Iraqi Journal of Statistical Sciences (25) 2013
The 6th Scientific Conference of the College of Computer Sciences & Mathematics
pp [210-233]

Optimal CD-DY Conjugate Gradient Methods with Sufficient Descent Directions

Dr. Abbas Y. Al-Bayati* Mr. Hawraz N. Al-Khayat**

Abstract :

Conjugate Gradient (CG) methods are widely used for large scale unconstrained optimization problems. Most of CG-methods don't always generate a descent search direction, so the descent condition is usually assumed in the analysis and implementations. In this paper, we have studied several modified CG-methods based on the famous CD (CG-method), and show that our new proposed CG-methods produces sufficient descent and converges globally if the Wolfe conditions are satisfied. Moreover, they produces the original version of the CD (CG-method), if the line searches are exact. The numerical results show that the new methods are more effective and promising by comparing with the standard CD and DY (CG-methods).

Keywords: Conjugate Gradient, Global Convergence, Unconstrained Optimization, Sufficient Descent Direction, Conjugacy Condition.

طرق تدرج مترافق مثلى لـ CD-DY مع اتجاهات كافية الانحدار

الملخص:

تستخدم طرائق التدرج المترافق (CG) بشكل واسع لحل مسائل الأمثلية غير المقيدة ذات القياس الكبير. أغلب طرائق CG- لا تولد دائما اتجاه بحث منحدر، لذلك عادة ما يتم افتراض شرط الانحدار في التحليل والتنفيذ. في هذا البحث، درسنا العديد من طرائق CG- المطورة التي تعتمد على طريقة CD المعروفة (طريقة-CG)، وأثبتنا أن طرائق CG الجديدة المقترحة تنتج اتجاهات منحدرة كفاية وتتقارب شمولياً إذا توفرت الشروط Wolfe.

* Prof / Dept. Mathematics / Computer Sciences and Mathematics College / Mosul University

** Researcher/ Dept. Mathematics / Computer Sciences and Mathematics College / Mosul University

Received Date 8/10/2013 ——— Accept Date 21/11/2013

فضلا عن ذلك، فإنها تنتج طريقة CD القياسية (طريقة-CG)، إذا كان خط البحث مضبوط. تظهر النتائج العددية أن الطرق المقترحة الجديدة تكون أكثر فعالية و كفاءة من خلال مقارنتها مع طريقتي CD و DY القياسية (طرق-CG).

1. Introduction

We are concerned with the following unconstrained minimization problem:

$$\text{minimize } f(x) \tag{1.1}$$

where $f : R^n \to R$ is a continuously differentiable function and its gradient $g_k = \nabla f(x_k)$ is available. There are several kinds of numerical methods for solving (1.1), which include the Steepest Descent (SD) method, the Newton method and Quasi-Newton (QN) methods, for example. Among them, the CG-method is one choice for solving large scale problems, because it does not need any matrices [13, 15]. CG-methods are iterative methods and at the k-th iteration, it's general form is given by:

$$x_{k+1} = x_k + \alpha_k d_k \tag{1.2}$$

where the step-length α_k is positive and the directions d_k are computed by:

$$\begin{aligned} d_0 &= -g_0 \\ d_k &= -g_k + \beta_k d_{k-1}, \quad k \geq 0 \end{aligned} \tag{1.3}$$

where g_k denotes $\nabla f(x_k)$ and $\beta_k \in R$ is a scalar parameter which characterizes the CG-method. If f is a strictly convex quadratic function and the line search is exact, then the iterative method (1.2)-(1.3) is called linear CG-method. Well-known formulas for β_k are the Fletcher-Reeves (FR), Polak-Ribiere-Polyak (PRP) and Hestenes-Stiefel (HS) formulas (see [8]; [15], [16]; and [11] respectively), Conjugate Descent (CD) [9], Liu-Storey (LS) [13] and Dai-Yuan (DY) [6] formulas and they are given by:

$$\beta_k^{FR} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \tag{1.4}$$

$$\beta_k^{PRP} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2} \tag{1.5}$$

$$\beta_k^{HS} = \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} \tag{1.6}$$

$$\beta_k^{CD} = \frac{\|g_k\|^2}{-g_{k-1}^T d_{k-1}} \tag{1.7}$$

$$\beta_k^{DY} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}} \tag{1.8}$$

$$\beta_k^{LS} = \frac{g_k^T y_{k-1}}{-g_{k-1}^T d_{k-1}} \tag{1.9}$$

where $\|.\|$ denotes the Euclidean norm, and $y_{k-1} = g_k - g_{k-1}$. The global convergence properties of the FR, PRP and HS methods without regular restarts have been studied by many researchers, including Al-Baali [1] and Gilbert and Nocedal [10], Zoutendijk [18], Liu et al [12], Dai and Yuan [5], Powell [17] and Dai and Yuan [4]. To establish the convergence results of these methods, it is normally required that the step-length α_k satisfies the following strong Wolfe conditions:

$$f(x_k) - f(x_k + \alpha_k d_k) \geq -\delta \alpha_k g_k^T d_k, \tag{1.10}$$

$$\left| g(x_k + \alpha_k d_k)^T d_k \right| \leq -\sigma g_k^T d_k \tag{1.11}$$

where $0 < \delta < 0.5 < \sigma < 1$. Some convergence analysis even require that the step-size α_k can be computed by an exact line search, namely:

$$f(x_k + \alpha_k d_k) = \min_{\alpha_k > 0} f(x_k + \alpha_k d_k). \tag{1.12}$$

On the other hand, many other numerical methods for unconstrained optimization are proved to be convergent under the standard Wolfe conditions (1.10):

$$g(x_k + \alpha_k d_k)^T d_k > \sigma \; g_k^T d_k. \tag{1.13}$$

For example, see Fletcher[9]. Hence, it is interesting to investigate whether there exists a CG-method that converges under the standard Wolfe

conditions. Recently CG-methods which satisfy the sufficient descent condition independent of line searches are paid attention to, and such researches are classified by, two types. The first one modifies the parameter β_k, and the second one modifies the search direction d_k.

