Laboratoire d'Arithmétique,
Calcul formel et d'Optimisation

UMR CNRS 6090

Modification of the Wolfe line search rules to satisfy the descent condition in the Polak-Ribière-Polyak conjugate gradient method

Paul Armand

Rapport de recherche n° 2003-10
Déposé le 18 décembre 2003

Université de Limoges, 123 avenue Albert Thomas, 87060 Limoges Cedex

Tél. 05 55 45 73 23 - Fax. 05 55 45 73 22 - laco@unilim.fr
http://www.unilim.fr/laco/

Université de Limoges, 123 avenue Albert Thomas, 87060 Limoges Cedex

Tél. 05 55 45 73 23 - Fax. 05 55 45 73 22 - laco@unilim.fr
http://www.unilim.fr/laco/

Modification of the Wolfe line search rules to satisfy the descent condition in the Polak-Ribière-Polyak conjugate gradient method

Paul Armand[†]

Decembre 15, 2003

Abstract. The Polak-Ribière-Polyak conjugate gradient algorithm is a useful tool of unconstrained numerical optimization. Efficient implementations of the algorithm usually perform line searches satisfying the strong Wolfe conditions. It is well known that these conditions do not guarantee that the successive computed directions are descent directions. This paper proposes a relaxation of the strong Wolfe line search conditions to guarantee the descent condition at each iteration. It is proved that the resulting algorithm preserves the usual convergence properties. In particular, it is shown that the Zoutendijk condition holds under mild assumptions and that the algorithm is convergent under a strong convexity assumption. Numerical tests are presented.

Key words. unconstrained optimization, conjugate gradient method, line search algorithm

AMS subject classification. 90C06, 90C30, 65K05

1 Introduction

The Polak-Ribière-Polyak [22, 23] (PRP) conjugate gradient algorithm is an iterative method for computing an unconstrained minimizer of a continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$. The method is particularly useful for solving problems when n is large, because it requires very few memory storages. The algorithm generates a sequence of iterates according to

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, 2, \ldots \tag{1.1}$$

where $d_k \in \mathbb{R}^n$ is the line search direction and $\alpha_k > 0$ is a step length. The direction is defined by

$$d_k = \begin{cases} -g_k & \text{for } k = 0, \\ -g_k + \beta_k d_{k-1} & \text{for } k \geq 1, \end{cases} \tag{1.2}$$

where g_k denotes the gradient $\nabla f(x_k)$ and β_k is a scalar defined by

$$\beta_k = \frac{(g_k - g_{k-1})^\top g_k}{\|g_{k-1}\|^2}. \tag{1.3}$$

[†]LACO–CNRS, Université de Limoges, Faculté des Sciences, 123, avenue Albert Thomas, 87060 Limoges (France) ; e-mail: armand@unilim.fr.

The step length is computed by performing a line search along d_k. In practice, a relevant choice is to compute α_k according to the realization of the strong Wolfe conditions, namely

$$f(x_{k+1}) \le f(x_k) + \omega_1 \alpha_k g_k^\top d_k \tag{1.4}$$

and

$$|g_{k+1}^\top d_k| \le -\omega_2 g_k^\top d_k. \tag{1.5}$$

where $0 < \omega_1 < \omega_2 < 1$, see [7, 21].

The PRP method belongs to class of descent algorithms. This means that each new direction d_{k+1} must satisfy the descent condition

$$g_{k+1}^\top d_{k+1} < 0. \tag{1.6}$$

However, and contrary to other formulæ of β_k, such as Fletcher-Reeves [8], conjugate descent [7] or Dai-Yuan [5], the realization of the strong Wolfe conditions does not guarantee that (1.6) is satisfied at each iteration. The following example is given in [4] (see [17] for a similar example). Consider $f(x) = ax^2$, where $a = \min\{(1+\omega_2)/2, 1-\omega_1\}$, with $0 < \omega_2 \le 1/2$. From the starting point $x_0 = 1$, the step length $\alpha_0 = 1$ satisfies (1.4) and (1.5), but $g_1^\top d_1 = 4a^2(2a-1)^3 > 0$. In particular, this example shows that the algorithm defined by (1.1)–(1.5), does not converge, even if the function is strongly convex. Note that the same behavior occurs when formula (1.3) is replaced by the Hestenes-Stiefel [15] formula.

This negative result has several consequences. From a theoretical point of view, all convergence analyses related to the PRP algorithm have been done with the additional assumption that the descent condition is satisfied at each iteration, see [3, 6, 12]. It would be more satisfactory to obtain convergence results without this assumption on the behavior of the algorithm. From a practical point of view, several line search algorithms have been proposed to satisfy the Wolfe conditions [1, 7, 16, 19, 20, 25, 26], but, and as emphasized by Gilbert and Nocedal in [12, p. 39], none of them is guaranteed to satisfy the strong Wolfe conditions and also provide a descent direction for the PRP method. To force the realization of (1.6) in the numerical tests, the strategy proposed in [12] is the following. At first, a step length that satisfies (1.4) and (1.5) is computed by means of the Moré-Thuente [20] algorithm, then the line search iterations are pursued until (1.6) is also satisfied. The code CG+, developed by Liu, Nocedal and Waltz [18], uses this technique. Most the time, this technique succeeds in satisfying the conditions (1.4)–(1.6) at each iteration, except for a problem of the CUTEr collection [13], for which we observed a failure due to the non realization of the descent condition. We will come back to this example at the end of the paper.

These considerations motivated us to propose a modification of the strong Wolfe conditions to satisfy the descent condition in the PRP conjugate gradient algorithm, while preserving the convergence properties and the numerical efficiency. Our line search technique, described in Section 2, is based on a relaxation of (1.4) whenever it is necessary during the line search procedure, allowing the trial steps to go in a neighborhood of a minimizer of the line search function, where both conditions (1.5) and (1.6) are satisfied. In fact, this technique has his roots in previous works on line searches for quasi-Newton methods in the framework of SQP algorithms, see [2, 9, 10, 11]. The general idea is to use the sufficient decrease condition (1.4) as a guide to produce trial steps that converge to a stationary point of the line search function, without supposing the existence of a step satisfying all the line search conditions. In Section 3, we will show that the resulting PRP algorithm satisfies the Zoutendijk condition under mild assumptions and that it is globally convergent when the function f is strongly convex. Next we present some numerical experiments in Section 4, to show the feasibility and the efficiency of this new line search technique.

