An inexact and nonmonotone proximal method for smooth unconstrained minimization*

Sandra A. Santos[†] Roberto C. M. Silva[‡]

November 7, 2013

Abstract

An implementable proximal point algorithm is established for the smooth nonconvex unconstrained minimization problem. At each iteration, the algorithm minimizes approximately a general quadratic by a truncated strategy with step length control. The main contributions are: (i) a framework for updating the proximal parameter; (ii) inexact criteria for approximately solving the subproblems; (iii) a nonmonotone criterion for accepting the iterate. The global convergence analysis is presented, together with numerical results that validate and put into perspective the proposed approach.

Keywords: Proximal point algorithms; regularization; nonconvex problems; unconstrained minimization; global convergence; nonmonotone line search; numerical experiments.

AMS Classification: 49M37; 65K05; 90C30.

*This work was partially supported by CNPq (Grants 163123/2011-0 and 304032/2010-7), FAPESP (Grants 2013/05475-7 and 2013/07375-0) and PRONEX-CNPq/FAPERJ (E-26/111.449/2010-APQ1).

[†]Department of Applied Mathematics, University of Campinas, Campinas, São Paulo, Brazil. E-mail: sandra@ime.unicamp.br.

[‡]Department of Mathematics, Federal University of Amazonas, Manaus, Amazonas, Brazil. E-mail: rmesquita@ufam.edu.br.

1 Introduction

Originally introduced by Martinet [17] and disseminated by Rockafellar [19, 20] for convex problems, the proximal point methods are tools for the solution of a wide class of mathematical problems, including ill-posed and ill-conditioned ones, being a standard regularization technique in optimization. In the vast literature of proximal methods, an area of active research (cf. the survey of Parikh and Boyd [18] and references therein), most of the contributions rest upon convex analysis and monotone operator theory. In practical terms, such assumptions allow the solution of the proximal operator to have closed-form or to be obtained very quickly, with suitable methods. These features are exploited, for instance, in the algorithms FISTA [3] and ADMM [5], to mention a few.

Three recent contributions [11, 15, 16] have motivated our investigation. Fuentes, Malick and Lemaréchal [11] have focused on the stopping rules for developing an inexact proximal algorithm for smooth optimization. Humes and Silva [16] have presented the theory of the proximal point algorithm under the perspective of descent methods for unconstrained optimization. Hager and Zhang [15] exploited self-adaptive proximal point methods specially developed for degenerate optimization problems, either with multiple minimizers, or with singular Hessian at a local solution.

In this work, an inexact nonmonotone proximal point algorithm – INPPA – for unconstrained smooth minimization is proposed. Is has been developed to address degenerate problems, in the aforementioned sense, without any convexity assumption. The proximal parameter is automatically updated, according with the obtained progress towards optimality. The unconstrained quadratic (not necessarily convex) subproblems are inexactly solved by means of the truncated Newton framework of Dembo and Steihaug [7], with the Steihaug-Toint step length control [24, 25]. Whenever the prospective step is bounded away from orthogonality with the gradient, it is accepted. If a relaxed condition of sufficient decrease fails, a nonmonotone line search is performed, aligned with elements from Zhang and Hager [26] (see also [21] and references therein). In this case, the rationale is to overcome a possibly premature stopping of the conjugate-gradient scheme and, instead of improving the current step further within the Krylov spaces [12], a simple surrogate model is used to define the initial step length [22, 23].

Although our algorithm employs trust-region ingredients, it is not a genuine trust-region algorithm. Thus, its convergence analysis is not a straighforward adaptation of trust-region convergence results. Indeed, it is supported by the line search framework, based on standard assumptions upon the problem and on the boundedness of the proximal parameter. For further comments on the connection between proximal point and trust-region methods, see [6, §6.1].

We have performed a comparison with the numerical results of [11] for singular or ill-conditioned problems from the CUTEr collection [13], which have also been solved by the approach developed in [15]. A comparison with results of [14] and [10] is made as well, for a set of general unconstrained minimization problems.

This paper is organized as follows. In Sect. 2, we present the main ingredients for establishing the proximal-point scheme. In Sect. 3, we describe the Algorithm INPPA, some inherent properties and auxiliary results. In Sect. 4, we prove that the proposed algorithm is globally convergent. The computational results are presented and analyzed in Sect. 5, and the final remarks are stated in Sect. 6. An Appendix with the conjugate gradient algorithm of Steihaug is included, for the readers convenience.

2 Preliminaries

In this section we present the ingredients to establish a proximal-point scheme for smooth non-convex unconstrained minimization. The unconstrained minimization problem under analysis is given by

$$\min_{x\in \mathbb{R}^n} f(x), \tag{1}$$

with $f : \mathbb{R}^n \to \mathbb{R}, f \in C^2$. Without any convexity assumption, a local solution of (1) may be characterized as a vector $x_* \in \mathbb{R}^n$ for which the second-order sufficiency optimality conditions hold, i.e., $\nabla f(x_*) = 0$ and $\nabla^2 f(x_*)$ is positive definite. Nevertheless, to encompass degenerate solutions, we will only assume that $f(x_*) \le f(x)$ for all x in a neighborhood of x_*, a point for which $\nabla f(x_*) = 0$.

A proximal-point method for solving (1) will generate a sequence $\{x_k\}$ of approximations to a local solution x_* by means of a sequence of proximal parameters $\{t_k\} \subset \mathbb{R}_+$ and the associated local solution of the proximal-point subproblems:

$$\min_{x\in \mathbb{R}^n} f(x) + \frac{1}{2t_k}\|x - x_k\|^2 \tag{2}$$

where $\|\cdot\|$ denotes the Euclidean norm. The objective function of problem (2) will be denoted by

$$\widetilde{f}_k(x) := f(x) + \frac{1}{2t_k}\|x - x_k\|^2. \tag{3}$$

For convenience, we use the following short notation: $f_k := f(x_k)$, $g_k := \nabla f(x_k)$ and $H_k := \nabla^2 f(x_k)$. The quadratic model for f around x_k is given by

$$q_k(x) := f_k + g_k^\top (x - x_k) + \frac{1}{2}(x - x_k)^\top B_k(x - x_k), \tag{4}$$

where $B_k \in \mathbb{R}^{n\times n}$ is any symmetric matrix that approximates H_k, including H_k itself.

A Newtonian strategy for solving (2) is based on the approximation $f(x) \approx q_k(x)$, so that the following sequence of quadratic subproblems must be addressed

$$\min_{x\in \mathbb{R}^n} \widetilde{q}_k(x) \tag{5}$$

where $\widetilde{q}_k(x) := q_k(x) + \frac{1}{2t_k}\|x - x_k\|^2$.

Without any convexity hypothesis for the problem (1), the matrices B_k may not be positive definite, especially far from a local solution x^*. The Hessians $\nabla^2 \widetilde{q}_k(x) = B_k + \frac{1}{t_k} I$, on the other hand, might provide positive curvature to the model, for appropriate choices of the proximal parameter t_k. In our approach, however, it is not mandatory to ensure the convexity of the kth-model (5), as clarified next.

3 The algorithm

Our goal is to build an implementable inexact nonmonotone proximal point algorithm, that we call the Algorithm INPPA. After the description of the algorithm, we set up the standard assumptions, and present some properties and auxiliary results for establishing that it is well defined.

The stationary points of (2), solutions of the nonlinear equations $\nabla f(x) + \frac{1}{t_k}(x - x_k) = 0$, may be approximated by the Newtonian linear system

$$\left(B_k + \frac{1}{t_k} I\right) s = -g_k, \tag{6}$$

in which $s := x - x_k$ defines the step for updating the current approximation x_k.

For a given tolerance $\varepsilon > 0$, a given forcing sequence $\{\eta_k\} \subset I\!R_+$ such that

$$\lim_{k\to\infty} \eta_k = 0, \tag{7}$$

and a given proximal parameter $t_k > 0$, whenever $\|g_k\| > \varepsilon$, it is possible to compute s_k satisfying $\|s_k\| \leq t_k \|g_k\|$ and either

$$\|s_k\| = t_k \|g_k\| \tag{8}$$

or

$$\left\| \left(B_k + \frac{1}{t_k} I\right) s_k + g_k \right\| \leq \eta_k \|g_k\|. \tag{9}$$

These conditions are the heart of our approach.

