SIAM J. OPTIM.
Vol. 10, No. 4, pp. 1196-1211

NONMONOTONE SPECTRAL PROJECTED GRADIENT METHODS ON CONVEX SETS*

ERNESTO G. BIRGIN†, JOSÉ MARIO MARTÍNEZ†, AND MARCOS RAYDAN‡

Abstract. Nonmonotone projected gradient techniques are considered for the minimization of differentiable functions on closed convex sets. The classical projected gradient schemes are extended to include a nonmonotone steplength strategy that is based on the Grippo–Lampariello–Lucidi nonmonotone line search. In particular, the nonmonotone strategy is combined with the spectral gradient choice of steplength to accelerate the convergence process. In addition to the classical projected gradient nonlinear path, the feasible spectral projected gradient is used as a search direction to avoid additional trial projections during the one-dimensional search process. Convergence properties and extensive numerical results are presented.

Key words. projected gradients, nonmonotone line search, large-scale problems, bound constrained problems, spectral gradient method

AMS subject classifications. 49M07, 49M10, 65K, 90C06, 90C20

PII. S1052623497330963

1. Introduction.

We consider the projected gradient method for the minimization of differentiable functions on nonempty closed and convex sets. Over the last few decades, there have been many different variations of the projected gradient method that can be viewed as the constrained extensions of the optimal gradient method for unconstrained minimization. They all have the common property of maintaining feasibility of the iterates by frequently projecting trial steps on the feasible convex set. This process is in general the most expensive part of any projected gradient method. Moreover, even if projecting is inexpensive, as in the box-constrained case, the method is considered to be very slow, as is its analogue, the optimal gradient method (also known as steepest descent), for unconstrained optimization. On the positive side, the projected gradient method is quite simple to implement and very effective for large-scale problems.

This state of affairs motivates us to combine the projected gradient method with two recently developed ingredients in optimization. First we extend the typical globalization strategies associated with these methods to the nonmonotone line search schemes developed by Grippo, Lampariello, and Lucidi [17] for Newton's method. Second, we propose to associate the spectral steplength, introduced by Barzilai and Borwein [1] and analyzed by Raydan [26]. This choice of steplength requires little computational work and greatly speeds up the convergence of gradient methods. In fact, while the spectral gradient method appears to be a generalized steepest descent method, it is clear from its derivation that it is related to the quasi-Newton family of methods through an approximated secant equation. The fundamental difference is

*Received by the editors December 3, 1997; accepted for publication (in revised form) June 15, 1999; published electronically June 13, 2000.

http://www.siam.org/journals/siopt/10-4/33096.html

†Departamento de Matemática Aplicada, IMECC-UNICAMP, CP 6065, 13081-970 Campinas SP, Brazil (ernesto@ime.unicamp.br, martinez@ime.unicamp.br). The research of the first author was sponsored by FAPESP (grant 95/2452-6). The research of the second author was sponsored by FAPESP (grant 90/3724-6), CNPq, and FAEP-UNICAMP.

‡Departamento de Computación, Facultad de Ciencias, Universidad Central de Venezuela, Ap. 47002, Caracas 1041-A, Venezuela (mraydan@reacciun.ve). This author's research was sponsored by FAPESP, FAEP-UNICAMP, and CDCH-UCV.

that it is a two-point method while the steepest descent method is not. The main idea behind the spectral choice of steplength is that the steepest descent method is very slow but it can be accelerated by taking, instead of the stepsize that comes from the minimization of the function along the gradient of the current iteration, the one that comes from the one-dimensional minimization at the previous step. See Glunt, Hayden, and Raydan [15] for a relationship with the shifted power method to approximate eigenvalues and eigenvectors and also for an interesting chemistry application. See also Raydan [27] for a combination of the spectral choice of steplength with nonmonotone line search techniques to solve unconstrained minimization problems. A successful application of this technique can be found in [5].

Therefore, it is natural and rather easy to transport the spectral gradient idea with a nonmonotone line search to the projected gradient case in order to speed up the convergence of the projected gradient method. In particular, in this work we extend the practical version of Bertsekas [2] that enforces an Armijo-type condition along the curvilinear projection path. This practical version is based on the original version proposed by Goldstein [16] and Levitin and Polyak [19]. We also apply the new ingredients to the feasible continuous projected path that will be properly defined in section 2.

The convergence properties of the projected gradient method for different choices of stepsize have been extensively studied. See, e.g., [2, 3, 7, 11, 16, 19, 22, 30]. For an interesting review of the different convergence results that have been obtained under different assumptions, see Calamai and Moré [7]. For a complete survey see Dunn [12].

In section 2 of this paper we define the spectral projected gradient algorithms and prove global convergence results. In section 3 we present numerical experiments. This set of experiments shows that, in fact, the spectral choice of the steplength represents considerable progress in relation to constant choices and that the nonmonotone framework is useful. Some final remarks are presented in section 4. In particular, we elaborate on the relationship between the spectral gradient method and the quasi-Newton family of methods.

2. Nonmonotone gradient-projection algorithms.

The nonmonotone spectral gradient-projection algorithms introduced in this section apply to problems of the form

$$\text{minimize } f(x) \quad \text{subject to} \quad x \in \Omega,$$

where Ω is a closed convex set in $\mathbb{R}^n$. Throughout this paper we assume that f is defined and has continuous partial derivatives on an open set that contains Ω. Throughout this work $\|\cdot\|$ denotes the 2-norm of vectors and matrices, although in some cases it can be replaced by an arbitrary norm.

Given $z \in \mathbb{R}^n$ we define $P(z)$ as the orthogonal projection on Ω. We denote $g(x) = \nabla f(x)$. The algorithms start with $x_0 \in \Omega$ and use an integer $M \geq 1$, a small parameter $\alpha_{\min} > 0$, a large parameter $\alpha_{\max} > \alpha_{\min}$, a sufficient decrease parameter $\gamma \in (0, 1)$, and safeguarding parameters $0 < \sigma_1 < \sigma_2 < 1$. Initially, $\alpha_0 \in [\alpha_{\min}, \alpha_{\max}]$ is arbitrary. Given $x_k \in \Omega$ and $\alpha_k \in [\alpha_{\min}, \alpha_{\max}]$, Algorithms 2.1 and 2.2 describe how to obtain x_{k+1} and α_{k+1} and when to terminate the process.

ALGORITHM 2.1.

Step 1. *Detect whether the current point is stationary*

If $\|P(x_k - g(x_k)) - x_k\| = 0$, stop, declaring that x_k is stationary.

Step 2. *Backtracking*

Step 2.1. Set $\lambda \leftarrow \alpha_k$.

Step 2.2. Set $x_+ = P(x_k - \lambda g(x_k))$.
Step 2.3. If

$$f(x_+) \leq \max_{0 \leq j \leq \min\{k, M-1\}} f(x_{k-j}) + \gamma \langle x_+ - x_k, g(x_k) \rangle, \tag{1}$$

then define $\lambda_k = \lambda$, $x_{k+1} = x_+$, $s_k = x_{k+1} - x_k$, $y_k = g(x_{k+1}) - g(x_k)$, and go to Step 3.

If (1) does not hold, define

$$\lambda_{new} \in [\sigma_1 \lambda, \sigma_2 \lambda], \tag{2}$$

set $\lambda \leftarrow \lambda_{new}$, and go to Step 2.2.
Step 3.

Compute $b_k = \langle s_k, y_k \rangle$.

