An Approach to Variable Metric Bundle Methods

Claude Lemaréchal and Claudia Sagastizábal

INRIA, BP 105, 78153 Le Chesnay (France)

ABSTRACT

To minimize a convex function f, we state a penalty-type bundle algorithm, where the penalty uses a variable metric. This metric is updated according to quasi-Newton formulae based on Moreau-Yosida approximations of f. In particular, we introduce a "reversal" quasi-Newton formula, specially suited for our purpose. We consider several variants in the algorithm and discuss their respective merits. Furthermore, we accept a degenerate penalty term in the Moreau-Yosida regularization.

Key words. Bundle methods, convex optimization, mathematical programming, proximal point, quasi-Newton algorithms, variable metric.

AMS Subject Classification. Primary: 65K05. Secondary: 90C30, 90C25.

1 Introduction

This paper addresses the numerical minimization of a (finite-valued) convex function $f : \mathbb{R}^N \to \mathbb{R}$, characterized by a black box which, for any $x \in \mathbb{R}^N$, computes $f(x)$ and some subgradient $g(x) \in \partial f(x)$. Our approach employs bundle methods, so we briefly recall their basic principles here. At the current iteration of the algorithm, the black box has computed the sample values $f(y_i)$ and $g_i \in \partial f(y_i)$ for $i = 1, \ldots, k$; the *cutting-plane* model of f is then

$$\check{f}_k(y) := \max\{f(y_i) + \langle g_i, y - y_i\rangle \ : \ i = 1, \ldots, k\} .$$

In the cutting-plane method [CG59], [Kel60], the next iterate y_{k+1} is a minimizer of $\check{f}_k$. However this method is notoriously unstable (see an example of A.S. Nemirovski, described in Section XV.1.1 of [HULL93]). Bundle methods offer a stabilizing device based on the following ingredients:

(i) a sequence $\{x_n\}$ of stabilized iterates;
(ii) a test deciding whether a new stabilized iterate has been found and/or whether the model $\check{f}_k$ should be enriched;
(iii) a sequence $\{M_n\}$ of positive definite matrices defining a scalar product and its associated norm.

A number of different approaches have been developed according to the above principles. We give now a close description of one of them, namely the *proximal* form, which we consider in this paper:

(i) A *candidate* y^c is computed as the minimizer of the penalized model

$$\check{f}_k(y) + \tfrac{1}{2}\langle M_n(y - x_n), y - x_n\rangle . \tag{1}$$

(ii) A *nominal decrease*

$$\delta_n := f(x_n) - \check{f}_k(y^c) - \tfrac{1}{2}\langle M_n(y^c - x_n), y^c - x_n\rangle$$

controls the update of x_n and/or the enrichment of $\check{f}_k$. More specifically, a fixed parameter $m \in]0, 1[$ being chosen, we perform the descent test

$$f(y^c) \le f(x_n) - m\delta_n . \tag{2}$$

If (2) holds, then we set $x_{n+1} = y_{k+1} = y^c$; n and k are increased by 1. Otherwise n is kept fixed, we set $y_{k+1} = y^c$ and k is increased by 1; in some improved versions (see the considerations in §6 below), an additional test is made before increasing k.

(iii) The choices of the *norming* $\{M_n\}$ given so far in the literature are:

- an abstract sequence, as in [Lem78],
- $M_n \equiv \mathrm{I}$, as in [Kiw83],
- $M_n = \mu_n \mathrm{I}$, with heuristic rules for computing μ_n; see [Kiw90], [SZ92].

An essential feature of our present development consists of a quasi-Newton update of M_n [DM77] using the so-called Moreau-Yosida regularization ([Mor65], [Yos64]); we also pay some attention to the updates used by Shor [Sho85].

The next section is devoted to the Moreau-Yosida regularization; we recall a few results, slightly generalized in the sense that we admit semi-positive definite matrices M_n. Then we state the algorithm and give some of its basic properties. In §4 we consider some quasi-Newton formulae, which exploit two possible ideas:

- First, we choose a move in x and we compute the corresponding move in the gradient of a smooth function, namely the regularization $\check{F}_k$ coming out of (1).
- Second, a "reversal" idea starts from a move in the gradient space; a corresponding move in x is then computed to estimate the curvature of the smoothened objective function.

The last two sections assess the approach, they show that the second idea has several merits related to implementation and convergence issues.

2 Some Basic Results

Given a semi-positive definite matrix M, we denote by

$$H(x) := \inf \{h(y) + \tfrac{1}{2} \langle M(y-x), y-x \rangle\} \tag{3}$$

the corresponding Moreau-Yosida regularization of a function h. Allowing a degenerate M in the quadratic perturbation departs from the classical framework, but we show that the essential results concerning this regularization can be reproduced.

Notationally, block letters $H, F, G, \ldots$ will be used to designate the regularized versions of objects such as $h, f, g, \ldots$

Theorem 1. *Let h in (3) be a closed convex function.*

(i) If $\operatorname{dom} h^* \cap \operatorname{Im} M = \emptyset$ then $H(x) = -\infty$ for all x.

(ii) If $\operatorname{dom} h^* \cap \operatorname{Im} M \neq \emptyset$ then $H(x) > -\infty$ for all x, and H is a convex function defined on the whole of $\mathbb{R}^N$.

Assume case (ii) and denote by M^- the pseudo-inverse of M. The dual problem of (3)

$$\min_{g \in \operatorname{Im} M} [h^*(g) - \langle g, x \rangle + \tfrac{1}{2} \langle M^- g, g \rangle] = -H(x) \tag{4}$$

has a unique solution $G(x)$. The solution set in (3) is then

$$P(x) = \partial h^*(G(x)) \cap [x - M^- G(x) + \operatorname{Ker} M]; \tag{5}$$

when $\operatorname{ri} \operatorname{dom} h^* \cap \operatorname{Im} M \neq \emptyset$, or when h is polyhedral, this set is nonempty (for all x).

Proof. The function H is the infimum of a function that is jointly convex in (x,y); in view of §IV.2.4 of [HULL93], H is convex (in x); also $H(x)$ is clearly $< +\infty$ (for all x) but we have to cope with the case $H(x) = -\infty$.