The computed direction s_k is considered satisfactory if, given $\theta \in (0, 1)$, the angle-related condition

$$g_k^\top s_k \leq -\theta \|g_k\| \|s_k\| \tag{10}$$

holds, meaning that s_k is not only of descent but it is also uniformly bounded away from orthogonality with the gradient g_k. In case s_k does not satisfy (10), the parameter t_k must be reduced and another direction s_k verifying $\|s_k\| \leq t_k \|g_k\|$, and either (8) or (9), should be computed.

The acceptance of the trial point as the new iterate is conditioned to the fulfilment of a sufficient descent condition, established as follows. We define a sequence $\{C_k\}$ as in [26], namely, $C_0 = f(x_0)$, $Q_0 = 1$ and given $0 \leq \xi_{\min} \leq \xi_{\max} \leq 1$, we choose $\xi_k \in [\xi_{\min}, \xi_{\max}]$ and update

$$Q_{k+1} = \xi_k Q_k + 1, \tag{11}$$

and

$$C_{k+1} = \frac{\xi_k Q_k C_k + f(x_{k+1})}{Q_{k+1}}. \tag{12}$$

Let $m_k(s) := q_k(x_k + s) - f_k$. For a given $\gamma_1 \in (0, 1)$, if the Armijo-like sufficient descent condition $f(x_k + s_k) \leq C_k + \gamma_1 m_k(s_k)$ is satisfied then $x_{k+1} = x_k + s_k$. Otherwise, a line search along the direction s_k is performed. In this case, due to the possible non convexity of (5), instead of a plain backtracking, we have rested upon what we call a *surrogate model* with strictly positive curvature along s_k [22, 23] to predict the initial step length. More specifically, for a given nonnegative integer i, the surrogate model is defined by $\widehat{m}_k(s) := m_k(s) + \frac{i}{2}\|s\|^2$. We compute i as the smallest nonnegative integer that fulfills

$$s_k^\top \widehat{B}_k s_k = s_k^\top B_k s_k + i \|s_k\|^2 > 0, \tag{13}$$

and set

$$\sigma_k := \frac{-g_k^\top s_k}{s_k^\top \widehat{B}_k s_k} \tag{14}$$

as the optimal step length obtained from the surrogate model along s_k, that is, the minimizer of the scalar function $\varphi(\sigma) := \widehat{m}_k(\sigma s_k)$.

We choose $\beta \in (0,1)$ and define $\alpha^{(k)}$ as the largest scalar in $\{\sigma_k, \beta\sigma_k, \beta^2\sigma_k, \ldots\}$ for which

$$f(x_k + \alpha^{(k)} s_k) \leq C_k + \gamma_1 m_k(\alpha^{(k)} s_k). \tag{15}$$

Thus, the next iterate is $x_{k+1} = x_k + \alpha^{(k)} s_k$. The practical effects of the line search and of adopting such a choice for the step lengths are illustrated in our numerical experiments. It is worth mentioning that the scalar σ_k might be either below or above 1.

Combining the aforementioned ingredients with the necessary safeguards, the proposed algorithm is stated next:

Algorithm INPPA (Inexact Nonmonotone Proximal Point Algorithm)

Initialization. Given $\varepsilon > 0$, $\gamma_0, \gamma_1, \beta, \theta \in (0,1)$, $\gamma_2 > 1$, $0 < t_{\min} < t_{\max} < +\infty$, $0 \leq \xi_{\min} \leq \xi_{\max} \leq 1$, $x_0 \in I\!R^n$ and $t_0 \in [t_{\min}, t_{\max}]$, compute g_0 and B_0, set $k = 0$, $Q_0 = 1$, $C_0 = f(x_0)$ and define $\{\eta_k\} \subset I\!R_+$ satisfying (7).

While $\|g_k\| > \varepsilon$ do

1. Solve (5) approximately, obtaining a step s_k such that

   $\|s_k\| \leq t_k \|g_k\|$ and either (8) or (9) holds.

2. If $g_k^\top s_k > -\theta \|g_k\| \|s_k\|$ then
   $x_{k+1} = x_k$, $t_{k+1} = \gamma_0 \frac{\|s_k\|}{\|g_k\|}$, $Q_{k+1} = Q_k$, $C_{k+1} = C_k$ and $k = k+1$.

   Otherwise,

   2.1 set $\alpha^{(k)} = 1$.

   2.2 If $f(x_k + \alpha^{(k)} s_k) > C_k + \gamma_1 m_k(\alpha^{(k)} s_k)$ then compute i satisfying (13) and $\alpha^{(k)}$ the largest α in $\{\sigma_k, \beta\sigma_k, \beta^2\sigma_k, \ldots\}$ for which (15) is verified.

   2.3 Set $x_{k+1} = x_k + \alpha^{(k)} s_k$, compute g_{k+1} and B_{k+1}, set $t^{(k)} = \frac{\|\alpha^{(k)} s_k\|}{\|g_k\|}$ and $t_{k+1} = \min\{t_{\max}, \max\{t_{\min}, \gamma_2 t^{(k)}\}\}$, update Q_{k+1}, C_{k+1} using (11)-(12) and set $k = k+1$.

The Step 1 of the Algorithm INPPA will be accomplished by means of the truncated Newton algorithm of Dembo and Steihaug (cf. [7]), enhanced by the step length control of [24] (see the Appendix). By resting upon matrix-vector products, this approach is particularly suitable for large-scale and structured problems.

Although the *model for the objective decrease* $m_k(s)$ is used instead of the regularized model that defines the subproblem (5), notice that both quadratic functions are related by

$$\widetilde{q}_k(x_k + s) = m_k(s) + f_k + \frac{1}{2t_k}\|s\|^2. \tag{16}$$

The next assumptions are essential to establish the convergence properties of the Algorithm INPPA and have been used in the smooth unconstrained minimization scenario (cf. [22], see also [1]).

Assumption A1. The objective function f has a lower bound in $I\!R^n$ and its gradient ∇f is uniformly continuous on an open convex set Ω that contains the level set $L(x_0) = \{x \in I\!R^n | f(x) \leq f(x_0)\}$, where $x_0 \in I\!R^n$ is a given initial iterate.

Assumption A2. The matrices B_k are uniformly bounded, i.e., there exists a constant $M > 0$ such that, $\|B_k\| \leq M$ for all k.

Remark 3.1. *If the function f is twice continuously differentiable and the level set $L(x_0)$ is bounded , then Assumption A1 implies that the Hessian $\nabla^2 f$ is uniformly continuous and bounded on an open bounded convex set Ω that contains $L(x_0)$. Therefore, there exists L such that $\|\nabla^2 f(x)\| \leq L$, $\forall x \in \Omega$ and thus $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$, $\forall x, y \in \Omega$. Moreover, if ∇f is Lipschitz continuous on Ω then Assumption A1 holds.*

Remark 3.2. *Assumption A2, standard in the literature of smooth unconstrained minimization, implies that the Hessians of the surrogate model $\widehat{B}_k$ are also uniformly bounded. In fact, if $\|B_k\| \leq M$ for all k, then*

$$\|\widehat{B}_k\| = \|B_k + iI\| \leq 2M + 1 \tag{17}$$

because $M < i \leq M + 1$ implies that (13) holds.

Concerning the approximate solution of problem (5), the following algorithmic hypothesis is made:

Assumption A3. The direction s_k, computed in Step 1 of the Algorithm PPTLRS, is obtained by the Algorithm CG-Steihaug, stated in the Appendix.

For convenience, we define two index sets as follows:

$$\mathcal{I} = \{k : g_k^\top s_k \leq -\theta\|g_k\|\|s_k\|\} \text{ and } \mathcal{J} = \{k : g_k^\top s_k > -\theta\|g_k\|\|s_k\|\}.$$

We refer to x_{k+1} as a *successful iterate* if

$$x_{k+1} = x_k + \alpha^{(k)} s_k \neq x_k$$

and as an *unsuccessful iterate* if

$$x_{k+1} = x_k,$$

so that, according with the Step 2 of the Algorithm INPPA, the associated indices belong, respectively, to the sets $\mathcal{I}$ and $\mathcal{J}$.

3.1 The Algorithm INPPA is well defined

In this subsection we present the results concerning the consistency of the Algorithm INPPA.

The following lemma ensures that the direction computed at Step 1 of the Algorithm PPTLRS is of descent. It is worth pointing out that its proof is similar to the one of Lemma A.2 of [7] in case (9) occurs, and it is included here to encompass, as well, the analysis of directions s_k for which (8) holds.