If $b_k \leq 0$, set $\alpha_{k+1} = \alpha_{\max}$; else, compute $a_k = \langle s_k, s_k \rangle$ and

$$\alpha_{k+1} = \min\{\alpha_{\max}, \max\{\alpha_{\min}, a_k/b_k\}\}.$$

The one-dimensional search procedure of Algorithm 2.1 (called SPG1 from now on) takes into account points of the form $P(x_k - \lambda g(x_k))$ for $\lambda \in (0, \alpha_k]$, which, in general, form a curvilinear path (piecewise linear if Ω is a polyhedral set). For this reason, the scalar product $\langle x_+ - x_k, g(x_k) \rangle$ in the nonmonotone Armijo condition (1) must be computed for each trial point x_+. Moreover, in the SPG1 formulation the distance between two consecutive trial points could be very small or even null in the vicinity of corner points of the set Ω. In fact the distance between the projections of two points on the feasible set can be small, even if the points are distant from each other. Clearly, to evaluate the objective function on two close points represents a bad use of available information. Of course, proximity of two consecutive trial points can be computationally detected at the expense of $O(n)$ operations or comparisons.

These observations motivated us to define Algorithm 2.2. This algorithm is also based on the spectral projected gradient direction $P(x_k - \alpha_k g(x_k)) - x_k$, with α_k as the safeguarded "inverse Rayleigh quotient" $\frac{\langle s_{k-1}, s_{k-1} \rangle}{\langle s_{k-1}, y_{k-1} \rangle}$. (Observe that $\frac{\langle s_{k-1}, y_{k-1} \rangle}{\langle s_{k-1}, s_{k-1} \rangle}$ is in fact a Rayleigh quotient corresponding to the average Hessian matrix $\int_0^1 \nabla^2 f(x_{k-1} + t s_{k-1}) dt$.) However, in the case of rejection of the first trial point, the next ones are computed along the same direction. As a consequence, $\langle x_+ - x_k, g(x_k) \rangle$ must be computed only at the first trial and the projection operation must be performed only once per iteration. So, Algorithm 2.2, which will be called SPG2 in the rest of the paper, coincides with SPG1 except at the backtracking step, whose description is given below.

Algorithm 2.2.
Step 2 (*Backtracking*)
Step 2.1. Compute $d_k = P(x_k - \alpha_k g(x_k)) - x_k$. Set $\lambda \leftarrow 1$.
Step 2.2. Set $x_+ = x_k + \lambda d_k$.
Step 2.3. If

$$f(x_+) \leq \max_{0 \leq j \leq \min\{k, M-1\}} f(x_{k-j}) + \gamma \lambda \langle d_k, g(x_k) \rangle, \tag{3}$$

then define $\lambda_k = \lambda$, $x_{k+1} = x_+$, $s_k = x_{k+1} - x_k$, $y_k = g(x_{k+1}) - g(x_k)$, and go to Step 3.

If (3) does not hold, define λ_{new} as in (2), set $\lambda \leftarrow \lambda_{new}$, and go to Step 2.2.

In both algorithms the computation of λ_{new} uses one-dimensional quadratic interpolation and it is safeguarded taking $\lambda \leftarrow \lambda/2$ when the minimum of the one-dimensional quadratic lies outside $[0.1, 0.9\lambda]$. Notice also that the line search conditions (1) and (3) guarantee that the sequence $\{x_k\}$ remains in $\Omega_0 \equiv \{x \in \Omega : f(x) \leq f(x_0)\}$.

It will be useful in our theoretical analysis to define the *scaled projected gradient* $g_t(x)$ as

$$g_t(x) = [P(x - tg(x)) - x]$$

for all $x \in \Omega$, $t > 0$. If x is an iterate of SPG1 or SPG2 and $t = \alpha_k$ the scaled projected gradient is the *spectral projected gradient* (SPG) that gives the name to our methods. If $t = 1$, the scaled projected gradient is the *continuous projected gradient* whose ∞-norm $\|g_1(x)\|_\infty$ is used for the termination criterion of the algorithms. In fact, the annihilation of $g_t(x)$ is equivalent to the satisfaction of first-order stationary conditions. This property is stated in the following lemma, whose proof is a straightforward consequence of the convexity of Ω.

LEMMA 2.1. *For all $x \in \Omega$, $t \in (0, \alpha_{\max}]$,*

(i) $\langle g(x), g_t(x)\rangle \leq -\frac{1}{t}\|g_t(x)\|_2^2 \leq -\frac{1}{\alpha_{\max}}\|g_t(x)\|_2^2$.

(ii) *The vector $g_t(\bar{x})$ vanishes if and only if $\bar{x}$ is a constrained stationary point.*

Now, let us prove that both algorithms are well defined and have the property that every accumulation point $\bar{x}$ is a constrained stationary point, *i.e.*, that

$$\langle g(\bar{x}), x - \bar{x}\rangle \geq 0 \quad \text{for all} \quad x \in \Omega.$$

The proof of our first theorem relies on Proposition 2.3.3 in Bertsekas [3], which is related to the Armijo condition along the projection arc. This proposition was originally shown in [14]. For completeness we include in the next lemma some technical results from [3] that will be used in our proof.

LEMMA 2.2. (i) *For all $x \in \Omega$ and $z \in \mathbb{R}^n$, the function $h : [0, \infty) \to \mathbb{R}$ given by*

$$h(s) = \frac{\|P(x + sz) - x\|}{s} \quad \textit{for all } s > 0$$

is monotonically nonincreasing.

(ii) *For all $x \in \Omega$ there exists $s_x > 0$ such that for all $t \in [0, s_x]$ it holds that*

$$f(P(x - tg(x))) - f(x) \leq \gamma\langle g(x), g_t(x)\rangle.$$

Proof. See Lemma 2.3.1 and Theorem 2.3.3 (part (a)) in [3].

THEOREM 2.3. *Algorithm SPG1 is well defined, and any accumulation point of the sequence $\{x_k\}$ that it generates is a constrained stationary point.*

Proof. From Lemma 2.2(ii), we have for all $\lambda \in [0, \min\{s_{x_k}, \alpha_{\min}\}]$ that

$$\begin{aligned} f(P(x_k - \lambda g(x_k))) - \max_{0\leq j\leq M-1} f(x_{k-j}) &\leq f(P(x_k - \lambda g(x_k))) - f(x_k) \\ &\leq \gamma\langle g(x_k), g_\lambda(x_k)\rangle. \end{aligned}$$

Therefore, a stepsize satisfying (1) will be found after a finite number of trials, and Algorithm SPG1 is well defined.

Let $\bar{x} \in \Omega$ be an accumulation point of $\{x_k\}$, and relabel $\{x_k\}$ a subsequence converging to $\bar{x}$. We consider two cases.

Case 1. If $\inf \lambda_k = 0$, then there exists a subsequence $\{x_k\}_K$ such that

$$\lim_{k \in K} \lambda_k = 0.$$

In that case, from the way λ_k is chosen in (1), there exists an index $\bar{k}$ sufficiently large such that for all $k \geq \bar{k}$, $k \in K$, there exists ρ_k, $0 < \sigma_1 \leq \rho_k \leq \sigma_2$, for which $\psi_k \equiv \lambda_k/\rho_k > 0$ fails to satisfy condition (1), i.e.,

$$f(P(x_k - \psi_k g(x_k))) > \max_{0 \leq j \leq M-1} f(x_{k-j}) + \gamma \langle g(x_k), P(x_k - \psi_k g(x_k)) - x_k \rangle$$

$$\geq f(x_k) + \gamma \langle g(x_k), P(x_k - \psi_k g(x_k)) - x_k \rangle.$$

Therefore, it follows that

$$f(P(x_k - \psi_k g(x_k))) - f(x_k) > \gamma \langle g(x_k), g_{\psi_k}(x_k) \rangle. \tag{4}$$

By the mean value theorem we obtain

$$f(P(x_k - \psi_k g(x_k))) - f(x_k) = \langle g(x_k), g_{\psi_k}(x_k) \rangle + \langle g(\xi_k) - g(x_k), g_{\psi_k}(x_k) \rangle, \tag{5}$$

where ξ_k lies along the line segment connecting x_k and $P(x_k - \psi_k g(x_k))$.