For fixed x, denote by $y \mapsto \varphi(y)$ the objective function in (3); it is the sum of two closed convex functions, one of them is finite everywhere. The conjugates of these two functions are respectively (see Example X.1.1.4 in [HULL93], I is the indicator function)

$$h^*(g) \text{ and } \langle g, x\rangle + \tfrac{1}{2}\langle M^- g, g\rangle + \mathrm{I}_{\mathrm{Im}\, M}(g)\,,$$

which have respectively the subdifferentials

$$\partial h^*(g) \text{ and } \{x + M^- g\} + \mathrm{Ker}\, M\,. \tag{6}$$

According to Theorem X.2.3.2 of [HULL93] the infimal convolution

$$\varphi^*(s) = \inf_g [h^*(g) + \langle s-g, x\rangle + \tfrac{1}{2}\langle M^-(s-g), s-g\rangle + \mathrm{I}_{\mathrm{Im}\, M}(s-g)]$$

is a closed convex function, which is the conjugate of φ. It follows in particular that $\varphi^*(0) = -H(x)$; to say that this number is not $+\infty$ is to say that there is some $g \in \mathrm{dom}\, h^*$ such that $0 - g \in \mathrm{Im}\, M$; (i) and (ii) are proved.

Now (4), precisely, just expresses the relation $\varphi^*(0) = -H(x)$; observing that M^- is an isomorphism on $\mathrm{Im}\, M$, the infimum is attained at a unique $G(x)$.

Finally, the primal solution set $P(x)$ is $\partial\varphi^*(0)$; by Theorem XI.3.4.1 of [HULL93], this is (5). Under a suitable qualification assumption such as those stated, $\partial\varphi^*(g)$ is the sum of the subdifferentials in (6). Then the optimality condition $0 \in \partial\varphi^*(G(x))$ gives

$$0 \in \partial h^*(G(x)) - x + M^- G(x) + \mathrm{Ker}\, M\,;$$

this just expresses the nonemptiness of (5). □

Among other things, this result reveals an important property of our "extended" Moreau-Yosida regularization: finiteness of the value $H(x)$ depends only on the geometry of h and M, but not on the particular value of x. Furthermore, when h is polyhedral (the only case of interest to us), existence of a candidate $y^c \in P(x)$ in (1) also does not depend on the particular x.

Remark 2. Suppose that the dual solution $G(x)$ is available, together with a multiplier $\check{w} \in \mathrm{Ker}\, M$ of the constraint $g \in \mathrm{Im}\, M$. This means that there is some $\check{z} \in \partial h^*(G(x))$ such that $0 = \check{z} - x + M^- G(x) + \check{w}$, so that $\check{y} := x - M^- G(x) - \check{w} \in P(x)$; then it is rather clear that the whole $P(x)$ is the closed convex set

$$P(x) = \{y = x - M^- G(x) - w \;:\; w \in \mathrm{Ker}\, M \text{ and } h(y) \le h(\check{y})\}. \tag{7}$$

□

We now check that the well-known regularity properties of H are preserved.

Theorem 3. *Assume that, for all x, the infimum in (3) is attained on some nonempty set $P(x)$. Then the convex function H has at all x a gradient given by*

$$\nabla H(x) = G(x) = M(x - y), \quad \text{for arbitrary } y \in P(x), \tag{8}$$

where $G(x)$ solves (4). Furthermore, ∇H is Lipschitzian. More precisely, for all x_1, x_2:

$$\|\nabla H(x_1) - \nabla H(x_2)\|^2 \leq \frac{1}{\Lambda} \langle \nabla H(x_1) - \nabla H(x_2), x_1 - x_2 \rangle , \tag{9}$$

where Λ is the largest eigenvalue of M.

Proof. Under the stated conditions, Corollary VI.4.5.3 of [HULL93] can be applied. It gives $\nabla H(x) = M(x - y)$, no matter how y is chosen in the optimal set $P(x)$. Then use (7): for such an optimal y and $w \in \mathrm{Ker}\, M$, we have

$$M(y - x) = M(x - M^- G(x) + w - x) = G(x) .$$

As for the Lipschitz property, it is essentially proved in Theorem X.4.3.1 of [HULL93]. □

Remark 4. When h is the piecewise affine function $\check{f}_k$, the regularized value $\check{F}_k(x_n)$ in (1) is easily computed, via the resolution of a quadratic problem. A particular y^c can even be selected: for stability reasons, it is advisable to choose one as close as possible to the current center x_n. For this, it suffices to solve a projection problem onto the polyhedron defined by (7).

When M is positive definite, the solution set $P(x)$ reduces to a singleton, called the *proximal* point of x and denoted by $p(x)$. The update $x_{n+1} = y^c$ therefore appears as the proximal point of x_n, associated with the cutting-plane model $\check{f}_k$. For the general case of a degenerate M, we still use the notation $p(x)$ for an arbitrary point in $P(x)$. Actually, this notation is slightly simplistic; in particular it neglects the matrix M and the function h.

Finally, we recall that, in the classical Moreau-Yosida regularization, minimizing h is equivalent to minimizing H; this property is conserved when M is singular, we omit the proof which is simple. □

3 Description of the Algorithm

In this section we concretize the principles exposed in §1. We state a schematic algorithm and introduce some of its properties. By $M_+ = Up(M, u, v)$ we mean an updated matrix using M and two vectors u and v; for example "Up" may be a formula based on the *quasi-Newton* equation $M_+ u = v$. We will actually consider two variants; in the first, M_+ is imposed to be proportional to the identity matrix, while the second will use a full matrix.

Algorithm 5.

Step 0 (Initialization). Choose $m \in]0,1[$, $x_0 \in \mathbb{R}^N$. Set $y_0 = x_0, M_0 = \mathrm{I}, n = k = 0$.

