ABSTRACT

Conjugate Gradient (CG) methods, which we have investigated in this study, were widely used in optimization, especially for large scale optimization problems, because it does not need the storage of any matrix. In this paper, we have constructed a new combined CG-memoryless BFGS algorithm. Our new proposed algorithm which is suitable for solving large scale optimization problem has been constructed by interleaving the modified CG-method due to Liu and Li (2012) with the standard memoryless BFGS update. Numerical results, showed that the new algorithm has been proved to be an effective algorithm in solving large scale optimization problems and gave us a very good numerical results and this algorithm always produce descent search directions and were shown to be globally convergent under some assumptions.

Key Words: Unconstrained Optimization, Conjugate Gradients, Memoryless BFGS Updates, Hybrid Search Directions, Descent Directions, Globally Convergent Methods.

2000 AMS Subject Classification: 47H17; 47H05; 47H09.

1. GENERAL INTRODUCTION

Consider the unconstrained optimization problem defined by

$$\text{Min } f(x), \ x \in \mathbb{R}^n$$

(1)

where \( f(x) \) is non-linear, continuous and differentiable whose gradient denoted by \( g(x) \). The unconstrained optimization methods are iterative in character, this means that we can construct a finite or infinite sequence of a points \( x_k \), for \( k=0,1,\ldots \) which convergence to a solution \( x^* \) of the problem (1). The points of the sequence are related by linear recurrence equation 

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

where \( d_k \) is the search direction and \( \alpha_k \) referred to as step-size, therefore the description of any line search method for solving unconstrained optimization problems consists in establishing a method of choosing the search direction \( d_k \) and step size \( \alpha_k \). It should be noted that the choice of the vector \( d_k \) determines the rate of convergence of the process and the choice of step-size \( \alpha_k \) has an important influence on the amount of calculations at each iteration (Gill et al., 1981). In this paper our attention is focused on establishing a method for determining the search direction \( d_k \). For the computing \( \alpha_k \) we consider an efficient strategy studied by Wolfe, see for example, (Gilbert and Nocedal, 1992) and (Wolfe, 1969, 1971), consisting of accepting a positive step length \( \alpha_k \), if the objective function:

$$f(x) \in C^2$$

(2)

and the Hessian matrix

$$G = \nabla^2 f(x)$$

(3)

is available and symmetric, positive definite, then ideal choice for \( d_k \) is the Newton direction (Fletcher, 1993) given by:

$$d_{k+1} = -G_k^{-1} g_{k+1}$$

(4)
Newton's method has superior convergence properties if the starting point is near the solution. However, the method is not guaranteed to converge to the solution if we start a way from it (Edwin and Stanislaw, 2001). Another type of line search descent methods for solving problem (1) are the Quasi-Newton (QN) methods, they avoid costly computations of Hessian matrices and perform well in the practice, several kinds of them have been proposed, but since the 1970's the BFGS method become more and more popular and today it is accepted as the best QN-method which defines the search directions as:

\[ d_{k+1} = -H_{k+1} g_{k+1} \]  
\[ H_{k+1} = \begin{pmatrix} I_{n \times n} - s_k y_k^T/s_k^T y_k \\ I_{n \times n} - t y_k^T/s_k^T y_k \end{pmatrix} \]  
(5)

where \( H_{k+1} \) is symmetric and positive definite defined by:

\[ y_k = g_{k+1} - g_k \quad \text{and} \quad s_k = x_{k+1} - x_k, \text{ often } H_0 \text{ is taken as an identity matrix, for more details on BFGS see } (\text{Kinsella, 2008}). \]

In spite of these desirable properties of BFGS, (Walter, 2004) show that the BFGS method and other methods in the Broyden class with exact line searches (ELS) may fail for non-convex objective functions. The other drawback for QN-methods are dealing with \( n \times n \) matrix. CG-methods are very useful for solving (1) especially when \( n \) is large. In the CG methods the search directions are defined as:

\[ d_0 = -g_0 \quad k = 0 \]
\[ d_{k+1} = -g_{k+1} + \beta_k d_k \quad k \geq 0 \]  
(7)

were \( \beta_k \) is a scalar. The best known formulas for \( \beta_k \), see for example, (Yabe, 2004) and (Zhang, 2009) are called Fletcher-Reeves (FR), Polack-Ribiere (PR), Hestenes-Stiefel (HS), Dai-Liao (DL):

\[ \beta_{FR} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} \]
\[ \beta_{PR} = \frac{g_{k+1}^T y_k}{g_k^T y_k} \]
\[ \beta_{HS} = \frac{g_{k+1}^T y_k}{d_k^T y_k} \]
\[ \beta_{DL} = \frac{g_{k+1}^T (y_k - t s_k)}{d_k^T y_k} \]  
(8)