Combining (4) and (5) we obtain for all $k \in K$ sufficiently large that

$$(1 - \gamma) \langle g(x_k), g_{\psi_k}(x_k) \rangle > \langle g(x_k) - g(\xi_k), g_{\psi_k}(x_k) \rangle. \tag{6}$$

Using Lemmas 2.1 and 2.2, we have

$$\langle g(x_k), g_{\psi_k}(x_k) \rangle \leq -\frac{1}{\psi_k} \|g_{\psi_k}(x_k)\|_2^2 \leq -\frac{1}{\alpha_k} \|g_{\alpha_k}(x_k)\|_2 \, \|g_{\psi_k}(x_k)\|_2, \tag{7}$$

where α_k is the initial stepsize at iteration k. Combining (6) and (7) and using the Schwartz inequality, we obtain for $k \in K$ sufficiently large

$$\frac{(1-\gamma)}{\alpha_k} \|g_{\alpha_k}(x_k)\|_2 \, \|g_{\psi_k}(x_k)\|_2 < \langle g(\xi_k) - g(x_k), g_{\psi_k}(x_k) \rangle$$

$$\leq \|g(\xi_k) - g(x_k)\|_2 \, \|g_{\psi_k}(x_k)\|_2.$$

Using that $\|g_{\psi_k}(x_k)\|_2 \neq 0$, we have

$$\frac{(1-\gamma)}{\alpha_k} \|g_{\alpha_k}(x_k)\|_2 < \|g(\xi_k) - g(x_k)\|_2. \tag{8}$$

Since $\psi_k \to 0$ and $x_k \to \bar{x}$ as $k \to \infty$, $k \in K$, then $\xi_k \to \bar{x}$ as $k \to \infty$, $k \in K$. Taking a convenient subsequence $\bar{K} \subseteq K$ such that $\{\alpha_k\}$ is convergent to $\bar{\alpha} \in [\alpha_{\min}, \alpha_{\max}]$, and taking limits in (8) as $k \to \infty$, $k \in \bar{K}$, we deduce that

$$\|g_{\bar{\alpha}}(\bar{x})\|_2 \leq 0.$$

Therefore, $g_{\bar{\alpha}}(\bar{x}) = 0$, and $\bar{x}$ is a constrained stationary point.

Case 2. Assume that $\inf \lambda_k \geq \rho > 0$. Let us suppose by way of contradiction that $\bar{x}$ is not a constrained stationary point. Therefore $\|g_\lambda(\bar{x})\| > 0$ for all $\lambda \in (0, \alpha_{\max}]$.

By continuity and compactness, there exists $\delta > 0$ such that $\|g_\lambda(\bar{x})\| \geq \delta > 0$ for all $\lambda \in [\rho, \alpha_{\max}]$. Using the first part of the proof of the theorem in [17, p. 709], we obtain a monotonically nonincreasing sequence $\{f(x_{l(k)})\}$. Indeed, let $l(k)$ be an integer such that $k - \min\{k, M-1\} \leq l(k) \leq k$ and

$$f(x_{l(k)}) = \max_{0 \leq j \leq \min\{k, M-1\}} f(x_{k-j}).$$

From (1) it follows that, for $k > M - 1$ (see [17] for details),

$$f(x_{l(k)}) \leq f(x_{l(l(k)-1)}) + \gamma \langle g(x_{l(k)-1}), g_{\lambda_{l(k)-1}}(x_{l(k)-1}) \rangle.$$

By continuity, for $k \geq \bar{k}$ sufficiently large, $\|g_\lambda(\bar{x}_k)\| \geq \delta/2$. Hence, using Lemma 2.1, we obtain

$$f(x_{l(k)}) \leq f(x_{l(l(k)-1)}) - \frac{\gamma}{\alpha_{\max}} \|g_{\lambda_{l(k)-1}}(x_{l(k)-1})\|_2^2 \leq f(x_{l(l(k)-1)}) - \frac{\gamma \delta^2}{4\alpha_{\max}}.$$

When $k \to \infty$, clearly $f(x_{l(k)}) \to -\infty$, which is a contradiction. In fact, f is a continuous function and so $f(x_k)$ converges to $f(\bar{x})$. ∎

THEOREM 2.4. *Algorithm SPG2 is well defined, and any accumulation point of the sequence $\{x_k\}$ that it generates is a constrained stationary point.*

Proof. If x_k is not a constrained stationary point, then by Lemma 2.1

$$\langle g(x_k), d_k \rangle = \langle g(x_k), g_{\alpha_k}(x_k) \rangle \leq -\frac{1}{\alpha_{\max}} \|g_{\alpha_k}(x_k)\|_2^2 < 0,$$

and the search direction is a descent direction. Hence, a stepsize satisfying (3) will be found after a finite number of trials, and Algorithm SPG2 is well defined.

Let $\bar{x} \in \Omega$ be an accumulation point of $\{x_k\}$, and relabel $\{x_k\}$ a subsequence converging to $\bar{x}$. We consider two cases.

Case 1. Assume that $\inf \lambda_k = 0$. Suppose, by contradiction, that $\bar{x}$ is not stationary. By continuity and compactness, there exists $\delta > 0$ such that

$$\left\langle g(\bar{x}), \frac{P(\bar{x} - \alpha g(\bar{x})) - \bar{x}}{\|P(\bar{x} - \alpha g(\bar{x})) - \bar{x}\|} \right\rangle < -\delta \quad \text{for all} \quad \alpha \in [\alpha_{\min}, \alpha_{\max}].$$

This implies that

$$\left\langle g(x_k), \frac{P(x_k - \alpha g(x_k)) - x_k}{\|P(x_k - \alpha g(x_k)) - x_k\|} \right\rangle < -\delta/2 \quad \text{for all} \quad \alpha \in [\alpha_{\min}, \alpha_{\max}] \tag{9}$$

and k large enough on the subsequence that converges to $\bar{x}$.

Since $\inf \lambda_k = 0$, there exists a subsequence $\{x_k\}_K$ such that

$$\lim_{k \in K} \lambda_k = 0.$$

In that case, from the way λ_k is chosen in (3), there exists an index $\bar{k}$ sufficiently large such that for all $k \geq \bar{k}$, $k \in K$, there exists ρ_k, $0 < \sigma_1 \leq \rho_k \leq \sigma_2$, for which $\lambda_k/\rho_k > 0$ fails to satisfy condition (3); i.e.,

$$f\left(x_k + \frac{\lambda_k}{\rho_k} d_k\right) > \max_{0 \leq j \leq M-1} f(x_{k-j}) + \gamma \frac{\lambda_k}{\rho_k} \langle g(x_k), d_k \rangle \geq f(x_k) + \gamma \frac{\lambda_k}{\rho_k} \langle g(x_k), d_k \rangle.$$

Hence,

$$\frac{f(x_k + \frac{\lambda_k}{\rho_k} d_k) - f(x_k)}{\lambda_k/\rho_k} > \gamma \langle g(x_k), d_k \rangle.$$

By the mean value theorem, this relation can be written as

$$\langle g(x_k + t_k d_k), d_k \rangle > \gamma \langle g(x_k), d_k \rangle \quad \text{for all } k \in K,\ k \geq \bar{k}, \tag{10}$$

where t_k is a scalar in the interval $[0, \lambda_k/\rho_k]$ that goes to zero as $k \in K$ goes to infinity.

Taking a convenient subsequence such that $d_k/\|d_k\|$ is convergent to d, and taking limits in (10), we deduce that $(1-\gamma)\langle g(\bar{x}), d \rangle \geq 0$. (In fact, observe that $\{\|d_k\|\}_K$ is bounded and so $t_k\|d_k\| \to 0$.) Since $(1-\gamma) > 0$ and $\langle g(x_k), d_k \rangle < 0$ for all k, then $\langle g(\bar{x}), d \rangle = 0$.

