Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 4, No. 3, pp. 571–604, September 2008

A PROXIMITY CONTROL ALGORITHM TO MINIMIZE NONSMOOTH AND NONCONVEX FUNCTIONS

Dominikus Noll, Olivier Prot and Aude Rondepierre

Abstract: We present a new proximity control bundle algorithm to minimize nonsmooth and nonconvex locally Lipschitz functions. In contrast with the traditional oracle-based methods in nonsmooth programming, our method is model-based and can accommodate cases where several Clarke subgradients can be computed at reasonable cost. We propose a new way to manage the proximity control parameter, which allows us to handle nonconvex objectives. We prove global convergence of our method in the sense that every accumulation point of the sequence of serious steps is critical. Our method is tested on a variety of examples in H_∞-controller synthesis.

Key words: *nonsmooth programming, nonconvex programming, bundle methods, proximity control, global convergence, H_∞-synthesis*

Mathematics Subject Classification: *49J52, 90C26, 90C22, 93B36*

1 Introduction

It is a long standing objective of nonsmooth optimization to develop methods for the unconstrained minimization program

$$\min_{x\in\mathbb{R}^n} f(x) \tag{1.1}$$

where $f:\mathbb{R}^n\to\mathbb{R}$ is locally Lipschitz and neither smooth nor convex. The goal is to compute local solutions $\bar{x}$ of (1.1) in the sense that the first order necessary optimality condition

$$0\in\partial f(\bar{x}) \tag{1.2}$$

is satisfied, where $\partial f(x)$ is the Clarke subdifferential of f at x [8]. In this paper we propose a bundle algorithm based on proximity control for (1.1), which solves this problem in the sense that for an arbitrary starting point, every accumulation point of the sequence of serious iterates is a critical point of f, i.e. satisfies (1.2).

Bundling is one of the principal techniques in nonsmooth optimization. It is well suited for convex problems, but falls somewhat short without convexity. The reason for this is that cutting planes, the driving mechanism behind bundling, is basically a convexity tool and becomes artificial and delicate to handle without this prerequisite. In response, Fuduli *and al.* [12, 13] propose a new type of bundle methods, where tangent planes at trial points are classified in two groups to create an upper and a lower model of f, and to use this information to build a trust region around the current iterate. The authors obtain a global convergence result, but do not completely solve the principal problem in bundling, the overflow of memory caused by accumulating information. This is usually addressed by Kiwiel's idea of aggregation [16–18].

Older approaches which combine bundling and proximity control such as Schramm [28] or Schramm and Zowe [29] are essentially based on the convex case. For nonconvex f the authors propose to downshift tangent planes at trial points if they are no longer cutting planes, and they prove that at least one of the accumulation points $\bar{x}$ of the sequence of serious steps is critical. The idea to downshift tangent planes seems to work well as a rule, as witnessed by the BT-codes [33] or Lemaréchal's M2FC1, but a rigourous justification seems lacking.

Our present approach is based on a new idea, motivated by practical experience gained from the study of nonsmooth programs in eigenvalue optimization and automatic control [1–4, 24, 25]. We start with the concept of a local model $\Phi(\cdot, x)$ of f in the neighbourhood of the current iterate x, which may be understood as a nonsmooth generalization of the Taylor expansion. The interplay between f and its local model is used to create descent steps for f away from x by a proximity control mechanism. Our investigation shows that a local model should have the form

$$\Phi(y, x) = \phi(y, x) + \tfrac{1}{2}(y - x)^\top Q(x)(y - x),$$

where $\phi(\cdot, x)$ is the first order model of f at x and $\frac{1}{2}(y - x)^\top Q(x)(y - x)$ the second order part. It turns out that the first order part $\phi(\cdot, x)$ is convex, but can be nonsmooth. It has to be contingent with the objective at the current x, which is to say $\partial_1\phi(x, x) \subset \partial f(x)$. In contrast, the second order term is smooth, even quadratic, but need not be convex.

Notice that the idea of a local model Φ has also been used in [9] or [27]. What is new in our approach is that we do not use $\Phi(\cdot, x)$ directly to generate search steps, because this may be too costly. Instead we build a so-called working model $\Phi_k(\cdot, x)$, which has the form

$$\Phi_k(y, x) = \phi_k(y, x) + \tfrac{1}{2}(y - x)^\top Q(x)(y - x),$$

and can be thought of as a crude approximation of the ideal model $\Phi(\cdot, x)$. The first order part $\phi_k(y, x)$ is improved iteratively using cutting planes and aggregation. The second order part $\frac{1}{2}(y - x)^\top Q(x)(y - x)$ is kept fixed during the inner loop and only updated between serious steps.

Let us highlight an important aspect of using the models $\Phi(y, x)$ and $\Phi_k(y, x)$. Applications in eigenvalue optimization and automatic control differ in many respects from the traditional use of nonsmooth optimization in Lagrangian relaxation [22]. In that class the objectives f are convex, but one only disposes of the function value $f(y)$ and one particular subgradient $g(y) \in \partial f(y)$ at every trial step y, a situation referred to as *calling an oracle.* In contrast, applications in control allow to know many subgradients, and it may then be a good idea to build a more sophisticated working model Φ_k. This is what makes the concepts of the models Φ and Φ_k attractive, because it leaves their choice to the user, who may have special knowledge in a given application.