2 The line search algorithm

Using (1.2) and (1.6), the direction d_{k+1} is a descent direction provided that

$$\beta_{k+1} g_{k+1}^\top d_k < \|g_{k+1}\|^2.$$

To find a step length that satisfies this inequality, a possible strategy would be to reduce enough the size of the step. Since the gradient is supposed to be continuous, β_{k+1} goes to zero with α_k. Such an approach has been used by Grippo and Lucidi [14]. They proposed line search conditions that accept short steps and proved a strong convergence result for the resulting PRP algorithm. From a practical point of view, the computation of short steps is not really efficient. Moreover, this strategy can not be adopted in our case, because (1.5) is not satisfied for small steps. A second strategy would be to choose a step length near a local minimizer of the line search function (note that in this case a global convergence result cannot be expected because of the counterexample of Powell [24]). In that case the scalar product $g_{k+1}^\top d_k$ goes to zero, and unless g_{k+1} goes also to zero, the descent condition can be satisfied. Note that (1.5) is also satisfied in a neighborhood of a minimum, but this is not necessarily the case for condition (1.4). To solve this kind of conflict, we propose to relax the decrease condition (1.4) in the following way. Starting from an initial trial step, we first use a backtracking technique until (1.4) is satisfied. Let $\alpha_{k,1}$ the corresponding step. Suppose that either (1.5) or (1.6) is not satisfied at the new point, say $x_{k,1} = x_k + \alpha_{k,1} d_k$. If $g_{k,1}^\top d_k > 0$, then a minimum of the line search function $\alpha \to f(x_k + \alpha d_k)$ has been bracketed. In this case it is easy to find a step near this minimum while satisfying the decrease condition, see for example the sectioning phase described in [7, Section 2.6] or the modified updating algorithm described in [20, Section 3]. We adopt the same strategy here. Suppose now that $g_{k,1}^\top d_k < 0$. In this case, the decrease condition is reformulated as follows:

$$f(x_k + \alpha d_k) \le f(x_k) + \alpha_{k,1}\omega_1 g_k^\top d_k + (\alpha - \alpha_{k,1})\omega_1 \max\{g_k^\top d_k, g_{k,1}^\top d_k\}.$$

From $\alpha_{k,1}$, a forward (or extrapolation) step is computed and it is followed by a sequence of backtracking steps until a new step, say $\alpha_{k,2}$, satisfies the above condition. If $\alpha_{k,2}$ satisfies the other line search conditions, then it is an acceptable step, otherwise the decrease condition is reformulated as previously and so on. By repeating this procedure, it is only asked to the function value $f(x_{k+1})$, to be not greater than the value of a decreasing and convex piecewise linear function. We will show that it is sufficient to obtain global convergence properties for the descent algorithm. Note that this algorithmic scheme operates without assuming the existence of a step length which satisfies all conditions (1.4) and (1.5)–(1.6). In the sequel, the latter conditions will be replaced by only one set of conditions, called the stopping criterion of the line search. The only assumption is that the stopping criterion is satisfied at a stationary point of the line search function.

To give a description of the line search algorithm, we denote by ϕ the line search function. With the previous notation we would have $\phi(\alpha) = f(x_k + \alpha d_k)$.

Assumptions 2.1 *(i)* The function $\phi : \mathbb{R} \to \mathbb{R}$ is continuously differentiable on $[0, \infty)$ and $\phi'(0) < 0$. *(ii)* If $\phi'(\alpha) = 0$, then the step length α satisfies the line search stopping criterion.

The algorithm generates a sequence $\{\alpha_i\}$ of steps in the following way. While $\phi'(\alpha_i) < 0$ the algorithm stays in Phase I. From α_i, a first extrapolation trial step is computed and is followed by a sequence of backtracking trial steps, until a sufficient decrease of ϕ is obtained. This defines the next step α_{i+1}. As soon as a new step satisfies $\phi'(\alpha_{i+1}) > 0$, this means that a minimizer has been bracketed in $[\alpha_i, \alpha_{i+1}]$, then the algorithm enters in Phase II. In this second phase, the algorithm generates a sequence of intervals of the form $[\alpha_i, \alpha_{i+1}]$ or $[\alpha_{i+1}, \alpha_i]$ that contain the minimizer and whose lengths tend to zero. It is assumed that the following constants are given: $\omega_1 \in (0, 1)$, $\tau_{\mathrm{E}} > 0$ and $\tau_{\mathrm{I}} \in (0, \frac{1}{2})$.

Line search algorithm

Phase I

1. Set $i = 0$, $\alpha_0 = 0$ and $b_0 = \infty$.
2. Find a step length such that a sufficient decrease of ϕ is obtained:
   - 2.1. Set $j = 0$, $b_{i,0} = b_i$ and $s_i = \max\{\phi'(\alpha_l) : 0 \le l \le i\}$.
   - 2.2. If $b_i = \infty$, choose $\tau_{i,0} \ge \tau_{\mathrm{E}}$ and set $\alpha_{i,1} = \alpha_i + \tau_{i,0}$, else choose $\tau_{i,0} \in [\tau_{\mathrm{I}}, (1-\tau_{\mathrm{I}})]$ and set $\alpha_{i,1} = \alpha_i + \tau_{i,0}(b_i - \alpha_i)$. Go to Step 2.4.
   - 2.3. Choose $\tau_{i,j} \in [\tau_{\mathrm{I}}, (1-\tau_{\mathrm{I}})]$ and set $\alpha_{i,j+1} = \alpha_i + \tau_{i,j}(\alpha_{i,j} - \alpha_i)$.
   - 2.4. If $\phi(\alpha_{i,j+1}) \le \phi(0) + \omega_1 \sum_{l=0}^{i-1} (\alpha_{l+1} - \alpha_l) s_l + \omega_1 (\alpha_{i,j+1} - \alpha_i) s_i$, go to Step 3.
   - 2.5. If $\phi'(\alpha_{i,j+1}) > 0$, set $b_{i,j+1} = \alpha_{i,j+1}$, else set $b_{i,j+1} = b_{i,j}$.
   - 2.6. Set $j = j + 1$ and go to Step 2.3.
3. Set $\alpha_{i+1} = \alpha_{i,j+1}$. If the stopping criterion holds, stop.

4. If $\phi'(\alpha_{i+1}) < 0$, set $b_{i+1} = b_{i,j}$, $i = i + 1$ and go to Step 2, else set $b_{i+1} = \alpha_i$, $i = i + 1$ and go to Step 5.

PHASE II

5. Find a step length such that a decrease of ϕ is obtained:
   - 5.1. Set $j = 0$ and $b_{i,0} = b_i$.
   - 5.2. Choose $\tau_{i,0} \in [\tau_{\text{I}}, (1 - \tau_{\text{I}})]$ and set $\alpha_{i,1} = \alpha_i + \tau_{i,0}(b_i - \alpha_i)$. Go to Step 5.4.
   - 5.3. Choose $\tau_{i,j} \in [\tau_{\text{I}}, (1-\tau_{\text{I}})]$ and set $\alpha_{i,j+1} = \alpha_i + \tau_{i,j}(\alpha_{i,j} - \alpha_i)$.
   - 5.4. If $\phi(\alpha_{i,j+1}) \le \phi(\alpha_i)$, go to Step 6.
   - 5.5. Set $b_{i,j+1} = \alpha_{i,j+1}$.
   - 5.6. Set $j = j + 1$ and go to Step 5.3.
6. Set $\alpha_{i+1} = \alpha_{i,j+1}$. If the stopping criterion holds, stop.
7. If $\phi'(\alpha_{i+1})(\alpha_{i+1} - \alpha_i) < 0$, set $b_{i+1} = b_{i,j}$, else set $b_{i+1} = \alpha_i$.
8. Set $i = i + 1$ and go to Step 5.