Lemma 3.3. *Suppose that Assumptions A2 and A3 hold and* $0 < t_{\min} \le t_k \le t_{\max} < +\infty$. *Then, there exist positive constants* ρ_1 *and* ρ_2 *so that*

$$g_k^\top s_k \le -\rho_1 \|g_k\|^2 \tag{18}$$

and

$$\|s_k\| \le \rho_2 \|g_k\|. \tag{19}$$

Proof. The Algorithm CG-Steihaug has three possible reasons for stopping that shall be analyzed separately. The indices k and i denote, respectively, the outer (INPPA) and the inner (CG-Steihaug) iteration counters. Summarizing, for some $i \ge 0$, the direction computed at Step 1 of the Algorithm INPPA is such that

$$s_k = \begin{cases} s_{i+1} & \text{and} \quad \|r_{i+1}\| \le \eta_k \|g_k\| & \text{(from Step 4) or} \\ s_i + \tau d_i & \text{and} \quad \|s_k\| = t_k \|g_k\| \text{ with } \gamma_i > \epsilon \delta_i & \text{(from Step 3) or} \\ s_i + \tau d_i & \text{and} \quad \|s_k\| = t_k \|g_k\| \text{ with } \gamma_i \le \epsilon \delta_i & \text{(from Step 2).} \end{cases}$$

First, in case s_k has been computed at Step 4 of the Algorithm CG-Steihaug then it satisfies

$$s_k = s_{i+1} = \sum_{j=0}^{i} \alpha_j d_j. \tag{20}$$

Notice that the relationship $d_j^\top q_j > \epsilon \delta_j$ holds for all $j = 0, \dots, i$, where $q_j = \left(B_k + \frac{1}{t_k} I\right) d_j$, so that Theorem 7.1 (Appendix) applies. From (52), it follows that

$$\alpha_j = \frac{d_j^\top r_0}{d_j^\top q_j}. \tag{21}$$

Hence using (20) and (21) we obtain

$$s_k^\top g_k = -s_{i+1}^\top r_0 = -\sum_{j=0}^{i} \frac{(d_j^\top r_0)^2}{d_j^\top q_j} \le \frac{-d_0^\top r_0}{d_0^\top q_0} d_0^\top r_0. \tag{22}$$

But $d_0 = r_0 = g_k$ and

$$\left\| B_k + \frac{1}{t_k} I \right\| \le \|B_k\| + \frac{1}{t_k} \le M + \frac{1}{t_{\min}} = \frac{M t_{\min} + 1}{t_{\min}}, \tag{23}$$

so that

$$\frac{d_0^\top r_0}{d_0^\top q_0} = \frac{d_0^\top d_0}{d_0^\top \left(B_k + \frac{1}{t_k} I\right) d_0} \ge \frac{1}{\left\| B_k + \frac{1}{t_k} I \right\|} \ge \frac{t_{\min}}{M t_{\min} + 1}. \tag{24}$$

Therefore, (18) follows from (22) and (24), with

$$\rho_1 := \frac{t_{\min}}{M t_{\min} + 1}. \tag{25}$$

Also, note from (20) and (21) that

$$s_{i+1} = \sum_{j=0}^{i} \frac{d_j^\top r_0}{d_j^\top q_j} d_j = \left[\sum_{j=0}^{i} \frac{1}{d_j^\top q_j} d_j d_j^\top \right] r_0.$$

Hence, using (55) we have

$$\|s_k\| = \|s_{i+1}\| \leq \left[\sum_{j=0}^{i} \frac{d_j^\top d_j}{|d_j^\top q_j|}\right] \|r_0\| < \frac{i+1}{\epsilon}\|r_0\| = \frac{i+1}{\epsilon}\|g_k\| \leq \frac{n}{\epsilon}\|g_k\|.$$

Because we have $\|s_k\| \leq t_k\|g_k\| \leq t_{\max}\|g_k\|$ as well, the desired result (19) holds with

$$\rho_2 := \min\{n/\epsilon, t_{\max}\}.$$

Now, assume that s_k has been computed at Step 3 of the Algorithm CG-Steihaug. Then $s_k = s_i + \tau d_i$ and $\gamma_i = d_i^\top q_i > \epsilon\delta_i$, so that Theorem 7.1 also holds and the previous reasoning applies for both s_i and s_{i+1}. Therefore, inequalities (18) and (19) are valid along the segment $[s_i, s_{i+1}]$, and particularly at s_k.

Finally, if s_k has been computed at Step 2 of the Algorithm CG-Steihaug then $s_k = s_i + \tau d_i$ and $\gamma_i = d_i^\top q_i \leq \epsilon\delta_i$, so that d_i is safely not a direction of positive curvature. Consequently, the quadratic model decreases along this direction and the result is true in this case as well. □

Remark 3.4. *The two inequalities proved in Lemma 3.3 ensure that the sequence $\{s_k\}$ generated by the Algorithm* INPPA *is* gradient related *(cf. [4, p.35]). In [26], the inequalities (18) and (19) are included as a* Direction Assumption*, essential for the global convergence result. In [8], the authors assume (19) as a requirement on the step to prove global convergence. Likewise, in [2], the relationship (19) is required upon the trial step as the additional assumption (H3) to reach the global convergence of the algorithm.*

The next lemma establishes that the parameter σ_k has a lower bound for every k.

Lemma 3.5. *Under Assumptions A2 and A3, the parameter σ_k defined in (14) is bounded away from zero for all k.*

Proof. From (14), (18) and (19) we have that

$$\sigma_k = \frac{-g_k^\top s_k}{s_k^\top \widehat{B}_k s_k} \geq \frac{\rho_1\|g_k\|^2}{s_k^\top \widehat{B}_k s_k} \geq \frac{\rho_1\|s_k\|^2}{\rho_2^2 s_k^\top \widehat{B}_k s_k}.$$

Now, using (17) it follows from the previous inequality that

$$\sigma_k \geq \frac{\rho_1}{\rho_2^2\|\widehat{B}_k\|} \geq \frac{\rho_1}{\rho_2^2\widehat{M}} \tag{26}$$

where $\widehat{M} = 2M + 1$. □

The next result ensures that, if the current iterate is nonstationary, the Algorithm INPPA eventually reaches the Step 2.2.

Lemma 3.6. *Suppose that Assumptions A2 and A3 hold and the sequence $\{x_k\}$ is generated by the Algorithm* INPPA*. For every $k \in I\!N$, if $g_k \neq 0$ then there exists a nonnegative integer m such that $k + m \in \mathcal{I}$.*

Proof. Let x_k be such that $g_k \neq 0$. If $t_k \leq \rho_1/\theta$ then, by Lemma 3.3, as s_k is such that $\|s_k\| \leq t_k\|g_k\|$, we have

$$-\theta\|s_k\|\|g_k\| \geq \frac{-\rho_1}{t_k}\|s_k\|\|g_k\| \geq -\rho_1\|g_k\|^2 \geq g_k^\top s_k,$$

so that $k \in \mathcal{I}$.

Now, if $t_k > \rho_1/\theta$ then after a finite number of consecutive reductions of the proximal parameter at Step 2, denoted by m, we obtain $t_{k+m} \leq \rho_1/\theta$, with $x_{k+m} = x_k$. Hence s_{k+m} is such that $-\theta\|s_{k+m}\|\|g_k\| \geq g_k^\top s_{k+m}$, implying that $k + m \in \mathcal{I}$. □

The next result highlights the features of the step computed by the Algorithm CG-Steihaug within our terminology.

Lemma 3.7. *Under Assumption A3 upon the current kth subproblem, let s_j, $j = 0, \ldots, i$, be the generated iterates of the Algorithm* CG-Steihaug *and $\bar{s}$ be the direction that verifies either (8) or (9). Then $\widetilde{q}_k(x_k + s_j)$ is strictly decreasing for $j = 0, \ldots, i$, and*

$$\widetilde{q}_k(x_k + \bar{s}) \leq \widetilde{q}_k(x_k + s_i).$$

Further, $\|s_j\|$ is strictly increasing for $j = 0, \ldots, i,$, and

$$\|\bar{s}\| > \|s_i\|.$$

Proof. See Theorem 2.1 of [24]. □

The next lemma will be useful not only for the consistency but also for the analysis of convergence of Algorithm INPPA. Ideas from the Lemma 2.3 of [22] are employed in the first part of the proof.