Step 1 (Computation of the candidate). If the stopping criterion is not satisfied, find $y^c = p(x_n)$ i.e., solve

$$y^c \in \mathrm{Argmin}[\check{f}_k(y) + \tfrac{1}{2}\langle M_n(y - x_n), y - x_n\rangle], \tag{10}$$

and compute

$$\delta_n = \delta_n^k := f(x_n) - \check{f}_k(y^c) - \tfrac{1}{2}\langle M_n(y^c - x_n), y^c - x_n\rangle . \tag{11}$$

Step 2 (Descent-step). If $f(y^c) \leq f(x_n) - m\delta_n$, then: update $x_{n+1} = y^c$. Choose two points z and z' and a closed convex function h; compute the corresponding dual solutions $G(z)$ and $G(z')$ from (4); set $u := z' - z$ and $v := G(z') - G(z)$; update $M_{n+1} = Up(M_n, u, v)$ and increase n.

Step 3 (Null-step). Set $y_{k+1} = y^c$, increase k. Loop to 1. □

Before starting the theoretical study of this algorithm, let us mention some implementation issues:

(i) The above description neglects numerical technicalities such as:
- an explicit stopping test (which can use (14) below),
- an elaborate choice of the initial matrix,
- an aggregation mechanism to avoid storing the whole bundle when k becomes large,
- a safeguard to prevent awkward candidates, since the quadratic problem in Step 1 may have no solution (y^c "at infinity"); the end of this section suggests a possible safeguarding technique.
- We will see that, for efficiency, some sort of line-search should be inserted before looping to the next iteration; it is in this sense that the above description is only schematic.

(ii) The matrix update in Step 2 will be explained in §4 below; we will consider two possibilities for (z, z', h) and two possibilities for the formula symbolized by "Up".

(iii) Concerning the existence of solutions to (10), we recall here a result of [FW56]: being piecewise quadratic, the objective function has a minimum point if and only if it is bounded from below. A detailed answer to this existence question was given in Theorem 1, which can be particularized now to our present situation:

Proposition 6. *At iteration (n,k), let Γ be the convex hull of the subgradients $g_1, \ldots, g_k$ and assume $\Gamma \cap \mathrm{Ker}\, M_n \neq \emptyset$. Then* (10) *has a solution of the form*

$$y^c = x_n - M_n^- G(x_n) + \tilde{w}_k \tag{12}$$

with $\tilde{w}_k \in \mathrm{Ker}\, M_n$ and $G(x_n) \in \partial \check{f}_k(y^c)$ is given by (4).

The following relations hold:

$$\delta_n^k = \tfrac{1}{2}\langle G(x_n), M_n^- G(x_n)\rangle + \varepsilon_n^k \tag{13}$$

with

$$\varepsilon_n^k := f(x_n) - \check{f}_k(y^c) - \langle G(x_n), M_n^- G(x_n)\rangle$$

and, for all $y \in \mathbb{R}^N$,

$$f(y) \geq f(x_n) + \langle G(x_n), y - x_n\rangle - \varepsilon_n^k \,. \tag{14}$$

Proof. Use Theorem 1 with $h = \check{f}_k$ and $M = M_n$. First of all, the domain of $(\check{f}_k)^*$ is the convex hull Γ of the subgradients g_i making up $\check{f}_k$ (see for example §X.3.4 in [HULL93]). When $\Gamma \cap \operatorname{Ker} M \neq \emptyset$, the optimal value in (3) is a finite number and, because $\check{f}_k$ is a polyhedral function, an optimal solution y^c exists; its expression (12) comes from the characterization (5). To obtain the form (13) of δ_n, plug the value (12) of y^c into (11) and use the property $M_n^- \tilde{w}_k = 0$.

Finally express that $G(x_n) \in \partial \check{f}_k(y^c)$: for all $z \in \mathbb{R}^N$,

$$f(z) \geq \check{f}_k(z) \geq \check{f}_k(y^c) + \langle G(x_n), z - y^c\rangle$$

and perform some straightforward algebraic manipulations to obtain (14). □

Observe that, when the existence condition $\Gamma \cap \operatorname{Im} M \neq \emptyset$ holds, it holds for every subsequent iteration, as long as M is not updated. In the particular case when $0 \in \Gamma$ (which corresponds to $\check{f}_k$ having a minimum point), the existence condition holds at every subsequent iteration.

We recall that the subdifferential of the max-function $\check{f}_k$ is the convex hull of the active subgradients g_i. In other words, for some set of convex multipliers α_i,

$$\begin{gathered} G(x_n) = \textstyle\sum_{i \in I_k} \alpha_i g_i \text{ where} \\ I_k := \{i = 1, \ldots, k : f(y_i) + \langle g_i, y^c - y_i\rangle = \check{f}_k(y^c)\} \,. \end{gathered} \tag{15}$$

Remark 7. When $G(x_n)$ and ε_n^k are both close to 0, (14) shows that x_n satisfies an approximate optimality property. In view of (13), the aim of the algorithm is thus to force δ_n^k to 0 and to avoid "large" matrices M_n. □

The next result is motivated by the introduction of semi-definite matrices.

Lemma 8. *Let H be a closed convex function. If, for some x_1, x_2, there are $g_i \in \partial H(x_i)$, $i = 1, 2$, such that*

$$\langle g_1 - g_2, x_1 - x_2\rangle = 0 \,, \tag{16}$$

then H is affine on the segment $[x_1, x_2]$ and ∂H is constant on $]x_1, x_2[$. If H is (finite-valued and) differentiable, $g_1 = g_2$.

Proof. Take $x := x_1 + \alpha(x_2 - x_1)$ with $\alpha \in [0, 1]$; write the subgradient inequalities

$$H(x) \geq H(x_1) + \alpha \langle g_1, x_2 - x_1\rangle$$

$$H(x) \geq H(x_2) - (1 - \alpha) \langle g_2, x_2 - x_1\rangle$$

and obtain by convex combination, using (16),

$$H(x) \geq (1-\alpha)H(x_1) + \alpha H(x_2)\,.$$

Since the convexity of H gives the converse inequality, H is affine on $[x_1, x_2]$.

Now, restrict the above α to $]0,1[$ and take $g \in \partial H(x)$. Then the affinity of H means that, for any $x' := x_1 + \alpha'(x_2 - x_1)$ with $\alpha' \in]0,1[$,

$$H(x') = H(x) + \langle g, x' - x\rangle\ .$$

Additionally, for all z,

$$H(z) \geq H(x) + \langle g, z - x\rangle = H(x') + \langle g, z - x'\rangle - \varepsilon\,,$$

where $\varepsilon := H(x') - H(x) - \langle g, x - x'\rangle = 0$. Thus $\partial H(x) \subset \partial H(x')$; the other inclusion is established likewise, exchanging x and x'.

