International Journal of Pure and Applied Mathematics
Volume 63 No. 3 2010, 285-296

A NEW HYBRID CONJUGATE GRADIENT METHOD AND ITS GLOBAL CONVERGENCE FOR UNCONSTRAINED OPTIMIZATION

Shaogang Li[1 §], Zhongbo Sun[2]

[1]School of Mathematics and Computational Science
Guilin University of Electronic Technology
Guilin, 541004, P.R. CHINA
e-mail: flybird_pf@guet.edu.cn

[2]Department of Mathematical Education
College of Humanities and Sciences
Northeast Normal University
Changchun, 130117, P.R. CHINA
e-mail: szb21971@yahoo.com.cn

Abstract: In this paper, a new hybrid conjugate gradient method is proposed for solving unconstrained optimization problems. The parameter β_k is computed as a convex combination of β_k^{FR} and β_k^* algorithms, i.e. $\beta_k^N = (1 - \theta_k)\beta_k^{FR} + \theta_k\beta_k^*$. The parameter θ_k is computed in such a way so that the direction corresponding to the conjugate gradient algorithm to be the Newton equation $\nabla^2 f(x^{k+1})s^k = y^k$. It is sufficient descent at every iteration. The theoretical analysis shows that the algorithm is global convergence under some suitable conditions. Numerical results show that this new algorithm is effective in unconstrained optimization problems.

AMS Subject Classification: 90C30
Key Words: hybrid conjugate gradient method, large scale matrix, quasi-Newton matrix, sufficient descent direction

1. Introduction

The conjugate gradient method is useful and powerful approach method for

Received: April 6, 2010

§Correspondence author

solving minimization problems. It has widely application in many fields, such as control science, engineering, management science and operation research [1]. In this paper, we will consider the following optimization problem

$$\min\ f(x),\ x \in R^n, \tag{1.1}$$

where $f(x) : R^n \to R$ is continuously differentiable function.

Conjugate gradient method is very efficient for solving (1.1), especially, when the dimension is large. Generally, this method generates a sequence of iteration

$$x^{k+1} = x^k + \alpha_k d^k, \quad k = 0, 1, \cdots, \tag{1.2}$$

where the step size α_k is obtained by carrying out some line search rules. The direction d^k is defined by

$$d^k = \begin{cases} -g^k & \text{if } k = 0, \\ -g^k + \beta_k d^{k-1} & \text{if } k > 0, \end{cases} \tag{1.3}$$

where β_k is parameter.

In (1.3) β_k is known as the conjugate gradient parameter $s^k = x^{k+1} - x^k$, $g^k = \nabla f(x^k)$ and $y^k = g^{k+1} - g^k$. There are many methods to solve the problem (1.1). On the one hand, such as, Fletcher and Reeves, see [2], of Dai and Yuan, see [3] and the Conjugate Descent (CD), proposed by Fletcher, see [4]:

$$\beta_k^{FR} = \frac{\|g^{k+1}\|^2}{\|g^k\|^2}, \quad \beta_k^{DY} = \frac{\|g^{k+1}\|^2}{y_k^T s^k}, \quad \beta_k^{CD} = -\frac{\|g^{k+1}\|^2}{g^{kT} s^k}$$

have strong convergence properties, but they may have modest practical performance due to jamming phenomenon. On the other hand, the methods of Hestenes and Stiefel, see [5], of Polak-Ribiere-Polyak, see [6], [7], or of Liu and Storey, see [7]:

$$\beta_k^{HS} = \frac{g^{(k+1)T} y^k}{y_k^T s^k}, \quad \beta_k^{PRP} = \frac{g^{(k+1)T} y^k}{\|g^k\|^2}, \quad \beta_k^{LS} = -\frac{g^{(k+1)T} y^k}{g^{kT} s^k}$$

may not always be convergent, but they often have better computational performances. Wei et al, see [8], proposed a conjugate gradient formula β_k^*, where $\beta_k^* = \frac{\|g^{k+1}\|^2 - \frac{\|g^{k+1}\|}{\|g^k\|} g^{(k+1)T} g^k}{\|g^k\|^2}$. They proved the formula not only $\beta_k^* \geq 0$, but also global convergence properties under different line search rules.