2. A New Type of CD (CG-Method).

In this section we have, first, to investigate how to determine a descent direction of the objective function. Let x_k be the current iterate and d_k will be defined by:

$$d_k = \begin{cases} -g_k, & if \quad k = 0 \\ -g_k + \beta_k^{New} d_{k-1}, & if \quad k \geq 1 \end{cases} \tag{2.1}$$

where

$$\beta_k^{New} = \frac{\|g_k\|^2}{\mu\ d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} \tag{2.2}$$

and $1 \geq \mu \geq 0$. We note that, the new method reduces to the standard CD method if the line search is exact. Furthermore, if $\mu = 0$ then $\beta_k^{New} = \beta_k^{CD}$, and if $\mu = 1$ then $\beta_k^{New} = \beta_k^{DY}$. But generally we refer to use the inexact line search (s.t. Wolfe line search). We first prove that d_k is a sufficiently descent direction by using the new β_k^{New} only for $1 \geq \mu \geq 0$.

2.1 Lemma

Suppose that d_k is given by (2.1)-(2.2). We assume that α_k satisfies strong Wolfe condition (1.10)-(1.11) with $0.5 < \sigma < 1$ and $1 \geq \mu \geq 0$ Then, the following result:

$$g_k^T d_k \leq -c \|g_k\|^2 \tag{2.3}$$

holds for any $k \geq 0$

Proof.

If $k = 0$, then $d_k^T g_k = -\|g_k\|^2$. Then, from (2.1)-(2.2), it is follows that

$$g_k^T d_k = -\|g_k\|^2 + \beta_k^{New} g_k^T d_{k-1}$$

$$g_k^T d_k = -\|g_k\|^2 + \frac{\|g_k\|^2}{\mu\, d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} g_k^T d_{k-1} \qquad (2.4)$$

By using strong Wolfe Powell condition (1.11) in (2.4) yields:

$$g_k^T d_k \le -\|g_k\|^2 + \frac{\|g_k\|^2}{\mu\, d_{k-1}^T g_k + \frac{1}{\sigma} d_{k-1}^T g_k} g_k^T d_{k-1}$$

$$g_k^T d_k \le -\|g_k\|^2 + \frac{\sigma\, \|g_k\|^2}{\sigma\, \mu\, d_{k-1}^T g_k + d_{k-1}^T g_k} g_k^T d_{k-1}$$

$$g_k^T d_k \le -\|g_k\|^2 + \frac{\sigma}{\sigma\, \mu\, +1} \|g_k\|^2$$

$$g_k^T d_k \le -(\, 1 - \frac{\sigma}{\sigma\, \mu\, +1}\,)\, \|g_k\|^2$$

where

$$c = 1 - \frac{\sigma}{\sigma\, \mu + 1} > 0$$

and $$g_k^T d_k \le -c\|g_k\|^2$$

Hence, from **Lemma 2.1**, it is known that d_k is a **sufficient descent direction** of f at x_k .

2.2 Computations of The New Scalar μ

We can compute the value of μ by **three different approaches** so the results yields three different CG-methods:

1. Descent Direction

By **Lemma 2.1,** d_k which is defined in (2.3) is a descent direction. Now, we have:

$$g_k^T d_k = -\|g_k\|^2 + \beta_k^{New} g_k^T d_{k-1}$$

$$g_k^T d_k = -\|g_k\|^2 + \frac{\|g_k\|^2}{\mu\ d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} g_k^T d_{k-1}$$

$$-\|g_k\|^2 + \frac{\|g_k\|^2}{\mu\ d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} g_k^T d_{k-1} \le 0 \qquad (2.5)$$

By dividing (2.5) on $\|g_k\|^2$

$$-1 + \frac{g_k^T d_{k-1}}{\mu\ d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} \le 0$$

$$\frac{g_k^T d_{k-1}}{\mu\ d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} \le 1$$

$$\mu\ d_{k-1}^T g_k \ge g_k^T d_{k-1} + d_{k-1}^T g_{k-1}$$

$$\mu_1 \ge \frac{g_k^T d_{k-1} + d_{k-1}^T g_{k-1}}{d_{k-1}^T g_k} \qquad (2.6a)$$

Or

$$\mu_1 \ge 1 + \frac{d_{k-1}^T g_{k-1}}{d_{k-1}^T g_k} \quad , \ s.t. \quad 0 < \mu_1 < 1 \qquad (2.6b)$$

Putting (2.6) in (2.2) yields:

$$\beta_k^{New1} = \frac{\|g_k\|^2}{(\frac{g_k^T d_{k-1} + d_{k-1}^T g_{k-1}}{d_{k-1}^T g_k}) d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} \qquad (2.7)$$

2.3. Outline of The New[1] CG-Algorithm:

Step 1: Initialization: Take $x_0 \in R^n$ and the parameter $0 < \delta \le \sigma < 1$. Compute

$f(x_0)$ and $g_0 = \nabla f(x_0)$ and set $d_0 = -g_0$ for $k = 0$.

Step 2: Computation of the Line Search: Compute α_k satisfying strong Wolfe conditions (1.10)-(1.11) and then evaluate $x_{k+1} = x_k + \alpha_k d_k$

Step 3: Test for Convergence: If $(\|g_k\|_\infty \le 10^{-5}$ or $|\alpha_k g_k^T d_k| \le 10^{-10} |f_k|)$

is satisfied then the iterations are stopped.

Step 4: If $\mu \geq 1$ then put $\mu = 1$ and set $\beta_k^{New1} = \beta_k^{DY}$, go to **Step 6**.
Else if $\mu \leq 0$ then put $\mu = 0$ and set $\beta_k^{New1} = \beta_k^{CD}$, go to **Step 6**.

Step 5: Computation of the New Scalar Parameter: Compute the new parameter

β_k^{New1} from (2.7).

Step 6: Search Direction: Compute the new search direction d_k as:

$$d_k = -g_k + \beta_k^{New1} d_{k-1}$$

If the restart criterion of Powell, s.t. $\left|g_k^T g_{k-1}\right| \geq 0.2\left\|g_k\right\|^2$ is satisfied, then set $d_k = -g_k$; otherwise, define $d_k = d$.

Step 7: Set *k=k+1* and go to **Step 2**.