Finally, if the convex function H is differentiable, it is continuously differentiable and the equality $\nabla H(x) = \nabla H(x')$ extends to the endpoints x_1 and x_2. □

We conclude this section with a word concerning the computation of y^c. When M_n is singular, the objective function in (10) may be unbounded from below; some safeguarding technique is therefore advisable. Rather than loading the diagonal of M_n, we prefer to perturb the function $\check{f}_k$ temporarily, just to eliminate candidates that are blatantly too far from the current stability center.

Safeguarding Technique Suppose an estimated lower bound for $f(x_{n+1})$ is at hand; we can take for example

$$\ell := f(x_n) - \frac{f(x_{n-1}) - f(x_n)}{m}\,. \tag{17}$$

Then the constant function of value ℓ can be appended to the affine functions making up $\check{f}_k$; the perturbed model is bounded from below and existence of a (perturbed) proximal point is guaranteed.

If the safeguard is active, $\check{f}_k(y^c) = \ell$, a supposedly very small value; this implies that $\check{f}_k$ does not approximate the actual objective f properly. Thus, our safeguard should not significantly disturb the algorithm. □

4 Matrix Updates

To compute M_{n+1} in Step 2 of Algorithm 5, we need to specify the pair of vectors u and v, as well as the actual formula for "Up".

4.1 Choice of a Regularizing Scheme

The vectors u and v are uniquely determined from a triple (z, z', h), knowing that

$$u = z' - z \quad \text{and} \quad v = \nabla H(z') - \nabla H(z)\,.$$

We consider two alternatives for (z, z', h).

Model regularization (x-$\check{f}$) A first natural idea is to take $z := x_n$, $z' := x_{n+1} = y^c$; then we need the two corresponding gradients of some smooth function H. For implementability reasons, H has to be the Moreau-Yosida regularization $\check{F}_k$ of the current model $\check{f}_k$ (the updated model $\check{f}_{k+1}$ could also be taken). Then $G(x_n) = \nabla \check{F}_k(x_n)$ is available and $\nabla \check{F}_k(y^c) = G(y^c)$ is obtained via one more resolution of the quadratic problem:

$$p(y^c) = y^{cc} \in \operatorname{Argmin}\{\check{f}_k(y) + \tfrac{1}{2}\,\langle M_n(y - y^c), y - y^c\rangle\}\,,$$

or rather of its dual. □

Our second choice manages to regularize f itself, thanks to a backward process. We take $v := g(y^c) - g(x_n)$ and we compute x_-, y_- such that $g(x_n) = \nabla F(x_-)$ and $g(y^c) = \nabla F(y_-)$. This amounts to inverting the proximal mapping, an operation which can be performed explicitly:

Proposition 9. *We use the notation of §2; assume that $z \in \mathbb{R}^N$ is such that $\partial h(z) \cap \operatorname{Im} M$ contains some point G. Then, for any $z_- \in \{z + M^- G\} + \operatorname{Ker} M$,*

$$G = G(z_-) = \nabla H(z_-)\,. \tag{18}$$

In fact, z solves (3) for $x = z_-$, i.e., $p(z_-) = z$.

Proof. With z_- as stated, set $w := z_- - z - M^- G \in \operatorname{Ker} M$ and consider the set

$$\partial h^*(g) - z_- + M^- G + w = \partial h^*(G) - z\,.$$

This set contains 0 because $G \in \partial h(z)$, i.e., $z \in \partial h^*(G)$. Together with the property $G \in \operatorname{Im} M$, we see that G satisfies the optimality condition of (4) with $x = z_-$.

Furthermore $G = M(z_- - z)$, hence z satisfies the optimality condition for (3). □

This result can be exploited with $h = f$, thus giving our second option:

Objective regularization (g-f) Suppose M_n is nonsingular. Having on hand the two successive iterates x_n and y^c, we also have the corresponding subgradients $g(x_n)$ and $g(y^c)$ of f. Then we simply compute the points at which these two subgradients are gradients of F. We therefore take

$$u := y^c + M_n^{-1} g(y^c) - [x_n + M_n^{-1} g(x_n)]\,, \quad v := g(y^c) - g(x_n)\,,$$

which can be suitably rewritten as

$$\Delta x := x_{n+1} - x_n\ , \quad v = g(x_{n+1}) - g(x_n)\ , \quad u = \Delta x + M_n^{-1} v\ . \tag{19}$$

□

Knowing that our real problem is to minimize f, i.e., F, this last strategy actually appears as more direct than (x-$\tilde{f}$): it tries to apply the algorithm

$$x_{n+1} = x_n - \nabla^{-2} F(x_n) \nabla F(x_n)\ , \tag{20}$$

$\nabla F(x_n)$ and $\nabla^2 F(x_n)$ being replaced by $\nabla \tilde{F}_k(x_n)$ and M_n respectively. Furthermore, we will see in §6.1 that positive definiteness can easily be preserved; we will also see that this additional advantage is desirable.

4.2 Choice of an Explicit Formula

Let us now turn to the possible formulae for "Up". First of all, it is important to remember that the property $v = \nabla H(z') - \nabla H(z)$ ensures $\langle v, u \rangle \geq 0$, and the situation $\langle v, u \rangle = 0$ is described by Lemma 8. An important inequality is

$$\frac{\|v\|^2}{\langle v, u \rangle} \leq \Lambda_n \tag{21}$$

where Λ_n is the largest eigenvalue of M_n. To obtain it, apply (9) with $M = M_n$.

Another useful inequality is

$$\frac{\|M_n u\|^2}{\langle M_n u, u \rangle} \leq \Lambda_n \quad \text{for } u \notin \operatorname{Ker} M\ .$$

Indeed, set $z := M_n^{1/2} u$ and observe that

$$\frac{\|M_n u\|^2}{\langle M_n u, u \rangle} = \frac{\langle M_n z, z \rangle}{\|z\|^2}\ .$$

We consider two variants for "Up", based on the quasi-Newton principle.