An important class of conjugate gradient methods is the hybrid conjugate gradient algorithms. Dai and Yuan, see [3], combined their algorithm with that of Hestenes and Stiefel, see [5], and suggested the following two hybrid methods:

$$\beta_k^{hDY} = \max\{-c\beta_k^{DY},\ \min\{\beta_k^{HS}, \beta_k^{DY}\}\}$$

and

$$\beta_k^{hDYz} = \max\{0,\ \min\{\beta_k^{HS}, \beta_k^{DY}\}\},$$

where $c = \frac{1-\pi}{1+\pi}$. Zhang and Zhou, see [9], proposed two hybrid conjugate gradient methods, which produce sufficient descent search direction at every iteration, namely,

$$d^{k+1} = -(1 + \beta_k^{H1}\frac{g^{(k+1)^T}d^k}{\|g^{k+1}\|^2})g^k + \beta_k^{H1}d^k$$

and

$$d^{k+1} = -(1 + \beta_k^{H2}\frac{g^{(k+1)^T}d^k}{\|g^{k+1}\|^2})g^k + \beta_k^{H2}d^k,$$

where $\beta_k^{H1} = \max\{0,\ \min\{\beta_k^{PRP},\ \beta_k^{FR}\}\}$ and $\beta_k^{H2} = \max\{0,\ \min\{\beta_k^{HS},\ \beta_k^{DY}\}\}$. Recently, N. Andrei [10], [11] proposed lots of hybrid conjugate gradient methods. These algorithms generated sufficient descent directions. The directions satisfied the sufficient descent condition, when the jamming phenomenon was holded.

In contrast to the hybrid methods β_k^{hDY} and β_k^{hDYz} this paper presents another hybrid conjugate gradient where the parameter β_k is computed as a convex combination of β_k^{FR} and β_k^*. In practice this method often performs better than the β_k^* and we use it in order to have a good practical conjugate gradient algorithm. Because of β_k^{FR} has strong convergence properties, we use it to analysis the convergence of the new hybrid conjugate gradient method.

The structure of this paper is as follows. Section 2 introduces our hybrid conjugate gradient algorithm, and proves that it generates direction satisfying the sufficient descent condition under some conditions. A new hybrid conjugate gradient method is proposed in Section 3. Its convergence analysis is shown in Section 4. Some numerical experiments are reported in Section 5.

2. A New Hybrid Conjugate Gradient Algorithm

In this section, we will describe a new hybrid conjugate gradient algorithm. In order to obtain the sufficient descent direction, we will compute θ_k as follows. Our algorithm generates the iterate sequence $\{x^k\}$ computed by means of the recurrence (1.2). The step size $\alpha_k > 0$ is determined according to the Wolfe line search conditions as follows,

$$f(x^k + \alpha_k d^k) - f(x^k) \leq \rho\alpha_k g^{kT} d^k, \tag{2.1}$$

and

$$g^{(k+1)T}d^k \geq \sigma g^{kT}d^k, \tag{2.2}$$

where d^k is a descent direction and $0 < \rho \leq \sigma < 1$. The directions d^k are generated by the rule:

$$d^{k+1} = -g^{k+1} + \beta_k^N s^k, \quad d^0 = -g^0, \tag{2.3}$$

where

$$\begin{aligned}\beta_k^N &= (1-\theta_k)\beta_k^{FR} + \theta_k\beta_k^* \\ &= (1-\theta_k)\frac{\|g^{k+1}\|^2}{\|g^k\|^2} + \theta_k\frac{\|g^{k+1}\|^2 - \frac{\|g^{k+1}\|}{\|g^k\|}g^{(k+1)T}g^k}{\|g^k\|^2}\end{aligned} \tag{2.4}$$