The following lemma gives some elementary properties of the line search algorithm.

Lemma 2.2 *Suppose that Assumptions 2.1 hold. For any index $i \ge 0$, if $\phi'(\alpha_i) \neq 0$, then the following properties hold:*

(*i*) $\phi'(\alpha_i)(b_i - \alpha_i) < 0$.

(*ii*) *If $b_i < \infty$, then $\phi(\alpha_i) < \phi(b_i)$ or $\phi'(\alpha_i)\phi'(b_i) < 0$.*

(*iii*) *If $b_i < \infty$, then the interval bounded by α_i and b_i contains at least one local minimizer of ϕ.*

(*iv*) *For sufficiently large j, the decrease conditions at Step 2.4 and Step 5.4 are satisfied. In particular, there exists $j_i \ge 0$ such that*

$$\begin{cases} \alpha_i < \alpha_{i+1} = \alpha_{i,j_i+1} < \cdots < \alpha_{i,1} < b_i & \text{if } \phi'(\alpha_i) < 0, \\ b_i < \alpha_{i,1} < \cdots < \alpha_{i,j_i+1} = \alpha_{i+1} < \alpha_i & \text{otherwise.} \end{cases}$$

(*v*) *If the algorithm entered in Phase II, then $|\alpha_{i+1} - b_{i+1}| \le (1 - \tau_I)|\alpha_i - b_i|$.*

Proof. To prove (*i*), consider first that during Phase I one has $\phi'(\alpha_i) < 0$ (Step 4) and $\alpha_i < b_i$ (Step 2). During Phase II, the first time the algorithm goes to Step 5, one has $\phi'(\alpha_i) > 0$ and $b_i = \alpha_{i-1} < \alpha_i$ (Step 4). For the next iterations, the property follows that at the end of Step 7 one has either $\phi'(\alpha_{i+1})(\alpha_{i+1} - \alpha_i) > 0$ and $b_{i+1} = \alpha_i$, or $\phi'(\alpha_{i+1})(\alpha_{i+1} - \alpha_i) < 0$ and $b_{i+1} = b_{i,j} < \alpha_{i+1} < \alpha_i$ or $\alpha_i < \alpha_{i+1} < b_{i,j} = b_{i+1}$.

To prove property (*ii*), suppose that $b_i < \infty$. During Phase I one has $\phi'(\alpha_i) < 0$ (Step 4). The value of b_i depends on steps 2.1, 2.5 and 4. As long as $\phi'(\alpha_i) < 0$ one has $\phi'(b_i) > 0$. Consider now the sequence of values generated during Phase II. The first time the algorithm goes to Step 5, one has $\phi'(\alpha_i) > 0$ and $\phi'(b_i) < 0$ (Step 4). For the next iterations, suppose that

Property (*ii*) is satisfied at rank i. By Assumption 2.1 (*ii*), one has $\phi'(\alpha_{i+1}) \neq 0$. Suppose at first that $\phi'(\alpha_{i+1})(\alpha_{i+1} - \alpha_i) > 0$. One has $b_{i+1} = \alpha_i$ (Step 7) and $\phi'(\alpha_i)(b_i - \alpha_i) < 0$ (Property (*i*)). But $(\alpha_{i+1} - \alpha_i)$ and $(b_i - \alpha_i)$ are of the same sign, it follows that $\phi'(\alpha_{i+1})\phi'(b_{i+1}) < 0$. Suppose now that $\phi'(\alpha_{i+1})(\alpha_{i+1} - \alpha_i) < 0$. One has either $b_{i+1} = \alpha_{i,j}$, for some $j \geq 1$ or $b_{i+1} = b_i$. In the former case, one has $\phi(\alpha_{i,j}) > \phi(\alpha_i)$ and since $\phi(\alpha_i) \geq \phi(\alpha_{i+1})$, we obtain $\phi(b_{i+1}) > \phi(\alpha_{i+1})$. In the latter case, from the induction hypothesis one has either $\phi(b_{i+1}) = \phi(b_i) > \phi(\alpha_i) \geq \phi(\alpha_{i+1})$ or $\phi'(\alpha_i)\phi'(b_{i+1}) < 0$. In this last case, since $\phi'(\alpha_i)(b_i - \alpha_i) < 0$, $\phi'(\alpha_{i+1})(\alpha_{i+1} - \alpha_i) < 0$ and $(b_i - \alpha_i)$ is of the same sign that $(\alpha_{i+1} - \alpha_i)$, we obtain $\phi'(\alpha_i)\phi'(\alpha_{i+1}) > 0$, and thus $\phi'(\alpha_{i+1})\phi'(b_{i+1}) < 0$.

Property (*iii*) is a straightforward consequence of the first two properties.

The choices of $\alpha_{i,j}$ (Steps 2.3 and 5.3), imply that $|\alpha_{i,j+1} - \alpha_i| \leq (1 - \tau_{\text{I}})^j |\alpha_{i,1} - \alpha_i|$ for $j \geq 1$, and thus $\alpha_{i,j} \to \alpha_i$ when $j \to \infty$. Since $\phi'(\alpha_i)(b_i - \alpha_i) < 0$ and $0 < \omega_1 < 1$, the tests at Steps 2.4 and 5.4 are satisfied for sufficiently large j, from which Property (*iv*) follows.

To prove (*v*), suppose that the algorithm entered is Phase II. At Step 7, if $b_{i+1} = b_{i,j}$, then from Step 5.5 one has $b_{i,j} = \alpha_{i,j}$ and thus $\alpha_{i+1} = \alpha_i + \tau_{i,j_i}(b_{i+1} - \alpha_i)$. Since b_{i+1} is in the interval bounded by α_i and b_i, one has $|\alpha_{i+1} - b_{i+1}| \leq (1 - \tau_{\text{I}})|b_{i+1} - \alpha_i| \leq (1 - \tau_{\text{I}})|b_i - \alpha_i|$. On the other hand, if $b_{i+1} = \alpha_i$, then $|\alpha_{i+1} - b_{i+1}| = |\alpha_{i+1} - \alpha_i| \leq (1 - \tau_{\text{I}})|b_i - \alpha_i|$. □

Suppose that α_{i+1} was computed during Phase I. One has $\alpha_i < \alpha_{i+1}$ and

$$\phi(\alpha_{i+1}) \leq \phi(0) + \omega_1 \sum_{j=0}^{i} (\alpha_{j+1} - \alpha_j) \max\{\phi'(\alpha_l) : 0 \leq l \leq j\}. \tag{2.1}$$

On the other hand, if α_{i+1} was computed during Phase II, one has $\phi(\alpha_{i+1}) \leq \phi(\alpha_i)$.

