A descent modified Polak–Ribière–Polyak conjugate gradient method and its global convergence

LI ZHANG†
Institute of Mathematics, Changsha University of Science and Technology, Changsha 410077, China

AND

WEIJUN ZHOU‡ and DONG-HUI LI§
College of Mathematics and Econometrics, Hunan University, Changsha 410082, China

[Received on 4 July 2005; revised on 31 March 2006]

In this paper, we propose a modified Polak–Ribière–Polyak (PRP) conjugate gradient method. An attractive property of the proposed method is that the direction generated by the method is always a descent direction for the objective function. This property is independent of the line search used. Moreover, if exact line search is used, the method reduces to the ordinary PRP method. Under appropriate conditions, we show that the modified PRP method with Armijo-type line search is globally convergent. We also present extensive preliminary numerical experiments to show the efficiency of the proposed method.

Keywords: PRP method; MPRP method; Armijo-type line search; global convergence.

1. Introduction

The conjugate gradient methods are successful methods for solving optimization problems. They are particularly efficient for solving large-scale problems due to their simplicity and low storage (Nocedal, 1996; Polak, 1997).

Consider the following unconstrained optimization problem:

$$\min f(x), \quad x \in \mathbb{R}^n,$$

where $f: \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function whose gradient will be denoted by $g$.

The iterative process of the conjugate gradient method is given by

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, \ldots,$$

where $x_k$ is the current iterate, $\alpha_k$ is called the step length which is determined by some line search and $d_k$ is the search direction defined by

$$d_k = \begin{cases} -g_k & \text{if } k = 0, \\ -g_k + \beta_k d_{k-1} & \text{if } k > 0. \end{cases}$$

†Email: zl606@tom.com
‡Corresponding author. Email: weijunzhou@126.com
§Email: dhli@hnu.cn

© The author 2006. Published by Oxford University Press on behalf of the Institute of Mathematics and its Applications. All rights reserved.
The parameter $\beta_k$ in (1.3) is chosen so that when applied to minimize a strongly convex quadratic function, the directions $d_k$ and $d_{k-1}$ are conjugate with respect to the Hessian of the quadratic function. Well-known conjugate gradient methods include the Fletcher–Reeves (FR) method (Fletcher & Reeves, 1964), the Polak–Ribiére–Polyak (PRP) method (Polak & Ribiere, 1969; Polyak, 1969), the Hestenes–Stiefel (HS) method (Hestenes & Stiefel, 1952), the Dai–Yuan method (Dai & Yuan, 2000) and the conjugate descent method (Fletcher, 1987). We are particularly interested in the PRP method in which the parameter $\beta_k$ is defined by

$$
\beta_{k}^{\text{PRP}} = \frac{g_k^T y_{k-1}}{\|g_k\|^2}.
$$

Here and throughout, $\|\cdot\|$ stands for the Euclidean norm of vectors and $y_{k-1} = g_k - g_{k-1}$.

Much effort has been devoted to the global convergence analysis of the conjugate gradient methods. One approach for ensuring global convergence is to use the steepest descent direction if the angle condition

$$
-g_k^T d_k \geq \varepsilon \|g_k\| \|d_k\|
$$

is violated (Dixon, 1970), where $\varepsilon > 0$ is a prefixed constant. This results in a hybrid method. It is clear that the direction $d_k$ of the method is a descent direction of $f$ at $x_k$ Consequently, the Armijo-type line search or Wolfe-Powell-type line search can be used. This hybrid method has been shown to be globally convergent (Dixon, 1970).