and θ_k is a scalar parameter satisfying $0 \leq \theta_k \leq 1$. Obviously, if $\theta_k = 0$, then $\beta_k^N = \beta_k^{FR}$, and if $\theta_k = 1$, then $\beta_k^N = \beta_k^*$. On the other hand, if $0 < \theta_k < 1$, then β_k^N is a convex combination of β_k^{FR} and β_k^*. From (2.3) and (2.4) it is obvious that

$$d^{k+1} = \begin{cases} -g^{k+1}, & \text{if } k = 1, \\ -g^{k+1} + (1-\theta_k)\dfrac{\|g^{k+1}\|^2}{\|g^k\|^2}s^k \\ \qquad +\theta_k\dfrac{\|g^{k+1}\|^2 - \frac{\|g^{k+1}\|}{\|g^k\|}g^{(k+1)T}g^k}{\|g^k\|^2}s^k, & \text{if } k > 1. \end{cases} \tag{2.5}$$

Our motivation is to choose the parameter θ_k in such a way so that the direction d^{k+1} given by (2.5) to be the Newton direction. Therefore, from the equation

$$-\nabla^2 f(x^{k+1})^{-1}g^{k+1} = d^{k+1}, \tag{2.6}$$

we will get θ_k as follows. Now, we will deduce the parameter θ_k. Multiplying two sides of the equation (2.6) with s^{kT}, then we get

$$\begin{aligned}&- s^{kT}\nabla^2 f(x^{k+1})^{-1}g^{k+1} \\ &= -g^{(k+1)T}s^k + [(1-\theta_k)\frac{\|g^{k+1}\|^2}{\|g^k\|^2} + \theta_k\frac{\|g^{k+1}\|^2 - \frac{\|g^{k+1}\|}{\|g^k\|}g^{(k+1)T}g^k}{\|g^k\|^2}]s^{kT}s^k \\ &= -g^{(k+1)T}s^k + [\frac{\|g^{k+1}\|^2}{\|g^k\|^2} - \theta_k\frac{\|g^{k+1}\|\|g^{(k+1)T}g^k}{\|g^k\|^3}]\|s^k\|^2 \\ &= -g^{(k+1)T}s^k + \frac{\|g^{k+1}\|^2}{\|g^k\|^2}\|s^k\|^2 - \theta_k\frac{\|g^{k+1}\|\|g^{(k+1)T}g^k}{\|g^k\|^3}\|s^k\|^2,\end{aligned}$$

namely,

$$\theta_k = \frac{-\|g^k\|^3[g^{(k+1)T}s^k - s^{kT}\nabla^2 f(x^{k+1})^{-1}g^{k+1}] + \|s^k\|^2\|g^{k+1}\|^2\|g^k\|}{\|g^{k+1}\|\|s^k\|^2 g^{(k+1)T}g^k}. \quad (2.7)$$

However, in this formula the salient point is the presence of the Hessian. 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 in each iteration.

Algorithm 2.1. *Step 0.* Initialization and date. Given constant $\rho \in (0, \frac{1}{2})$, $\sigma \in (0,1)$, $0 < \rho \leq \sigma < 1$. Choose an initial point $x^0 \in R^n$, and compute $f(x^0)$ and g^0. Consider $d^0 = -g^0$ and set $\alpha_0 = \frac{1}{\|g^0\|}$.

Step 1. Test for termination of iterations. If $\|g^k\| \leq \epsilon$, then stop.

Step 2. Line search for α_k. Compute $\alpha_k > 0$ satisfying the Wolfe line search conditions (2.1) and (2.2) and update the variables

$$x^{k+1} = x^k + \alpha_k d^k.$$

Compute $f(x^{k+1})$, g^{k+1} and $s^k = x^{k+1} - x^k$, $y^k = g^{k+1} - g^k$.

Step 3. Parameter θ_k computation. If $\|s^k\|\|g^{k+1}\|g^{(k+1)T}g^k = 0$, then set $\theta_k = 0$, otherwise compute θ_k as in (2.7).