\( \beta_k \) were \( t \in [0, \infty) \). If the objective function is quadratic and step-size exact i.e.

\[ g_{k+1}^T d_k = 0 \]  
(12)

All these methods are equivalent, yet, they are different performance on non-quadratic functions. The methods mentioned earlier (Newton, QN and CG) are called conjugate direction methods since they are generate conjugate directions i.e.

\[ d_i^T A d_j = 0 \quad \forall \quad i \neq j \]  
(13)

where A is \( n \times n \) symmetric and positive definite matrix. Furthermore, these methods generates descent directions i.e.

\[ g_k^T d_k < 0, \quad \forall \quad k \]  
(14)

the conjugacy condition given in (13) can be replaced to the following equation:

\[ d_k^T y_{k+1} = 0 \]  
(15)

which is called pure conjugacy condition. (Dai and Liao, 2004) show that if \( \alpha_k \) is not exact the condition in (15) is written as

\[ d_k^T y_{k+1} = -t g_{k+1}^T s_k \]  
(16)

where \( t \) is positive scalar. Therefore the conjugacy condition (16) is more suitable for inexact line searches (ILS). Dai and Liao proved, for any symmetric and positive definite matrix \( H_k \) the secant equation can be written as:
we see from conjugacy condition (16) and secant equation (17) a close relationship between them, we use this relation to define a new scaled CG method. Besides of CG-methods the following gradient type methods:

$$d_{k+1} = \begin{cases} 
-g_k & k = 0 \\
-\theta_k g_{k+1} + \alpha_k d_k & k \geq 1
\end{cases}$$

have also been studied extensively by many authors. Here $\theta_k$ and $\beta_k$ are two parameters. If $\theta_k = 1 \forall k$, then (18) becomes the Perry-CG method defined by:

$$\beta_k^{\text{Perry}} = \frac{g_k^T (y_k - s_{k-1})}{d_{k-1}^T y_{k-1}}$$

and for any scalar $\theta_k$ (for example $\theta_k = \frac{s_k^T y_k - s_{k-1}}{s_k^T y_{k-1}}$), (18) becomes:

$$\beta_k = \frac{g_k^T (\theta_k y_k - s_{k-1})}{d_{k-1}^T y_{k-1}}$$

and is called the spectral CG-method (Birgin and Martinez, 2001) or scaled CG-method (Andrei, 2007).

1.1 INTRODUCTION TO QN-METHODS

BFGS QN-method has a reliable and efficient performance in solving optimization problems for the unconstrained minimization of a smooth nonlinear function $f: \mathbb{R}^n \rightarrow \mathbb{R}$. However, the need to store an $n \times n$ approximate Hessian has limited their application to problems with a small to medium number of variables. For large $n$ it is necessary to use methods that do not require the storage of a full $n \times n$ matrix. Sparse QN-updates can be applied if the Hessian has a significant number of zero entries, see for example, (Powell and Toint, 1979) and (Fletcher, 1995). In nonlinearly constrained optimization, other methods must be used. Such methods include CG-methods, limited-memory (LM) and QN methods, and LM reduced-Hessian QN methods (Gill and Michael., 2000).

1.2 VARIABLE METRIC (VM) METHODS

We have seen that in order to obtain a super linearly convergent method. How can we do this without actually evaluating the Hessian matrix at every iteration? The answer was discovered by Dixon, and was subsequently developed and popularized by (Fletcher and Powell, 1963). It consists of starting with any approximation to the Hessian matrix, and at each iteration, update this matrix by incorporating the curvature of the problem measured along the step. If this update is done appropriately, one obtains some remarkably robust and efficient methods, called Variable Metric (VM) methods. They revolutionized nonlinear optimization by providing an alternative to Newton's method, which is too costly for many applications. There are many VM-methods, but since 1970, the BFGS method has been generally considered to be the most effective. The BFGS method is a line search method. At the $k-th$ iteration, a symmetric and positive definite matrix $B_k$ is given, and a search direction is computed by:

$$d_k = -B_k^{-1} g_k$$

The next iterate is Given by:

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

where the step-size $\lambda_k$ satisfies Wolfe's line search conditions:

$$f(x_k + \lambda_k d_k) \leq f(x_k) + \sigma_1 \lambda_k g_k^T d_k$$

and

$$g(x_k + \lambda_k d_k)^T d_k \geq \sigma_2 g_k^T d_k$$

where $0 < \sigma_1 < \sigma_2 < 1$.