2.4. Conjugacy Property

First, we define conjugate directions, "the set of non-zero vectors $\{d_0, d_1, \ldots, d_n\}$ called conjugate about the nonsingular matrix G if the following property is satisfied":

$$d_i^T G d_j = 0 \ , \quad i \neq j \tag{2.8}$$

The second way to compute the value of μ by use conjugate property. Dai and Liao [3] proved that we can write (2.8) for quadratic functions exact line search by:

$$d_k^T y_{k-1} = 0 \tag{2.9}$$

and with an inexact line search by:

$$d_k^T y_{k-1} = -t g_k^T s_{k-1}, \quad t \geq 0 \tag{2.10}$$

In this paper we suggest another value of μ by using (2.9):

$$d_k = -g_k + \frac{\left\|g_k\right\|^2}{\mu\ d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} d_{k-1} \tag{2.11}$$

$$d_k^T y_{k-1} = -g_k^T y_{k-1} + \frac{\left\|g_k\right\|^2}{\mu\ d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} d_{k-1}^T y_{k-1}$$

$$\frac{-g_k^T y_{k-1}(\mu\ d_{k-1}^T g_k - d_{k-1}^T g_{k-1}) + \|g_k\|^2 d_{k-1}^T y_{k-1}}{\mu\ d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} = 0$$

Since

$$\mu\ d_{k-1}^T g_k - d_{k-1}^T g_{k-1} \neq 0$$

we get

$$-g_k^T y_{k-1}(\mu\ d_{k-1}^T g_k - d_{k-1}^T g_{k-1}) + \|g_k\|^2 d_{k-1}^T y_{k-1} = 0$$

$$\mu\ d_{k-1}^T g_k - d_{k-1}^T g_{k-1} = \|g_k\|^2 \frac{d_{k-1}^T y_{k-1}}{g_k^T y_{k-1}}$$

$$\mu\ \frac{d_{k-1}^T g_k}{d_{k-1}^T g_{k-1}} - 1 = \frac{\|g_k\|^2}{\underbrace{d_{k-1}^T g_{k-1}}_{-\beta_k^{CD}}}\ \frac{d_{k-1}^T y_{k-1}}{\underbrace{g_k^T y_{k-1}}_{1/\beta_k^{HS}}}$$

$$\mu\ \frac{d_{k-1}^T g_k}{d_{k-1}^T g_{k-1}} - 1 = -\frac{\beta_k^{CD}}{\beta_k^{HS}}$$

$$\mu\ \frac{d_{k-1}^T g_k}{d_{k-1}^T g_{k-1}} = \frac{\beta_k^{HS} - \beta_k^{CD}}{\beta_k^{HS}}$$

$$\mu_2 = \frac{\beta_k^{HS} - \beta_k^{CD}}{\beta_k^{HS}} \frac{d_{k-1}^T g_{k-1}}{d_{k-1}^T g_k} \tag{2.12}$$

Putting (2.12) in (2.2) yields:

$$\beta_k^{New\,2} = \frac{\|g_k\|^2}{(\frac{\beta_k^{HS} - \beta_k^{CD}}{\beta_k^{HS}} \frac{d_{k-1}^T g_{k-1}}{d_{k-1}^T g_k}) d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} \tag{2.13}$$

2.5. Outline of The New^2 CG-Algorithm:

Step 1: Initialization: Take $x_0 \in R^n$ and the parameter $0 < \delta \leq \sigma < 1$. Compute

$f(x_0)$ and $g_0 = \nabla f(x_0)$ and set $d_0 = -g_0$ for $k = 0$.

Step 2: Computation of the Line Search: Compute α_k satisfying strong Wolfe conditions (1.10)-(1.11) and then evaluate $x_{k+1} = x_k + \alpha_k d_k$

Step 3: Test for Convergence: If $(\|g_k\|_\infty \le 10^{-5}$ or $|\alpha_k g_k^T d_k| \le 10^{-10}|f_k|)$

is satisfied then the iterations are stopped.

Step 4: If $\mu \ge 1$ then put $\mu = 1$ and set $\beta_k^{New2} = \beta_k^{DY}$, go to **Step 6**.
Else if $\mu \le 0$ then put $\mu = 0$ and set $\beta_k^{New2} = \beta_k^{CD}$, go to **Step 6**.

Step 5: Computation of the New Scalar Parameter: Compute the new parameter β_k^{New2} from (2.13).

Step 6: Search Direction: Compute the new search direction d_k as:

$$d_k = -g_k + \beta_k^{New2} d_{k-1}$$

If the restart criterion of Powell, s.t. $|g_k^T g_{k-1}| \ge 0.2\|g_k\|^2$ is satisfied, then set $d_k = -g_k$; otherwise, define $d_k = d$.

Step 7: Set *k=k+1* and go to **Step 2**.

2.6. A Parallel Direction to the Newton direction

The third way to compute another new value of the scalar μ by assuming a parallel direction to the Newton direction :

$$d_k = -G_k^{-1} g_k \ , \ \forall k \ge 0 \qquad (2.14)$$

As we know, when the initial point x_0 is close enough to a local minimum point x^* , then the best direction to be followed in the current point x_k is the Newton direction $-G_k^{-1} g_k$. Therefore, our motivation is to choose the parameter β_k^{New} in (2.2) so that the direction d_k can be the best direction we know, i.e. the Newton direction. Hence, using the Newton direction (2.14) in (2.11):

$$-G_k^{-1} g_k = -g_k + \frac{\|g_k\|^2}{\mu\ d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} d_{k-1} \qquad (2.15)$$

Multiple both sides of the equation (2.15) by $s_{k-1}^T G_k$ have get:

$$-s_{k-1}^T g_k = -s_{k-1}^T G_k\ g_k + \frac{\|g_k\|^2}{\mu\ d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} s_{k-1}^T G_k\ d_{k-1} \qquad (2.16)$$

Observe that the Newton direction is being used here only as a motivation for formula μ. However, in formula (2.16) the main drawback is the presence of the Hessian. One of the first CG-algorithm using the Hessian matrix was given by Daniel [7], where $\beta_k = (g_k^T G_k d_{k-1} / d_{k-1}^T G_k d_{k-1})$. For large-scale problems, choices for the update parameter that do not require the evaluation of the Hessian matrix are often preferred in practice to the methods that require the Hessian. As we know, QN-methods an approximation matrix H_{k-1} to the Hessian G_{k-1} is used and updated so that the new matrix H_k satisfies the secant $H_k s_{k-1} = y_{k-1}$ condition. This leads us to a motivate CG-algorithm, where:

$$-s_{k-1}^T g_k = -y_{k-1}^T g_k + \frac{\|g_k\|^2}{\mu\, d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} d_{k-1}^T y_{k-1}$$

$$-s_{k-1}^T g_k + y_{k-1}^T g_k = \frac{\|g_k\|^2}{\mu\, d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} d_{k-1}^T y_{k-1}$$