By continuity and the definition of d_k this implies that for k large enough on that subsequence we have that

$$\left\langle g(x_k), \frac{P(x_k - \alpha_k g(x_k)) - x_k}{\|P(x_k - \alpha_k g(x_k)) - x_k\|} \right\rangle > -\delta/2,$$

which contradicts (9).

Case 2. Assume that $\inf \lambda_k \geq \rho > 0$. Let us suppose by way of contradiction that $\bar{x}$ is not a constrained stationary point. Therefore $\|g_\lambda(\bar{x})\| > 0$ for all $\lambda \in (0, \alpha_{\max}]$. By continuity and compactness, there exists $\delta > 0$ such that $\|g_\lambda(\bar{x})\| \geq \delta > 0$ for all $\lambda \in [\rho, \alpha_{\max}]$.

As in the proof of the second case of Theorem 2.3,

$$f(x_{l(k)}) = \max_{0 \leq j \leq \min\{k, M-1\}} f(x_{k-j})$$

is a monotonically nonincreasing sequence. From (3) it follows that, for $k > M-1$,

$$f(x_{l(k)}) \leq f(x_{l(l(k)-1)}) + \gamma \lambda_{l(k)-1} \langle g(x_{l(k)-1}), g_{\alpha_{l(k)-1}}(x_{l(k)-1}) \rangle.$$

By continuity, for $k \geq \bar{k}$ sufficiently large, $\|g_{\alpha_k}(\bar{x_k})\| \geq \delta/2$. Hence, using Lemma 2.1, we obtain

$$f(x_{l(k)}) \leq f(x_{l(l(k)-1)}) - \frac{\gamma\,\rho}{\alpha_{\max}} \|g_{\alpha_{l(k)-1}}(x_{l(k)-1})\|_2^2 \leq f(x_{l(l(k)-1)}) - \frac{\gamma \delta^2 \rho}{4\alpha_{\max}}.$$

When $k \to \infty$, clearly $f(x_{l(k)}) \to -\infty$, which is a contradiction. In fact, f is a continuous function and so $f(x_k)$ converges to $f(\bar{x})$. ☐

3. Numerical results. The algorithms SPG1 and SPG2 introduced in the previous section compute at least one projection on the feasible set Ω per iteration. Therefore, these algorithms are especially interesting in the case in which this projection is easy to compute. An important situation in which the projection is trivial is when Ω is an n-dimensional box, possibly with some infinite bounds. In fact, good algorithms for box constrained minimization are the essential tool for the development of efficient augmented Lagrangian methods for general nonlinear programming (see [8, 10, 13]). With this in mind, we implemented the spectral projected gradient algorithms for the case in which Ω is described by bounds on the variables. In order to assess the reliability of SPG algorithms, we tested them against the well-known

Table 1
Problem sets according to the CUTE classification.

Set #	Objective type	Problem interest
1	other	academic
2	other	modeling
3	other	real application
4	sum of squares	academic
5	sum of squares	modeling
6	quadratic	academic
7	quadratic	modeling
8	quadratic	real application

package LANCELOT [9] using *all* the bound constrained problems with more than 50 variables from the CUTE [10] collection. Only problem GRIDGENA was excluded from our tables because it gives an "exception error" when evaluated at some point by SPG algorithms. For all the problems with variable dimension, we used the largest dimension that is admissible without modification of the internal variables of the "double large" installation of CUTE.

Altogether, we solved 50 problems. The horizontal lines in Tables 2–5 divide the CUTE problems into 8 classes according to objective function type (quadratic, sum of squares, other) and problem interest (academic, modeling, real application). All problems are bound constrained only, twice continuously differentiable, and with more than 50 variables. The 8 sets, in the order in which they appear in the tables, are described in Table 1.

In the numerical experiments we used the default options for LANCELOT, i.e.,

- `exact-second-derivatives-used`,
- `bandsolver-preconditioned-cg-solver-used 5`,
- `exact-cauchy-point-required`,
- `infinity-norm-trust-region-used`,
- `gradient-tolerance 1.0D-05`.

We are deeply concerned with the reproducibility of the numerical results presented in this paper. For this reason, all the used codes are available by e-mail request to any of the authors, who are also available to discuss computational details.

All the experiments were run in a SPARCstation Sun Ultra 1, with an UltraSPARC 64-bit processor, 167 MHz clock and 128 MBytes of RAM memory. SPG codes are in Fortran77 and were compiled with the optimization compiler option -O4.

For the SPG methods we used $\gamma = 10^{-4}$, $\alpha_{min} = 10^{-30}$, $\alpha_{max} = 10^{30}$, $\sigma_1 = 0.1$, $\sigma_2 = 0.9$, and $\alpha_0 = 1/\|g_1(x_0)\|_\infty$. After running a few problems with $M \in \{5, 10, 15\}$, we decided to use $M = 10$, as the tests did not show meaningful differences. To decide when to stop the execution of the algorithms declaring convergence we used the criterion $\|g_1(x_k)\|_\infty \leq 10^{-5}$. We also stopped the execution of SPG when 50,000 iterations or 200,000 function evaluations were completed without achieving convergence.

To complete the numerical insight into the behavior of SPG methods, we also ran the projected gradient algorithm (PGA), which turns out to be identical to SPG1, with the initial choice of steplength $\alpha_k \equiv 1$. In this case we implemented both the monotone version of PGA, which corresponds to $M = 1$, and the nonmonotone one with $M = 10$. The convergence of the nonmonotone version is a particular case of our Theorem 2.3. The performance of the nonmonotone version of PGA, which is more efficient than the monotone version, is reported in Table 2.

Table 2
Performance of nonmonotone ($M = 10$) projected gradient.