The study of global convergence of pure conjugate gradient methods has also made good progress. Global convergence results for the FR method have been obtained by Zoutendijk (1970) if exact line search was used. Al-Baali (1985) proved the global convergence of FR method with inexact line search. The global convergence of the PRP method with exact line search has been proved in Polak & Ribiere (1969) under strong convexity assumption on $f$. Somewhat surprisingly, Powell (1986) constructed an example which showed that the exact line search could result in a non-convergent sequence in the case of PRP and HS methods. Inspired by Powell’s work, Gilbert & Nocedal (1992) conducted an elegant analysis and showed that the PRP method is globally convergent if $\beta_{k}^{\text{PRP}}$ is restricted to be non-negative and $\alpha_k$ is determined by a line-search step satisfying the sufficient descent condition

$$
g_k^T d_k \leq -c \|g_k\|^2
$$

in addition to the standard Wolfe conditions. To investigate the global convergence property of the PRP method, Grippo & Lucidi (1997) proposed a new line search below. For given constants $\mu > 0$, $\delta > 0$ and $\rho \in (0, 1)$, let

$$
a_k = \max \left\{ \rho^j \frac{\mu \|g_k^T d_k\|}{\|d_k\|^2} ; j = 0, 1, \ldots \right\}
$$

satisfy

$$f(x_{k+1}) \leq f(x_k) - \delta a_k^2 \|d_k\|^2$$

and

$$-c_1 \|g(x_{k+1})\|^2 \leq g(x_{k+1})^T d_{k+1} \leq -c_2 \|g(x_{k+1})\|^2,$$

where $0 < c_2 < 1 < c_1$ are constants. Grippo & Lucidi (1997) proved that the PRP method with above line-search condition is globally convergent for non-convex minimization. The theoretical results
obtained in Grippo & Lucidi (1997) are very interesting. Recently, Hager & Zhang (2005) proposed a new conjugate gradient method with

$$\beta_k = \beta_k^{HZ} \triangleq \max \{ \beta_k^{HZ}, \eta_k \}, \quad \eta_k = \frac{-1}{\|d_{k-1}\| \min \{ \eta, \|g_{k-1}\| \}}$$

where \( \eta > 0 \) is a constant and

$$\beta_k^{HZ} = \frac{g_k^T \left( y_{k-1} - \frac{2\|y_{k-1}\|^2}{s_{k-1}^Ty_{k-1}} \right)}{d_{k-1}^Ty_{k-1}}$$

with \( s_{k-1} = x_k - x_{k-1} \). This method can guarantee that \( d_k \) provides a descent direction of \( f \) at \( x_k \). Moreover, \( d_k \) satisfies \( g_k^T d_k \leq -\frac{\eta}{2}\|g_k\|^2 \). This method can be regarded as a modified HS method. Hager & Zhang (2005) proved that this method with the Wolfe line search converges globally. Other interested results about the global convergence of conjugate gradient methods can be found in Beale (1972), Dai & Yuan (1996a,b, 2000), Dai (1998), Hager & Zhang (2006), Hu & Storey (1991), Liu et al. (1995), Sun & Zhang (2001), Touati-Ahmed & Storey (1990), etc.

Although the PRP method performs better than other conjugate gradient methods in practice, it is generally not a descent method if Armijo-type line search is used. The purpose of this paper is to overcome this drawback. We propose a modified Polak–Ribiére–Polyak (MPRP) method. An attractive property of the MPRP method lies in that the direction generated by the method is always a descent direction of the objective function. This property is independent of the line search used. Moreover, if exact line search is used, the MPRP method reduces to the standard PRP method. Under appropriate conditions, we show that the MPRP method with Armijo-type line search is globally convergent.

In Section 2, we present the MPRP method. To improve the efficiency of the method, we propose a way to estimate the initial step length in the line-search process in Section 2. In Section 3, we prove the global convergence of the method. In Section 4, we report some numerical results to test the proposed method and compare its performance with some existing methods.

2. Algorithm

In this section, we describe the MPRP method. In order to introduce our method, let us simply recall the well-known BFGS quasi-Newton method (see, e.g. Dennis & Moré, 1974, 1977). The direction \( d_k \) in the BFGS method is given by

$$d_k = -H_k g_k,$$

where \( H_k \) is obtained by the BFGS formula

$$H_k = H_{k-1} + \left( 1 + \frac{y_{k-1}^T H_{k-1} y_{k-1}}{s_{k-1}^T y_{k-1}} \right) \frac{s_{k-1} y_{k-1}}{s_{k-1}^T y_{k-1}} - \frac{s_{k-1} y_{k-1}^T H_{k-1} + H_{k-1} y_{k-1} s_{k-1}^T}{s_{k-1}^T y_{k-1}},$$