Diagonal quasi-Newton Variant (dqN) The matrices are restricted to being proportional to the identity: $M_n = \mu_n \mathrm{I}$. Given u and v, the updated matrix is $\mu_{n+1}\mathrm{I}$, where μ_{n+1} minimizes $1/2\|v/\mu - u\|^2$; we take

$$\mu_{n+1} := \begin{cases} \dfrac{\|v\|^2}{\langle v, u \rangle} & \text{if } \langle v, u \rangle > 0, \\ 0 & \text{if } \langle v, u \rangle = 0\ . \end{cases}$$

If $\langle v, u \rangle = 0$, then $v = 0$ (Lemma 8): the observed curvature of H along u is 0, which explains our choice $\mu_{n+1} = 0$. □

With relation to Remark 7, we have from (21)

$$\mu_{n+1} \leq \mu_n\ , \tag{22}$$

an inequality which holds independently of (z, z', h).

Full quasi-Newton Variant (fqN) The updated matrix is computed from the BFGS formula:

$$M_{n+1} := M_n + A - B\,,$$

where

$$A := \begin{cases} \dfrac{vv^\top}{\langle v, u\rangle} & \text{if } \langle v, u\rangle > 0, \\ 0 & \text{if not} \end{cases}$$

and

$$B := \begin{cases} \dfrac{M_n u u^\top M_n}{\langle M_n u, u\rangle} & \text{if } M_n u \neq 0, \\ 0 & \text{if not.} \end{cases}$$

Note that this variant is robust, since

$$\operatorname{tr} A \leq \Lambda_n \quad \text{and} \quad \operatorname{tr} B \leq \Lambda_n\,.$$

It is also consistent:

- the choice $A = 0$ was already explained in (dqN),
- the choice $B = 0$ is similar, namely the predicted curvature along u is 0 when $M_n u = 0$.
- As for the quasi-Newton equation, we have

$$M_{n+1}u = M_n u + Au - Bu = Au$$

and this is v in any case. □

Thus, all our formulae introduce smoothness while preserving implementability, two features which are not present in [BGLS93]. Note, however, that we may well have $G(y^c) = G(x_n)$ (think of the very first iteration $n = k = 0$!). Then the variant $(x\text{-}f)$ gives $v = 0$ and M_{n+1} degenerates. By contrast, M_{n+1} in $(g\text{-}f)$ can degenerate only when $g(y^c) = g(x_n)$, an unlikely event.

Remark 10. The difference of f-subgradients for v in $(g\text{-}f)$ suggests Shor's r-algorithm [Sho85]. In this variant, the matrix M_{n+1} dilates the space in the direction v. Having a coefficient $\beta_n > 1$, we take (see [Sko73])

$$M_{n+1} := M_n + \begin{cases} 0 & \text{if } v \in \operatorname{Ker} M_n, \\ \dfrac{\beta_n^2 - 1}{\langle v, M_n^{-1} v\rangle} vv^\top & \text{if not.} \end{cases}$$

We mention that this variant accommodates any vector v: we can also take $v = g(y^c)$, as in the ellipsoid-type algorithm [Sho70].

No matter how v is chosen, each matrix M_{n+1} is positive definite if M_n is such; again robustness is preserved:

$$\operatorname{tr} M_{n+1} \leq \operatorname{tr} M_n + (\beta_n^2 - 1)\Lambda_n\,.$$ □

5 Convergence Issues

As usual with bundle methods, we split our convergence analysis into two parts.

Theorem 11. *Suppose that Algorithm 5 generates a finite sequence $\{x_n, n = 0, 1, \ldots, n_f\}$ and that the last generated matrix M_{n_f} is nonsingular. Then x_{n_f} is optimal.*

Proof. Since M_{n_f} is nonsingular, Algorithm 5 becomes a standard bundle method and the proof of, for example, Theorem XV.3.2.4 of [HULL93] can be reproduced. For the sake of completeness, we give here a simplified version (which cannot be generalized when the bundle is aggregated).

From the definition (11) of δ^k, we have for k large enough and $i = 1, \ldots, k$

$$f(y_i) + \langle g_i, y_{k+1} - y_i \rangle + \tfrac{1}{2} \langle M_{n_f}(y_{k+1} - x_{n_f}), y_{k+1} - x_{n_f} \rangle \leq f(x_{n_f}) - \delta^k \quad (23)$$

(δ^k stands for $\delta^k_{n_f}$). On the other hand, non-descent implies

$$f(x_{n_f}) - m\delta_i \leq f(y_i) \quad \text{for } i \text{ large enough}$$

and we obtain by addition (neglecting the quadratic term):

$$\langle g_i, y_{k+1} - y_i \rangle \leq m\delta^i - \delta^k \quad \text{for large } i \text{ and } k, \quad \text{with } i \leq k\,. \qquad (24)$$

Now, there exists by construction an i such that $x_{n_f} = y_i$; taking this i in (23):

$$\langle g_i, y_{k+1} - x_{n_f} \rangle + \tfrac{1}{2} \langle M_{n_f}(y_{k+1} - x_{n_f}), y_{k+1} - x_{n_f} \rangle \leq -\delta^k \leq 0\,.$$

Because M_{n_f} is positive definite, this implies the boundedness of $\{y_k\}$, hence, from (15), of $\{g_k\}$: the left-hand side in (24) can be made arbitrarily close to 0. On the other hand, since $\check{f}_k \leq \check{f}_{k+1}$, the sequence $\{\delta^k\}$ is decreasing and has a limit, which therefore has to be 0. Then, from (13), $\check{g}_k$ and ε^k tend to 0 and (14) shows that x_{n_f} minimizes f. □

We now turn to the case of infinitely many descent-steps; our study will be limited to the diagonal variant (dqN). The result below is rather classical ([Kiw90], [SZ92], [CL93]), apart from the possible degeneracy of the quadratic term in the proximal problem (10). The proof suggests that the particular value $1/m$ of the safeguarding parameter in (17) is not totally innocent.

Theorem 12. *Consider Algorithm 5 with the following options:*

- *diagonal quasi-Newton update (dqN);*
- *safeguarded resolution of the quadratic subproblems, as explained at the end of §3.*

If an infinite sequence $\{x_n\}$ is generated, it is minimizing: $f(x_n) \to \inf f$.