Lemma 3.8. *Assume that $k \in \mathcal{I}$ and $0 < \alpha \leq \max\{\sigma_k, 1\}$.*

i) If $\sigma_k \geq 1$ then

$$m_k(\alpha s_k) \leq \frac{1}{2}\alpha g_k^\top s_k < 0. \tag{27}$$

ii) If $\sigma_k < 1$ then

$$m_k(\alpha s_k) \leq \frac{-1}{2t_k}(\alpha\|s_k\|)^2 < 0. \tag{28}$$

Proof. First, if $\sigma_k \geq 1$, using (13), the fact that $\alpha \leq \sigma_k$, the definition of σ_k, and $k \in \mathcal{I}$ we have

$$\begin{aligned} m_k(\alpha s_k) &= \alpha g_k^\top s_k + \tfrac{1}{2}\alpha^2 s_k^\top B_k s_k \\ &\leq \alpha g_k^\top s_k + \tfrac{1}{2}\alpha^2 s_k^\top \widehat{B}_k s_k \\ &\leq \alpha g_k^\top s_k - \tfrac{1}{2}\alpha g_k^\top s_k = \tfrac{1}{2}\alpha g_k^\top s_k < 0, \end{aligned}$$

so that (27) holds.

Now, assume that $\sigma_k < 1$ and let $\alpha \in (0, 1]$. From Lemma 3.7, the regularized quadratic model strictly decreases along the segment that joins x_k and $x_k + s_k$, that is, $\widetilde{q}_k(x_k + \alpha s_k) < \widetilde{q}_k(x_k) = f_k, \forall \alpha \in (0, 1]$. Therefore, from (16) we obtain

$$\widetilde{q}_k(x_k + \alpha s_k) - f_k = m_k(\alpha s_k) + \frac{1}{2t_k}(\alpha\|s_k\|)^2 < 0.$$

Hence, (28) is verified, completing the proof. □

The next result establishes the monotonicity of the auxiliary sequence $\{C_k\}$ that controls the nonmonotone line search, and completes the analysis of the consistency of the Algorithm INPPA.

Proposition 3.9. *Let $\{x_k\}$ be the sequence generated by the Algorithm* INPPA*. Then for all k we have*

$$f_{k+1} \le C_{k+1} \le C_k, \quad \forall k \in I\!N. \tag{29}$$

Moreover, the Step 2.2 of the Algorithm INPPA *is well defined for each $k \in \mathcal{I}$.*

Proof. The result will be proved by induction. For $k = 0$, we have $C_0 = f_0$ from the initialization. Besides, C_1 is a convex combination between f_1 and C_0, thus we have either

$$f_1 \le C_1 \le C_0 \tag{30}$$

or

$$C_0 \le C_1 \le f_1. \tag{31}$$

If $0 \in \mathcal{J}$ then the inequalities (30) hold as equalities. In case $0 \in \mathcal{I}$, by Lemma 3.8, we have

$$m_0(\alpha s_0) < 0, \forall \alpha \in (0, \max\{\sigma_0, 1\}].$$

If

$$f(x_0 + s_0) \le C_0 + \gamma_1 m_0(s_0) \tag{32}$$

then $x_1 = x_0 + s_0$ and $f_1 < C_0$, that contradicts (31). Hence, the relationship (30) holds. If (32) is not valid, notice that

$$f(x_0 + \alpha s_0) = f(x_0) + \alpha s_0^\top \nabla f(x_0) + o(\alpha) = C_0 + \alpha s_0^\top g_0 + o(\alpha),$$

where $\lim_{\alpha \downarrow 0} \frac{o(\alpha)}{\alpha} = 0$. For α small enough, the term $\alpha^2 s_0^\top B_0 s_0/2$ of the quadratic model $m_0(\alpha s_0)$ is of order $o(\alpha)$. Therefore, because $s_0^\top g_0 < 0$, for α sufficiently small, it holds

$$f(x_0 + \alpha s_0) \le C_0 + \gamma_1 m_0(\alpha s_0). \tag{33}$$

For the given $\beta \in (0, 1)$, let $\alpha^{(0)}$ be the largest scalar that simultaneously belongs to the set $\{\sigma_0, \beta\sigma_0, \beta^2\sigma_0, \ldots\}$ and to the interval $(0, \max\{\sigma_0, 1\}]$ for which (33) is valid and define $x_1 = x_0 + \alpha^{(0)} s_0$, so that $f_1 < C_0$. Then (31) cannot occur and (30) holds. As a result, the Step 2.2 is well defined if $k = 0 \in \mathcal{I}$.

Assume, as the inductive hypothesis, that

$$f_k \le C_k \le C_{k-1}, \tag{34}$$

and the Step 2.2 is well defined, and thus a successful iterate x_k is computed at Step 2.3 if $k - 1 \in \mathcal{I}$.

By (11)-(12), $C_{k+1} = (\xi_k Q_k C_k + f_{k+1})/(\xi_k Q_k + 1)$, that is, C_{k+1} is a convex combination between C_k and f_{k+1}. Therefore, we have either

$$C_k \le C_{k+1} \le f_{k+1} \tag{35}$$

or

$$f_{k+1} \le C_{k+1} \le C_k. \tag{36}$$

If $k \in \mathcal{J}$ then, from (34) we have $f_{k+1} = f_k \leq C_k$, so the inequalities (35) can only hold as equalities and thus (36) also holds. Now, assuming that $k \in \mathcal{I}$ then, by Lemma 3.8,

$$m_k(\alpha s_k) < 0, \forall \alpha \in (0, \max\{\sigma_k, 1\}].$$

Reasoning as before, there exists a successful iterate $x_{k+1} = x_k + \alpha^{(k)} s_k$, with either $\alpha^{(k)} = 1$ or $\alpha^{(k)}$ the largest scalar in $\{\sigma_k, \beta\sigma_k, \beta^2\sigma_k, \ldots\}$, such that

$$f_{k+1} = f(x_k + \alpha^{(k)} s_k) \leq C_k + \gamma_1 m_k(\alpha^{(k)} s_k).$$

Therefore, $f_{k+1} < C_k$, so that (35) is not valid and (36) holds. Consequently, x_{k+1} is well defined for $k \in \mathcal{I}$ by the Steps 2.1-2-3 and the proof is complete. □

4 Global convergence

In this section we analyze the global convergence of Algorithm INPPA, employing elements from [26]. The notation adopted for updating the proximal parameter along the successful iterations is extended to encompass the unsuccessful iterations as well:

$$t^{(k)} := \begin{cases} \frac{\|\alpha^{(k)} s_k\|}{\|g_k\|}, & k \in \mathcal{I} \\ \frac{\|s_k\|}{\|g_k\|}, & k \in \mathcal{J}. \end{cases} \tag{37}$$

The following result guarantees that the whole sequence $\{t^{(k)}\}_{k \in \mathbb{N}}$ is bounded away from zero and it is essential in the analysis of convergence of Algorithm INPPA.

Lemma 4.1. *Suppose that Assumptions A1, A2 and A3 hold and $0 < t_{\min} \leq t_k \leq t_{\max} < +\infty$. For k large enough,*

i) if $k \in \mathcal{I}$ and s_k satisfies (8) then

$$t^{(k)} \geq \frac{\bar{\beta}\rho_1 t_{\min}}{\rho_2^2 \widehat{M}}; \tag{38}$$

ii) if $k \in \mathcal{I}$ and s_k satisfies (9) then

$$t^{(k)} \geq \frac{\bar{\beta}\rho_1^2}{2\rho_2^2 \widehat{M}}, \tag{39}$$

where $\widehat{M} = 2M + 1$ and $\bar{\beta}$ is the largest scalar in $\{1, \beta, \beta^2, \ldots\}$ for which (15) is verified with $\alpha^{(k)} = \bar{\beta}\sigma_k$. Moreover, if $k \in \mathcal{J}$, for s_k satisfying (8) or (9) it holds

$$t^{(k)} \geq \frac{t_{\min}}{2(M t_{\min} + 1)}. \tag{40}$$

Proof. First notice that $\bar{\beta}$ is well defined due to the uniformity of ∇f from Assumption A1. Consider $k \in \mathcal{I}$ and s_k satisfying (8). So, using the definition of $\bar{\beta}$, we have from (37) that

$$t^{(k)} = \frac{\|\alpha^{(k)} s_k\|}{\|g_k\|} = \alpha^{(k)} t_k \geq \bar{\beta}\sigma_k t_k.$$

By (26) and $t_{\min} \le t_k$ we obtain (38).