where \( s_{k-1} = x_k - x_{k-1} = \alpha_{k-1} d_{k-1}, y_{k-1} = g_k - g_{k-1} \). If \( H_{k-1} = I \) (here \( I \) is the identity matrix), then the above BFGS method becomes the L-BFGS method (Nocedal, 1980) or Memoryless BFGS method (Shanno, 1978). In this case, \( d_k \) can be written as

$$d_k = -g_k + \left( \frac{y_{k-1}^T g_k}{s_{k-1}^T y_{k-1}} - \left( 1 + \frac{y_{k-1}^T y_{k-1}}{s_{k-1}^T y_{k-1}} \right) \frac{s_{k-1}^T g_k}{s_{k-1}^T y_{k-1}} \right) \alpha_{k-1} d_{k-1} + \frac{s_{k-1}^T g_k}{s_{k-1}^T y_{k-1}} y_{k-1},$$
which shows that $d_k$ possesses the following form:

$$d_k = -g_k + \beta_k d_{k-1} - \theta_k y_{k-1}.$$

The above observation motivates us to construct a PRP conjugate gradient-like method where $d_k$ takes the following form:

$$d_k = \begin{cases} -g_k & \text{if } k = 0, \\ -g_k + \beta_k^{PRP} d_{k-1} - \theta_k y_{k-1} & \text{if } k > 0. \end{cases} \quad (2.1)$$

If we take

$$\theta_k = \frac{g_k^T d_{k-1}}{\|g_k\|^2}, \quad (2.2)$$

it is easy to see from (2.1) and (2.2) that

$$d_k^T g_k = -\|g_k\|^2. \quad (2.3)$$

This implies that $d_k$ provides a descent direction of $f$ at $x_k$. It is also clear that if exact line search is used, then we have $\theta_k = 0$. Consequently, the above method reduces to the standard PRP method. We summarize these obvious results as the following theorem.

**Theorem 2.1** Let $d_k$ be defined by (2.1). Then $d_k$ is a descent direction of $f$ at $x_k$. Moreover, if exact line search is used, then $d_k$ is the same as the direction generated by the standard PRP method.

Based on Theorem 2.1, we propose a MPRP method, as follows.

**Algorithm 2.1 (MPRP Method)**

**Step 0:** Choose constants $\rho, \delta \in (0, 1)$. Choose an initial point $x_0 \in \mathbb{R}^n$. Let $k := 0$.

**Step 1:** Compute $d_k$ by (2.1).

**Step 2:** Determine $\alpha_k = \max\{\rho^j, j = 0, 1, 2, \ldots\}$ satisfying

$$f(x_k + \alpha_k d_k) \leq f(x_k) - \delta \alpha_k^2 \|d_k\|^2. \quad (2.4)$$

**Step 3:** Let the next iterate be $x_{k+1} = x_k + \alpha_k d_k$.

**Step 4:** Let $k := k + 1$. Go to Step 1.

**Remark 2.1** It is easy to see that Algorithm 2.1 is well-defined. Moreover, it follows from (2.4) that the function value sequence $\{f(x_k)\}$ is decreasing. We also have from (2.4) that

$$\sum_{k=0}^{\infty} \alpha_k^2 \|d_k\|^2 < \infty, \quad (2.5)$$

if $f$ is bounded from below. In particular, we have

$$\lim_{k \to \infty} \alpha_k \|d_k\| = 0. \quad (2.6)$$

**Remark 2.2** In Step 2, the initial step length is set to 1. This may make the search directions tend to be poorly scaled. The line-search process may perform more function evaluations to obtain a suitable
step length $a_k$. It is common to see that the step lengths can be larger or smaller than 1 depending on how the problem is scaled. Moreover, the size of $a_k$ tends to vary in an unpredictable manner. This is in sharp contrast to quasi-Newton and limited memory methods which accept the unit step length most of the time, and thus usually require only one function evaluation per search direction. Nocedal (1996) pointed out that the efficiency of non-linear conjugate gradient methods would be greatly improved if one could design a variant of the FR method or the PRP method that produced well-scaled search directions without increasing the storage requirements of the iteration. In what follows, we propose a way to improve the initial step length.