Problem	n	IT	FE	GE	Time	$f(x)$	$\|g_1(x)\|_\infty$
BDEXP	5000	13065	13066	13066	459.99	3.464D−03	9.999D−06
EXPLIN	120	30608	200001	30609	15.08	−7.238D+05	7.768D−05
EXPLIN2	120	19581	126328	19582	9.87	−7.245D+05	8.192D−06
EXPQUAD	120	7899	200001	7900	22.06	−3.626D+06	3.875D−03
MCCORMCK	10000	16080	47939	16081	2755.50	−9.133D+03	2.485D−09
PROBPENL	500	888	10249	889	11.39	3.992D−07	7.265D−06
QRTQUAD	120	3464	38175	3465	3.76	−3.625D+06	5.303D−06
S368	100	2139	12532	2140	317.55	−7.085D+01	9.966D−06
HADAMALS	1024	1808	11468	1809	157.88	3.067D+04	9.611D−06
CHEBYQAD	50	5287	50893	5288	607.89	5.386D−03	9.918D−06
HS110	50	1	2	2	0.00	−9.990D+09	0.000D+00
LINVERSE	1999	19563	200001	19564	1465.91	6.820D+02	9.202D−02
NONSCOMP	10000	3737	25220	3738	559.04	7.632D−13	9.933D−06
QR3DLS	610	17272	200001	17273	735.62	3.051D−01	3.638D−01
SCON1LS	1002	40237	200001	40238	1512.18	6.572D+01	8.501D−02
DECONVB	61	6536	35665	6537	10.00	2.713D−03	1.814D−06
BIGGSB1	1000	50001	104775	50002	190.46	1.896D−02	1.362D−03
BQPGABIM	50	2222	22640	2223	1.68	−3.790D−05	9.972D−06
BQPGASIM	50	1247	12394	1248	0.94	−5.520D−05	9.334D−06
BQPGAUSS	2003	13482	200001	13483	986.07	−1.294D−01	1.037D+00
CHENHARK	1000	50001	173351	50002	323.09	−2.000D+00	5.299D−04
CVXBQP1	10000	1	2	2	0.10	2.250D+06	0.000D+00
HARKERP2	100	100	304	101	0.26	−5.000D−01	0.000D+00
JNLBRNG1	15625	13681	28689	13682	3332.51	−1.806D−01	5.686D−06
JNLBRNG2	15625	21444	107760	21445	8427.10	−4.150D+00	9.624D−06
JNLBRNGA	15625	12298	27172	12299	2666.47	−2.685D−01	5.388D−06
JNLBRNGB	15625	32771	200001	32772	12672.71	−5.569D+00	3.744D+00
NCVXBQP1	10000	1	2	2	0.10	−1.986D+10	0.000D+00
NCVXBQP2	10000	18012	200001	18013	4053.97	−1.334D+10	5.798D−01
NCVXBQP3	10000	15705	200001	15706	3955.02	−6.559D+09	2.609D+00
NOBNDTOR	14884	3649	7300	3650	718.13	−4.405D−01	8.604D−06
OBSTCLAE	15625	5049	11402	5050	1119.07	1.901D+00	1.000D−05
OBSTCLAL	15625	2734	6838	2735	634.97	1.901D+00	9.986D−06
OBSTCLBL	15625	3669	9084	3670	846.45	7.296D+00	9.995D−06
OBSTCLBM	15625	2941	7634	2942	694.42	7.296D+00	9.983D−06
OBSTCLBU	15625	3816	9403	3817	880.51	7.296D+00	9.981D−06
PENTDI	1000	50001	199995	50002	460.38	−7.500D− 01	2.688D−05
TORSION1	14884	4540	9082	4541	890.47	−4.257D−01	6.673D−06
TORSION2	14884	8704	17294	8705	1703.87	−4.257D−01	6.599D−06
TORSION3	14884	1941	4525	1942	406.85	−1.212D+00	9.957D−06
TORSION4	14884	4273	9062	4274	862.93	−1.212D+00	9.897D−06
TORSION5	14884	672	1651	673	144.80	−2.859D+00	9.813D−06
TORSION6	14884	1569	3322	1570	316.06	−2.859D+00	9.908D−06
TORSIONA	14884	4155	8312	4156	953.30	−4.184D−01	8.980D−06
TORSIONB	14884	8274	16417	8275	1899.52	−4.184D−01	8.829D−06
TORSIONC	14884	1933	4563	1934	476.48	−1.204D+00	9.976D−06
TORSIOND	14884	4325	9218	4326	1013.10	−1.204D+00	9.854D−06
TORSIONE	14884	688	1695	689	172.87	−2.851D+00	9.727D−06
TORSIONF	14884	1493	3143	1494	349.72	−2.851D+00	9.712D−06
ODNAMUR	11130	13222	200001	13223	5249.00	1.209D+04	5.192D+00

The complete performance of LANCELOT on this set of problems is reported in Table 3. In Tables 4 and 5 we show the behavior of SPG1 and SPG2, respectively.

For LANCELOT, we report the number of outer iterations (or function evaluations) (IT_{out}-FE), gradient evaluations (GE), conjugate gradient (or inner) iterations

TABLE 3
Performance of LANCELOT.

Problem	n	IT_{out}-FE	GE	IT_{in}-CG	Time	$f(x)$	$\|g_1(x)\|_\infty$
BDEXP	5000	10	11	26	3.19	1.964D−03	6.167D−06
EXPLIN	120	13	14	50	0.08	−7.238D+05	5.183D−09
EXPLIN2	120	11	12	24	0.07	−7.245D+05	1.012D−06
EXPQUAD	120	18	16	52	0.14	−3.626D+06	1.437D−06
MCCORMCK	10000	7	6	5	4.71	−9.133D+03	5.861D−06
PROBPENL	500	1	2	0	0.17	3.992D−07	3.424D−07
QRTQUAD	120	168	137	187	1.23	−3.625D+06	3.568D−06
S368	100	7	7	11	2.19	−1.337D+02	3.314D−06
HADAMALS	1024	33	34	5654	157.60	7.444D+02	7.201D−06
CHEBYQAD	50	65	48	829	5.41	5.386D−03	7.844D−06
HS110	50	1	2	0	0.02	−9.990D+09	0.000D+00
LINVERSE	1999	35	30	2303	77.52	6.810D+02	8.407D−06
NONSCOMP	10000	8	9	9	4.74	3.055D−14	9.749D−09
QR3DLS	610	255	226	25036	434.02	3.818D−08	4.051D−06
SCON1LS	1002	1604	1372	1357	56.51	7.070D−10	8.568D−06
DECONVB	61	17	16	233	0.40	1.236D−08	2.147D−06
BIGGSB1	1000	501	502	500	6.17	1.500D−02	4.441D−16
BQPGABIM	50	3	4	10	0.03	−3.790D−05	6.120D−06
BQPGASIM	50	3	4	9	0.03	−5.520D−05	5.733D−06
BQPGAUSS	2003	8	9	2345	42.60	−3.626D−01	4.651D−06
CHENHARK	1000	205	206	484	5.02	−2.000D+00	6.455D−06
CVXBQP1	10000	1	2	1	3.69	2.250D+06	0.000D+00
HARKERP2	100	1	2	2	0.11	−5.000D−01	7.514D−13
JNLBRNG1	15625	24	25	1810	217.19	−1.806D−01	4.050D−06
JNLBRNG2	15625	14	15	912	108.93	−4.150D+00	9.133D−07
JNLBRNGA	15625	21	22	1327	155.93	−2.685D−01	1.191D−06
JNLBRNGB	15625	10	11	329	42.58	−6.281D+00	2.602D−06
NCVXBQP1	10000	1	2	0	3.27	−1.986D+10	0.000D+00
NCVXBQP2	10000	3	4	407	6.62	−1.334D+10	5.821D−11
NCVXBQP3	10000	5	6	360	6.67	−6.558D+09	2.915D−06
NOBNDTOR	14884	36	37	790	117.34	−4.405D−01	2.758D−06
OBSTCLAE	15625	4	5	7409	1251.08	1.901D+00	1.415D−06
OBSTCLAL	15625	24	25	480	58.05	1.901D+00	5.323D−06
OBSTCLBL	15625	18	19	2761	397.58	7.296D+00	1.996D−06
OBSTCLBM	15625	5	6	1377	233.70	7.296D+00	2.243D−06
OBSTCLBU	15625	19	20	787	112.55	7.296D+00	1.529D−06
PENTDI	1000	1	2	0	0.20	−7.500D−01	0.000D+00
TORSION1	14884	37	38	793	96.88	−4.257D−01	1.237D−06
TORSION2	14884	9	10	4339	722.28	−4.257D−01	4.337D−06
TORSION3	14884	19	20	241	27.36	−1.212D+00	2.234D−06
TORSION4	14884	15	16	5639	894.13	−1.212D+00	6.469D−07
TORSION5	14884	9	10	72	10.48	−2.859D+00	3.186D−06
TORSION6	14884	10	11	4895	579.62	−2.859D+00	8.124D−07
TORSIONA	14884	37	38	795	103.70	−4.184D−01	9.590D−07
TORSIONB	14884	10	11	4025	722.79	−4.184D−01	1.329D−06
TORSIONC	14884	19	20	241	29.77	−1.205D+00	2.236D−06
TORSIOND	14884	9	10	9134	1369.14	−1.205D+00	5.184D−06
TORSIONE	14884	9	10	72	11.25	−2.851D+00	3.201D−06
TORSIONF	14884	10	11	5008	631.14	−2.851D+00	8.796D−07
ODNAMUR	11130	11	12	26222	1416.03	9.237D+03	7.966D−06

(IT_{in}-CG), CPU time in seconds (Time), functional value at the final iterate ($f(x)$), and ∞-norm of the "continuous projected gradient" at the final iterate ($\|g_1(x)\|_\infty$). For SPG methods, we report number of iterations (IT), function evaluations (FE), gradient evaluations (GE), CPU time in seconds (Time), best function value found ($f(x)$),

TABLE 4
Performance of SPG1.