By Dividing both sides on $[(g_k^T g_k) d_{k-1}^T y_{k-1}]$ we get :

$$\frac{-s_{k-1}^T g_k + y_{k-1}^T g_k}{g_k^T g_k d_{k-1}^T y_{k-1}} = \frac{1}{\mu\, d_{k-1}^T g_k - d_{k-1}^T g_{k-1}}$$

$$\mu\, d_{k-1}^T g_k - d_{k-1}^T g_{k-1} = \frac{g_k^T g_k\ d_{k-1}^T y_{k-1}}{-s_{k-1}^T g_k + y_{k-1}^T g_k}$$

$$\mu\, d_{k-1}^T g_k = \frac{g_k^T g_k\ d_{k-1}^T y_{k-1}}{-s_{k-1}^T g_k + y_{k-1}^T g_k} + d_{k-1}^T g_{k-1}$$

$$\mu\, d_{k-1}^T g_k = \frac{g_k^T g_k\ d_{k-1}^T y_{k-1} + d_{k-1}^T g_{k-1}(-s_{k-1}^T g_k + y_{k-1}^T g_k)}{-s_{k-1}^T g_k + y_{k-1}^T g_k}$$

$$\mu_3 = \frac{g_k^T g_k\ d_{k-1}^T y_{k-1} + d_{k-1}^T g_{k-1}(-s_{k-1}^T g_k + y_{k-1}^T g_k)}{d_{k-1}^T g_k(-s_{k-1}^T g_k + y_{k-1}^T g_k)} \qquad (2.17)$$

Then putted μ_3 from (2.17) in (2.2) we get:

$$\beta_k^{New3} = \frac{\|g_k\|^2}{\left(\frac{g_k^T g_k d_{k-1}^T y_{k-1} + d_{k-1}^T g_{k-1}(-s_{k-1}^T g_k + y_{k-1}^T g_k)}{d_{k-1}^T g_k(-s_{k-1}^T g_k + y_{k-1}^T g_k)}\right) d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} \qquad (2.18)$$

2.7. Outlines of The New^3 CG-Algorithm:

Step 1: Initialization: Take $x_0 \in R^n$ and the parameter $0 < \delta \le \sigma < 1$. Compute

$f(x_0)$ and $g_0 = \nabla f(x_0)$ and set $d_0 = -g_0$ for $k = 0$.

Step 2: Computation of the Line Search: Compute α_k satisfying strong Wolfe conditions (1.10)-(1.11) and then evaluate $x_{k+1} = x_k + \alpha_k d_k$

Step 3: Test for Convergence: If $(\|g_k\|_\infty \le 10^{-5}$ or $|\alpha_k g_k^T d_k| \le 10^{-10} |f_k|)$

is satisfied then the iterations are stopped.

Step 4: If $\mu \ge 1$ then put $\mu = 1$ and set $\beta_k^{New3} = \beta_k^{DY}$, go to **Step 6**
Else if $\mu \le 0$ then put $\mu = 0$ and set $\beta^{New3} = \beta_k^{CD}$, go to **Step 6**

Step 5: Computation of the New Scalar Parameters: compute the new parameter

β^{New3} from (2.18).

Step 6: Search Direction: Compute the new search direction d_k as:

$$d_k = -g_k + \beta_k^{New3} d_{k-1}$$

If the restart criterion of Powell, s.t. $|g_k^T g_{k-1}| \ge 0.2\|g_k\|^2$ is satisfied, then set $d_k = -g_k$; otherwise, define $d_k = d$.

Step 7: Set *k=k+1* and go to **Step 2**.

3. Convergence Analysis

In this section, we are in a position to study the global convergence of **new proposed CG-method with different versions**. We first state the following mild assumptions, which will be used in the proof of global convergence property.

Assumption (H).

(i) The level set $S = \{x : x \in R^n, f(x) \le f(x_1)\}$ is bounded, where x_1 is the starting point.

(ii) In a neighborhood Ω of S, f is continuously differentiable and its gradient g is Lipschitz continuously, namely, there exists a constant $L \ge 0$ such that:

$$\| g(x) - g(x_k) \| \le L \| x - x_k \|, \forall x, x_k \in \Omega \qquad (3.1)$$

Obviously, from the Assumption (H, i) there exists a positive constant D such that:

$$D = \max\{\|x - x_k\|, \forall x, x_k \in S\} \tag{3.2}$$

where D is the diameter of Ω. From Assumption (H, ii), we also know that there exists a constant $\Gamma \geq 0$, such that:

$$\|g(x)\| \leq \Gamma, \ \forall x \in S \tag{3.3}$$

On some studies of the CG-methods, the sufficient descent or descent condition plays an important role. Unfortunately, this condition is hard to hold.

3.1 Theorem

In the CG-algorithm (1.2), (2.1), (2.6) and (2.7), assume that α_k is determined by the Wolfe line search (1.10)-(1.11). If, $0 < \mu_1 < 1$, then the direction d_k given by (2.1) is a **descent direction.**

Proof

Since, $0 < \mu_1 < 1$, from (2.1) and (2.7) we get:

$$g_k^T d_k = -\|g_k\|^2 + \beta_k^{New1} g_k^T d_{k-1}$$

$$g_k^T d_k = -\|g_k\|^2 + \frac{\|g_k\|^2}{\mu_1 d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} g_k^T d_{k-1}$$

$$g_k^T d_k \leq -\|g_k\|^2 + \frac{\|g_k\|^2}{\left[\frac{g_k^T d_{k-1} + d_{k-1}^T g_{k-1}}{d_{k-1}^T g_k}\right] d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} g_k^T d_{k-1}$$

$$g_k^T d_k \leq -\|g_k\|^2 + \frac{\|g_k\|^2 . g_k^T d_{k-1}}{g_k^T d_{k-1} + d_{k-1}^T g_{k-1} - d_{k-1}^T g_{k-1}}$$

$$g_k^T d_k \leq -\|g_k\|^2 + \frac{\|g_k\|^2 . g_k^T d_{k-1}}{g_k^T d_{k-1}} \Rightarrow g_k^T d_k \leq 0$$

Hence the direction d_k is a descent one.