Proposition 2.3 *Suppose that Assumptions 2.1 hold. If the algorithm does not terminate, then an infinite sequence of step lengths $\{\alpha_i\}$ is built and either the sequence $\{\phi(\alpha_i)\}$ tends to $-\infty$ or $\liminf_{i\to\infty} \phi'(\alpha_i) = 0$.*

Proof. While the stopping criterion at Step 3 or Step 6 is not satisfied, Assumption 2.1 (*ii*) implies that $\phi'(\alpha_i) \neq 0$ and Property (*iv*) of Lemma 2.2 implies that the next step α_{i+1} is well defined.

Suppose at first that $\phi'(\alpha_i) > 0$ for some $i \geq 1$, then the algorithm entered in Phase II. Properties (*iii*) and (*v*) of Lemma 2.2 imply that the sequences $\{\alpha_i\}$ and $\{b_i\}$ converge to a common limit point $\bar{\alpha}$ such that $\phi'(\bar{\alpha}) = 0$.

Suppose now that $\phi'(\alpha_i) < 0$ for all $i \geq 1$. We proceed by contradiction, by supposing that ϕ is bounded from below and that there exists $\varepsilon > 0$ such that

$$\phi'(\alpha_i) \leq -\varepsilon, \quad \text{for all } i \geq 0. \tag{2.2}$$

Since the algorithm stays in Phase I, the sequence $\{\alpha_i\}$ is increasing and Property (2.1) implies that

$$\begin{aligned}\phi(\alpha_i) &\leq \phi(0) + \omega_1 \sum_{j=0}^{i-1} (\alpha_{j+1} - \alpha_j) \max\{\phi'(\alpha_l) : 0 \leq l \leq j\}, \qquad (2.3)\\ &\leq \phi(0) - \omega_1 \sum_{j=0}^{i-1} (\alpha_{j+1} - \alpha_j)\varepsilon,\\ &= \phi(0) - \omega_1 \varepsilon \alpha_i.\end{aligned}$$

Since ϕ is bounded from below, the sequence $\{\alpha_i\}$ is convergent to some limit point $\bar{\alpha}$.

The distance between the first trial step $\alpha_{i,1}$ and α_i is bounded away from zero. Indeed, either $b_i = \infty$ for all i and then $\alpha_{i,1} - \alpha_i \geq \tau_{\rm E} > 0$, or $b_i < \infty$ for some iteration i. In the latter case, the choice of $\alpha_{i,1}$ at Step 2.2 implies that $\alpha_{i,1} - \alpha_i \geq \tau_{\rm I}(b_i - \alpha_i)$. But we know from Lemma 2.2 (*iii*) that the interval $[\alpha_i, b_i]$ contains a local minimizer of ϕ, therefore $(b_i - \alpha_i)$ is bounded away from zero because of (2.2). Since $(\alpha_{i+1} - \alpha_i) \to 0$ when $i \to \infty$, for sufficiently large i the decrease condition at Step 2.4 is not satisfied at $\alpha_{i,1}$. It follows that the sequence $\{\alpha_{i,j_i}\}$ is such that $\alpha_{i+1} = \alpha_i + \tau_{i,j_i}(\alpha_{i,j_i} - \alpha_i)$ and

$$\begin{aligned}\phi(\alpha_{i,j_i}) &> \phi(0) + \omega_1 \sum_{j=0}^{i-1} (\alpha_{j+1} - \alpha_j) \max\{\phi'(\alpha_l) : 0 \leq l \leq j\}\\ &\quad + \omega_1(\alpha_{i,j_i} - \alpha_i) \max\{\phi'(\alpha_l) : 0 \leq l \leq i\}.\end{aligned}$$

Subtracting $\phi(\alpha_i)$ on both sides, using (2.3) and next (2.2), we obtain

$$\frac{\phi(\alpha_{i,j_i}) - \phi(\alpha_i)}{\alpha_{i,j_i} - \alpha_i} - \phi'(\alpha_i) \geq (1 - \omega_1)\varepsilon.$$

By taking limit $i \to \infty$, we obtain $0 \geq (1-\omega_1)\varepsilon$ a contradiction with $\omega_1 < 1$. □

This line search technique allows to compute a step length that satisfies the Wolfe or strong Wolfe conditions. It suffices to use it with one of both following stopping criteria:

$$\phi'(\alpha_i) \geq \omega_2 \phi'(0) \qquad (2.4)$$

or

$$|\phi'(\alpha_i)| \leq \omega_2 |\phi'(0)|, \qquad (2.5)$$

where $\omega_2 \in (0, 1)$.

Proposition 2.4 *Suppose that Assumption 2.1 (i) holds and that ϕ is bounded from below. If (2.4) (resp. (2.5)) is used as line search stopping criterion, then the line search algorithm terminates at some α_i and $\phi(\alpha_i) \leq \phi(0) + \omega_1\omega_2\alpha_i\phi'(0)$.*

Proof. Since the stopping criteria (2.4) and (2.5) are satisfied in a neighborhood of a stationary of ϕ, Assumptions 2.1 are satisfied, and thus Proposition 2.3 implies that the line search terminates after a finite number of iterations.

While (2.4) (resp. (2.5)) is not satisfied, one has $\phi'(\alpha_i) < \omega_2\phi'(0)$ (resp. $\phi'(\alpha_i) < \omega_2\phi'(0)$ or $\phi'(\alpha_i) > -\omega_2\phi'(0) > 0$). Using Property (2.1) and the remarks which go with, we obtain the result. □

3 Application to the PRP conjugate gradient method

Let x_0 be the starting point of the minimization method. We assume that the following standard assumptions are satisfied.

Assumptions 3.1 (i) *The level set* $\mathcal{L} = \{x : f(x) \leq f(x_0)\}$ *is bounded.* (ii) *The gradient of* f *is Lipschitzian in a neighborhood* $\mathcal{N}$ *of* $\mathcal{L}$, *that is, there exists* $L > 0$ *such that* $\|g(x) - g(y)\| \leq L\|x - y\|$ *for all* $(x, y) \in \mathcal{N} \times \mathcal{N}$.

We consider the following PRP conjugate gradient algorithm. Starting from x_0, a sequence $\{x_k\}$ is generated according to (1.1)–(1.3), the step length α_k being computed with the line search algorithm described in Section 2, where the line search function is defined by $\alpha \to f(x_k + \alpha d_k)$ and where the line search stopping criterion is composed of both conditions (1.5) and (1.6). At each iteration k, the line search algorithm generates intermediate step lengths $\alpha_{k,i}$ satisfying the decrease conditions at Steps 2.4 or 5.4. We denote by $x_{k,i} = x_k + \alpha_{k,i}d_k$ the corresponding intermediate iterates and we use the notation $g_{k,i} = g(x_{k,i})$.

The following result suggests to insert the stopping test of the PRP algorithm during the line search procedure.

Proposition 3.2 *Suppose that Assumption 3.1 hold. At iteration* k, *either the line search is finite or* $\liminf_{i\to\infty} \|g_{k,i}\| = 0$.