It has been found that it is best to implement BFGS with a very loose line search: typical values for parameters in (23), (24) are $\sigma_1 = 10^{-4}$ and $\alpha_2 = 0.9$. The Hessian approximation is updated by:
A global convergence result for the BFGS method can be obtained by careful consideration of these eigen value shifts. This done by (Powell, 1976) who uses the trace and the determinant to measure the effect of the two rank-one corrections on $B_k$. He is able to show that if $f$ is convex, then for any positive definite starting matrix $B_1$ and any starting point $x_1$, the BFGS method gives 

$$
\lim \inf_k g_k = 0.
$$

If in addition the sequence $\{x_k\}$ converges to a solution point at which the Hessian matrix is positive definite, then the rate of convergence is super-linear. This analysis has been extended by (Byrd, et al., 1987) to the restricted Broyden class of QN-methods in which (25) is replaced by:

$$
B_{k+1} = B_k - \frac{B_k s_k y_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{\phi^2}
$$

where:

$$
\phi \in [0,1], \quad v_k = \left[ \frac{y_k}{y_k^T s_k} - \frac{B_k s_k}{s_k^T B_k s_k} \right].
$$

The choice $\phi = 0$ gives rise to the BFGS update, whereas $\phi = 1$ defines the DFP method, the first VM-method proposed by Davidon, Fletcher and Powell. (Byrd et al., 1987) prove the global and super-linear convergence on convex problems, for all methods in the restricted Broyden class, except for DFP. Their approach breaks down when $\phi = 1$, and leaves that case unresolved. Indeed the following question has remained unanswered since 1976, when Powell published his study on the BFGS method (Nocedal, 1991).

### 1.2 LIMITED MEMORY BFGS METHOD FOR CONVEX FUNCTIONS

QN-methods are a class of numerical methods that are similar to Newton's method except that the inverse of Hessian $(G(x_k)^{-1})$ is replaced by a n by n symmetric matrix $H_k$, which satisfies the QN-condition, see (June and Abu Hassan, 2005):

$$
H_k y_{k-1} = s_{k-1},
$$

where

$$
s_{k-1} = x_k - x_{k-1} = \lambda_{k-1} d_{k-1}, \quad y_{k-1} = g_k - g_{k-1}
$$

The step-size $\lambda_{k-1} > 0$. Assuming $H_k$ nonsingular, we define $B_k = H_k^{-1}$. It is easy to see that the QN step:

$$
d_k = -H_k g_k
$$

Is a stationary point of the following problem:

$$
\min_{d \in \mathbb{R}^n} \phi_k(d) = f(x_k) + d^T g_k + \frac{1}{2} d^T B_k d
$$

which is an approximation to problem $\min_{x \in \mathbb{R}^n} f(x)$ near the current iterate $x_k$, since $\phi_k(d) \approx f(x_k + d)$ for small $d$. In fact, the definition of $\phi_k(d)$ in (31) implies that

$$
\phi_k(0) = f(x_k), \nabla \phi_k(0) = g(x_k)
$$

and the QN-condition (28) is equivalent to:

$$
\nabla \phi_k(x_{k-1} - x_k) = g(x_{k-1}).
$$

Thus, $\phi_k(x - x_k)$ is a quadratic interpolation of $f(x)$ at $x_k$ and $x_{k-1}$, satisfying conditions (31)-(32). The matrix $B_k$ (or $H_k$) can be updated so that the QN-condition is satisfied. One well known update formula is the BFGS formula which updates $B_{k-1}$ from $B_k$, $s_k$ and $y_k$ in the following way:
The approximate function $\phi_k(d)$ in (31) is required to satisfy the interpolation condition:

$$\phi_k(x_{k-1} - x_k) = f(x_{k-1})$$

instead of (33). This change was inspired from the fact that for one-dimensional problem, using (35) gives a slightly faster local convergence if we assume $\lambda_k = 1$ for all $k$. Equation (35) can be rewritten as:

$$s_{k-1}^T B_k s_{k-1} = 2\left[f(x_{k-1}) - f(x_k) + s_{k-1}^T g_k\right].$$

In order to satisfy (36), the BFGS formula is modified as follows:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{s_k^T y_k}$$

where

$$t_k = \frac{2}{s_k^T y_k} \left[f(x_k - f(x_{k+1}) + s_{k+1}^T g_{k+1}\right].$$