Step 4. Conjugate gradient parameter β_k^N computation. If $0 < \theta_k < 1$, then compute β_k^N as in (2.4). If $\theta_k \geq 1$, then set $\beta_k^N = \beta_k^{FR}$. If $\theta_k \leq 0$, then set $\beta_k^N = \beta_k^*$.

Step 5. Direction d^k computation. Compute $d^{k+1} = -g^{k+1} + \beta_k^N s^k$. If the restart criterion of Powell

$$|g^{(k+1)T}g^k| \geq \tau\|g^{k+1}\|^2 \quad (2.8)$$

is satisfied, where $0 < \tau < 1$, then restart, i.e. set $d^{k+1} = -g^{k+1}$, otherwise define $d^{k+1} = d^{k+1}$. Compute the initial guess $\alpha_k = \frac{\alpha_{k-1}\|s^k\|}{\|d^k\|}$, set $k = k+1$ and continue with Step 1.

It is well known that if f is bounded along the direction d^k then there exists a step size α_k satisfying the Wolfe line search conditions (2.1) and (2.2). In our algorithm, when the Powell restart condition is satisfied, then we restart the algorithm with the negative gradient $-g^{k+1}$. Under reasonable assumptions, condition (2.1) and (2.2) and Powell's criterion are sufficient to prove the global convergence of the algorithm.

Based on Algorithm 2.1, we will discuss its well defined at every iteration.

Lemma 2.1. *If g^k is proposed by above Algorithm 2.1, then we have*

$$g^{(k+1)T}g^k \geq 0. \quad (2.9)$$

Proof. See [12]. □

Lemma 2.2. *In the algorithm, assume that α_k is determined by the Wolfe line search (2.1) and (2.2). If $0 < \theta_k < 1$, then the direction d^{k+1} given by (2.5) is a descent direction, namely,*

$$g^{(k+1)T} d^{k+1} < 0. \tag{2.10}$$

Proof. We prove equation (2.4) by considering Algorithm 2.1. d^{k+1} is defined by Step 5. Now, we consider two cases as follows

Case 1. If $k = 1$ and $d^{k+1} = -g^{k+1}$, then

$$g^{(k+1)T} d^{k+1} = g^{(k+1)T}(-g)^{k+1} = -\|g^{k+1}\|^2 < 0.$$

Lemma 2.2 is true.

Case 2. If $k > 1$ and

$$d^{k+1} = -g^{k+1} + (1-\theta_k)\frac{\|g^{k+1}\|^2}{\|g^k\|^2} s^k + \theta_k \frac{\|g^{k+1}\|^2 - \frac{\|g^{k+1}\|}{\|g^k\|} g^{(k+1)T} g^k}{\|g^k\|^2} s^k,$$

combining Lemma 2.2 and Wolfe line search, then we have

$$\begin{aligned} g^{(k+1)T} d^{k+1} &= -\|g^{k+1}\|^2 + [(1-\theta_k)\beta_k^{FR} + \theta_k \beta_k^*] g^{(k+1)T} s^k \\ &\le -\|g^{k+1}\|^2 + \beta_k^{FR} g^{(k+1)T} s^k + \beta_k^* g^{(k+1)T} s^k \\ &= -\|g^{k+1}\|^2 + \frac{\|g^{k+1}\|^2}{\|g^k\|^2} g^{(k+1)T} s^k + \frac{\|g^{k+1}\|^2 - \frac{\|g^{k+1}\|}{\|g^k\|} g^{(k+1)T} g^k}{\|g^k\|^2} g^{(k+1)T} s^k \\ &= -\|g^{k+1}\|^2 - \frac{\|g^{k+1}\| g^{(k+1)T} g^k g^{(k+1)T} s^k}{\|g^k\|^2} \le -\|g^{k+1}\|^2 < 0. \end{aligned}$$

Therefore, d^k is a descent direction, then Algorithm 2.1 is well defined. □

3. Global Convergence Analysis

Throughout this section, we assume that:

H1. The level set $\Omega = \{x \in R^n \mid f(x) \le f(x^0)\}$ is below bounded.