Problem	n	IT	FE	GE	Time	$f(x)$	$\|g_1(x)\|_\infty$
BDEXP	5000	12	13	13	0.45	2.744D−03	7.896D−06
EXPLIN	120	66	75	67	0.01	−7.238D+05	3.100D−06
EXPLIN2	120	48	54	49	0.01	−7.245D+05	9.746D−07
EXPQUAD	120	92	107	93	0.03	−3.626D+06	4.521D−06
MCCORMCK	10000	16	17	17	1.78	−9.133D+03	4.812D−06
PROBPENL	500	2	7	3	0.01	3.992D−07	1.721D−07
QRTQUAD	120	1693	5242	1694	0.74	−3.625D+06	5.125D−06
S368	100	8	14	9	0.67	−1.200D+02	1.566D−07
HADAMALS	1024	33	42	34	1.49	3.107D+04	4.828D−08
CHEBYQAD	50	970	1545	971	35.52	5.386D−03	9.993D−06
HS110	50	1	2	2	0.00	−9.990D+09	0.000D+00
LINVERSE	1999	1707	2958	1708	45.42	6.810D+02	9.880D−06
NONSCOMP	10000	43	44	44	2.28	3.419D−10	7.191D−06
QR3DLS	610	50001	106513	50002	884.18	2.118D−04	9.835D−03
SCON1LS	1002	50001	75083	50002	882.43	1.329D+01	7.188D−03
DECONVB	61	1786	2585	1787	1.68	4.440D−08	9.237D−06
BIGGSB1	1000	6820	11186	6821	23.15	1.621D−02	9.909D−06
BQPGABIM	50	30	39	31	0.01	−3.790D−05	8.855D−06
BQPGASIM	50	32	39	33	0.01	−5.520D−05	9.100D−06
BQPGAUSS	2003	50001	86373	50002	930.52	−3.623D−01	1.930D−02
CHENHARK	1000	3563	6113	3564	14.89	−2.000D+00	9.993D−06
CVXBQP1	10000	1	2	2	0.10	2.250D+06	0.000D+00
HARKERP2	100	33	46	34	0.06	−5.000D−01	0.000D+00
JNLBRNG1	15625	1335	1897	1336	283.55	−1.806D−01	9.624D−06
JNLBRNG2	15625	1356	2121	1357	296.46	−4.150D+00	9.738D−06
JNLBRNGA	15625	629	933	630	116.77	−2.685D−01	9.809D−06
JNLBRNGB	15625	8531	13977	8532	1635.15	−6.281D+00	9.903D−06
NCVXBQP1	10000	1	2	2	0.10	−1.986D+10	0.000D+00
NCVXBQP2	10000	60	83	61	3.47	−1.334D+10	8.219D−06
NCVXBQP3	10000	112	118	113	5.31	−6.558D+09	6.019D−06
NOBNDTOR	14884	568	817	569	99.62	−4.405D−01	9.390D−06
OBSTCLAE	15625	749	1028	750	136.98	1.901D+00	7.714D−06
OBSTCLAL	15625	290	411	291	53.56	1.901D+00	7.261D−06
OBSTCLBL	15625	354	500	355	65.52	7.296D+00	9.024D−06
OBSTCLBM	15625	249	343	250	45.74	7.296D+00	9.139D−06
OBSTCLBU	15625	325	468	326	60.44	7.296D+00	7.329D−06
PENTDI	1000	12	14	13	0.07	−7.500D−01	8.523D−07
TORSION1	14884	574	832	575	101.00	−4.257D−01	9.525D−06
TORSION2	14884	586	862	587	102.79	−4.257D−01	9.712D−06
TORSION3	14884	231	350	232	41.47	−1.212D+00	9.593D−06
TORSION4	14884	190	259	191	32.66	−1.212D+00	8.681D−06
TORSION5	14884	83	101	84	13.84	−2.859D+00	9.169D−06
TORSION6	14884	82	97	83	13.58	−2.859D+00	7.987D−06
TORSIONA	14884	722	1057	723	147.94	−4.184D−01	8.590D−06
TORSIONB	14884	527	765	528	107.52	−4.184D−01	9.475D−06
TORSIONC	14884	190	270	191	38.50	−1.204D+00	9.543D−06
TORSIOND	14884	241	340	242	48.43	−1.204D+00	9.575D−06
TORSIONE	14884	57	76	58	11.42	−2.851D+00	8.700D−06
TORSIONF	14884	67	85	68	14.16	−2.851D+00	9.352D−06
ODNAMUR	11130	50001	82984	50002	4187.58	9.250D+03	9.690D−02

and ∞-norm of the continuous projected gradient at the final iterate ($\|g_1(x)\|_\infty$).

The numerical results of 10 problems deserve special comments:

(1) BDEXP ($n = 5,000$): LANCELOT obtained $f(x) = 1.964 \times 10^{-3}$ in 3.19 seconds, whereas SPG1 and SPG2 got $f(x) = 2.744 \times 10^{-3}$ in 0.45 seconds. Since the gradient norm is computed in LANCELOT only after each outer

TABLE 5
Performance of SPG2.

Problem	n	IT	FE	GE	Time	$f(x)$	$\|g_1(x)\|_\infty$
BDEXP	5000	12	13	13	0.45	2.744D−03	7.896D−06
EXPLIN	120	54	57	55	0.01	−7.238D+05	4.482D−06
EXPLIN2	120	56	59	57	0.01	−7.245D+05	5.633D−06
EXPQUAD	120	92	110	93	0.03	−3.626D+06	7.644D−06
MCCORMCK	10000	16	17	17	1.78	−9.133D+03	4.812D−06
PROBPENL	500	2	6	3	0.01	3.992D−07	1.022D−07
QRTQUAD	120	598	1025	599	0.19	−3.624D+06	8.049D−06
S368	100	16	19	17	1.15	−1.403D+02	1.963D−08
HADAMALS	1024	30	42	31	1.27	3.107D+04	2.249D−07
CHEBYQAD	50	1240	2015	1241	45.73	5.386D−03	8.643D−06
HS110	50	1	2	2	0.00	−9.990D+09	0.000D+00
LINVERSE	1999	1022	1853	1023	26.75	6.810D+02	8.206D−06
NONSCOMP	10000	43	44	44	2.22	3.419D−10	7.191D−06
QR3DLS	610	50001	107915	50002	869.25	2.312D−04	1.599D−02
SCON1LS	1002	50001	76011	50002	835.10	1.416D+01	1.410D−02
DECONVB	61	1670	2560	1671	1.38	4.826D−08	9.652D−06
BIGGSB1	1000	7571	12496	7572	24.41	1.626D−02	9.999D−06
BQPGABIM	50	24	37	25	0.01	−3.790D−05	8.640D−06
BQPGASIM	50	33	46	34	0.01	−5.520D−05	8.799D−06
BQPGAUSS	2003	50001	87102	50002	902.26	−3.624D−01	2.488D−03
CHENHARK	1000	2464	4162	2465	9.60	−2.000D+00	9.341D−06
CVXBQP1	10000	1	2	2	0.10	2.250D+06	2.776D−17
HARKERP2	100	33	46	34	0.06	−5.000D−01	1.110D−16
JNLBRNG1	15625	1664	2524	1665	349.19	−1.806D−01	6.265D−06
JNLBRNG2	15625	1443	2320	1444	309.22	−4.150D+00	9.665D−06
JNLBRNGA	15625	981	1530	982	180.92	−2.685D−01	6.687D−06
JNLBRNGB	15625	17014	28077	17015	3180.14	−6.281D+00	1.000D−05
NCVXBQP1	10000	1	2	2	0.10	−1.986D+10	2.776D−17
NCVXBQP2	10000	84	93	85	4.00	−1.334D+10	2.956D−06
NCVXBQP3	10000	111	117	112	5.13	−6.558D+09	2.941D−06
NOBNDTOR	14884	566	834	567	98.52	−4.405D− 01	8.913D−06
OBSTCLAE	15625	639	936	640	116.86	1.901D+00	9.343D−06
OBSTCLAL	15625	176	243	177	31.69	1.901D+00	6.203D−06
OBSTCLBL	15625	321	460	322	58.49	7.296D+00	3.731D−06
OBSTCLBM	15625	143	192	144	25.63	7.296D+00	8.294D−06
OBSTCLBU	15625	311	449	312	56.72	7.296D+00	9.703D−06
PENTDI	1000	1	3	2	0.01	−7.500D−01	0.000D+00
TORSION1	14884	685	1023	686	119.38	−4.257D−01	9.404D−06
TORSION2	14884	728	1117	729	127.62	−4.257D−01	9.616D−06
TORSION3	14884	183	264	184	31.72	−1.212D+00	6.684D−06
TORSION4	14884	226	325	227	38.99	−1.212D+00	9.398D−06
TORSION5	14884	73	105	74	12.68	−2.859D+00	8.751D−06
TORSION6	14884	63	75	64	10.39	−2.859D+00	9.321D−06
TORSIONA	14884	496	756	497	100.13	−4.184D−01	6.442D−06
TORSIONB	14884	584	866	585	116.70	−4.184D−01	7.917D−06
TORSIONC	14884	247	350	248	48.81	−1.204D+00	9.683D−06
TORSIOND	14884	226	317	227	44.62	−1.204D+00	9.467D−06
TORSIONE	14884	65	89	66	12.90	−2.851D+00	9.459D−06
TORSIONF	14884	68	84	69	13.07	−2.851D+00	9.302D−06
ODNAMUR	11130	50001	80356	50002	3927.97	9.262D+03	4.213D−01