Proof. Suppose that at some iteration k the line search does not terminate. For all index $i \geq 1$ one has

$$|g_{k,i}^\top d_k| > -\omega_2 g_k^\top d_k \quad \text{or} \quad \frac{(g_{k,i} - g_k)^\top g_{k,i}}{\|g_k\|^2} g_{k,i}^\top d_k \geq \|g_{k,i}\|^2.$$

By Assumptions 3.1 and Proposition 2.3, there exist a subset I of indices such that $\{x_{k,i}\}_{i\in I}$ is convergent and $g_{k,i}^\top d_k \to 0$ when $i \to \infty$ in I. By taking limits in the above inequalities, we deduce that $\|g_{k,i}\| \to 0$ when $i \to \infty$ in I. □

The following result is a useful tool for the convergence analysis of descent algorithms. Property (3.1) is called the Zoutendijk condition. It was proved by Zoutendijk [29] and Wolfe [27, 28].

Proposition 3.3 *Suppose that Assumption 3.1 hold. If the number of line search iterations is finite for all iteration* k, *then*

$$\sum_{k\geq 0} (g_k^\top d_k)^2 \|d_k\|^{-2} < \infty. \tag{3.1}$$

Proof. Since for all $k \geq 0$, the number of line search iterations is finite, there exists $w_k \geq 1$ such that

$$g_{k,i}^\top d_k < \omega_2 g_k^\top d_k \quad \text{for all } 1 \leq i < w_k \quad \text{and} \quad g_{k,w_k}^\top d_k \geq \omega_2 g_k^\top d_k. \tag{3.2}$$

In particular, for all $1 \leq i < w_k$ one has $g_{k,i}^\top d_k < 0$ and $0 = \alpha_{k,0} < \alpha_{k,1} < \ldots < \alpha_{k,w_k}$. By using the properties (2.1) and (3.2), we deduce that

$$\begin{aligned} f(x_{k+1}) &\leq f(x_k) + \omega_1 \sum_{i=0}^{w_k-1} (\alpha_{k,i+1} - \alpha_{k,i}) \max\{g_{k,l}^\top d_k : 0 \leq l \leq i\} \\ &\leq f(x_k) + \omega_1 \sum_{i=0}^{w_k-1} (\alpha_{k,i+1} - \alpha_{k,i}) \omega_2 g_k^\top d_k \\ &= f(x_k) + \omega_1 \omega_2 \alpha_{k,w_k} g_k^\top d_k. \end{aligned} \tag{3.3}$$

Using the second inequality of (3.2) and the Lipschitz continuity of g one has

$$(1 - \omega_2)|g_k^\top d_k| \leq (g_{k,w_k} - g_k)^\top d_k \leq L\alpha_{k,w_k} \|d_k\|^2.$$

Combining this last inequality with (3.3) we obtain

$$L^{-1}\omega_1\omega_2(1 - \omega_2)(g_k^\top d_k)^2 \|d_k\|^{-2} \leq f(x_k) - f(x_{k+1}).$$

To conclude the proof, it suffices to note that Assumptions 3.1 imply that f is bounded from below. $\square$

We assume now that the function f is strongly convex on $\mathcal{L}$, that is, there exists $\kappa > 0$ such that the function $f(\cdot) - \frac{\kappa}{2}\|\cdot\|^2$ is convex on $\mathcal{L}$. For a differentiable function, the strong convexity is equivalent to the the strong monotonicity of its gradient, namely $\tau\|x-y\|^2 \leq (g(x) - g(y))^\top(x-y)$ for all $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$. Note that a strongly convex function satisfies (i) of Assumptions 3.1.

Lemma 3.4 *Suppose that φ is continuously differentiable and strongly convex. If $\{\xi_j\}$ is a sequence in $\mathbb{R}^n$ such that $\{\varphi(\xi_j)\}$ is nonincreasing and a subsequence of $\{\nabla\varphi(\xi_j)\}$ tends to zero, then the whole sequence $\{\xi_j\}$ converges to the unique minimum of φ.*

Proof. The strong convexity of φ implies that its level sets are bounded. Since $\{\varphi(\xi_j)\}$ is nonincreasing, the sequence $\{\xi_j\}$ is bounded. Moreover, φ is bounded from below and thus $\lim_{j\to\infty} \varphi(\xi_j) = \varphi^*$ exists. The continuity of φ implies that all the convergent subsequences of $\{\xi_j\}$ have the same limit point ξ^* such that $\varphi(\xi^*) = \varphi^*$, therefore $\lim_{j\to\infty} \xi_j = \xi^*$. By assumption, a subsequence of the gradient values $\nabla\varphi(\xi_j)$ goes to zero, therefore ξ^* is the unique minimum of φ. $\square$

Theorem 3.5 *Suppose that f is strongly convex and that g is Lipschitz continuous on a neighborhood of the level set $\{x : f(x) \leq f(x_0)\}$. The whole sequence $\{x_{k,i}\}$ converges to the unique minimum of f.*

Proof. Suppose that at some iteration k, the line search does not terminate. By Proposition 3.2, there exists a subset I of indices such that $\|g_{k,i}\| \to 0$ when $i \to \infty$ in I. Using the strong convexity of f one has $\|g_{k,i}\| \geq \tau \|x_{k,i} - x_*\|$, where x_* is the unique minimum of f. If the line search algorithm stays in Phase I, then the sequence $\{\alpha_{k,i}\}$ is increasing, bounded and thus $x_{k,i} = x_k + \alpha_{k,i} d_k$ converges to x_*. On the other hand, if the line search is entered in Phase II, the values $f(x_{k,i})$ become decreasing and by Lemma 3.4 we obtain $x_{k,i} \to x_*$ when $i \to \infty$.

Suppose now that for each iteration k the line search terminates. Proposition 3.3 implies that the Zoutendijk condition (3.1) is satisfied. By the curvature condition (1.5) one has $g_{k+1}^\top d_k \leq -\omega_2 g_k^\top d_k$. Subtracting $g_k^\top d_k$ on both sides and using the strong convexity of f, we obtain $\alpha_k \|d_k\|^2 \leq \tau^{-1}(1+\omega_2)|g_k^\top d_k|$. This inequality and the Zoutendijk condition (3.1) imply that $\alpha_k \|d_k\| \to 0$. We proceed now by contradiction by supposing that there exits $\epsilon > 0$ such that

$$\|g_k\| \geq \epsilon, \quad \text{for all } k \geq 0. \tag{3.4}$$

Assumptions 3.1 imply that there exist a positive constant c such that $\|g(x)\| \leq c$ for all $x \in \mathcal{L}$. From the definition of β_{k+1} and the Lipschitz continuity of g, we deduce that

$$\begin{aligned} |\beta_{k+1}| &\leq \|g_k\|^{-2} \|g_{k+1} - g_k\| \|g_{k+1}\| \\ &\leq Lc\epsilon^{-2} \alpha_k \|d_k\|, \end{aligned}$$

therefore $\beta_k \to 0$. By (1.2) one has $\|d_k\| \leq c + |\beta_k| \|d_{k-1}\|$, and thus the sequence $\{d_k\}$ is bounded. The Zoutendijk condition (3.1) and the boundedness of $\|d_k\|$ imply that $g_k^\top d_k \to 0$. Finally, by using the definition of d_{k+1} and the curvature condition (1.5), we obtain

$$\begin{aligned} \|g_{k+1}\|^2 &\leq |g_{k+1}^\top d_{k+1}| + |\beta_{k+1}| |g_{k+1}^\top d_k| \\ &\leq |g_{k+1}^\top d_{k+1}| + \omega_2 |\beta_{k+1}| |g_k^\top d_k|. \end{aligned}$$

