjugate pair, the minimum is found in at most n iterations, moreover, x_n lies in the subspace generated by x_0 and $v_o, v_1, \ldots, v_{n-1}$.

Proof:

The first result is established by proving that

$$g_{i+1}^T v_j = 0$$

for all i<n and j=0,1,…,i, by induction. For i=0,

$$g_i^T v_0 = (Gx_1 + b)^T v_0 = [G(x_0 + \alpha_i^{**} u_0 + b]^T v$$

$$= g_0^T v_0 + \alpha_i^{**} u_0^T G v_0$$

$$= g_0^T v_0 - \left(\frac{g_0^T v_0}{v_0^T G u_0}\right) u_0^T G v_0 = 0$$

now suppose that

$$g_i^T v_j = 0$$

for some i and j=0,1,…,i-1, then for j=0,1,…,i

$$g_{i+1}^T v_j = g_i^T v_i + \alpha_i^{**} v_i^T G u_i \,.$$

Due to the induction hypothesis and the conjugacy,

$$g_i^T v_j = 0 = v_i^T G u_i$$

for all j<i, and for the case j=i

$$g_i^T v_i - \left(\frac{v_i^T g_i}{v_i^T G u_i}\right) v_i^T G u_i = 0$$, (see, Wyk, 1977).

7. Numerical results

We tested the HS method (10), Perry method (24), DL method (25) and our new pair conjugate gradient method (49) All results are obtained using Pentium 4 workstation and all programs are written in Fortran language. Our line search subroutine computes α_k such that the strong Wolfe condition (13)-(14) hold with $\delta = 0.001$ and $\sigma = 0.9$. The initial value of α_k is always compute by a cubic fitting procedure which was described in details by Bunday (Bunday, 1982) used as a line search procedure. Although our line search cannot always ensure the descent property of d_k for all three methods, uphill search directions seldom occur in our numerical experiments. In the case when an uphill search direction does occur, we restart the algorithm by setting $d_k = -g_k$. For the DL method (25) $t = 0.1$ is selected. (see Dai and Liao, 2001).

We have test ten function with different dimension n=100, 1000 and 10000. The numerical results are given in the form of NOF and NOI where NOF denote the numbers of function evaluations, and NOI denote the numbers of iterations. The stopping condition is $\|g_{k+1}\| \le 1*10^{-5}$.

Comparing the new pair method (49) with HS method, Peery method, DL method we could say that the new method is more efficient than all especially for Powell function, Wood function, Helical function, Powell3 function, Helical function, Edeger function and Resip function from the ten function test in this section as we see from the Tabel (7.1), (7.2), (7.3).

Table (6.1A)

Numerical comparisons of the new CG method with n=100

	HS method		**Perry method**		**DL method**		**New method**	
function	NOF	NOI	NOF	NOI	NOF	NOI	NOF	NOI
Powell	180	60	131	48	143	49	123	40
Wood	103	49	103	49	103	49	71	25
Powell3	43	20	32	15	48	23	35	14
Helical	250	123	246	121	250	123	82	33
Edger	16	6	14	5	16	6	15	6
Recip	31	11	27	10	31	11	16	5
Tolal	623	269	553	248	591	261	342	123

Table (6.1B)

Performance Percentage for the new pair CG algorithm compared with others and for n=100

Tools %	**HS method**	**Perry method**	**DL method**	**New method**
NOF %	100 %	86	95	55
NOI %	100 %	92	97	55

Table (6.2A)

Numerical comparisons of the new CG method with n=1000

	HS method		**Perry method**		**DL method**		**New method**	
Function	NOF	NOI	NOF	NOI	NOF	NOI	NOF	NOI
Powell	219	66	131	48	143	49	140	41
Wood	103	49	103	49	103	49	77	27
Powell3	49	23	35	16	52	25	35	14
Helical	270	133	268	123	272	134	82	33
Edger	18	7	17	6	18	7	15	6
Recip	33	12	33	12	33	12	16	5

Tolal	692	290	587	254	621	276	365	126

Table (6.2B)

Performance Percentage for the new pair CG algorithm compared with others and for n=1000

Tools %	**HS method**	**Perry method**	**DL method**	**New method**
NOF %	100 %	92	90	53
NOI %	100 %	88	95	43

Table (6.3A)