If $H_{k+1}$ is the inverse of $B_{k+1}$, then

$$H_{k+1} = H_k + \frac{1}{s_k^T y_k} \left[\alpha_k + \frac{y_k^T H_k y_k}{s_k^T y_k}\right] s_k s_k^T - s_k y_k^T H_k - H_k y_k s_k^T.$$

with

$$\alpha_k = \frac{1}{t_k}$$

Assume that $B_k$ is positive definite and that $s_k^T y_k > 0$, $B_{k+1}$ definite by (37) is positive definite if and only if $t_k > 0$. The inequality $t_k > 0$ is trivial if $f$ is strictly convex, and it is also true if the step-length $\lambda_k$ is chosen by an exact line search, which requires $s_k^T g_{k+1} = 0$. For a uniformly convex function, it can be easily shown that there exists a constant $\delta > 0$ such that $t_k \in [\delta, 2]$ for all $k$, and consequently global convergence proof of the BFGS method for convex functions with inexact line searches. However, for a general nonlinear function $f$, inexact line searches do not imply the positively of $t_k$, hence (Yuan, 1991) truncated $t_k$ to the interval $[0.01, 100]$, and showed that the global convergence of the modified BFGS algorithm is preserved for convex functions. If the objective function $f$ is cubic along the line segment between $x_{k-1}$ and $x_k$ then we have the following relation

$$s_{k-1}^T G(x_k) s_{k-1} = 4s_{k-1}^T g_k + 2s_{k-1}^T g_{k+1} - 6\left[f(x_{k-1}) - f(x_k)\right].$$

By considering the Hermit interpolation on the line between $x_{k-1}$ and $x_k$. Hence it is reasonable to require that the new approximate Hessian satisfies condition:

$$s_{k-1}^T B_k s_{k-1} = 4s_{k-1}^T g_k + 2s_{k-1}^T g_{k+1} - 6\left[f(x_{k-1}) - f(x_k)\right]$$

Instead of (18). (Biggs, 1993) gives the update of (39) with the value $t_k$ chosen so that (42) holds. The respected value of $t_k$ is given by

$$t_k = \frac{6}{s_k^T y_k} \left[f(x_k - f(x_{k+1}) + s_k^T g_{k+1}\right] - 2.$$
For one-dimensional problems, it is well known that the convergence rate of secant method is $\frac{1 + \sqrt{5}}{2}$ which is approximately 1.618 and less than 2. The limited memory BFGS method is described by (Nocedal, 1980), where it is called the SQN method. The user specifies the number $m$ of BFGS corrections that are to be kept, and provides a sparse symmetric and positive definite matrix $H_0$, which approximates the inverse Hessian of $f$. During the first $m$ iterations the method is identical to the BFGS method. For $k > m$, $H_k$ is obtained by applying $m$ BFGS updates to $H_0$ using information from the $m$ previous iterations. The method uses the inverse BFGS formula in the form (see Biggs, 1973).

$$H_{k+1} = V_k^T H_k V_k + \rho_k s_k s_k^T,$$

where

$$\rho_k = \frac{1}{y_k^T s_k}, \quad V_k = I - \rho_k y_k s_k^T.$$

### 1.3 LIMITED MEMORY BFGS METHOD FOR NON-CONVEX FUNCTIONS

All the results for the BFGS method discussed so far depend on the assumption that the objective function $f$ is convex. At present, few results are available for the case in which $f$ is a more general nonlinear function. Even though the numerical experience of many years suggests that the BFGS method always converges to a solution point, this has not been proved. Consider the BFGS method with a line search satisfying the Wolfe conditions (23) and (24).

Assume that $f$ is twice continuously differentiable and bounded below. Do the iterates satisfy $\lim \inf 0 = kg$, for any starting point $x_1$ and any positive definite starting matrix $B_1$? It is remarkable that the answer to this question has not yet been found. Nobody has been able to construct an example in which the BFGS method fails, and the most general result available now.