Observe that if s_k satisfies (9), for any k, then

$$\|g_k\| - \left\| \left(B_k + \frac{1}{t_k} I\right) s_k \right\| \le \left\| \left(B_k + \frac{1}{t_k} I\right) s_k + g_k \right\| \le \eta_k \|g_k\|,$$

or, equivalently, due to (23),

$$(1-\eta_k)\|g_k\| \le \left\| \left(B_k + \frac{1}{t_k} I\right) s_k \right\| \le \|B_k s_k\| + \frac{\|s_k\|}{t_k} \le \left(M + \frac{1}{t_k}\right) \|s_k\|,$$

and hence, using (25) we have

$$(1-\eta_k)\rho_1 = \frac{(1-\eta_k)t_{\min}}{Mt_{\min}+1} \le \frac{(1-\eta_k)t_k}{Mt_k+1} \le \frac{\|s_k\|}{\|g_k\|}. \tag{41}$$

Moreover, assuming that k is large enough so that $\eta_k \le 1/2$, it follows that $1-\eta_k \ge 1/2$.

Therefore, from (41), for $k \in \mathcal{I}$ and s_k satisfying (9) we have

$$\frac{\rho_1}{2} \le \frac{\|s_k\|}{\|g_k\|}. \tag{42}$$

Thus, the bound (39) is obtained from (37), the definition of $\bar\beta$, (42) and (26):

$$t^{(k)} = \frac{\|\alpha^{(k)} s_k\|}{\|g_k\|} \ge \frac{\rho_1 \bar\beta \sigma_k}{2} \ge \frac{\bar\beta \rho_1^2}{2\widehat{M}\rho_2^2}.$$

Now, let $k \in \mathcal{J}$ and s_k satisfying (8). Then, the bound (40) follows directly from (37) as $0 < t_{\min} \le t_k = t^{(k)}$.

Finally, assume that $k \in \mathcal{J}$ and s_k satisfies (9). In this case, from (37), (41) and for k large enough we obtain

$$t^{(k)} = \frac{\|s_k\|}{\|g_k\|} \ge \frac{t_{\min}}{2(Mt_{\min}+1)},$$

what concludes the proof. □

Remark 4.2. *The bound (40) obtained in Lemma 4.1 is an additional property of the generated sequence, included for completeness. It is not essential for the convergence analysis, that solely rests upon the bounds (38) and (39).*

The global convergence of Algorithm INPPA is established next.

Theorem 4.3. *Suppose that Assumptions A1, A2 and A3 hold,* $0 \le \xi_{\min} \le \xi_k \le \xi_{\max} \le 1$, $0 < t_{\min} \le t_k \le t_{\max}$ *and that* $\|\nabla f(x_k)\| \ne 0$, *for all* $k \in I\!N$. *Then the sequence* $\{x_k\}$ *generated by the Algorithm* INPPA *has the property that*

$$\liminf_{k\to\infty} \|\nabla f(x_k)\| = 0. \tag{43}$$

Moreover, if $\xi_{\max} < 1$, *then*

$$\lim_{k\to\infty} \nabla f(x_k) = 0. \tag{44}$$

Proof. Initially note that according with the updating scheme of the Algorithm INPPA, as $\|\nabla f(x_k)\| \neq 0$ for all k, it is sufficient to analyze the convergence of the generated infinite sequence of distinct points, identified as $\{x_k\}_{k\in I\!N} = \{x_k\}_{k\in\mathcal{I}}$. From Lemma 3.6 we know that the set $\mathcal{I}$ is infinite.

First we will prove that (43) holds by combining the cases (i) and (ii) of Lemma 4.1 with the two bounds of Lemma 3.8. Indeed, due to $k \in \mathcal{I}$, the fact that $\alpha^{(k)}\|s_k\| = t^{(k)}\|g_k\|$ (cf. Step 2.3) and from Lemmas 3.8 and 4.1, for k large enough,

$$f_{k+1} \leq C_k - \tau\|g_k\|^2, \tag{45}$$

where

$$\tau = \begin{cases} \frac{\gamma_1\theta\bar{\beta}\rho_1 t_{\min}}{2\rho_2^2\widehat{M}} & \text{if (8) holds and } \alpha^{(k)} \leq \sigma_k, \\ \frac{\gamma_1}{2t_{\max}}\left(\frac{\bar{\beta}\rho_1 t_{\min}}{\rho_2^2\widehat{M}}\right)^2 & \text{if (8) holds and } \alpha^{(k)} \leq 1, \\ \frac{\gamma_1\theta\bar{\beta}\rho_1^2}{4\rho_2^2\widehat{M}} & \text{if (9) holds and } \alpha^{(k)} \leq \sigma_k, \\ \frac{\gamma_1}{8t_{\max}}\left(\frac{\bar{\beta}\rho_1^2}{\rho_2^2\widehat{M}}\right)^2 & \text{if (9) holds and } \alpha^{(k)} \leq 1. \end{cases}$$

Combining (11), (12) and (45), we find

$$C_{k+1} \leq \frac{\xi_k Q_k C_k + C_k - \tau\|g_k\|^2}{Q_{k+1}} = C_k - \frac{\tau\|g_k\|^2}{Q_{k+1}}. \tag{46}$$

From (29), the sequence $\{C_k\}$ is nonincreasing. By Assumption $A1$, we have that $\{C_k\}$ is convergent. It follows from (46) that

$$\sum_{k=0}^{\infty} \frac{\|g_k\|^2}{Q_{k+1}} < \infty. \tag{47}$$

By (11), the initialization $Q_0 = 1$, and the fact that $\xi_k \in [0, 1]$ we have

$$Q_{k+1} = 1 + \sum_{j=0}^{k}\prod_{i=0}^{j}\xi_{k-i} \leq k + 2. \tag{48}$$

So, if $\|g_k\|$ were bounded away from zero, (47) would be violated since $Q_{k+1} \leq k + 2$ by (48). Therefore, (43) holds.

Finally, if $\xi_{\max} < 1$, then by (48),

$$Q_{k+1} = 1 + \sum_{j=0}^{k}\prod_{i=0}^{j}\xi_{k-i} \leq 1 + \sum_{j=0}^{k}\xi_{\max}^{j+1} \leq \sum_{j=0}^{\infty}\xi_{\max}^{j} = \frac{1}{1-\xi_{\max}}.$$

Consequently, (47) implies (44) and the theorem is proved. □

5 Numerical experiments

To investigate the efficiency and the robustness of the Algorithm INPPA, we have implemented it in `fortran` and solved selected problems from the `CUTEr` collection [13]. They are not exhaustive, but were prepared to illustrate and highlight the main features of the proposed algorithm. The comparisons are based on data provided in the original references, justifying the adopted comparative measures in each case

We have four sets of experiments. In the first one, we have made a comparative analysis of the performance of the Algorithm INPPA for solving selected difficult problems reported in [11]. In the second set of experiments, we have investigated the effect of the nonmonotone acceptance criterion (15), by adopting distinct strategies to control the degree of nonmonotonicity. In the third set, we have analyzed the effect of the line search based on (14) as compared with the backtracking scheme that simply decreases the initial step length set as $\alpha^{(k)} = 1$ in Step 2.1; results without performing any line search were included as well. Finally, in the fourth set of experiments, we have compared the performance of the Algorithm INPPA with the results of the filter trust-region algorithms of Gould, Sainvitu and Toint [14] and of Fatemi and Mahdavi-Amiri [10], to assess the tuning of the combined inexact and nonmonotone proximal approach for solving a more general family of problems.