Consider the following quadratic model:

$$q_k(t) = f(x_k) + t g_k^T d_k + \frac{t^2}{2} d_k^T \nabla^2 f(x_k) d_k.$$  \hspace{1cm} (2.7)

Note that if $\varepsilon > 0$ is sufficiently small, then the following approximate relation holds:

$$\nabla^2 f(x_k) d_k \approx \frac{g(x_k + \varepsilon d_k) - g(x_k)}{\varepsilon}. \hspace{1cm}$$

Let

$$z_k = \frac{g(x_k + \varepsilon d_k) - g(x_k)}{\varepsilon}. \hspace{1cm} (2.8)$$

Then, we get

$$q_k(t) \approx f(x_k) + t g_k^T d_k + \frac{t^2}{2} d_k^T z_k, \hspace{1cm} (2.9)$$

which implies

$$q'_k(t) \approx g_k^T d_k + t d_k^T z_k. \hspace{1cm} (2.10)$$

Let $g_k^T d_k + t d_k^T z_k = 0$. We get

$$t = -\frac{g_k^T d_k}{d_k^T z_k}. \hspace{1cm}$$

So, a reasonable choice of the initial step length is

$$t_k = \left| \frac{g_k^T d_k}{d_k^T z_k} \right|. \hspace{1cm} (2.11)$$

Based on the above argument, we can change the line-search step (Step 2) of Algorithm 2.1 as follows:

If $|d_k^T z_k| > 0$ and $f(x_k + t_k d_k) \leq f(x_k) - \delta \|t_k d_k\|^2$, we choose $a_k = t_k$;

Otherwise, we compute step length $a_k$ by Step 2 in the Algorithm 2.1. \hspace{1cm} (2.12)

We shall see in Section 4 that this initial step length works well.
3. Global convergence

In this section, we prove the global convergence of Algorithm 2.1 under the following assumptions.

**ASSUMPTION A**

1. The level set \( \Omega = \{ x \in \mathbb{R}^n | f(x) \leq f(x_0) \} \) is bounded.
2. In some neighbourhood \( N \) of \( \Omega \), \( f \) is continuously differentiable and its gradient is Lipschitz continuous, namely, there exists a constant \( L > 0 \) such that
   \[
   \| g(x) - g(y) \| \leq L \| x - y \| \quad \forall x, y \in N.
   \] (3.1)

Since \( \{ f(x_k) \} \) is decreasing, it is clear that the sequence \( \{ x_k \} \) generated by Algorithm 2.1 is contained in \( \Omega \). In addition, we get from Assumption A that there is a constant \( \gamma_1 > 0 \), such that
\[
\| g(x) \| \leq \gamma_1 \quad \forall x \in \Omega.
\] (3.2)

In the latter part of the paper, we always suppose that the conditions in Assumption A hold. Without specification, we let \( \{ x_k \} \) and \( \{ d_k \} \) be the iterative sequence and the direction sequence generated by Algorithm 2.1, respectively.

**LEMMA 3.1** If there exists a constant \( \varepsilon > 0 \) such that
\[
\| g_k \| \geq \varepsilon \quad \forall k,
\] (3.3)
then there exists a constant \( M > 0 \) such that
\[
\| d_k \| \leq M \quad \forall k.
\] (3.4)

**Proof.** By the definition of \( d_k \), we get from (3.1), (3.2) and (3.3) that
\[
\| d_k \| \leq \| g_k \| + \frac{2 \| g_k \| \| y_{k-1} \| \| d_{k-1} \|}{\| g_{k-1} \|^2} \leq \gamma_1 + \frac{2 \gamma_1 L a_{k-1} \| d_{k-1} \|}{\varepsilon^2} \| d_{k-1} \|.
\]
Since \( a_k d_k \to 0 \) as \( k \to \infty \), there exist a constant \( r \in (0, 1) \) and an integer \( k_0 \), such that the following inequality holds for all \( k \geq k_0 \):
\[
\frac{2 L \gamma_1}{\varepsilon^2} a_{k-1} \| d_{k-1} \| \leq r.
\]