Proof. The key is (22): $\{\mu_n\}$ is a nonincreasing sequence, hence $t_n := 1/\mu_n$ forms a divergent series. We consider two cases.

Suppose first $\mu_n > 0$ for all n. Then the proof is classical: we can reproduce for example Proposition 2.2 in [CL93] or Theorem XV.3.2.2 in [HULL93].

Now assume $\mu_n = 0$ for some n. In view of (22), $\mu_p = 0$ for all $p \geq n$. As long as the safeguard of §3 does not come into play, y^c minimizes $\check{f}_k$ and the situation is essentially the same as before. From (13), $\delta_n^k = f(x_n) - \check{f}_k(y^c)$. Since $\check{f}_k \leq f$ and the descent test forces $\delta_n^k \to 0$, the conclusion still holds.

The last possibility is when $\mu_n = 0$ with an active safeguard. Then (14) cannot be used because, after perturbation by ℓ, the model $\check{f}_k$ is no longer below f. Rather, combine (17) with the descent test to obtain

$$f(x_{n+1}) \leq f(x_n) - m[f(x_n) - \ell] = f(x_n) - [f(x_{n-1}) - f(x_n)]\,.$$

If this holds infinitely often, $f(x_n) \to -\infty$ and the conclusion still holds. □

We do not know if the above result can be proved for the variant (fqN). The usual technique for BFGS updates is to bound the trace of M_n from above. Here the inequality

$$\operatorname{tr} M_{n+1} \leq \operatorname{tr} M_n + \operatorname{tr} A \leq \operatorname{tr} M_n + \Lambda_n$$

is easily obtained from (21). However, it is not sharp enough to establish the divergence of the series $\{1/\Lambda_n\}$ (a key argument, see for example [BGLS93]).

Our last result is related to speed of convergence. In fact, consider (20), which is the basis for all our development. We are trying to minimize a function F which, despite appearances, depends on the iteration index n, through the matrix M_n. We should therefore check whether the whole idea makes any sense. We do this in an ideal situation: assume that, for given x_n and M_n, the regularized values $F(x_n)$ and $\nabla F(x_n)$ can be exactly computed. Limiting ourselves to the combination (dqN)-(g-f), we obtain the following simplification of Algorithm 5:

Algorithm 13.

Step 0 (Initialization). Choose $x_0 \in \mathbb{R}^N$ and compute $g_0 := g(x_0)$. Set $\mu_0 = 1, n = 0$.

Step 1 (Computation of the proximal point). If the stopping criterion is not satisfied, solve

$$x_{n+1} \in \operatorname{Argmin}[f(x) + \tfrac{1}{2}\mu_n \langle x - x_n, x - x_n\rangle]\,.$$

Compute $g_{n+1} := g(x_{n+1})$ and set

$$\Delta x := x_{n+1} - x_n,\ v := g_{n+1} - g_n,\ u := \Delta x + \frac{1}{\mu_n} v\,.$$

Step 2 (Descent-step). Set

$$\mu_{n+1} := \begin{cases} \dfrac{|v|^2}{\langle v, u\rangle} & \text{if } \langle v, u\rangle > 0\,, \\ 0 & \text{if not}\,. \end{cases}$$

Increase n and loop to 1. □

The above expression for u comes from (19). We can also write the update as

$$\frac{1}{\mu_{n+1}} = \frac{1}{\mu_n} + \frac{\langle v, \Delta x\rangle}{|v|^2} \tag{25}$$

whenever $\langle v, u\rangle > 0$.

Theorem 14. *If ∇f is locally Lipschitzian, then $\mu_n \to 0$. Make the following additional assumptions: f has a (unique) minimal point $\bar{x}$ and a quadratic growth condition holds: for some $\alpha > 0$,*

$$f(x) \geq f(\bar{x}) + \alpha|x - \bar{x}|^2 .$$

Then $f(x_n)$ tends to $f(\bar{x})$ q-superlinearly.

Proof. If $\mu_n = 0$ for some n, x_{n+1} is obviously a minimizer of f, so we can assume in Step 2 that (25) holds for all n. If ∇f has the local Lipschitz constant L, we can apply [Pow76] or Theorem X.4.2.2 of [HULL93]: $\langle v, \Delta x\rangle / |v|^2 \geq 1/L$ and $1/\mu_n \to +\infty$.

Now apply the subgradient inequality:

$$f(x_n) \geq f(x_{n+1}) + \langle g_{n+1}, x_n - x_{n+1}\rangle = f(x_{n+1}) + \frac{|g_{n+1}|^2}{\mu_n} .$$

On the other hand, our growth condition implies (Lemma 4.3 of [BGLS93]):

$$\frac{1}{\alpha}|g_{n+1}|^2 \geq f(x_{n+1}) - f(\bar{x})$$

hence

$$f(x_n) - f(\bar{x}) \geq f(x_{n+1}) - f(\bar{x}) + \frac{\alpha}{\mu_n}[f(x_{n+1}) - f(\bar{x})] .$$

The conclusion follows, since $\frac{1}{1+\alpha/\mu_n} \to 0$. □

Note that we have

$$\alpha|x - \bar{x}|^2 \leq f(x) - f(\bar{x}) \leq L|x - \bar{x}|^2 ,$$

so $\{f(x_n)\}$ and $\{x_n\}$ converge at the same speed. The above result may seem artificial since, in our ideal situation, the ideal value for the penalty is $\mu = 0$. To become really convincing, the proof should be extended to the variant (fqN). Let us say that, at least, (25) gives a constructive (and hopefully reasonable) way of driving μ to 0.