The tests were performed using the `gfortran` compiler (32-bits), version gcc-4.6, in a notebook Sony Vaio VGN-SR140E, Intel Centrino 2, 2.26GHz, RAM of 3 Gb and Cache of 3Mb. The algorithmic parameters were set as follows:

a. The proximal parameter is initialized as $t_0 := 1$ and updated within the problem-depending bounds $t_{\min} := \min\{10^{-4}, \|g_0\|^{-1}\}$ and $t_{\max} := \max\{10^4, \|g_0\|\}$.

b. The forcing sequence (7) is set as in [7]: $\eta_k := \min\{1/k, \|g_k\|\}$.

c. We have used the exact second order derivatives, i.e. $B_k := \nabla^2 f(x_k)$.

d. In the computation of the parameter σ_k defined in (14), the integer i in (13) is set to 1 if $s_k^T B_k s_k \geq 0$ or computed as $\lceil -s_k^\top B_k s_k / s_k^\top s_k \rceil$, if $s_k^T B_k s_k < 0$ and $-s_k^\top B_k s_k / s_k^\top s_k \leq 10^9$. In case $-s_k^\top B_k s_k / s_k^\top s_k$ exceeds the upper bound 10^9, to avoid numerical overflow we simply use $\sigma_k = 1$.

e. We have defined the updating sequence for the nonmonotone line search parameter by the constant value $\xi_k \equiv 0.85$, as suggested in [26].

f. The tolerance for the absolute stopping criterion $\|g_k\| \leq \varepsilon$ is set to $\varepsilon = 10^{-6}$ (compatible with [11]). The maximum allowed number of iterations is $\max\{5000, 100n\}$. The remaining parameters are defined as $\gamma_0 = 0.1$, $\gamma_1 = 0.1$, $\gamma_2 = 10^2$, $\theta = 10^{-4}$ and $\beta = 0.5$.

5.1 Comparative results with the instances of [11]

The numerical results are presented in Tables 1 and 2. The former contains problems for which the pure quasi-Newton fails in [11]. The latter is constituted of ill-conditioned problems, addressed not only in [11] but also in [15]. We report the problem name, its dimension n, and #iter denotes the number of outer iterations, that coincides with the number of calls to the algorithm CG-Steihaug for approximately solving the subproblem (5). The number of functional evaluations, gradient (and Hessian) computations and total iterations of the algorithm CG-Steihaug are referred to as $\#f$, $\#gH$ and #CG, respectively. Notice that $\#f$ corresponds to the total number of computed trial points, whereas $\#gH$ is the number of distinct points of the generated sequence. We also report the demanded CPU time, in seconds.

Table 1: Performance of Algorithm INPPA on the instances of [11].

Problem	n	#iter	$\#f$	$\#gH$	#CG	CPU
DJTL	2	688	1842	689	1015	0.316
BROWNDEN	4	13	14	14	33	0.008
TOINTGOR	50	11	12	12	176	0.008
SENSORS	100	14	73	15	35	0.364
NCB20	210	34	68	35	709	0.376
BDQRTIC	1000	15	16	16	97	0.264
CRAGGLVY	2000	17	18	18	196	0.480
FREUROTH	5000	17	29	18	57	2.084
BROYDN7D	5000	217	990	218	3889	30.65
SINQUAD	5000	17	50	15	39	2.196
SCHMVETT	5000	51	236	52	449	7.556

Table 2: Performance of Algorithm INPPA on the instances of [11] and [15].

Problem	n	#iter	$\#f$	$\#gH$	#CG	CPU
SPARSINE	1000	88	732	73	11586	3.548
SPARSINE	2000	105	904	87	38418	21.05
NONDQUAR	500	160	161	161	10375	0.572
NONDQUAR	1000	175	176	176	10954	2.092
EIGENALS	420	55	63	56	1016	8.376
EIGENBLS	420	175	218	176	13630	32.87
EIGENCLS	462	313	1168	314	21519	73.23
NCB20	510	59	163	54	843	1.444

Whenever $\#f = \#gH$, the step length $\alpha^{(k)} = 1$ has always been accepted, indicating the generation of favourable directions by the Algorithm INPPA. In Table 1 this ocurred for 4 out of the 11 problems, whereas in Table 2 this happened for 2 out of the 8 problems. By computing the ratio $(\#f - \#gH)/\#gH$ for each problem, we have measured the average additional number of functional evaluations necessary to obtain a successful iteration. The worst case scenario was 3.9 for the problem SENSORS of Table 1 and 9.4 for the problem SPARSINE, $n = 2000$, of Table 2.

We should also observe that $\#\text{iter} + 1 - \#gH$ indicates the cardinality of the set $\mathcal{J}$, that is, how many unsuccessful iterations were generated. This value is zero for 10 out of the 11 problems of Table 1, and 5 out of the 8 problems of Table 2. Moreover, the largest values for Tables 1 and 2, respectively, were 3 (problem SINQUAD) and 19 (problem SPARSINE, $n = 2000$).

The cost of the Algorithm INPPA is contextualized in terms of the functional evaluations demanded. The obtained values for $\#f$ and $\#gH$ are compared with the number of simulations #sim reported in [11], that corresponds to the total $\#f = \#g$, as the limited-memory quasi-Newton solver m1qn3 used in [11] systematically computes ∇f together with f.

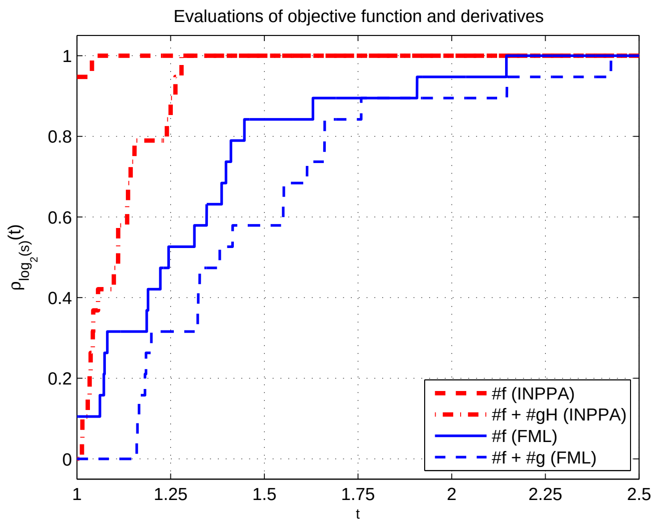

Figure 1: Performance profiles for the functional evaluations demanded by INPPA and those reported in [11] for FML.

The performance profiles [9] of Figure 1 depict the $\log_2$ scaled curves for the aforementioned measures for the whole set of test problems of Tables 1 and 2. We have jointly plot the four combinations $\#f$ and $\#f + \#g$ for INPPA and FML (that stands for Fuentes, Malick and Lemaréchal). The graphs reveal that INPPA is more efficient than FML, possibly due to our usage of the true Hessian in the quadratic models. In terms of robustness, both strategies are able to successfully solve this joint set of difficult problems.

5.2 Assessing the acceptance criterion

To analyze the effect of the default updating scheme within the nonmonotone strategy ($\xi_k \equiv 0.85$), we have compared its results with those of the monotone strategy ($\xi_k \equiv 0$) and of a very tolerant one ($\xi_k \equiv 0.99$), for the problems reported in Tables 1 and 2. We have also devised and tested an expression for obtaining a variable and less tolerant strategy as the iterations proceed, that starts from $\xi_0 = 0.85$, namely

$$\xi_k = 0.75 \exp\left(-\left(\frac{k}{15}\right)^2\right) + 0.1. \tag{49}$$

The bounds $0 < \xi_{\min} = 0.1$ and $\xi_{\max} = 0.85 < 1$ are safely ensured.

The $\log_2$ scaled performance profiles are depicted in Figure 2. The graphs on the right provide a zoomed view of the ones on the left, so that the fact that the default adopted strategy is slightly more efficient than its counterparts can be better visualized. The robusteness of the nonmonotone strategy can be observed from the graphs on the left, in which it is evident that the solution of these difficult problems is clearly benefited by relaxed acceptance criteria. The Armijo-like line search ($\xi_k \equiv 0$) has stopped due to lack of progress (measured by $|m_k(s)| \leq \varepsilon^{2.5} = 10^{-15}$) for two problems (FREUROTH and SINQUAD). It is worth mentioning that the performance profiles for the corresponding total iterations of the algorithm CG-Steihaug, another measure of spent effort, are very similar to those of Figure 2, so they are omitted.