Hence, we have, for any \( k > k_0 \),
\[
\| d_k \| \leq \gamma_1 + r \| d_{k-1} \| \leq \gamma_1 (1 + r + r^2 + \cdots + r^{k-k_0-1}) + r^{k-k_0} \| d_{k_0} \| \leq \frac{\gamma_1}{1-r} + \| d_{k_0} \|.
\]

Let \( M = \max \{ \| d_1 \|, \| d_2 \|, \ldots, \| d_{k_0} \|, \frac{\gamma_1}{1-r} + \| d_{k_0} \| \} \) to get (3.4).

Now we can establish the following global convergence theorem for Algorithm 2.1.

**THEOREM 3.2** We have
\[
\liminf_{k \to \infty} \| g_k \| = 0.
\] (3.5)

**Proof.** For the sake of contradiction, we suppose that the conclusion is not true. Then there exists a constant \( \varepsilon > 0 \) such that
\[
\| g_k \| \geq \varepsilon \quad \forall k.
\] (3.6)
If $\liminf_{k \to \infty} \alpha_k > 0$, we get from (2.3) and (2.6) that $\liminf_{k \to \infty} \|g_k\| = 0$. This contradicts assumption (3.6).

Suppose $\liminf_{k \to \infty} \alpha_k = 0$. That is, there is an infinite index set $K$ such that

$$\lim_{k \in K, \ k \to \infty} \alpha_k = 0. \quad (3.7)$$

It follows from Step 2 of Algorithm 2.1 that when $k \in K$ is sufficiently large, $\rho^{-1} \alpha_k$ does not satisfy (2.4). This means,

$$f(x_k + \rho^{-1} \alpha_k d_k) - f(x_k) > -\delta \rho^{-2} \alpha_k^2 \|d_k\|^2. \quad (3.8)$$

By the mean-value theorem, Lemma 3.1, (3.1) and (2.3), there is a $h_k \in (0, 1)$ such that

$$f(x_k + \rho^{-1} \alpha_k d_k) - f(x_k) = \rho^{-1} \alpha_k g(x_k + h_k \rho^{-1} \alpha_k d_k)^T d_k$$

$$= \rho^{-1} \alpha_k g_k^T d_k + \rho^{-1} \alpha_k (g(x_k + h_k \rho^{-1} \alpha_k d_k) - g_k)^T d_k$$

$$\leq \rho^{-1} \alpha_k g_k^T d_k + L \rho^{-2} \alpha_k^2 \|d_k\|^2,$$

where $L > 0$ is the Lipschitz constant of $g$. Substituting the last inequality into (3.8), we get for all $k \in K$ sufficiently large,

$$\|g_k\|^2 \leq \rho^{-1} (L + \delta) \alpha_k \|d_k\|^2.$$

Since $\{d_k\}$ is bounded and $\lim_{k \in K, \ k \to \infty} \alpha_k = 0$, the last inequality implies $\lim_{k \in K, \ k \to \infty} \|g_k\| = 0$. This also yields a contradiction. The proof is then complete. □

4. Numerical experiments

In this section, we report some preliminary numerical experiments. The test problems are the unconstrained problems in the CUTE (Bongartz et al., 1995) test problem library. We test almost all problems tested by Hager & Zhang (2005). The parameters in the MPRP method are specified as follows. We set $\delta = 10^{-4}$, $\rho = 0.5$ and $\varepsilon = 10^{-8}$. We stop the iteration if the iteration number exceeds $2 \times 10^4$ or the function evaluation number exceeds $4 \times 10^5$ or the inequality $\|g(x_k)\|_{\infty} \leq 10^{-6}$ is satisfied. All numerical results are listed on the web site http://blog.sina.com.cn/u/4928efd5010003cz.