### 1.4 OUTLINE OF THE LIMITED MEMORY BFGS ALGORITHM

**Step 1:** Choose, and initial matrix $H_0 = I$. Set $k = 0$.

**Step 2:** Compute:

$$d_k = -H_k g_k$$

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

**Step 3:** Let $m = \min[k, m-1]$.

Update $H_0$ for $m + 1$ times by using the pairs $\{y_j, s_j\}_{j=k-m}$, i.e. let

$$H_{k+1} = (V_k^T \ldots V_{k-m}^T) H_0 (V_{k-m} \ldots V_k) + \rho_{k-m} (V_k \ldots V_{k-m+1}) s_{k-m} s_{k-m}^T (V_{k-m+1} \ldots V_k) + \rho_{k-m+1} (V_k \ldots V_{k-m+2}) s_{k-m+1} s_{k-m+1}^T (V_{k-m+2} \ldots V_k) + \rho_k s_k s_k^T$$

**Step 4:** If $\|g_{k+1}\| < \varepsilon$ then stop, otherwise, set $k = k+1$ and go to **Step 2**.

### 2. LIU-LI MODIFIED PRCG-METHOD

In (Liu and Li, 2012), a class of new CG-method with variable parameters is proposed to solve unconstrained optimization problems on the base of PRCG-method. Under the strong Wolfe line searches, they proved the global convergence of their new method without the given sufficient descent condition. Many numerical experiments show that their new method was efficient. Recently, Liu and Li in (2012) had modified PRCG-method with the following hybrid technique and as follows:

$$d_k = \begin{cases} -g_0, & \text{if } k = 0, \\ -g_k + \beta_k^{MPR} d_{k-1}, & \text{if } k \geq 1, \end{cases}$$

(46)
where \( \beta_k^{MPR} \) is defined by:

\[
\beta_k^{MPR} = \frac{g_k^T g_k - \rho g_k^T g_{k-1}}{g_{k-1}^T g_{k-1} + u d_{k-1}^T g_{k-1}} \quad \text{if} \quad \|g_k\| \geq \|g_{k-1}\|,
\]

\[
\beta_k^{MPR} = 0 \quad \text{else}
\]

with \( u = \rho = 0.25 \)

### 2.1 A NEW COMBINED CG-MEMORYLESS BFGS METHOD

In this section, we have constructed a new combined CG-memoryless BFGS algorithm. The purpose of this construction is to find a new CG-type methods suitable for solving large scale optimization problems under special conditions.

### 2.2 OUTLINE OF THE NEW PROPOSED COMBINED ALGORITHM

**Step 1:** Choose \( x_0 \) as initial point; \( H_0 = 0 \); let \( \epsilon_0 > 0 \).

**Step 2:** Put \( k = 0 \), repeat.

**Step 3:** Compute

\[
d_k = \begin{cases} -g_0, & \text{if } k = 0, \\ -g_k + \beta_k^{MPR} d_{k-1}, & \text{if } k \geq 1, \end{cases}
\]

\[
\beta_k^{MPR} = \frac{g_k^T g_k - \rho g_k^T g_{k-1}}{g_{k-1}^T g_{k-1} + u d_{k-1}^T g_{k-1}} \quad \text{if} \quad \|g_k\| \geq \|g_{k-1}\|,
\]

\[
\beta_k^{MPR} = 0 \quad \text{else}
\]

and set \( x_{k+1} = x_k + \lambda_k d_k \); where \( \lambda_k \) satisfies Wolfe conditions (23),(24).

**Step 4:** If Powell restarting criterion is satisfied, i.e. \( \|g_k\| \geq 0.2 \|g_{k-1}\| \), then compute the next iteration step by a memoryless BFGS direction.

**Step 5:** Compute \( H_{k+1} \) of the BFGS update in a vector form by considering:

\[
H_{k+1} = H_k + \frac{1}{s_k^T y_k} \left[ \left( 1 + \frac{y_k^T H_k y_k}{s_k^T y_k} \right) s_k s_k^T - s_k y_k^T H_k - H_k y_k s_k^T \right]
\]

And the matrix \( H_{k+1} \) must be computed by a memoryless BFGS update:

\[
H_{k+1} = (V_k^T \cdots V_{k-3}^T) H_0 (V_{k-3} \cdots V_k) + \rho_k (V_{k-2}^T \cdots V_{k-1}^T) s_{k-2} s_{k-2}^T (V_{k-2} \cdots V_k) + \rho_k s_k s_k^T
\]

**Step 6:** If \( \|g_k\| < \epsilon_0 \) then stop, otherwise, put \( k = k+1 \) and Go to Step (3).

### 2.3 CONVERGENCE ANALYSIS

It is clear that the new proposed algorithm is a hybrid or combined algorithm from two well-known conjugate direction methods. The convergence property of the BFGS update had been proved by many authors, see for example (Biggs, ...
1973), while the convergence properties of the modified PRCG method had been proved by (Liu and Li, 2012). So since exactly these search directions are used in our new proposed algorithm, this implies that the new algorithm satisfies the global convergence property.