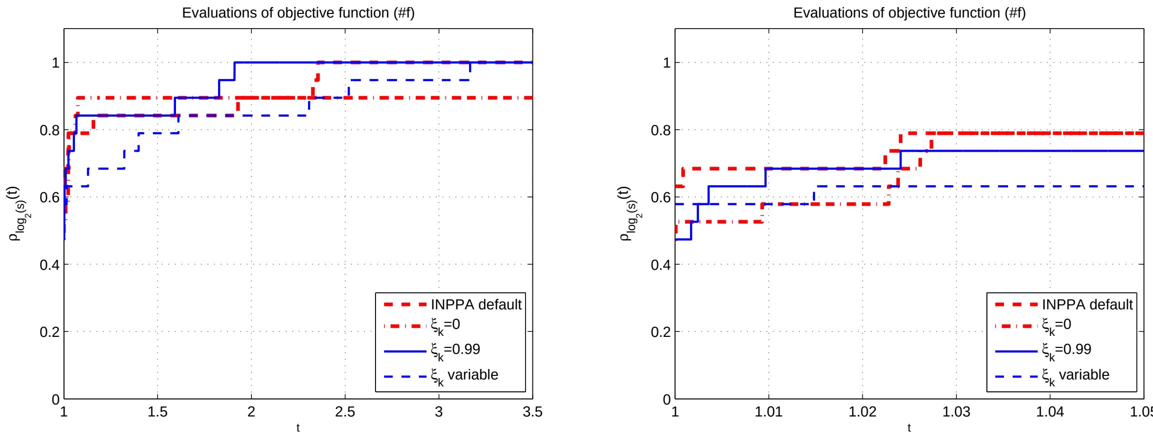

Figure 2: Performance profiles of the objective function evaluations for the default updating of the Algorithm INPPA ($\xi_k \equiv 0.85$) compared with the Armijo-like line search ($\xi_k \equiv 0$), a strongly nonmonotone version ($\xi_k \equiv 0.99$) and the variable updating of (49).

5.3 Analyzing the line search scheme

The effect of adopting the line search based on the parameter σ_k defined by (14) was assessed by the comparison with the plain choice $\sigma_k \equiv 1$ and without performing any line search, for the problems reported in Tables 1 and 2. The results are shown in the $\log_2$ scaled performance profiles of Figure 3. Clearly, for this set of test problems, the variable choice for σ_k has generated a more efficient version of the algorithm. Moreover, the line search strategy has a significant impact, not only improving efficiency but also robustness. Without performing any line search, the algorithm could not solve 10 out of the 19 problems of Tables 1 and tab:2, that stopped by reaching the maximum allowed number of iterations. This lack of robustness shows the role of the line search procedure to ensure the global convergence of the Algorithm INPPA.

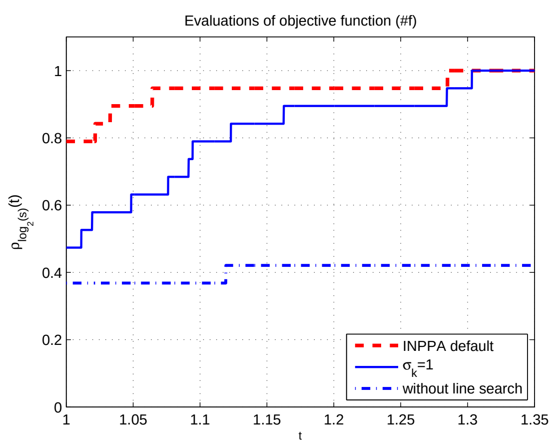

Figure 3: Performance profiles of the default version of the Algorithm INPPA compared with both the usage of $\sigma_k \equiv 1$ and inhibiting the line search.

5.4 Further comparison with results of [14] and [10]

With the aim of widen the analysis of the performance of the Algorithm INPPA for solving general unconstrained minimization problems, we have selected 34 unconstrained problems from the `CUTEr` collection, for which the number of iterations of conjugate gradient is reported in [10] for the two filter trust-region algorithms FTRA [14] and FTRA-2 [10], specially developed for tackling nonconvex problems as well. The default version of the Algorithm INPPA was run, adopting the same parameters described at the beginning of this section, except for the stopping criterion, that was changed to $\|g_k\| \leq \sqrt{n}\varepsilon$, $\varepsilon = 10^{-6}$ (compatible with [10], where the exact second order derivatives are used as well). The numerical results are presented in Table 3, that follows the pattern of the previous tables. The problem dimensions are those established in [14].

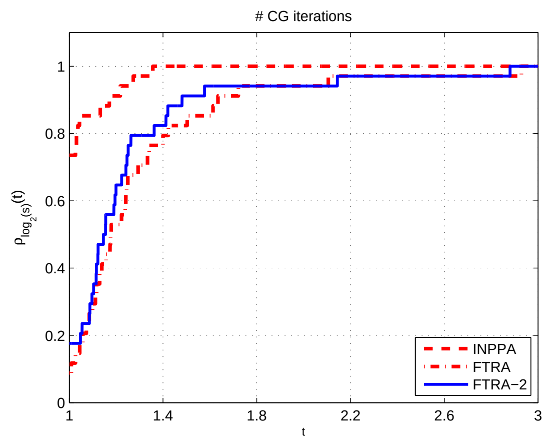

Figure 4: Performance profiles of the number of iterations of the Algorithm CG-Steihaug of the default version of the Algorithm INPPA compared with the number of iterations of conjugate gradient demanded by the filter trust-region algorithms of [14] (FTRA) and [10] (FTRA-2).

Analyzing the results of Table 3, we shall notice that the the ratios $(\#f - \#gH)/\#gH$ remain below 1 for 29 out of the 34 problems, being inferior to 0.5 for 21 problems, revealing that, in general, very few additional objective function evaluations are demanded to produce a successful iteration. The worst case is 6.2, for problem `HAIRY`. Moreover, for 27 out of the 34 problems, $\#\text{iter} + 1 - \#gH$ is zero, so that no unsuccessful iterations were generated. The largest value for this indicator is 8 (problem `NCB20`).

Adopting #CG as the measure of the effort employed by INPPA, and based on the data reported in [10], Figure 4 depicts the comparative $\log_2$ scaled profiles. The plots reveal that for this set of test problems, the Algorithm INPPA is equally robust, and clearly more efficient than the filter-trust-region counterparts.

6 Final remarks

We have introduced an inexact nonmonotone proximal point algorithm for unconstrained smooth minimization. It is not a general-purpose algorithm, being mainly developed for ill-conditioned or singular problems.The novelty is the combination of: (i) a framework for updating the proximal parameter; (ii) inexact criteria for approximately solving the not necessarily convex subproblems; (iii) a nonmonotone criterion for accepting the iterate. We have analyzed its consistency

Table 3: Performance of Algorithm INPPA on instances of [14] and [10].

Problem	n	#iter	$\#f$	$\#gH$	#CG	CPU
CHNROSNB	10	25	28	26	151	0.012
CHNROSNB	50	42	50	43	540	0.020
COSINE	10000	8	12	9	12	3.872
DENSCHND	3	39	45	40	96	0.016
DIXMAANB	9000	8	9	9	9	3.196
DIXMAANG	9000	18	21	19	754	7.340
DIXMAANH	9000	61	112	61	1161	21.33
DIXMAANL	9000	68	82	64	2901	24.54
EIGENALS	2550	79	87	80	3150	2428.
ERRINROS	50	77	89	78	607	0.028
EXPFIT	2	11	22	12	17	0.008
EXTROSNB	1000	217	219	218	2291	1.852
FMINSRF2	5625	298	549	299	16669	54.55
FMINSURF	49	70	124	71	1167	0.048
FREUROTH	5000	16	28	17	50	1.976
GENROSE	500	256	1568	256	3144	0.668
GROWTHLS	3	277	339	278	718	0.084
GULF	3	97	109	98	251	0.048
HAIRY	2	40	275	38	79	0.048
HATFLDD	3	20	21	21	50	0.012
HEART8LS	8	64	114	61	261	0.028
KOWOSB	4	18	27	19	59	0.012
MARATOSB	2	1136	2185	1137	2141	0.408
MSQRTALS	1024	903	1320	904	469660	2558.
MSQRTBLS	1024	422	571	423	203181	1176.
NCB20	5010	130	563	123	1079	43.61
NCB20B	5000	32	38	33	2039	29.70
PALMER7C	8	12	13	13	84	0.008
SNAIL	2	66	72	67	121	0.020
SPMSRTLS	4900	62	109	63	1584	8.500
TOINTPSP	50	19	62	20	125	0.016
VIBRBEAM	8	188	430	189	983	0.136
WOODS	10000	42	43	42	136	17.40
YFITU	3	282	287	283	783	0.080

and global convergence, resting upon standard assumptions. We have illustrated its numerical behaviour by means of a comparison with results from [11] using selected and difficult unconstrained problems from the CUTEr collection. Further, we have investigated the effects of the adopted nonmonotone acceptance criterion and of the employed line search scheme, together with a comparative analysis of the effort employed by the Algorithm CG-Steihaug against the conjugate-gradient iterations demanded by two filter trust-region algorithms. Our computational results exhibit a promising numerical performance for degenerate problems. Moreover, our approach seems to be quite well tuned: the overall demanded effort for the family of more general problems solved were surprisingly competitive.