H2. In some neighborhood N of Ω. f is continuously differentiable. Its gradient is Lipschitz continuous, namely, there exists a constant $L > 0$, such that

$$\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\|, \quad \forall x, y \in N.$$

Since $\{f(x^k)\}$ is decrease, it is clear that the sequence $\{x^k\}$ generated by Algorithm 2.1 is contained in Ω. In addition, we can obtain from the algorithm, there is a constant $\kappa_1,\ \kappa_2 > 0$, such that $\kappa_1 \leq \|\nabla f(x)\| \leq \kappa_2\ ,\ \forall x \in \Omega$.

Lemma 3.1. *Let assumptions H1 and H2 hold and consider any conjugate gradient method (1.2) and (1.3), where d^{k+1} is a descent direction and α_k is obtained by the strong Wolfe line search. If*

$$\sum_{k\geq 1} \frac{1}{\|d^k\|^2} = \infty, \tag{3.1}$$

then

$$\lim_{k\to\infty} \inf \|g^k\| = 0. \tag{3.2}$$

Proof. See [13]. □

For uniformly convex functions which satisfy the above assumptions, we can prove the norm of d^{k+1} given by (2.5) is bounded. Assume that the function f is a uniformly convex function, i.e. there exists a constant $\gamma \geq 0$ such that, for all $x^{k+1},\ x^k \in \Omega$,

$$(\nabla f(x^{k+1}) - \nabla f(x^k))^T (x^{k+1} - x^k) \geq \gamma \|x^{k+1} - x^k\|^2, \tag{3.3}$$

and the step size α_k is obtained by the strong Wolfe line search.

$$f(x^k + \alpha_k d^k) - f(x^k) \leq \rho \alpha_k g^{kT} d^k, \tag{3.4}$$

and

$$|g^{(k+1)T} d^k| \leq -\sigma g^{kT} d^k, \tag{3.5}$$

Using Lemma 3.1 the following result can be proved.

Theorem 3.1. *Suppose that the assumptions H1 and H2 hold. Consider Algorithm 2.1, where $0 < \theta_k < 1$ and α_k is obtained by the strong Wolfe line search. If s^k tends to zero and there exists nonnegative constants λ_1 and λ_2 such that*

$$\|g^k\| \geq \lambda_1 \|s^k\|,\ \|g^{k+1}\| \leq \lambda_2 \|s^k\|, \tag{3.6}$$

and f is a uniformly convex function, then

$$\lim_{k\to\infty} g^k = 0. \tag{3.7}$$

Proof. Now, since $0 < \theta_k < 1$, from uniform convexity and (3.7), we have

$$|\beta_k^N| = |(1-\theta_k)\beta_k^{FR} + \theta_k \beta_k^*| \leq |\beta_k^{FR}| + |\beta_k^*|$$

$$= \frac{\|g^{k+1}\|^2}{\|g^k\|^2} + |\frac{\|g^{k+1}\|^2 - \frac{\|g^{k+1}\|}{\|g^k\|} g^{(k+1)T} g^k}{\|g^k\|^2}| \leq 2\frac{\|g^{k+1}\|^2}{\|g^k\|^2} + \frac{\|g^{k+1}\|^2 \|g^{k+1}\|}{\|g^k\|^2}$$
$$= \frac{(2 + \|g^{k+1}\|)\|g^{k+1}\|^2}{\|g^k\|^2} \leq \frac{2\lambda_2^2}{\lambda_1^2}.$$

Since $\|s^k\| \to 0$, when $k \to \infty$, there exist $\zeta > 0$, such that $\|s^k\| < \zeta$, then we have

$$\|d^{k+1}\| \leq \|g^{k+1}\| + |\beta_k^N| \|s^k\| \leq \lambda_2 + \frac{2\lambda_2^2}{\lambda_1^2}\|s^k\| \leq \lambda_2 + \frac{2\lambda_2^2}{\lambda_1^2}\zeta,$$

which implies that (3.1) is true. Therefore, by Lemma 3.1 we have (3.2), which for uniformly convex function is equivalent to (3.7). The proof is completed. □

Now, we will prove the global convergence properties under the Wolfe line search as follows.