iteration, which involves considerable computer effort, LANCELOT usually stops at points where this norm is considerably smaller than the tolerance 10^{-5}. On the other hand, SPG methods, which test the projected gradient more frequently, stop when $\|g_1(x)\|_\infty$ is slightly smaller than that tolerance. In a small number of cases this affects the quality of the solution, reflected in the objective function value.

(2) S368 ($n = 100$): LANCELOT, SPG1, and SPG2 arrived at different solutions, the best of which was the one obtained by SPG2. SPG1 was the winner in terms of computer time.

(3) HADAMALS ($n = 1,024$): LANCELOT obtained $f(x) = 74.44$ in 157.6 seconds. SPG1 and SPG2 obtained stationary points with $f(x) = 31,070$ in less than 2 seconds.

(4) NONSCOMP ($n = 10,000$): As in BDEXP, the SPG methods found a solution slightly worse than the one found by LANCELOT but used less computer time.

(5) QR3DLS ($n = 610$): LANCELOT found a better solution ($f(x) \approx 4 \times 10^{-8}$ against $f(x) \approx 2.3 \times 10^{-4}$) and used less computer time than the SPG methods.

(6) SCON1LS ($n = 1,002$): LANCELOT found the solution whereas the SPG methods did not converge after 50,000 iterations.

(7) DECONVB ($n = 61$): LANCELOT found the (slightly) best solution and used less computer time than the SPG methods.

(8) BIGGSB1 ($n = 1,000$): LANCELOT found $f(x) = 0.015$ in 6.17 seconds, whereas the SPG methods got $f(x) \approx 0.016$ in ≈ 24 seconds.

(9) BQPGAUSS ($n = 2,003$): LANCELOT beat SPG methods in this problem, in terms of both computer time and quality of solution.

(10) ODNAMUR ($n = 11,130$): LANCELOT obtained a better solution than the SPG methods for this problem and used less computer time.

Four of the problems considered above (QR3DLS, SCON1LS, BQPGAUSS, and ODNAMUR) can be considered failures of both SPG methods, since convergence to a stationary point was not attained after 50,000 iterations. In the four cases, the final point seems to be in the local attraction basin of a local minimizer, but local convergence is very slow. In fact, in the first three problems, the final projected gradient norm is $\approx 10^{-2}$, and in ODNAMUR the difference between $f(x)$ and its optimal value is ≈ 0.1 %. Slow convergence of SPG methods when the Hessian at the local minimizer is very ill conditioned is expected, and preconditioning schemes tend to alleviate this inconvenient. See [21].

In the remaining 40 problems, LANCELOT, SPG1, and SPG2 found the same solutions. In terms of computer time, SPG1 was faster than LANCELOT in 29 problems (72.5%) and SPG2 outperformed LANCELOT also in 29 problems. There are no meaningful differences between the performances of SPG1 and SPG2.

Excluding problems where the difference in CPU time was less than 10%, SPG1 beat LANCELOT 28-9 and SPG2 beat LANCELOT 28-11.

Excluding, from the 40 problems above, the ones in which the 3 algorithms converged in less than 1 second, we are left with 31 problems. Considering this set, SPG1 beat LANCELOT 20-11 (or 19-9 if we exclude, again, differences smaller than 10%) and SPG2 beat LANCELOT 20-11 (or 19-11).

As we mentioned above, we also implemented the projected gradient algorithm PGA, using the same framework as SPG in terms of interpolation schemes, both with monotone and nonmonotone strategies. The performance of both alternatives is very poor, in comparison to the algorithms SPG1 and SPG2 and other box-constraint minimizers. The performance of the nonmonotone version is given in Table 2. This confirms that the spectral choice of the steplength is the essential feature that puts efficiency in the projected gradient methodology.

4. Final remarks. It is customary to interpret the first trial step of a minimization algorithm as the minimizer of a quadratic model $q(x)$ on the feasible region or an approximation to it. It is always imposed that the first-order information at the current point should coincide with the first order information of the quadratic model. So, the quadratic approximation at x_{k+1} should be

$$q(x) = \frac{1}{2}\langle x - x_{k+1}, B_{k+1}(x - x_{k+1})\rangle + \langle g(x_{k+1}), x - x_{k+1}\rangle + f(x_{k+1})$$

and

$$\nabla q(x) = B_{k+1}(x - x_{k+1}) + g(x_{k+1}).$$

Secant methods are motivated by the interpolation condition $\nabla f(x_k) = \nabla q(x_k)$. Let us impose here the weaker condition

$$D_{s_k} q(x_k) = D_{s_k} f(x_k), \tag{11}$$

where $D_d\varphi(x)$ denotes the directional derivative of φ along the direction d (so $D_d\varphi(x) = \langle\nabla\varphi(x), d\rangle$). A short calculation shows that condition (11) is equivalent to

$$\langle s_k, B_{k+1}s_k\rangle = \langle s_k, y_k\rangle. \tag{12}$$

Clearly, the spectral choice

$$B_{k+1} = \frac{\langle s_k, y_k\rangle}{\langle s_k, s_k\rangle} I \tag{13}$$

(where I is the identity matrix) satisfies (12). Now, suppose that z is orthogonal to s_k and that x belongs to $\mathcal{L}_k$, the line determined by x_k and x_{k+1}. Computing the directional derivative of q along z at any point $x \in \mathcal{L}_k$, and using (13), we obtain

$$D_z q(x) = \langle B_{k+1}(x - x_{k+1}) + g(x_{k+1}), z\rangle = \langle g(x_{k+1}), z\rangle = D_z f(x_{k+1}).$$

Moreover, the properties (12) and

$$D_z q(x) = D_z f(x_{k+1}) \quad \text{for all} \quad x \in \mathcal{L}_k \quad \text{and} \quad z \perp s_k \tag{14}$$

imply that s_k is an eigenvector of B_{k+1} with eigenvalue $\langle s_k, y_k\rangle / \langle s_k, s_k\rangle$. Clearly, (13) is the most simple choice that satisfies this property. Another remarkable property of (13) is that the resulting algorithms turn out to be invariant under change of scale of both f and the independent variables.