3.2 Theorem

If $0 < \mu_2, \mu_3 < 1$, then the direction d_k given by (2.1) satisfies the **sufficient descent direction**, for inexact line searches, (i.e. strong Wolfe line search (1.10)-(1.11)),

$$g_k^T d_k \leq -c\|g_k\|^2; c > 0$$

Proof

CaseI: Since, $0<\mu_2<1$, from (2.1) and (2.13) then yield:

$$g_k^T d_k = -\|g_k\|^2 + \beta_k^{New2} g_k^T d_{k-1}$$

$$g_k^T d_k = -\|g_k\|^2 + \frac{\|g_k\|^2}{\mu_2 d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} g_k^T d_{k-1}$$

$$g_k^T d_k = -\|g_k\|^2 + \frac{\|g_k\|^2}{\left[\frac{\beta_k^{HS}-\beta_k^{CD}}{\beta_k^{HS}}\frac{d_{k-1}^T g_{k-1}}{d_{k-1}^T g_k}\right] d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} g_k^T d_{k-1}$$

$$g_k^T d_k = -\|g_k\|^2 + \|g_k\|^2 \frac{g_k^T d_{k-1}}{d_{k-1}^T g_{k-1}} \frac{1}{\left[\frac{\beta_k^{HS}-\beta_k^{CD}}{\beta_k^{HS}} - 1\right]}$$

$$= -\|g_k\|^2 + \|g_k\|^2 \frac{g_k^T d_{k-1}}{d_{k-1}^T g_{k-1}} \left[\frac{\beta_k^{HS}}{\beta_k^{HS}-\beta_k^{CD}-\beta_k^{HS}}\right]$$

$$= -\|g_k\|^2 + \|g_k\|^2 \frac{g_k^T d_{k-1}}{d_{k-1}^T g_{k-1}} \left[\frac{\beta_k^{HS}}{-\beta_k^{CD}}\right]$$

$$= -\|g_k\|^2 + \|g_k\|^2 \frac{g_k^T d_{k-1}}{d_{k-1}^T g_{k-1}} \cdot \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} \cdot \frac{d_{k-1}^T g_{k-1}}{\|g_k\|^2}$$

$$g_k^T d_k = -\|g_k\|^2 + \|g_k\|^2 \frac{g_k^T d_{k-1} . g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}}$$

By the strong Wolfe condition (1.11) such that:

$$\sigma g_{k-1}^T d_{k-1} \le g_k^T d_{k-1} \le -\sigma g_{k-1}^T d_{k-1}$$

$$g_k^T d_k \le -\|g_k\|^2 + \frac{(-\sigma) g_{k-1}^T d_{k-1} . [\|g_k\|^2 - g_k^T g_{k-1}]}{(-\sigma) g_{k-1}^T d_{k-1} - g_{k-1}^T d_{k-1}}$$

And using the Powell restarting criterion then get:

$$g_k^T d_k \le -\|g_k\|^2 + \frac{(-\sigma) g_{k-1}^T d_{k-1} . [\|g_k\|^2 - (0.2)\|g_k\|^2]}{-(\sigma+1) g_{k-1}^T d_{k-1}}$$

$$g_k^T d_k \le -(\frac{1+(0.2)\sigma}{1+\sigma})\|g_k\|^2$$

$g_k^T d_k \le -c_1 \|g_k\|^2$; $c_1 = \frac{1+0.2\sigma}{1+\sigma} > 0$. This completes the proof.

CaseII: Since, $0<\mu_3<1$, from (2.1) and (2.18):

$$g_k^T d_k = -\|g_k\|^2 + \beta_k^{New3} g_k^T d_{k-1}$$

$$g_k^T d_k = -\|g_k\|^2 + \frac{\|g_k\|^2}{\mu_3 d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} g_k^T d_{k-1}$$

Substituting μ_3 from (2.17) we get:

$$g_k^T d_k = -\|g_k\|^2 + \frac{\|g_k\|^2}{\left[\frac{g_k^T g_k \; d_{k-1}^T y_{k-1} + d_{k-1}^T g_{k-1}(-s_{k-1}^T g_k + y_{k-1}^T g_k)}{d_{k-1}^T g_k (-s_{k-1}^T g_k + y_{k-1}^T g_k)}\right] d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} g_k^T d_{k-1}$$

$$g_k^T d_k = -\|g_k\|^2 + \frac{\|g_k\|^2}{\frac{\|g_k\|^2 \; d_{k-1}^T y_{k-1} + d_{k-1}^T g_{k-1}(-s_{k-1}^T g_k + y_{k-1}^T g_k) - d_{k-1}^T g_{k-1}(-s_{k-1}^T g_k + y_{k-1}^T g_k)}{(-s_{k-1}^T g_k + y_{k-1}^T g_k)}} g_k^T d_{k-1}$$

$$g_k^T d_k = -\|g_k\|^2 + \frac{\|g_k\|^2 (-s_{k-1}^T g_k + y_{k-1}^T g_k)}{\|g_k\|^2 \; d_{k-1}^T y_{k-1}} g_k^T d_{k-1}$$

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

Then by the strong Wolfe condition (1.11) and use the Powell restarting criterion to get:

$$g_k^T d_k \le -\|g_k\|^2 + \frac{(-\sigma) g_{k-1}^T d_{k-1} (\alpha_k g_{k-1}^T g_k + g_k^T g_k - g_{k-1}^T g_k)}{(-\sigma \, d_{k-1}^T g_{k-1} - d_{k-1}^T g_{k-1})} g_k^T d_{k-1}$$

$$g_k^T d_k \le -\left[\frac{1 - \alpha_k (0.2)\sigma + (0.2)\sigma}{\sigma + 1}\right] \|g_k\|^2 \text{ ; where } c_2 = \frac{1 - \alpha_k (0.2)\sigma + (0.2)\sigma}{\sigma + 1} > 0$$

Hence the proof is completed.