Theorem 3.2. *Let assumptions H1 and H2 hold. Assume that* $0 < \theta_k < 1$ *for every* $k \geq 1$. *Then, for the computational algorithm where* α_k *is determined by the Wolfe line search (2.1) and (2.2), either* $g^k = 0$ *for some* k *or* $\lim_{k\to\infty} \inf \|g^k\| = 0$.

Proof. Under the assumptions of H1 and H2, $0 < \theta_k < 1$, we have

$$\begin{aligned}
\|d^{k+1}\|^2 = d^{(k+1)T} d^{k+1} &= (-g^{k+1} + \beta_k^N s^k)^T (-g^{k+1} + \beta_k^N s^k) \\
&= \|g^{k+1}\|^2 - 2\beta_k^N g^{(k+1)T} s^k + (\beta_k^N)^2 \|s^k\|^2 \\
&= \|g^{k+1}\|^2 - 2[(1-\theta_k)\beta_k^{FR} + \theta_k \beta_k^*] g^{(k+1)T} s^k + (\beta_k^N)^2 \|s^k\|^2 \\
&= \|g^{k+1}\|^2 - 2[\beta_k^{FR} + \theta_k \beta_k^*] g^{(k+1)T} s^k + 2\theta_k \beta_k^{FR} g^{(k+1)T} s^k + (\beta_k^N)^2 \|s^k\|^2 \\
&\leq \|g^{k+1}\|^2 + 2\theta_k \beta_k^{FR} g^{(k+1)T} s^k + (\beta_k^N)^2 \|s^k\|^2 \\
&= \|g^{k+1}\|^2 + 2\theta_k \beta_k^{FR} g^{(k+1)T} s^k + (1-\theta_k)^2 (\beta_k^{FR})^2 \|s^k\|^2 + \theta_k^2 (\beta_k^*)^2 \|s^k\|^2 \\
&\quad + 2(1-\theta_k)\beta_k^{FR} \theta_k \beta_k^* \|s^k\|^2 \\
&\leq \|g^{k+1}\|^2 + 2\theta_k \beta_k^{FR} g^{(k+1)T} s^k + (\beta_k^{FR})^2 \|s^k\|^2 + (\beta_k^*)^2 \|s^k\|^2 + 2(1-\theta_k)\beta_k^{FR} \theta_k \beta_k^* \|s^k\|^2 \\
&\leq [\|g^{k+1}\| + \beta_k^{FR} g^{(k+1)T} s^k]^2 + (\beta_k^*)^2 \|s^k\|^2 + 2\theta_k \beta_k^{FR} \theta_k \beta_k^* \|s^k\|^2 \\
&\leq \|g^{k+1}\| (1 + \frac{\|s^k\|}{\|g^k\|})^2 + \beta_k^* \|s^k\|^2 (1 + 2\beta_k^{FR}) \\
&\leq \kappa_2 (1 + \frac{M}{\kappa_1})^2 + (1 + 2\frac{\kappa_2}{\kappa_1}) M (\frac{\kappa_2}{\kappa_1})^2 \triangleq \Psi,
\end{aligned}$$

where $M = \max\{\|x^{k+1} - x^k\| \mid x^k,\ x^{k+1} \in \Omega\}$ is the diameter of the level set Ω. Now, we can write

$$\|d^{k+1}\| \leq \sqrt{\Psi}. \tag{3.8}$$

Since the level set Ω is bounded and the function f is bounded from below, using Wolfe line search it follows that