In contrast to the property (14), satisfied by the spectral choice of B_{k+1}, models generated by the secant choice have the property that the directional derivatives of the model coincide with the directional derivatives of the objective function *at* x_k. Property (14) says that the model was chosen in such a way that the first order information with respect to orthogonal directions to s_k is the same as the first order information of the true objective function at x_{k+1} *for all* the points on the line $\mathcal{L}_k$. This means that first order information at the current point is privileged in the construction of the quadratic model, in relation to second order information that comes from the previous iteration. Perhaps this is one of the reasons underlying the unexpected efficiency of spectral gradient algorithms in relation to some rather arbitrary secant methods. Needless to say, the special form of B_{k+1} trivializes the problem of minimizing the model on the feasible set when this is simple enough, a fact that is fully exploited in SPG1 and SPG2.

Boxes are not the only type of sets on which it is trivial to project. The norm-constrained regularization problem [18, 23, 24, 32], defined by

$$\text{minimize } f(x) \quad \text{subject to} \quad x^T Ax \leq r, \tag{15}$$

where A is symmetric positive definite, can be reduced to ball constrained minimization by a change of variables and, in this case, projections can be trivially computed. A particular case of (15) is the classical trust-region subproblem, where f is quadratic. Recently (see [20, 25]) procedures for escaping from nonglobal stationary points of this problem have been found, and so it becomes increasingly important to obtain fast algorithms for finding critical points, especially in the large-scale case. (See [28, 29, 31].)

Perhaps the most important characteristic of SPG algorithms is that they are extremely simple to code, to the point that anyone can write her or his own code using any scientific language in a couple of hours. (Fortran, C, and Matlab codes written by the authors are available by request.) Moreover, their extremely low memory requirements make them very attractive for large-scale problems. It is quite surprising that such a simple tool can be competitive with rather elaborate algorithms that use extensively tested subroutines and numerical procedures. The authors would like to encourage readers to write their own codes and to verify for themselves the nice properties of these algorithms in practical situations. Papers [6] and [4] illustrate the use of SPG methods in applications.

Acknowledgment. The authors are indebted to two anonymous referees whose comments helped considerably to improve this paper.

REFERENCES

[1] J. Barzilai and J. M. Borwein, *Two point step size gradient methods*, IMA J. Numer. Anal., 8 (1988), pp. 141–148.

[2] D. P. Bertsekas, *On the Goldstein-Levitin-Polyak gradient projection method*, IEEE Trans. Automat. Control, 21 (1976), pp. 174–184.

[3] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995.

[4] E. G. Birgin, R. Biloti, M. Tygel, and L. T. Santos, *Restricted optimization: A clue to a fast and accurate implementation of the common reflection surface stack method*, J. Appl. Geophys., 42 (1999), pp. 143–155.

[5] E. G. Birgin, I. Chambouleyron, and J. M. Martínez, *Estimation of the optical constants and the thickness of thin films using unconstrained optimization*, J. Comput. Phys., 151 (1999), pp. 862–880.

[6] E. G. Birgin and Y. G. Evtushenko, *Automatic differentiation and spectral projected gradient methods for optimal control problems*, Optim. Methods Softw., 10 (1998), pp. 125–146.

[7] P. H. Calamai and J. J. Moré, *Projected gradient methods for linearly constrained problems*, Math. Programming, 39 (1987), pp. 93–116.

[8] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *Global convergence of a class of trust region algorithms for optimization with simple bounds*, SIAM J. Numer. Anal., 25 (1988), pp. 433–460; see also SIAM J. Numer. Anal., 26 (1989), pp. 764–767.

[9] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*, Springer Ser. Comput. Math. 17, Springer-Verlag, New York, Berlin, Heidelberg, 1992.

[10] A. R. Conn, N. I. M. Gould, and Ph. L. Toint, *A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds*, SIAM J. Numer. Anal., 28 (1991), pp. 545–572.

[11] J. C. Dunn, *Global and asymptotic convergence rate estimates for a class of projected gradient processes*, SIAM J. Control Optim., 19 (1981), pp. 368–400.

[12] J. C. Dunn, *Gradient-related constrained minimization algorithms in function spaces: Convergence properties and computational implications*, in Large Scale Optimization: State

of the Art, W. W. Hager, D. W. Hearn, and P. M. Pardalos, eds., Kluwer, Dordrecht, the Netherlands, 1994.

[13] A. Friedlander, J. M. Martínez, and S. A. Santos, *A new trust region algorithm for bound constrained minimization*, Appl. Math. Optim., 30 (1994), pp. 235–266.

[14] E. M. Gafni and D. P. Bertsekas, *Convergence of a Gradient Projection Method*, Report LIDS-P-1201, Lab. for Info. and Dec. Systems, MIT, Cambridge, MA, 1982.

[15] W. Glunt, T. L. Hayden, and M. Raydan, *Molecular conformations from distance matrices*, J. Comput. Chem., 14 (1993), pp. 114–120.

[16] A. A. Goldstein, *Convex programming in Hilbert space*, Bull. Amer. Math. Soc., 70 (1964), pp. 709–710.

[17] L. Grippo, F. Lampariello, and S. Lucidi, *A nonmonotone line search technique for Newton's method*, SIAM J. Numer. Anal., 23 (1986), pp. 707–716.

[18] M. Heinkenschloss, *Mesh independence for nonlinear least squares problems with norm constraints*, SIAM J. Optim., 3 (1993), pp. 81–117.

[19] E. S. Levitin and B. T. Polyak, *Constrained Minimization Problems*, USSR Comput. Math. Math. Phys., 6 (1966), pp. 1–50.

[20] S. Lucidi, L. Palagi, and M. Roma, *On some properties of quadratic programs with a convex quadratic constraint*, SIAM J. Optim., 8 (1998), pp. 105–122.

[21] F. Luengo, M. Raydan, W. Glunt, and T. L. Hayden, *Preconditioned Spectral Gradient Method for Unconstrained Optimization Problems*, Technical Report R.T. 96-08, Computer Science Department, Universidad Central de Venezuela, Caracos Venezuela.

[22] G. P. McCormick and R. A. Tapia, *The gradient projection method under mild differentiability conditions*, SIAM J. Control, 10 (1972), pp. 93–98.

[23] J. M. Martínez and S. A. Santos, *A trust region strategy for minimization on arbitrary domains*, Math. Programming, 68 (1995), pp. 267–302.

[24] J. M. Martínez and S. A. Santos, *Convergence results on an algorithm for norm constrained regularization and related problems*, RAIRO Rech. Opér., 31 (1997), pp. 269–294.

[25] P. D. Tao and L. T. H. An, *D.C. optimization algorithm for solving the trust-region subproblem*, SIAM J. Optim., 8 (1998), pp. 476–505.

[26] M. Raydan, *On the Barzilai and Borwein choice of steplength for the gradient method*, IMA J. Numer. Anal., 13 (1993), pp. 321–326.

[27] M. Raydan, *The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem*, SIAM J. Optim., 7 (1997), pp. 26–33.

[28] F. Rendl and H. Wolkowicz, *A semidefinite framework to trust region subproblems with application to large scale minimization*, Math. Programming, 77 (1997), pp. 273–299.

[29] M. Rojas, S. A. Santos and D. C. Sorensen, *A new matrix-free algorithm for the large-scale trust-region subproblem*, SIAM J. Optim, to appear.

[30] A. Schwartz and E. Polak, *Family of projected descent methods for optimization problems with simple bounds*, J. Optim. Theory Appl., 92 (1997), pp. 1–31.

[31] D. C. Sorensen, *Minimization of a large-scale quadratic function subject to spherical constraint*, SIAM J. Optim., 7 (1997), pp. 141–161.

[32] C. R. Vogel, *A constrained least squares regularization method for nonlinear ill-posed problems*, SIAM J. Control Optim., 28 (1990), pp. 34–49.