It follows that $g_k \to 0$ when $k \to \infty$, a contradiction with (3.4), which implies that $\liminf \|g_k\| = 0$. By virtue of Lemma 3.4, the sequence $\{x_k\}$ converges to x_*. The convergence of the whole sequence $\{x_{k,i}\}$ to x_* is a consequence of the following inequalities:

$$\tau \|x_{k,i} - x_*\|^2 \leq f(x_{k,i}) - f(x_*) \leq f(x_k) - f(x_*).$$

The left inequality follows from the strong convexity of f and $\nabla f(x^*) = 0$, the right inequality is a property of the line search. □

The convergence result given by Polak-Ribière [22], which was obtain for exact line searches, is a consequence of Theorem 3.5.

Corollary 3.6 *Suppose that assumptions of Theorem 3.5 hold. Let α_k^* be the unique minimum of the function $\alpha \to f(x_k + \alpha d_k)$. If $\omega_1 > 0$ is sufficiently small, then conditions (1.4)-(1.6) are satisfied at $\alpha_k = \alpha_k^*$, in particular, the sequence $\{x_k\}$ defined by $x_{k+1} = x_k + \alpha_k^* d_k$, converges to the unique minimum of f.*

Proof. By definition of α_k^* one has $g(x_k + \alpha_k^* d_k)^\top d_k = 0$. Therefore, both conditions (1.5) and (1.6) are satisfied at α_k^*. It suffices to verify that the decrease condition (1.4) is satisfied at α_k^*. One has

$$-g_k^\top d_k = (g(x_k + \alpha_k^* d_k) - g(x_k))^\top d_k \leq L\alpha_k^* \|d_k\|^2.$$

By using the strong convexity of f and next by choosing $\omega_1 \leq \frac{\tau}{2L}$ on has

$$\begin{aligned} f(x_k + \alpha_k^* d_k) &\leq f(x_k) - \frac{\tau}{2}\|\alpha_k^* d_k\|^2 \\ &\leq f(x_k) + \frac{\tau}{2L}\alpha_k^* g_k^\top d_k \\ &\leq f(x_k) + \alpha_k^* \omega_1 g_k^\top d_k. \end{aligned}$$

□

4 Numerical experiments

We tested the algorithm described in this paper, which we call CGA (A stands for adapted), on some problems of the CUTEr [13] collection. The code was written in Fortran 77 with double precision on a DEC Alpha running under Compaq Tru64 UNIX V5.1. We made a comparison with the code CG+ [18] and took the same experimental framework as described in [12]. The stopping test at an iterate x was

$$\|g(x)\|_\infty < 10^{-6}(1 + |f(x)|). \tag{4.1}$$

For the line search, we used the values $\omega_1 = 10^{-4}$ and $\omega_2 = 0.1$. The first trial step was computed with the formula proposed by Shanno and Phua [26], namely $\alpha_{0,1} = 1/\|g_0\|$ and $\alpha_{k,1} = \alpha_{k-1}\|g_{k-1}\|/\|g_k\|$ for $k \geq 1$. To preserve the finite termination property of the conjugate gradient method when the function is quadratic, we proceeded as follows. Whenever the first step is acceptable for the line search, we compute the minimizer of the quadratic that interpolates $f(x_k)$, $g_k^\top d_k$ and $g_{k,1}^\top d_k$, and the one that interpolates $f(x_k)$, $g_k^\top d_k$ and $f(x_{k,1})$. If the two minimizers are nearly equal (up to 10^{-7}), a new line search iteration is performed, otherwise the conjugate gradient iterations are pursued. We found that this strategy performs better than a systematic quadratic (or cubic) interpolation at the first line search iteration. At the other line search iterations, the computations of the trial steps were done by using quadratic and cubic interpolations formulæ as described by Moré and Thuente in [20, Section 4]. The stopping criterion for the line search iterations was performed as follows. At first, condition (4.1) with $x = x_{k,i}$ is tested, if it does not hold, then the curvature condition (1.5) is checked and at last the descent condition (1.6). The runs were stopped whenever the number of function evaluations exceeds 9999. The tests with both codes were done without using a restart strategy.

The computational results are reported in Table 2. The column headings have the following meanings: `pname` is the name of the problem in CUTEr, `n` is the number of variables, `iter` is the number of conjugate gradient iterations, `nfev` is the number of function and gradient evaluations, f_* and $\|g_*\|_\infty$ are the function value and the infinity norm of the gradient at the last iteration. From the results, we see that both codes give comparable results. We observed two failures for CG+ due to a too large number of function evaluations (`nfev` > 9999). Note that, at each iteration, CG+ evaluates the function at least twice and computes a quadratic or cubic step to guarantee the quadratic termination property. If one modifies the code such that the first step can be accepted, the two problems ERRINROS and NONDQUAR are solved with success, but other failures can be observed, for example for the quadratic problems TESTQUAD and TRIDIA. This emphasizes the importance of a careful choice of the first trial step.

During our numerical tests with large problems, we did not observe a failure of CG+ because of a descent condition that would not have been satisfied. The only failure of this kind that we noticed, comes from a problem with only two variables, named HIMMELBB in CUTEr. The minimization function is of the form $f = p^2$ where $p(u, v) = uv(1-u)(1-v-u(1-u)^5)$. At the third conjugate gradient iteration, once the step length that satisfies the strong Wolfe conditions is computed, the descent condition (1.6) is tested. But at this stage, the gradient values become very small, because the second component of the variables goes to zero, the computed values of $\|g_{k+1}\|^2$ and $\beta_{k+1} g_{k+1}^\top d_k$ are of the same order, so that $g_{k+1}^\top d_{k+1}$ remain nonnegative (see Table 1). This example shows that it seems necessary to include the stopping test of the run during the line search procedure.

$\|g_{k+1}\|^2$	β_{k+1}	$g_{k+1}^\top d_k$	$g_{k+1}^\top d_{k+1}$
1.8111E−20	-5.1043E−06	-3.5783E−15	1.5405E−22
2.0874E−22	-5.4710E−07	-3.8416E−16	1.7766E−24
2.2541E−46	5.6945E−19	3.9920E−28	1.9185E−48

Table 1: Failure of the descent condition for HIMMELBB.

5 Conclusion

We have proposed a modification of the strong Wolfe line search technique to guarantee the descent condition at each iteration of the Polak-Ribière-Polyak conjugate gradient algorithm. We have shown that this technique allows to obtain the global convergence of the PRP algorithm for strongly convex function, without using the descent assumption. For nonlinear nonconvex function, we have shown that either a subsequence of the gradient values goes to zero during the line search or the Zoutendijk condition is satisfied. This result suggests to include a stopping test of the overall algorithm during the line search procedure. The numerical tests have shown that this line search technique is effective in practice.