To conclude this section, let us give some comments concerning the assumptions in Theorem 14. The growth condition is assessed by the following result:

Proposition 15. *Let Algorithm 13 be applied to the univariate function $f(x) = 1/3|x|^3$, starting from $x_0 > 0$. Then the convergence of $\{x_n\}$ to the solution 0 cannot be q-superlinear.*

Proof. First draw a picture to see that $x_n > x_{n+1} > 0$ for all n. The next iterate $p(x_n)$ satisfies the relation

$$p^2(x_n) + \mu_n[p(x_n) - x_n] = 0\,. \tag{26}$$

Divide succesively by $\mu_n p(x_n)$ and μ_n^2 to obtain

$$\frac{p(x_n)}{\mu_n} = \frac{x_n}{p(x_n)} - 1 \quad \text{and} \quad \frac{x_n}{\mu_n} = \frac{p^2(x_n)}{\mu_n^2} + \frac{p(x_n)}{\mu_n}\,. \tag{27}$$

Next, straightforward calculations in (25) and multiplication by $p(x_n)$ give

$$\frac{x_{n+1}}{\mu_{n+1}} = \frac{p(x_n)}{\mu_n} + \frac{p(x_n)}{x_n + p(x_n)}\,. \tag{28}$$

Now assume for contradiction that $\{x_n\}$ converges to 0 q-superlinearly, i.e., $p(x_n)/x_n \to 0$. Then $p(x_n)/\mu_n \to +\infty$ and $s_n := x_n/\mu_n \to +\infty$ because of (27); also, from (28), $s_{n+1} = p(x_n)/\mu_n + \varepsilon_n$, where $\varepsilon_n := p(x_n)/[x_n + p(x_n)]$ forms a convergent series. Finally, compute explicitly $p(x_n)$ from (26) and obtain the equalities

$$2s_{n+1} - \varepsilon_n + 1 = \frac{2p(x_n)}{\mu_n} + 1 = \sqrt{1 + 4x_n/\mu_n} = \sqrt{1 + 4s_n}\,.$$

Using the inequality $\sqrt{1 + 4s_n} \leq 1 + 2s_n$ and summing, we see that $\{s_n\}$ is bounded. This is the required contradiction. □

As for our smoothness assumption on f, its necessity is not obvious: we do not know if the property $\mu_n \to 0$ is really crucial. Such an asymmetry between the two assumptions in Theorem 14 can be explained:

- The growth condition appears natural: if it does not hold, the model F may become a gross approximation of f.
- Likewise, if f does not have a Lipschitzian gradient, its growth prevails over the quadratic perturbation.

6 Implementation Issues

We have already suggested that Algorithm 5 is only schematic, and that some line-search is needed. For this, two strategies may be adopted:

- *Standard line-search.* Once y^c is computed, the next iterate (x_{n+1} or y_{k+1}) is searched along the half-line $\{x_n + t(y^c - x_n) : t > 0\}$.
- *Curved search.* Adjust the norming, replacing the matrix M_n in (10) by M_n/t. Then the candidate y^c depends on $t > 0$, but the mapping $t \mapsto y^c$ is no longer positively homogeneous. The next iterate is searched along a curve, parametrized by the "stepsize" t.

6.1 Extrapolation

When a descent test is accepted, the matrix M_n is going to be updated, but it is advisable to avoid a degenerate M_{n+1}. In fact:

- The proof of Theorem 11 breaks down when M_{n_f} is degenerate; a very first difficulty is that the candidates y^c may become unbounded.
- In Theorem 12, the relevance of a degenerate M_n is questioned: there, such a degeneracy means $M_n = \mu_n \mathrm{I} = 0$ forever, unfortunate for an algorithm trying to identify a second order behaviour. Degeneracy appears as clumsiness, at least for the diagonal variant.
- Empirically, a degenerate matrix also presents a serious danger. Suppose an iteration where $M_n = 0$; then the algorithm starts a sequence of pure cutting-plane iterations. We may be in the situation of Nemirovski's counter-example: an enormous number of null-steps becomes necessary until a descent iterate is found.

Inspection of the update formulae in §4 shows that v should be nonzero to yield an invertible M_{n+1}. Guaranteeing this property seems difficult for the $(x\text{-}\check{f})$ variant, but it is straightforward with $(g\text{-}f)$:

Lemma 16. *Let u and v be given by (19) and take $m' < 1$. Then*

$$\langle g(x_{n+1}), x_{n+1} - x_n \rangle \geq -m'\delta_n \tag{29}$$

implies $\langle v, u \rangle > 0$.

Proof. The definition of $\check{f}_k$ implies

$$f(x_n) + \langle g(x_n), x_{n+1} - x_n \rangle \leq \check{f}_k(x_{n+1})\,;$$

add $1/2\ \langle M_n(x_{n+1} - x_n), x_{n+1} - x_n \rangle$ to both sides, to obtain

$$\langle g(x_n), x_{n+1} - x_n \rangle \leq \check{f}_k(x_{n+1}) + \tfrac{1}{2} \langle M_n(x_{n+1} - x_n), x_{n+1} - x_n \rangle - f(x_n) = -\delta_n\,.$$

Subtracting from (29), we get

$$\langle v, \Delta x \rangle = \langle g(x_{n+1}) - g(x_n), x_{n+1} - x_n \rangle \geq (1 - m')\delta_n > 0$$

and the conclusion follows due to (19). □

To guarantee positive definite matrices, it therefore suffices to find a candidate satisfying (29) as well as the descent condition. This is nothing but a Wolfe type criterion for the stepsize. With $m' \in]m, 1[$, this problem is classical for a standard line-search; as for the curved search, it is solved in (2.17) of [SZ92].

6.2 Interpolation

An awake reader may have already realized that the proof of Theorem 12 is "too easy to be true": its work-horse (22) is a luxury argument. Indeed the same proof would apply if $\{\mu_n\}$ were increasing with a moderate speed.

Remark 17. It can be proved that, if M_n is updated with the symmetric rank-one formula, then the same phenomenon occurs, namely

$$\operatorname{tr} M_{n+1} \leq \operatorname{tr} M_n .$$

Moreover, this formula preserves positive definiteness of M_{n+1}. Based on these properties, convergence of Algorithm 5 can be established when SR1 replaces (dqN) for "Up". □

Clearly the algorithm will be in trouble if M_0 is chosen unduly small. To struggle against this misbehaviour, we accept to increase the penalty through a division of the matrix by a short stepsize. Here comes a really delicate point: when the descent test (2) is not satisfied, a decision must be made: either to decrease the stepsize, or to enrich the bundle via a null-step (or even both, why not?). A possible strategy is as follows.