Numerical comparisons of the new CG method with n=10000

	HS method		**Perry method**		**DL method**		**New method**	
Function	NOF	NOI	NOF	NOI	NOF	NOI	NOF	NOI
Powell	253	72	133	49	178	57	186	47
Wood	105	50	105	50	105	50	77	27
Powell3	51	24	37	17	52	25	35	14
Helical	249	145	290	143	294	145	82	33
Edger	18	7	17	6	18	7	15	6
Recip	33	12	33	12	33	12	16	5
Tolal	709	310	615	277	680	296	411	123

Table (6.3B)

Performance Percentage for the new pair CG algorithm compared with others and for n=10000

Tools %	**HS method**	**Perry method**	**DL method**	**New method**
NOF %	100 %	87	96	60
NOI %	100 %	89	95	40

Appendix:

These test functions are famous and form general literature

1- Generalized Powell function:

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

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

2- Generalized Wood function:

$$f(x)=\sum_{i=1}^{n/4}100[(x_{4i-2}-x_{4i-3}^2)^2]+(1-x_{4i-3})^2+90(x_{4i}-x_{4i-1}^2)^2+(1-x_{4i-1})^2$$

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

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

3- Generalized Edeger function:

$$f(x)=\sum_{i=1}^{n}[(x_{2i-1}-2)^4+(x_{2i-1}-2)^2*{x_{2i}}^2+(x_{2i}+1)^2],$$

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

4- Generalized Powell3 function:

$$f(x)=\sum_{i=1}^{n}\left\{3-[\frac{1}{1+(x_i-x_{2i})^2}]-\sin(\frac{\pi\, x_{2i}\, x_{3i}}{2})-\exp[-(\frac{x_i+x_{3i}}{x_{2i}}-2)^2]\right\},$$

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

5- Generalized Helical function:

$$f(x)=100(x_{3i})-10(\frac{1}{2\pi}a\tan(\frac{x_{3i-1}}{x_{3i-2}}))^2+100(\sqrt{x_{3i-2}^2+x_{3i-1}^2}-1)^2+x_{3i}^2$$

$$x_0=(-1,0,0;\dots)^T .$$

6- Generalized Recip function:

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

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

REFERENCES

[1] Dai, Y. H. and Liao, L. Z. (2001), " New conjugate conditions and related nonlinear conjugate gradient methods", Applied Mathematics and Optimization, Vol. 43, pp. 87-101.

[2] Hestenes, M. R. and Stiefel, E. (1952), " Methods of conjugate gradients for solving linear systems", J. Res. Nat. Bur. Stds., Section B, Vol. 49, pp. 409-436.

[3] Flecher, R. and Reeves, C. M. (1964), "Function minimization by conjugate gradient", Computer Journal, Vol. 7, pp.149-154.

[4] Perry, A. (1978), "Amodified conjugate gradient algorithm", Operations Research, Vol. 26, pp. 1073-1078.

[5] Yuan, J. Y. Golub, H. G.,Plemmonso, J. R. and Cecilio, W. A. G. (2003), " Semi-Conjugate Direction Methods for Real Positive Definite Systems", Applied Mathematics and Optimization, Vol. 44, pp. 120-198.

[6] Wyk, V. D. J. (1977), "Generalization of Conjugate Direction Methods in the Optimization of Functions", Dural of optimization of theory and applications, Vol. 21, No. 4, PP. 435-449.

[7] Sun, J. and Zhang, J. (2004), " Global convergence of conjugate gradient methods without line search", Journal of Optimization Theory and application. ", Special communication.

[8] Bunday, B. (1984), "Basic Optimization Methods" Edward Arnold bedfor square, London.

[9] Shanno, F. D. (1978), " Conjugate gradient methods with inexact searches", Mathematics of Operations Research, Vol. 3, pp. 244-256.

[10] Nocedal, J. and Wright, J. S. (1999), " Numerical Optimization", Springer Series in Operations Research, Springer-Verlag, New Yourk.

[11] Zhang, Z. J. ,Deng, N. Y. and Chen, C. (1999), " New quasi-Newton equation and related methods for unconstrained optimization", Journal of Optimization Theory and Applications, Vol. 102, pp. 147-167.

[12] Gillbert, J. C. and Nocedal, J. (1992), “ Global convergence properties of conjugate gradient methods for optimization”, SIAMJournal on Optimization, Vol. 2, pp. 21-42.