3. NUMERICAL RESULTS

The main work of this section is to report the performance of the new proposed combined CG-memoryless BFGS method on a set of (35) test problems. The codes are written in Fortran and in double precision arithmetic. All the tests are performed on a PC. Our experiments are performed on the selected set of nonlinear unconstrained problems that have second derivatives available. These test problems are contributed in CUTE (Bongartiz et al.,1995) and their details are given in the Appendix. For each test function we have considered 10 numerical experiments with number of variables \( n = 100, ..., 1000 \). In order to assess the reliability of our new proposed methods, we have tested it against the new modified PRCG method introduced recently by (Liu and Li, 2012) using the same test problems. All these methods terminate when the following stopping criterion is met:

\[
\text{If } \left( \| g_k \|_\infty < \max (10^{-6}, 10^{-10} \| g_0 \|_\infty) \right) \quad (48)
\]

We also force these routines stopped if the iterations exceed 1000 or the number of function evaluations reach 2000 without achieving convergence. We use \( \delta = 10^{-2} \), \( \sigma = 0.1 \) in the Wolfe line search routine. Table (3.1) and Table (3.2) compare some numerical results for the new method against Liu-Li PRCG-method; these tables indicates for (n) as a dimension of the problem; (NOI) number of iterations; (NOFG) number of function and gradient evaluations; (IRS) number of restarts, i.e., number of used BFGS-updates; (LS) number of line searches used to complete the process and (TIME) the total time required to complete the evaluation process for each test problem.

From Table 3.1 it is clear that the new proposed combined CG-memoryless BFGS algorithm is very effective and robust compared with the new modified PRCG algorithm introduced by (Liu and Li, 2012) using \( u = \rho = 0.25 \). Namely, out of (35) cases it is clear from our table that the new method beats Liu-Li method in (32) cases while the other three cases are approximately comparable. This means that there was an improvement of (91.5)% in both NOI and NOFG Tools.

Table (3.2) presents our numerical results for the two algorithms according to different Tools. Here Liu-Li algorithm implemented with \( \rho = 1.0 \) and \( u = 0.2 \). From Table (3.2) we have found that the new proposed algorithm beats Liu-Li algorithm in about (51)% NOI; (82)% IRS ; (82)% NOFG; (33)%LS and (72)%TIME.

However, from these two tables we have concluded that the new proposed algorithm will be recommended for the purpose of the numerical implementations.