		CG+				CGA			
pname	n	iter	nfev	f_*	$\lVert g_* \rVert_\infty$	iter	nfev	f_*	$\lVert g_* \rVert_\infty$
ARGLINA	200	1	5	2.00E+02	3.50E−14	1	4	2.00E+02	4.05E−14
ARWHEAD	500	6	17	0.00E+00	3.65E−07	6	20	0.00E+00	6.37E−07
BDQRTIC	5000	445	932	2.00E+04	1.88E−02	345	782	2.00E+04	1.49E−02
BROWNAL	200	10	58	1.53E−15	1.45E−07	7	43	3.76E−14	3.89E−07
BRYBND	5000	54	123	1.85E−13	5.58E−07	34	81	1.02E−12	9.75E−07
CHAINWOO	1000	391	816	4.57E+00	4.70E−06	406	800	2.01E+01	1.94E−05
CHNROSNB	50	253	514	3.76E−13	8.64E−07	326	630	5.45E−14	5.75E−07
COSINE	10000	5	21	-1.00E+04	4.41E−03	5	19	-1.00E+04	8.08E−03
CRAGGLVY	5000	50	122	1.69E+03	1.55E−03	48	111	1.69E+03	1.65E−03
CURLY10	1000	1800	3614	-1.00E+05	9.56E−02	1718	3400	-1.00E+05	9.91E−02
CURLY20	1000	2473	4960	-1.00E+05	9.41E−02	2458	4911	-1.00E+05	8.53E−02
CURLY30	1000	2832	5679	-1.00E+05	8.57E−02	2652	5310	-1.00E+05	9.88E−02
DIXMAANA	3000	8	27	1.00E+00	1.13E−06	6	20	1.00E+00	1.32E−06
DIXMAANB	3000	7	25	1.00E+00	3.62E−08	6	24	1.00E+00	1.28E−07
DIXMAANC	3000	7	25	1.00E+00	5.43E−07	6	24	1.00E+00	8.44E−07
DIXMAAND	3000	9	29	1.00E+00	7.39E−07	7	27	1.00E+00	1.43E−06
DIXMAANE	3000	215	436	1.00E+00	1.96E−06	210	406	1.00E+00	1.98E−06
DIXMAANF	3000	159	327	1.00E+00	1.93E−06	225	360	1.00E+00	1.97E−06
DIXMAANG	3000	153	316	1.00E+00	1.94E−06	230	364	1.00E+00	1.94E−06
DIXMAANH	3000	155	321	1.00E+00	1.86E−06	221	358	1.00E+00	1.99E−06
DIXMAANI	3000	1993	3993	1.00E+00	1.95E−06	2243	3625	1.00E+00	1.95E−06
DIXMAANJ	3000	270	550	1.00E+00	1.99E−06	278	433	1.00E+00	1.84E−06
DIXMAANK	3000	208	426	1.00E+00	1.98E−06	250	403	1.00E+00	1.89E−06
DIXMAANL	3000	216	443	1.00E+00	1.98E−06	247	388	1.00E+00	1.54E−06
DIXON3DQ	1000	1000	2005	1.62E−18	4.26E−10	1013	2031	3.52E−12	8.57E−07
DQDRTIC	5000	5	15	1.24E−15	1.41E−08	16	47	6.13E−17	6.77E−08
DQRTIC	5000	24	85	1.07E−08	9.96E−07	35	111	8.64E−07	8.75E−07
EDENSCH	2000	13	44	1.20E+04	1.01E−02	14	43	1.20E+04	8.52E−03
EIGENALS	110	1502	3009	6.34E−13	9.39E−07	726	1413	1.31E−12	9.32E−07
EIGENBLS	110	394	794	6.69E−02	9.25E−07	357	641	6.69E−02	1.04E−06
EIGENCLS	462	1867	3746	2.32E−12	9.00E−07	2078	3909	4.90E−12	8.56E−07
ENGVAL1	5000	12	36	5.55E+03	5.49E−03	11	34	5.55E+03	5.00E−03
ERRINROS	50	4995	10002	3.99E+01	2.23E−02	938	1857	3.99E+01	3.93E−05
EXTROSNB	1000	69	170	4.67E−16	5.07E−07	77	166	6.43E+01	4.61E−05
FLETCBV2	5000	3938	7877	-5.00E−01	1.48E−06	2028	2874	-5.00E−01	1.50E−06
FLETCHCR	1000	4371	8767	5.94E−13	9.88E−07	4908	7908	7.51E−13	8.77E−07
FMINSRF2	5625	330	667	1.00E+00	1.68E−06	340	518	1.00E+00	1.75E−06
FMINSURF	5625	455	917	1.00E+00	1.98E−06	581	912	1.00E+00	1.92E−06
FREUROTH	5000	12	38	6.08E+05	3.90E−01	12	36	6.08E+05	5.50E−01
GENROSE	500	1113	2252	1.00E+00	1.80E−06	1177	2100	1.00E+00	1.22E−06
HILBERTB	50	6	14	1.00E−13	6.93E−07	6	20	1.39E−16	1.89E−08
LIARWHD	5000	16	43	1.33E−20	3.21E−08	18	50	1.30E−17	6.18E−07
MOREBV	5000	161	323	1.09E−10	9.94E−07	161	325	1.09E−10	9.94E−07
MSQRTALS	100	324	652	7.70E−12	8.47E−07	373	680	1.10E−11	9.91E−07
MSQRTBLS	100	338	680	2.64E−12	8.77E−07	383	695	7.98E−12	8.69E−07
NONCVXU2	5000	1395	2799	1.16E+04	1.15E−02	1644	2408	1.16E+04	1.05E−02
NONCVXUN	5000	2270	4550	1.16E+04	1.11E−02	2259	3371	1.16E+04	1.13E−02
NONDIA	5000	5	26	1.44E−17	6.58E−08	8	38	1.02E−19	9.52E−09
NONDQUAR	5000	4982	10001	9.66E−07	1.48E−05	2533	5051	3.24E−06	9.00E−07
PENALTY1	1000	40	164	9.69E−03	4.35E−07	42	147	9.69E−03	6.62E−07
POWELLSG	5000	154	358	8.99E−07	9.48E−07	105	237	2.32E−08	2.53E−07
POWER	10000	355	719	1.39E−09	9.68E−07	462	735	1.55E−09	9.55E−07
QUARTC	5000	24	85	1.07E−08	9.96E−07	35	111	8.64E−07	8.75E−07
SCHMVETT	5000	15	39	-1.50E+04	1.12E−02	14	33	-1.50E+04	1.50E−02
SENSORS	100	19	46	-2.11E+03	2.00E−03	18	44	-2.11E+03	1.20E−03
SINQUAD	5000	30	80	-6.76E+06	6.19E−01	26	70	-6.76E+06	8.61E−01
SPARSQUR	10000	35	111	6.77E−10	4.50E−07	34	99	1.79E−09	7.04E−07
SPMSRTLS	4999	212	430	2.85E−11	9.32E−07	224	410	2.92E−11	9.65E−07
SROSENBR	5000	10	28	1.22E−14	8.75E−08	10	33	3.86E−17	1.54E−09
TESTQUAD	5000	1652	3307	4.84E−13	8.08E−07	2137	4276	1.19E−13	9.61E−07
TOINTGOR	50	73	148	1.37E+03	1.30E−03	80	148	1.37E+03	1.02E−03
TOINTGSS	5000	1	7	1.00E+01	7.19E−07	1	7	1.00E+01	3.55E−15
TOINTPSP	50	87	235	2.26E+02	1.80E−04	90	217	2.26E+02	1.84E−04
TOINTQOR	50	20	42	1.18E+03	6.45E−04	22	45	1.18E+03	7.74E−04
TQUARTIC	5000	13	41	3.40E−17	3.09E−07	13	41	6.48E−21	2.50E−09
TRIDIA	5000	781	1565	4.72E−15	9.88E−07	783	1568	6.59E−15	9.98E−07
VARDIM	200	9	44	3.52E−24	7.50E−10	7	46	1.95E−23	1.75E−09
VAREIGVL	50	128	261	7.28E−12	8.74E−07	131	234	1.29E−11	9.09E−07
WOODS	4000	124	315	5.91E−13	5.92E−07	19	46	7.88E+03	3.82E−03
TOTAL		45099	91316			37884	68328		