$$0 < \sum_{k=1}^{\infty} \frac{g^{kT} d^k}{\|d^k\|^2} < \infty, \tag{3.9}$$

namely, the Zoutendijk condition holds. Therefore, from Lemma 2.2 we have

$$\sum_{k=1}^{\infty} \frac{{\kappa_1}^4}{\|d^k\|^2} \leq \sum_{k=1}^{\infty} \frac{\|g^k\|^4}{\|d^k\|^2} \leq \sum_{k=1}^{\infty} \frac{1}{\zeta^2} \frac{(g^{kT} d^k)^2}{\|d^k\|^2} < \infty \tag{3.10}$$

which contradicts (3.9). Hence, the conclusion is true. □

4. Numerical Experiments

In this section, we compare the performance of new modified conjugate gradient method to other classical conjugate gradient methods. Especially, we will compare this method with Liu-Storey conjugate gradient method in details. The algorithms are coded in *MATLAB 7.0*. The tests are performed on a PC computer with CPU *Pentium 4*, 2.40 GHz. The test results are listed in Table 1 and Table 2, where CPU-time represents the actual time costed in operation; NI represents the total iterations; NI-HCG, NI-FR and NI-WYLCG are represent hybrid conjugate gradient method, FR method and Wei et al method iteration number. - represents very large iteration number, then we will omit it. E represents the error of actual function solutions and approximate solutions; S represents procedure operation costing the actual CPU-time; f^* represents the problems approximately solutions which allowed in error range region; S-HCG, S-WYL and S-FR are represent hybrid conjugate gradient algorithm, FR method and Wei et al conjugate gradient method costing the actual CPU time; m represents variable number; n represents functional number. P represents the problems in [14]. We choose some test problems for our numerical experiments from the literature [14], and numerical results can be as follows: Table 1 and Table 2.

The parameters are $\delta = 0.35, \sigma = 0.9, \rho = 0.5, \varepsilon = 10^{-8}$. Firstly, we show Table 1 that contained the new hybrid conjugate gradient method and classical FR conjugate gradient method numerical results. In order to convenient, we put FR conjugate method's iteration times and CPU-time in the parentheses bellow the new modified method's.

From Table 1, we see that our method is better than the classical FR conjugate gradient method. For example, the new method's iteration times and

P	NI-HCG(NI-FR)	n	m	S-HCG(S-FR)	f^*	E
1	26(334)	2	2	4(31)	7.1256e-007	8.8968e-007
9	134(135)	15	3	34(33)	1.12	8.7402e-009
13	125(885)	4	4	23(94)	7.8111e-007	9.6169e-007
16	64(−)	4	4	14(-)	7.5249e-007	9.7860e-008
20	16(1079)	2	30	6(298)	2.5302e-007	9.7881e-007
23	11(−)	9	10	8(-)	2.7530e-005	8.2691e-009
24	66(−)	9	18	37(-)	2.8551e-007	8.7409e-009
24	11(1200)	2	4	2(135)	4.0589e-010	9.9906e-007
24	152(−)	200	400	36(-)	8.2253e-007	8.2833e-007
25	43(−)	4	4	8(-)	7.4125e-007	6.7948e-007
27	954(−)	4	4	52(76)	85822.21	5.6961e-009
29	73(970)	4	4	15(130)	9.4011e-007	9.6452e-007
30	33(719)	4	4	5(95)	8.2253e-007	8.2833e-007
30	58(−)	200	400	5(95)	8.2253e-007	8.2833e-007
31	45(−)	4	16	11(-)	7.1539e-007	9.7087e-007
31	754(−)	500	1000	98(-)	7.1539e-007	9.7087e-007

Table 1: Hybrid method and FR method numerical results

CPU time are less than the classical method. Obviously, when we test the larger scale problems, the new method is a better one. The error and approximate solutions are always the same at the terminal step. From problems 9 and 13, we know when the variables are less than 5, the new method and classical method can solve it at limited iteration times. When we enlarge the variables and functional numbers, the classical cannot solve it at limited iteration times, but ours method can solve it easily.