3.3. Theorem

Under Assumptions (H, i) and (H, ii) , suppose that d_k is given by (2.1) and (2.2) where α_k satisfies strong Wolfe condition (1.10)-(1.11) with $0.5 < \sigma < 1$ then it holds that:

$$\liminf_{k \to \infty} \|g_k\| = 0 \tag{3.4}$$

Proof

Suppose that there exists a positive constant $\varepsilon > 0$ such that,

$$\|g_k\| \ge \varepsilon \tag{3.5}$$

For all k. Then, from (2.1), it follows that:

$$d_k + g_k = \beta_k d_{k-1} \tag{3.6}$$

We have

$$\|d_k\|^2 = \beta_k^2 \|d_{k-1}\|^2 - \|g_k\|^2 - 2 g_k^T d_k \tag{3.7}$$

Then,

$$g_k^T d_k = -\|g_k\|^2 + \beta_k^{New} g_k^T d_{k-1}$$

$$= -\|g_k\|^2 + \frac{\|g_k\|^2}{\mu\ d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} g_k^T d_{k-1}$$

$$= \frac{-(\mu\ d_{k-1}^T g_k) + (d_{k-1}^T g_{k-1}) + (g_k^T d_{k-1})}{\mu\ d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} \|g_k\|^2$$

$$\leq \frac{d_{k-1}^T g_{k-1}}{\mu\ d_{k-1}^T g_k} \|g_k\|^2$$

Since $d_{k-1}^T g_{k-1} < 0$ and $d_k^T g_k < 0$, then

$$\|g_k\|^2 \leq \frac{\mu\ d_{k-1}^T g_k}{d_{k-1}^T g_{k-1}} g_k^T d_k$$

That is,

$$\beta_k^{New} = \frac{\|g_k\|^2}{\mu\ d_{k-1}^T g_k - d_{k-1}^T g_{k-1}} \leq \frac{\|g_k\|^2}{\mu\ d_{k-1}^T g_k} \leq \frac{d_k^T g_k}{d_{k-1}^T g_{k-1}} \tag{3.8}$$

Put (3.8) in (3.7) and we get

$$\frac{\|d_k\|^2}{(g_k^T d_k)^2} \leq \frac{\|d_{k-1}\|^2}{(g_{k-1}^T d_{k-1})^2} - \frac{\|g_k\|^2}{(g_k^T d_k)^2} - 2\frac{1}{g_k^T d_k}$$

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

$$\leq \frac{\|d_{k-1}\|^2}{(g_{k-1}^T d_{k-1})^2} + \frac{1}{\|g_k\|^2} \leq \frac{\|d_{k-1}\|^2}{(g_{k-1}^T d_{k-1})^2} + \frac{1}{\varepsilon^2}$$

Since $d_1 = -g_1$, so that

$$\frac{\|d_k\|^2}{(g_k^T d_k)^2} < \frac{\|d_1\|^2}{(g_1^T d_1)^2} + \frac{k-1}{\varepsilon^2} = \frac{1}{\|g_1\|^2} + \frac{k-1}{\varepsilon^2}$$
$$< \frac{1}{\varepsilon^2} + \frac{k-1}{\varepsilon^2} = \frac{k}{\varepsilon^2}$$

Thus

$$\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} > \sum_{k=1}^{\infty} \frac{\varepsilon^2}{k} = +\infty$$

which is contrary to proof this theorem. Hence , the proof is complete.

4. Numerical Experiments

The main work of this section is to report the performance of the new methods on a set of test problems. the codes were written in Fortran and in double precision arithmetic. All the tests were performed on a PC. Our experiments were performed on a set of **35**-nonlinear unconstrained problems that have second derivatives available. These test problems are contributed in CUTE [2] and their details are given in the Appendix. for each test function we have considered **10** numerical experiments with number of variable **n= 100,200,……,1000**. In order to assess the reliability of our new proposed methods, we have tested them against standard (CD & DY) classical CG-methods and using the same test problems. All these methods terminate when the following stopping criterion is met.

$$(\|g_k\|_\infty \le 10^{-5} \text{ or } |\alpha_k g_k^T d_k| \le 10^{-10} |f_k|) \qquad (4.1)$$

We also force these routines stopped if the iterations exceed 1000 or the number of function evaluations reach 2000 without achieving the minimum. We use $\delta = 1\times10^{-4}$, $\sigma = 0.1$ in the Wolfe line search routine. **Tables (4.1)-(4.3)** compare some numerical result for New^1 – New^3 CG-methods against (CD & DY) CG-methods respectively, these tables indicate for (**n**) as the dimension of the problem; (**NOI**) number of iterations; (**NOFG**) number of function and gradient evaluation; (**CPU**) the total time required to complete the evaluation process for each test problem. In **Table (4.4)-(4.9)** , we have compared the percentage performance of these New CG-methods against CD & DY methods taking over all the tools as **100%** . In order to

summarize our numerical results , we have concerned only on the total of different dimensions **n= 100, 200,……,1000**, for all tools used in these comparisons.

Table 4.1: Comparison between **New[1]**, CD and DY (CG methods) for the total of n different dimensions **n**= 100, 200, ... ,1000 for each test problems.

Prob.	**CD** method			**DY** method			**New[1]** method		
	NOI	**NOFG**	**CPU**	**NOI**	**NOFG**	**CPU**	**NOI**	**NOFG**	**CPU**
1	377	765	0.27	396	772	0.31	117	246	0.06
2	293	726	0.04	293	728	0.03	63	191	0
3	126	385	0.07	126	385	0.04	98	296	0.03
4	630	1263	0.3	545	1109	0.26	149	341	0.06
5	294	612	0.04	293	610	0.05	146	328	0.02
6	237	507	0.22	244	511	0.17	178	397	0.09
7	297	871	0.05	310	915	0.05	135	326	0
8	92	343	0.16	92	343	0.2	68	255	0.08
9	239	581	0.03	243	588	0.05	138	362	0
10	222	522	0.14	226	523	0.17	107	251	0.03
11	304	688	0.12	297	678	0.14	115	275	0.03
12	208	551	0.04	207	549	0.05	118	340	0.02
13	70	344	0.05	70	344	0.03	22	209	0.01
14	525	1049	0.11	517	1060	0.13	155	334	0
15	547	1515	0.14	534	1458	0.14	141	382	0.01
16	156	422	0.06	156	422	0.08	110	288	0.03
17	160	408	0.08	160	408	0.06	128	291	0.03
18	153	413	0.08	153	413	0.07	113	296	0.02
19	329	696	0.08	315	690	0.08	181	397	0.03
20	125	383	0.09	125	383	0.09	100	304	0.03
21	315	689	0.06	325	689	0.13	176	390	0.01
22	303	591	0.09	280	570	0.15	194	369	0.03
23	522	1153	0.21	519	1153	0.27	167	363	0.08
24	166	452	0.02	166	452	0.03	115	326	0.01