Having y^c, compute the linearization error

$$e^c := f(x_n) - [f(y^c) + \langle g(y^c), x_n - y^c \rangle] .$$

If e^c is large, the new affine piece in $\check{f}_k$ is going to have little influence on the computation of the new candidate. It is therefore reasonable to make a null-step only if

$$e^c \leq m'' \delta_n , \tag{30}$$

where m'' is a positive tolerance. When (30) is not satisfied, an interpolation is performed.

Remark 18. An important question is whether this backtracking procedure spoils the bundling mechanism. In other words, will repeated interpolations eventually produce a stepsize $t > 0$ such that (30) holds? It can be proved that the answer is yes; this actually relies on the semismoothness of convex functions [Mif77]. □

6.3 An Improved Algorithm

We end this paper with an example of algorithm including the refinements presented above. It uses the $(g\text{-}f)$ option for the update since the preceeding analysis reveals its definite advantages. Note an important detail: it is M_n/t and not M_n, which we update in the quasi-Newton formula, as in [OS76].

Algorithm 19.

Step 0 (Initialization). Choose $m \in]0,1[, m' \in]m,1[, m'' > 0, x_0 \in \mathbb{R}^N$. Set $y_0 = x_0, M_0 = 1, n = k = 0$.

Step 1 (t-adjustment). Execute Algorithm 20 to obtain $t > 0$ and y^c.
Step 2 (Descent-step). Update $x_{n+1} = y^c$. Set $\Delta x := x_{n+1} - x_n$, $v := g(x_{n+1}) - g(x_n)$, $u := \Delta x + tM_n^{-1}v$ and update $M_{n+1} = Up(M_n/t, u, v)$. Increase n.
Step 3 (Null-step). Set $y_{k+1} = y^c$, increase k. Loop to 1. □

A (dqN) strategy in Step 2 would result in an alternative to the proposals of [Kiw90] and [SZ92]. As for the t-adjustement, we have chosen a curved search, which we believe is more natural:

Algorithm 20. The data are $\check{f}_k, x_n, M_n$ and the tolerances m, m', m''.
Step 0. Set $t = 1, t_L = 0, t_R = +\infty$.
Step 1. Compute

$$y^c = y^c(t) := \operatorname{argmin}\left[\check{f}_k(y) + \tfrac{1}{2t}\langle M_n(y - x_n), y - x_n\rangle\right]$$
$$\delta := f(x_n) - \check{f}_k(y^c) - \tfrac{1}{2t}\langle M_n(y^c - x_n), y^c - x_n\rangle$$
$$e^c := f(x_n) - [f(y^c) + \langle g(y^c), x_n - y^c\rangle].$$

Step 2. If $f(y^c) > f(x_n) - m\delta$, go to Step 4.
Step 3. If $\langle g(y^c), y^c - x_n\rangle \geq m'\delta$, stop with a Descent-step.
Otherwise set $t_L = t$, compute a new t in $]t_L, t_R[$ and go to Step 1.
Step 4. If $t_L = 0$ and $e^c \leq m''\delta$, stop with a Null-step.
Otherwise set $t_R = t$, compute a new t in $]t_L, t_R[$ and go to Step 1. □

References

[BGLS93] J. Bonnans, J.Ch. Gilbert, C. Lemaréchal, and C. Sagastizábal. A family of Variable Metric Proximal methods. Rapport de Recherche 1851, INRIA, 1993.

[CG59] E. Cheney and A. Goldstein. Newton's method for Convex Programming and Tchebycheff approximations. *Numerische Mathematik*, 1:253–268, 1959.

[CL93] R. Correa and C. Lemaréchal. Convergence of some algorithms for convex minimization. Manuscript, INRIA, 78153 Le Chesnay Cedex (France), 1993.

[DM77] J.E. Dennis and J.J. Moré. Quasi-Newton methods, motivation and theory. *SIAM Review*, 19:46–89, 1977.

[FW56] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistic Quarterly*, 3:95–110, 1956.

[HULL93] J.B. Hiriart-Urruty and C. L-Lemaréchal. *Convex Analysis and Minimization Algorithms*. Springer-Verlag, 1993.

[Kel60] J. E. Kelley. The cutting plane method for solving convex programs. *J. Soc. Indust. Appl. Math.*, 8:703–712, 1960.

[Kiw83] K.C. Kiwiel. An aggregate subgradient method for nonsmooth convex minimization. *Mathematical Programming*, 27:320–341, 1983.

[Kiw90] K.C. Kiwiel. Proximity control in bundle methods for convex nondifferentiable minimization. *Mathematical Programming*, 46:105–122, 1990.

[Lem78] C. Lemaréchal. Bundle methods in nonsmooth optmization. In C. Lemaréchal and R. Mifflin, editors, *Nonsmooth optimization*. Pergamon Press, Oxford, 1978.

[Mif77] R. Mifflin. Semi-smooth and semi-convex functions in constrained optimization. *SIAM Journal on Control and Optimization*, 15:959–972, 1977.

[Mor65] J.J. Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93:273–299, 1965.

[OS76] S.S. Oren and E. Spedicato. Optimal conditioning of self-scaling variable metric algorithms. *Mathematical Programming*, 10:70–90, 1976.

[Pow76] M.J.D. Powell. Some global convergence properties of a variable metric algorithm for minimization without exact line searches. In R.W. Cottle and C.E. Lemke, editors, *Nonlinear Programming*, number 9 in SIAM-AMS Proceedings. American Mathematical Society, Providence, RI, 1976.

[Sho70] N. Shor. Utilization for the operation of space dilatation in the minimization of convex function. *Cybernetics*, 6:7–15, 1970.

[Sho85] N. Shor. *Minimization methods for non-differentiable functions*. Springer-Verlag, Berlin, 1985.

[Sko73] V. Skokov. Note on minimization methods employing space stretching. *Cybernetics*, 10:689–692, 1973.

[SZ92] H. Schramm and J. Zowe. A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results. *SIAM Journal on Optimization*, 2(1):121–152, 1992.

[Yos64] K. Yosida. *Functional Analysis*. Springer Verlag, 1964.

This article was processed using the LaTeX macro package with LMAMULT style