7 Appendix

For the sake of completeness and to standardize the notation, we include next the algorithm proposed in [24] and the associate convergence result, following the notation of [7]. It has been applied to approximately minimize the current quadratic subproblem (5) in the Step 1 of Algorithm INPPA, with the input data $g := g_k$, $B := B_k + \frac{1}{t_k} I$ and $\Delta := t_k\|g_k\|$. The tolerance $\epsilon > 0$ is used to detect a potential direction of non positive curvature and $\eta := \eta_k$ is the current term of the forcing sequence that fulfills (7).

Algorithm CG-Steihaug

Given $g \in \mathbb{R}^n$, $B \in \mathbb{R}^{n\times n}$, $\Delta > 0$, $\epsilon > 0$ and $\eta > 0$, compute $s \in \mathbb{R}^n$ such that either $\|s\| = \Delta$ or $\|Bs + g\| \le \eta\|g\|$:

Step 1. Set $s_0 = 0,\ r_0 = -g,\ d_0 = r_0,\ \delta_0 = r_0^\top r_0$ and $i = 0$.

Step 2. Compute $q_i = Bd_i$ and $\gamma_i = d_i^\top q_i$.
If $\gamma_i \le \epsilon\delta_i$ then compute $\tau > 0$ so that $\|s_i + \tau d_i\| = \Delta$,
set $s = s_i + \tau d_i$ and return.

Step 3. Compute $\alpha_i = r_i^\top r_i/\gamma_i$ and $s_{i+1} = s_i + \alpha_i d_i$.
If $\|s_{i+1}\| \ge \Delta$ then compute $\tau > 0$ so that $\|s_i + \tau d_i\| = \Delta$,
set $s = s_i + \tau d_i$ and return.

Step 4. Compute $r_{i+1} = r_i - \alpha_i q_i$.
If $\|r_{i+1}\| \le \eta\|g\|$ then set $s = s_{i+1}$ and return.

Step 5. Compute $\beta_i = r_{i+1}^\top r_{i+1}/r_i^\top r_i$, $d_{i+1} = r_{i+1} + \beta_i d_i$ and
$\delta_{i+1} = r_{i+1}^\top r_{i+1} + \beta_i^2\delta_i$. Set $i = i + 1$ and go to Step 2.

Let $m(s) := g^\top s + \frac{1}{2}s^\top Bs$ and $[v, w]$ denote the space spanned by v and w.

Theorem 7.1. *If $d_i^\top Bd_i > \epsilon\delta_i$, $i = 0, 1, \ldots, \ell$, then*

$$d_i^\top Bd_j = 0, \qquad i \ne j, \qquad i, j = 0, 1, \ldots, \ell, \tag{50}$$

$$d_i^\top r_j = 0, \qquad i < j, \qquad i, j = 0, 1, \ldots, \ell + 1, \tag{51}$$

$$r_i^\top d_j = r_j^\top r_j = r_0^\top d_j, \qquad i \le j, \qquad i, j = 0, 1, \ldots, \ell, \tag{52}$$

$$[d_0, d_1, \ldots, d_\ell] = [g, Bg, \ldots, B^{\ell-1}g], \tag{53}$$

$$m(s_{\ell+1}) = \min\{m(s) : s \in [d_0, \ldots, d_\ell]\}, \tag{54}$$

$$\delta_i = d_i^\top d_i, \qquad i = 0, 1, \ldots, \ell. \tag{55}$$

Proof. See Theorem A.1 of [7]. □

Acknowledgements. We are thankful to Paulo J. S. Silva, for pointing us valuable suggestions that have improved the presentation of this manuscript. We are also indebted to Benar F. Svaiter, for fruitful discussions and to Abel S. Siqueira, for helping us with the installation of the `CUTEr`.

References

[1] Ahookhosh, M., Amini, K.: A nonmonotone trust region method with adaptive radius for unconstrained optimization problems. *Comput. Math. Appl.* **60**, 411–422 (2010)

[2] Ahookhosh, M., Amini, K.: An efficient nonmonotone trust-region method for unconstrained optimization. *Numer. Algor.* **59**, 523–540 (2012)

[3] Beck, A., Teboulle, M.: A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM J. Imaging Sci.* **2**(1), 183–2002 (2009)

[4] Bertsekas, D. P.: *Nonlinear Programming.* 2nd ed. Athena Scientific, Belmont, Massachusetts (1999)

[5] Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn.***3**(1), 1–122 (2011)

[6] Conn, A. R., Gould, N. I. M, Toint, Ph. L.: *Trust-region methods.* SIAM, Philadelphia (2000)

[7] Dembo, R.S., Steihaug, T.: Truncated Newton algorithms for large scale unconstrained optimization. *Math. Program.* **26**(2), 190–212 (1983)

[8] Deng, N. Y., Xiao, Y., Zhou, F. J.: Nonmonotonic trust region algorithm. *J. Optim. Theory Appl.* **76**, 259–285 (1993)

[9] Dolan, E. D., Moré, J. J.: Benchmarking optimization software with performance profiles. *Math. Program.* **91**, 201–213 (2002)

[10] Fatemi, M., Mahdavi-Amiri, N.: A filter trust-region algorithm for unconstrained optimization with strong global convergence properties. *Comput. Optim. Appl.* **52**(1), 239–266 (2012)

[11] Fuentes, M., Malick, J., Lemaréchal, C.: Descentwise inexact proximal algorithms for smooth optimization. *Comput. Optim. Appl.* **53**(3), 755–769 (2012)

[12] Gould, N. I. M., Lucidi, S., Roma, M., Toint, Ph. L.: Solving the trust-region subproblem using the Lanczos method. *SIAM J. Optim.* **9**(2), 504–525 (1999)

[13] Gould, N. I. M., Orban, D., Toint, Ph. L.: CUTEr and SifDec: A Constrained and Unconstrained Testing Environment, revisited. *ACM Trans. Math. Software.* **29**, 373–394 (2003)

[14] Gould, N. I. M., Sainvitu, C., Toint, Ph. L.: A filter-trust-region method for unconstrained optimization. *SIAM J. Optim.* **16**(2), 341–357 (2005)

[15] Hager, W.W., Zhang, H.: Self-adaptive inexact proximal point methods. *Comput. Optim. Appl.* **39**, 161–181 (2008)

[16] Humes Jr., C., Silva, P.J.S.: Inexact Proximal Point Algorithms and Descent Methods in Optimization. *Optim. Eng.* **6**, 257–271 (2005)

[17] Martinet, B.: Regularisation dinéquations variationelles par approximations successives. *Revue Française d'Informatique et de Recherche Opérationnelle.* **4**, 154–158 (1970)

[18] Parikh, N., Boyd, S.: Proximal Algorithms. *Found. Trends Optim.* **1**(3) 123–231 (2013)

[19] Rockafellar, R.T.: Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Math. Oper. Res.* **2**, 97–116 (1976)

[20] Rockafellar, R.T.: Monotone operators and the proximal point algorithm. *SIAM J. Control* **14**, 877–898 (1976)

[21] Sachs, E. W., Sachs, S. M.: Nonmonotone line searches for optimization algorithms. *Control Cybernet.* **40**(4) 1059–1075 (2011)

[22] Shi, Z.-J., Guo, J.-H.: A new trust region method for unconstrained optimization. *J. Comput. Appl. Math.* **213**(2), 509–520 (2008)

[23] Shi, Z.-J., Shen, J.: Convergence of nonmonotone line search method. *J. Comput. Appl. Math.* **193**(2), 397–412 (2006)

[24] Steihaug, T.: The conjugate gradient method and trust regions in large scale optimization. *SIAM J. Numer. Anal.* **20**(3), 626–637 (1983)

[25] Toint, Ph. L.: Towards an efficient sparsity exploiting Newton method for minimization. In: Duff, I. S. (ed.) *Sparse Matrices and Their Uses*, pp. 57–88. Academic Press, London (1981)

[26] Zhang, H., Hager, W.W.: A nonmonotone line search technique and its application to unconstrained optimization. *SIAM J. Optim.* **14**(4), 1043–1056 (2004)