According to an anonymous referee’s suggestion, we first compare the performance of the MPRP method with that of the steepest descent method and our method with Armijo-type line search. Figure 1 shows the performance of these methods relative to CPU time, which were evaluated using the profiles of Dolan & Moré (2002). That is, for each method, we plot the fraction $P$ of problems for which the method is within a factor $\tau$ of the best time. The left side of the figure gives the percentage of the test problems for which a method is the fastest; the right side gives the percentage of the test problems that are successfully solved by each of the methods. The top curve is the method that solved the most problems in a time that was within a factor $\tau$ of the best time. In the figure, ‘steep’ stands for the steepest descent method with the following standard Armijo line search: compute step length $\alpha_k = \max\{\rho^j, j = 0, 1, 2, \ldots\}$ such that

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \delta \alpha_k g_k^T d_k; \quad (4.1)$$

‘mprp’ stands for the MPRP method with the same line-search rule; and ‘mprpa’ stands for the MPRP method with the modified Armijo line search (2.12).
We then compare the performance of the MPRP method with that of the L-BFGS (Liu & Nocedal, 1989) method, CG_DESCENT method (Hager & Zhang, 2005) and PRP+ (Gilbert & Nocedal, 1992) method with Wolfe-type line search. The L-BFGS (Liu & Nocedal, 1989) and PRP+ (Gilbert & Nocedal, 1992) codes were obtained from Nocedal’s web page at http://www.ece.northwestern.edu/nocedal/software.html, and the CG_DESCENT (Hager & Zhang, 2005) code can also be obtained from Hager’s web page at http://www.math.ufl.edu/hager/papers/CG. The L-BFGS code is authored by Nocedal, while the PRP+ code is coauthored by Liu, Nocedal and Waltz. The CG_DESCENT code is coauthored by Hager and Zhang. All codes were written in Fortran and run on a PC with 2.66 GHz CPU processor and 1 GB RAM memory and Linux operation system.

Figures 2–4 list the performance of the above methods relative to CPU time, the number of function evaluations and the number of gradient evaluations, respectively, where

- ‘cg’ stands for the CG_DESCENT method with the approximate strong Wolfe line search proposed by Hager & Zhang (2005);
- ‘mprp1’ stands for the MPRP method with the same line search as ‘cg’;
- ‘lbfgs’ stands for the L-BFGS method with strong Wolfe line search proposed by Moré & Thuente (1994);
- ‘PRP+’ is the PRP+ method with the same line search as ‘lbfgs’.
- ‘mprp2’ is the MPRP method with the same line search as ‘PRP+’.

Figure 1 shows that ‘mprpa’ performed much better than the others did. It outperforms the ‘mprp’ and the ‘steep’ for about 60% of the test problems. In addition, it solves 77% of the test problems.
FIG. 2. Performance profiles for CPU time.

FIG. 3. Performance profiles for the number of function evaluations.
successfully. This means that the MPRP method with line search (2.12) does have some advantage. While testing, we also found that in most cases, the iteration numbers for the ‘mprpa’ are almost equal to the function evaluations, which means that the initial step length $t_k = \frac{g_k^T d_k}{d_k^T z_k}$ is often accepted.

We see from Fig. 2 that the performance of ‘mprp1’ is almost as same as that of ‘cg’. These two methods perform much better than the others. They ultimately solve 100% of the test problems. Figure 3 shows that ‘lbfgs’ has the best performance for the number of function evaluations since it solves about 47% of the problems with the smallest number of function evaluations. We can see from Fig. 4 that ‘mprp1’ has the best performance for the number of gradient evaluations since it can solve about 53% of the problems with the smallest number of gradient evaluations which is shown in the vertical axis; ‘cg’ has the second best performance in which it solves about 48% in the same situation. All numerical results show that the efficiency of the MPRP method is encouraging.

Acknowledgements

The authors would like to thank two referees for giving us many valuable suggestions and comments, which improve this paper greatly. We are very grateful to Prof. J. Nocedal and Prof. W. W. Hager for providing us with their codes. The work was done while the authors were working in the State Key Laboratory of Advanced Design and Manufacture for Vehicle Body of Hunan University. This work was supported by the 973 project (2004CB719402) and the National Science Foundation (10471036) of China.
REFERENCES