25	255	589	0.04	262	595	0.03	147	346	0.02
26	499	1097	0.16	519	1155	0.16	175	385	0.05
27	147	405	0	147	406	0.02	89	260	0
28	131	403	0.1	131	403	0.06	97	300	0.04
29	122	376	0.09	122	376	0.08	90	277	0.05
30	128	471	0.09	128	471	0.08	56	245	0.05
31	222	522	0.14	226	523	0.1	107	251	0.03
32	497	1038	0.06	478	1026	0.03	132	301	0.01
33	10	30	0	10	30	0	10	30	0
34	90	110	0.03	90	110	0.02	80	100	0.02
35	218	514	0.03	218	514	0.03	150	350	0.01
Total	**9009**	**21484**	**3.29**	**8923**	**21362**	**3.39**	**4167**	**10402**	**0.99**

Table 4.2: Comparison between **New^2**, CD and DY (CG methods) for the total of n different dimensions **n**= 100, 200, ... ,1000 for each test problems

Prob.	**CD** method			**DY** method			**New^2** method		
	NOI	**NOFG**	**CPU**	**NOI**	**NOFG**	**CPU**	**NOI**	**NOFG**	**CPU**
1	377	765	0.27	396	772	0.31	125	258	0.06
2	293	726	0.04	293	728	0.03	63	191	0.02
3	126	385	0.07	126	385	0.04	98	296	0.03
4	630	1263	0.3	545	1109	0.26	309	629	0.16
5	294	612	0.04	293	610	0.05	178	383	0.03
6	237	507	0.22	244	511	0.17	166	370	0.12
7	297	871	0.05	310	915	0.05	140	454	0.02
8	92	343	0.16	92	343	0.2	68	255	0.11
9	239	581	0.03	243	588	0.05	136	355	0.01
10	222	522	0.14	226	523	0.17	150	341	0.07
11	304	688	0.12	297	678	0.14	196	455	0.04
12	208	551	0.04	207	549	0.05	117	325	0.02
13	70	344	0.05	70	344	0.03	22	209	0
14	525	1049	0.11	517	1060	0.13	239	496	0.03

15	547	1515	0.14	534	1458	0.14	127	443	0.02
16	156	422	0.06	156	422	0.08	110	288	0.03
17	160	408	0.08	160	408	0.06	128	291	0.03
18	153	413	0.08	153	413	0.07	113	296	0.05
19	329	696	0.08	315	690	0.08	177	393	0.01
20	125	383	0.09	125	383	0.09	100	304	0.03
21	315	689	0.06	325	689	0.13	167	380	0.03
22	303	591	0.09	280	570	0.15	189	367	0.06
23	522	1153	0.21	519	1153	0.27	270	594	0.11
24	166	452	0.02	166	452	0.03	115	326	0.02
25	255	589	0.04	262	595	0.03	148	340	0.03
26	499	1097	0.16	519	1155	0.16	302	661	0.11
27	147	405	0	147	406	0.02	93	269	0
28	131	403	0.1	131	403	0.06	97	300	0.08
29	122	376	0.09	122	376	0.08	90	277	0.06
30	128	471	0.09	128	471	0.08	56	245	0.05
31	222	522	0.14	226	523	0.1	150	341	0.06
32	497	1038	0.06	478	1026	0.03	273	576	0.05
33	10	30	0	10	30	0	10	30	0
34	90	110	0.03	90	110	0.02	80	100	0.01
35	218	514	0.03	218	514	0.03	150	360	0.02
Total	**9009**	**21484**	**3.29**	**8923**	**21362**	**3.39**	**4952**	**12198**	**1.58**

Table 4.3: Comparison between **New^3**, CD and DY (CG methods) for the total of n different dimensions **n**= 100, 200, ... ,1000 for each test problems

Prob.	**CD** method			**DY** method			**New^3** method		
	NOI	**NOFG**	**CPU**	**NOI**	**NOFG**	**CPU**	**NOI**	**NOFG**	**CPU**
1	377	765	0.27	396	772	0.31	125	258	0.04
2	293	726	0.04	293	728	0.03	63	191	0.02
3	126	385	0.07	126	385	0.04	98	296	0.03
4	630	1263	0.3	545	1109	0.26	303	613	0.1

5	294	612	0.04	293	610	0.05	177	380	0.02
6	237	507	0.22	244	511	0.17	166	370	0.11
7	297	871	0.05	310	915	0.05	184	552	0.03
8	92	343	0.16	92	343	0.2	68	255	0.08
9	239	581	0.03	243	588	0.05	135	354	0.01
10	222	522	0.14	226	523	0.17	138	331	0.05
11	304	688	0.12	297	678	0.14	227	501	0.05
12	208	551	0.04	207	549	0.05	108	292	0.02
13	70	344	0.05	70	344	0.03	22	209	0
14	525	1049	0.11	517	1060	0.13	239	490	0.04
15	547	1515	0.14	534	1458	0.14	145	479	0
16	156	422	0.06	156	422	0.08	110	288	0.03
17	160	408	0.08	160	408	0.06	128	291	0.03
18	153	413	0.08	153	413	0.07	113	296	0.04
19	329	696	0.08	315	690	0.08	177	393	0.03
20	125	383	0.09	125	383	0.09	100	304	0.03
21	315	689	0.06	325	689	0.13	168	386	0.02
22	303	591	0.09	280	570	0.15	188	357	0.05
23	522	1153	0.21	519	1153	0.27	361	689	0.08
24	166	452	0.02	166	452	0.03	115	326	0.01
25	255	589	0.04	262	595	0.03	150	360	0.02
26	499	1097	0.16	519	1155	0.16	363	736	0.09
27	147	405	0	147	406	0.02	93	269	0.02
28	131	403	0.1	131	403	0.06	97	300	0.04
29	122	376	0.09	122	376	0.08	90	277	0.05
30	128	471	0.09	128	471	0.08	56	245	0.03
31	222	522	0.14	226	523	0.1	138	331	0.06
32	497	1038	0.06	478	1026	0.03	285	621	0.03
33	10	30	0	10	30	0	10	30	0
34	90	110	0.03	90	110	0.02	80	100	0.02
35	218	514	0.03	218	514	0.03	160	370	0.01
Total	**9009**	**21484**	**3.29**	**8923**	**21362**	**3.39**	**5180**	**12540**	**1.29**

The percentage performance of the three new methods against 100% (**CD, DY**) methods respectively, as follows in **Tables (4.4)-(4.9).**

Table 4.4: Percentage Performance of **Table (4.1)** against CD-Method

Tools	CD Method	New[1] Method
NOI	100 %	46.2 %
NOFG	100 %	48.4 %
CPU	100 %	30.0 %

Clearly, from the above table, we have found that the new proposed method beats classical CD method in about (53.8)% NOI; (51.6)% NOFG and (70.0)%Time.