Table 2: Test results for CG+/CGA.

Acknowledgments

This paper is based on an earlier work presented at the ISMP at Lausanne in 1997. We would like to thank Jean-Charles Gilbert who brought us this subject and Mehi Albaali for some recent discussions, which motivated us to write this report. Special thanks to Guanghui Liu, Jorge Nocedal and Richard Waltz for the availability of their software CG+ and to Dominique Orban and Philippe Toint for their help to a correct installation of CUTEr.

References

[1] M. Al-Baali and R. Fletcher, *An efficient line search for nonlinear least squares*, Journal of Optimization Theory and Applications, 48 (1986), pp. 359–377.

[2] P. Armand and J. C. Gilbert, *A piecewise line-search technique for maintaining the positive definiteness of the matrices in the SQP method*, Comput. Optim. Appl., 16 (2000), pp. 121–158.

[3] Y. Dai, J. Han, G. Liu, D. Sun, H. Yin, and Y.-X. Yuan, *Convergence properties of nonlinear conjugate gradient methods*, SIAM J. Optim., 10 (2000), pp. 345–358 (electronic).

[4] Y. H. Dai and Y. Yuan, *Convergence properties of the Fletcher-Reeves method*, IMA J. Numer. Anal., 16 (1996), pp. 155–164.

[5] ———, *A nonlinear conjugate gradient method with a strong global convergence property*, SIAM J. Optim., 10 (1999), pp. 177–182 (electronic).

[6] Y.-h. Dai and Y.-x. Yuan, *A note on the nonlinear conjugate gradient method*, J. Comput. Math., 20 (2002), pp. 575–582.

[7] R. Fletcher, *Practical methods of optimization*, John Wiley & Sons Ltd., Chichester, second ed., 1987.

[8] R. Fletcher and C. M. Reeves, *Function minimization by conjugate gradients*, Comput. J., 7 (1964), pp. 149–154.

[9] J. C. Gilbert, *Maintaining the positive definiteness of the matrices in reduced secant methods for equality constrained optimization*, Mathematical Programming, 50 (1991), pp. 1–28.

[10] ———, *On the realization of the Wolfe conditions in reduced quasi-Newton methods for equality constrained optimization*, SIAM Journal on Optimization, 7 (1997), pp. 780–813.

[11] ———, *Piecewise line-search techniques for constrained minimization by quasi-Newton algorithms*, in Advances in Nonlinear Programming, Y. xiang Yuan, ed., Kluwer, 1998, pp. 73–103.

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

[13] N. I. M. Gould, D. Orban, and P. L. Toint, *CUTEr and SifDec: A constrained and unconstrained testing environment, revisited*, ACM Transactions on Mathematical Software, 29 (2003), pp. 373–394.

[14] L. Grippo and S. Lucidi, *A globally convergent version of the Polak-Ribière conjugate gradient method*, Mathematical Programming, 78 (1997), pp. 375–391.

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

[16] C. Lemaréchal, *A view on line searches*, in Optimization and Optimal Control, A. Auslender, W. Oettly, and J. Stoer, eds., vol. 30 of Lectures Notes in Control and Information Sciences, Berlin, 1981, Springer Verlag, pp. 59–78.

[17] Z. Li, J. Chen, and N. Deng, *A new conjugate gradient method and its global convergence properties*, Systems Sci. Math. Sci., 11 (1998), pp. 53–60.

[18] G. Liu, J. Nocedal, and R. Waltz, *CG+, version 1.1.* Optimization Technology Center, Argonne National Laboratory and Northwestern University, http://www.ece.northwestern.edu/˜nocedal/CG+.html, April 2000.

[19] J. J. Moré and D. C. Sorensen, *Newton's method*, in Studies in numerical analysis, vol. 24 of MAA Stud. Math., Math. Assoc. America, Washington, DC, 1984, pp. 29–82.

[20] J. J. Moré and D. J. Thuente, *Line search algorithms with guaranteed sufficient decrease*, ACM Transactions on Mathematical Software, 20 (1994), pp. 286–307.

[21] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer-Verlag, New York, 1999.

[22] E. Polak and G. Ribière, *Note sur la convergence de méthodes de directions conjuguées*, Revue Française d'Informatique et de Recherche Opérationnelle, 16 (1969), pp. 35–43.

[23] B. T. Polyak, *The conjugate gradient method in extremal problems*, USRR Comp. Maths. and Math. Phys., 9 (1969), pp. 94–112.

[24] M. J. D. Powell, *Nonconvex minimization calculations and the conjugate gradient method*, in Numerical analysis (Dundee, 1983), vol. 1066 of Lecture Notes in Math., Springer, Berlin, 1984, pp. 122–141.

[25] D. F. Shanno and K. H. Phua, *Algorithm 500: Minimization of unconstrained multivariate functions*, ACM Transactions on Mathematical Software, 2 (1976), pp. 87–94.

[26] ——, *Remark on algorithm 500: Minimization of unconstrained multivariate functions*, ACM Transactions on Mathematical Software, 6 (1980), pp. 618–622.

[27] P. Wolfe, *Convergence conditions for ascent methods*, 11 (1969), pp. 226–235.

[28] ——, *Convergence conditions for ascent methods II: some corrections*, 13 (1971), pp. 185–188.

[29] G. Zoutendijk, *Nonlinear programming, computational methods*, in Integer and Nonlinear Programming, J. Abadie, ed., North-Holland, Amsterdam, 1970, pp. 37–86.