### Table (3.1)

Comparison between the **New Combined** and **Liu-Li (2012)** methods for the total of (35) test problems with ten dimensions \( n = 100, 200, ..., 1000 \)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>345 587 0.16</td>
<td>596 972 0.37</td>
</tr>
<tr>
<td>2</td>
<td>486 985 0.05</td>
<td>336 800 0.05</td>
</tr>
<tr>
<td>3</td>
<td>388 856 0.05</td>
<td>325 777 0.05</td>
</tr>
<tr>
<td>4</td>
<td>173 353 0.03</td>
<td>117 242 0.02</td>
</tr>
<tr>
<td>5</td>
<td>267 3578 0.20</td>
<td>100 322 0.00</td>
</tr>
<tr>
<td>6</td>
<td>40 90 0.00</td>
<td>30 90 0.00</td>
</tr>
<tr>
<td>7</td>
<td>116 229 0.01</td>
<td>84 195 0.00</td>
</tr>
<tr>
<td>8</td>
<td>227 3174 0.86</td>
<td>58 126 0.05</td>
</tr>
<tr>
<td>9</td>
<td>521 835 0.06</td>
<td>515 831 0.07</td>
</tr>
<tr>
<td>10</td>
<td>40 80 0.02</td>
<td>30 80 0.02</td>
</tr>
<tr>
<td>11</td>
<td>40 90 0.06</td>
<td>30 90 0.03</td>
</tr>
<tr>
<td>-------</td>
<td>------------------------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>1</td>
<td>347/ 194/ 596/ 213/ 0.13</td>
<td>596/ 190/ 972/ 336/ 0.23</td>
</tr>
<tr>
<td>2</td>
<td>433/ 211/ 907/ 338/ 0.01</td>
<td>336/ 106/ 800/ 311/ 0.04</td>
</tr>
<tr>
<td>3</td>
<td>384/ 201/ 813/ 324/ 0.03</td>
<td>325/ 81/ 777/ 286/ 0.04</td>
</tr>
<tr>
<td>4</td>
<td>169/ 92/ 347/ 158/ 0.02</td>
<td>117/ 23/ 242/ 101/ 0.02</td>
</tr>
<tr>
<td>5</td>
<td>432/ 374/ 9223/ 392/ 0.23</td>
<td>100/ 24/ 322/ 109/ 0.00</td>
</tr>
<tr>
<td>6</td>
<td>40/ 40/ 90/ 40/ 0.00</td>
<td>30/ 0/ 90/ 40/ 0.02</td>
</tr>
<tr>
<td>7</td>
<td>116/ 61/ 229/ 103/ 0.01</td>
<td>84/ 13/ 195/ 91/ 0.02</td>
</tr>
<tr>
<td>8</td>
<td>210/ 153/ 2420/ 150/ 0.92</td>
<td>58/ 0/ 126/ 48/ 0.04</td>
</tr>
<tr>
<td>9</td>
<td>522/ 206/ 818/ 277/ 0.05</td>
<td>515/ 164/ 831/ 287/ 0.06</td>
</tr>
<tr>
<td>10</td>
<td>40/ 20/ 80/ 30/ 0.00</td>
<td>30/ 0/ 80/ 30/ 0.02</td>
</tr>
<tr>
<td>11</td>
<td>40/ 40/ 90/ 40/ 0.01</td>
<td>30/ 0/ 90/ 40/ 0.03</td>
</tr>
<tr>
<td>12</td>
<td>174/ 96/ 290/ 106/ 0.02</td>
<td>70/ 11/ 160/ 70/ 0.02</td>
</tr>
<tr>
<td>13</td>
<td>133/ 115/ 2125/ 133/ 0.39</td>
<td>70/ 10/ 160/ 70/ 0.03</td>
</tr>
<tr>
<td>14</td>
<td>829/ 273/ 1562/ 721/ 0.05</td>
<td>714/ 215/ 1373/ 636/ 0.06</td>
</tr>
</tbody>
</table>

Table (3.2): Comparison between the New Combined and Liu-Li (2012) methods for the total of (35) test problems according to different Tools.
4. CONCLUSIONS

We have presented a combined CG-Memoryless BFGS method which it is assumed to be an accelerations scheme for (Liu and Li, 2012) PRCG-method. The acceleration scheme is simple and proved to be robust in numerical experiments. For general functions the convergence of the method is coming Section 2.3 and the restart procedure. Therefore, if the Powell restart criterion is used, for general functions f bounded from below with bounded second partial derivatives and bounded level set, we have proved that the iterates converge to a point $x^*$. Under certain conditions we have proved that the new method has globally convergent property. For uniformly convex functions the reduction in the function values is significantly improved for a set of (35) test unconstrained optimization problems.

5. APPENDIX

The details of the test functions, used in this paper, can be found in Cute. The numbers (1-35) in our tables indicate to:

(1)-Extended Trigonometric Function.
(2)-Extended Rosenbrock Function
(3)-Extended White & Holst function
(4)-Extended Beale Function U63 (Matrix Rom) Function.
(5)-Extended Penalty Function.
(6)-Raydan 2 Function.
(7)-Generalized Tridiagonal-2 Function.
(8)-Diagonal4 Function.
(9)-Diagonal5 Function.
(10)-Extended Himmelblau Function.
(11)-Extended PSC1 Function.
(12)-Extended Block Diagonal. BD1 Function.
(13)-Extended Cliff Function.
(14)-Quadratic Diagonal Perturbed Function.
(15)-Quadratic Function QF1 Function.
(16)-Extended Quadratic Penalty QP1 Function.
(17)-Extended Quadratic Penalty QP2 Function.
(18)-Extended Tri-diagonal 2 Function.
(19)-DQDRTIC Function.
REFERENCES


© 2012, IJMA. All Rights Reserved


Source of support: Nil, Conflict of interest: None Declared