Table 4.5: Percentage Performance of **Table (4.1)** against DY-Method

Tools	DY Method	New[1] Method
NOI	100 %	46.6 %
NOFG	100 %	48.6 %
CPU	100 %	29.2 %

Clearly, from the above table, we have found that the new proposed method beats DY method in about (53.4)% NOI; (51.4)% NOFG and (70.8)%Time.

Table 4.6: Percentage Performance of **Table (4.2)** against CD-Method

Tools	CD Method	New[2] Method
NOI	100 %	54.9 %
NOFG	100 %	56.7 %
CPU	100 %	48.0 %

Clearly, from the above table, we have found that the new proposed method beats classical CD method in about (45.1)% NOI; (43.3)% NOFG and (52.0)%Time.

Table 4.7: Percentage Performance of **Table (4.2)** against DY-Method

Tools	DY Method	New[2] Method
NOI	100 %	55.4 %

NOFG	100 %	57.0 %
CPU	100 %	46.6 %

Clearly, from the above table, we have found that the new proposed method beats DY method in about (44.6)% NOI; (43.0)% NOFG and (53.4)%Time.

Table 4.8: Percentage Performance of **Table (4.3)** against CD-Method

Tools	**CD Method**	**New^3 Method**
NOI	100 %	57.4 %
NOFG	100 %	58.3 %
CPU	100 %	39.2 %

Clearly, from the above table, we have found that the new proposed method beats CD method in about (42.6) %NOI; (41.7)% NOFG and (60.8)%Time.

Table 4.9: Percentage Performance of **Table (4.3)** against DY-Method

Tools	**DY Method**	**New^3 Method**
NOI	100 %	58.0 %
NOFG	100 %	58.7 %
CPU	100 %	38.0 %

Clearly, from the above table, we have found that the new proposed method beats DY method in about (42.0)% NOI; (41.3)% NOFG and (62.0)%Time.

5. Concluding Remarks.

In this paper, we have proved the global convergence property of a new proposed nonlinear CG-method with new parameters (2.6),(2.12) and (2.17). They are, in general, generates sufficient descent directions independent of line searches. Numerical results show that our three new proposed CG-methods are effective for solving large-scale unconstrained optimization problems.

References

[1] Al-Baali, M., (1985), Descent property and global convergence of the Fletcher-Reeves method with inexact line search, IMA Journal of Numerical Analysis, 5, 121-124.

[2] Bongartz, K. E.; Conn, A.R., , Gould, N.I.M. and Toint, P.L., (1995), CUTE: constrained and unconstrained testing environments, ACM Trans, Math. Software, 21, 123-160.

[3] Dai, Y. H. and Liao, L.Z., (2001), New conjugacy conditions and related nonlinear conjugate gradient methods, Application Mathematical Optimization, 43, 87–101.

[4] Dai, Y. H. and Yuan, Y., (1995), Further studies on the Polak-Ribiere-Polyak method, Research report ICM-95-040, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences.

[5] Dai, Y. H. and Yuan, Y., (1996), Convergence properties of the Fletcher-Reeves method, IMAJ. Numer. Anal., 2 , 155-164.

[6] Dai, Y. H. and Yuan, Y., (1999), A nonlinear conjugate gradient method with a strong global convergence property, SIAM Journal on Optimization, 10 , 177-182.

[7] Daniel, J.W, (1967), The conjugate gradient method for linear and nonlinear operator equations, SIAM Journal Numerical Analysis,4, 10–26.

[8] Fletcher, R. and Reeves, C.M., (1964), Function minimization by conjugate gradients, The Computer Journal. 7 , 149–154.

[9] Fletcher, R., (1987), Practical methods of optimization, Unconstrained Optimization, John Wiley & Sons, New York, NY, USA.

[10] Gilbert, J. C. and Nocedal, J., (1992), Global convergence properties of conjugate gradient methods for optimization, SIAM Journal Optimization, 2, 21–42.

[11] Hestenes, M. R. and Stiefel E., (1952), Methods of conjugate gradients for solving linear systems, Journal of Research of the National Bureau of Standards. 49 , 409–436.

[12] Liu, G. H.; Han, J. Y. and Yin, H. X., (1993), Global convergence of the Fletcher-Reeves algorithm with an inexact line search, Report, Institute of Applied Mathematics, Chinese Academy of Sciences.

[13] Liu, Y. and Storey, C., (1991), Efficient generalized conjugate gradient algorithms, part 1: Theory, Journal of Optimization Theory and Applications, 69, 129-137.

[14] Nocedal, J. and Wright, S. J., (1999), Numerical Optimization (Springer Verlag), New York.

[15] Polak, E. and Ribiere, G., (1969), Note surla convergence des méthodes de directions conjuguées., 3(16), 35–43.

[16] Polyak, B. T., (1969), The conjugate gradient method in extreme problems, USSR Comp. Math. and Math. Phys., 94-112.

[17] Powell, M.J.D., (1977), Restart procedures for the conjugate gradient method, Mathematical Program. 12, 241–254.

[18] Zoutendijk, G., (1970), Nonlinear Programming, Computational Methods in Integer and Nonlinear Programming. North-Holland Amsterdam, 37-86.

Appendix

1)Trigonometric **2)**Penalty **3)**Raydan **4)**Hager **5)**Generalized Tri-diagonal **6)**Extended Three Exp-Terms **7)**Diagonal4 **8)**Diagonal **9)**Extended Himmelblau **10)**Extended PSC1 **11)**Extended BD1 **12)**Extended Quadratic Penalty QP1 **13)**Extended EP1 **14)**Extended Tridiagonal-2 **15)**ARWHEAD (CUTE) **16)**DIXMAANA (CUTE) **17)**DIXMAANB (CUTE) **18)**DIXMAANC (CUTE) **19)**EDENSCH (CUTE) **20)**DIAGONAL-6 **21)**ENGVAL1 (CUTE) **22)**DENSCHNA (CUTE) **23)**DENSCHNC (CUTE) **24)**DENSCHNB (CUTE) **25)**DENSCHNF (CUTE) **26)**Extended Block-Diagonal BD2 **27)**Generalized quarticGQ1 **28)**DIAGONAL-7 **29)**DIAGONAL-8 **30)**Full Hessian **31)**SINCOS **32)**Generalized quartic GQ2 **33)**ARGLINB (CUTE) **34)**HIMMELBG (CUTE) **35)**HIMMELBH (CUTE).