We use two classical conjugate gradient methods, which are FR and WYL conjugate gradient methods, to test the same problems in Table 1 and Table 2. Many test functions are not carry out by the classical method in limited iteration. But, our method is compute at less than 200 iterations. Certainly, we should further investigate more useful, powerful, and practical algorithms for solving large scale unconstrained optimization problems.

Acknowledgments

This work is supported by: NNSF (No. 10861005) and NSF of Guang Xi (No. 0728206, No. 0991238).

P	NI-WYLCG	m	n	S-WYLCG	f^*	E
1	30	2	2	4	7.1256e-007	8.8968e-007
9	135	15	3	35	1.12	8.7402e-009
13	692	4	4	84	7.8111e-007	9.6169e-007
16	168	4	4	14	7.5249e-007	9.7860e-008
20	17	2	30	5	2.5302e-007	9.7881e-007
23	–	9	10	-	2.7530e-005	8.2691e-009
24	–	9	18	-	2.8551e-007	8.7409e-009
24	22	2	4	2	4.0589e-010	9.9906e-007
24	189	200	400	78	8.2253e-007	8.2833e-007
25	103	4	4	9	7.4125e-007	6.7948e-007
27	149	4	4	16	85822.21	5.6961e-009
29	202	4	4	25	9.4011e-007	9.6452e-007
30	68	4	4	7	8.2253e-007	8.2833e-007
30	96	200	400	56	8.2253e-007	8.2833e-007
31	39	4	16	6	7.1539e-007	9.7087e-007
31	1456	500	1000	122	7.1539e-007	9.7087e-007

Table 2: The WYLCG method numerical results

References

[1] Jorge Nocedal, Stephen J. Wright, *Numerical Optimization*, Science Publishers, Beijing, China (2006).

[2] R. Fletcher, C. Reeves, Function minimization by conjugate gradients, *Comput. J.*, **7** (1964), 149-154.

[3] Y.H. Dai, Y. Yuan, An efficient hybrid conjugate gradient method for unconstrained optimization, *Ann. Oper. Res.*, **103** (2001), 33-47.

[4] R. Fletcher, *Unconstained Optimization. Practical Methods of Optimization*, Volume 1, Wiley, New York, USA (1987).

[5] M.R. Polak, G. Ribiere, Note sur la convergence des methodes de directions conjuguees, *Rev. Francaise Imformat Recherche Operionelle*, **16** (1969), 35-43.

[6] B.T. Poliak, The conjugate gradient method in extreme problems, *USSR Comput. Math. Math. Phys.*, **9** (1969), 94-112.

[7] Y. Liu, C. Storey, Efficient generalized conjugate gradient algorithms, Part 1, *J. Optim. Theorem. Appl.*, **69** (1991), 129-137.

[8] Z.X. Wei, S.W. Yao, L.Y. Liu, The convergence properties of some new conjugate gradient methods, *Appl. Math. Comput.*, **183** (2006), 1341-1350.

[9] L. Zhang, W.J. Zhou, Two descent hybrid conjugate gradient methods for optimization, *Comput. Appl. Math.*, **256** (2008), 251-264.

[10] N. Andrei, Hybrid conjugate gradient algorithm for unconstained optimization, *J. Optim. Theory. Appl.*, **141** (2009), 249-264.

[11] N. Andrei, Another hybrid conjugate gradient algorithm for unconstained optimization, *Numer. Algor.*, **47** (2008), 143-156.

[12] F.J. Duan, Z.B. Sun, A modified Liu-Storey conjugate gradient method and its global convergence for unconstrained optimization, In: *The 22-nd Chinese Control and Decision Conference* (2010), http://cms.amss.ac.cn/author.php?confid=6.

[13] Y.H. Dai, J.Y. Han, G.H. Liu, D.F. Yin, X. Yuan, Convergence properties of nonlinear conjugate gradient methods, *SIAM J. Optim.*, **21** (1999), 348-358.

[14] J.J. More, B.S. Garbow, K.E. Hillstrom, Testing unconstrained optimization software, *ACM Transactions on Mathematical Software*, **7** (1981), 17-41.