Let us look more closely at the bundling mechanism. At the current iterate x of the outer loop, the inner loop with counter k turns until a new serious step x^+ is found. At inner loop counter k we produce a trial step y^{k+1} by solving the tangent program with proximity control

$$\min_{y \in \mathbb{R}^n} \Phi_k(y, x) + \tfrac{\tau_k}{2}\|y - x\|^2, \tag{1.3}$$

where $\tau_k > 0$ is the proximity control parameter. If the solution y^{k+1} of (1.3) gives sufficient decrease in f, it becomes the new iterate $x^+ = y^{k+1}$. According to the standard terminology in nonsmooth optimization, this is called a serious step. Otherwise if y^{k+1} is not satisfactory, it is called a null step. In this case we keep x, but use the information transmitted by y^{k+1} to improve the first order part of the working model $\phi_{k+1}(\cdot, x)$. We then update the proximity control parameter τ_{k+1} and rerun the tangent program (1.3) to obtain a better trial step y^{k+2}. This updating procedure is crucial and uses cutting planes and aggregation, but based on information from the ideal model $\Phi(\cdot, x)$ and not directly from f. While we keep improving the first-order part of $\Phi_k(\cdot, x)$ in the inner loop, the second order term is kept invariant. It changes only between serious steps in the outer loop. Its rationale is to give our method an option to converge superlinearly if $f(x)$ has hidden smoothness properties, as is often the case in applications [5, 25].

Notation. Our terminology follows [8] or [15], where an introduction to bundle methods in the convex case can be found. For a function $\phi(y, x)$ of two arguments $\partial_1\phi(y, x)$ denotes the Clarke subdifferential of $\phi(\cdot, x)$ at y, and similarly for $\partial_2\phi(y, x)$.

Concerning the foundations of nonsmooth optimization we point the reader to the pioneering work by Wolfe [32], Lemaréchal [21], and Kiwiel [18]. An overview can be obtained from Lemaréchal [21], Kiwiel [17] or Polak [26], Ruszczyński [27], Shor [30].

$\boxed{2}$ First Order Model

What makes convex bundling so successful is that affine support functions of the objective f at unsuccessful trial steps (null steps) y^{k+1}, also known as cutting planes, are used to improve the working model. Without convexity it is less obvious in which way first-order affine approximations of f at the null steps y^{k+1} should be used to improve the working model, because these planes are no longer support functions of f. The principal idea of this paper is to generate cutting planes by means of a local model $\Phi(\cdot, x)$ of f in a neighbourhood of x. The model has slightly more structure than f itself, and is thereby amenable to bundling techniques, which were applied directly to f as long as the latter was convex. What we call a model of f in a neighbourhood of x could be thought of as a nonsmooth analogue of the Taylor expansion.

Definition 2.1. A function $\phi : \mathbb{R}^n \times \Omega \to \mathbb{R}$ is called a first-order model of f on the set $\Omega \subset \mathbb{R}^n$ if for every $x \in \Omega$ the function $\phi(\cdot, x) : \mathbb{R}^n \to \mathbb{R}$ is convex and the following conditions are satisfied:

(M_1) $\phi(x, x) = f(x)$ and $\partial_1\phi(x, x) \subset \partial f(x)$.

(M_2) For every $x \in \Omega$ and every $\epsilon > 0$ there exists $\delta > 0$ such that $f(y) - \phi(y, x) \le \epsilon\|y - x\|$ for every $y \in \mathbb{R}^n$ with $\|y - x\| \le \delta$.

(M_3) ϕ is jointly upper semicontinuous on $\mathbb{R}^n \times \Omega$, i.e., $(y_j, x_j) \to (y, x)$ in $\mathbb{R}^n \times \Omega$ implies $\limsup_{j\to\infty} \phi(y_j, x_j) \le \phi(y, x)$.

If $\Omega = \mathbb{R}^n$, then we simply say that ϕ is a first-order model of f.

Remark 2.2. Axiom (M_2) could be written $f(y) - \phi(y, x) \le \mathrm{o}(\|y - x\|)$ as $y \to x$.

Remark 2.3. Every locally Lipschitz function has a first order model, which we call the standard model:

$$\phi^\sharp(y, x) := f(x) + f^0(x; y - x),$$

where $f^0(x;d)$ is the Clarke directional derivative at x in direction d. Indeed, axiom (M_1) is satisfied because $\partial_2 f^0(x;0) = \partial f(x)$. Axiom (M_2) is immediate from the definition of f^0, and axiom (M_3) follows from the upper semicontinuity of the Clarke subdifferential.

Remark 2.4. The standard model $\phi^\sharp$ is the smallest model of f. That is, $\phi^\sharp \leq \phi$ for any other first-order model ϕ of f. In particular, this implies equality $\partial_1\phi(x,x) = \partial f(x)$ in axiom (M_1).

In order to prove $\phi^\sharp(\cdot,x) \leq \phi(\cdot,x)$ it suffices by convexity of $\phi(\cdot,x)$ to show $f^0(x,d) \leq \phi'(x,x;d)$ for every d, where $\phi'(x,x;d)$ is the directional derivative of $\phi(\cdot,x)$ at x in direction d. Let $g \in \partial f(x)$ be a limiting subgradient. That means there exist $x_j \to x$ such that f is differentiable at x_j and $g_j = \nabla f(x_j) \to g$ as $j \to \infty$ (see [8]). Since $\partial f(x)$ is the convex hull of the limiting subgradients, it suffices to show $g \in \partial_1\phi(x,x)$.

Clearly $g_j = \nabla f(x_j) \in \partial f(x_j)$. We claim that $\nabla f(x_j) \in \partial_1\phi(x_j,x_j)$. Indeed, by axiom (M_2) there exist $\epsilon_t \to 0^+$ as $t \to 0^+$ such that

$$t^{-1}\left(f(x_j+td) - f(x_j)\right) \leq t^{-1}\left(\phi(x_j+td,x_j) + \epsilon_t t\|d\| - f(x_j)\right).$$

Passing to the limit $t \to 0^+$ gives $\nabla f(x_j)^\top d \leq \phi'(x_j,x_j;d)$, hence $\nabla f(x_j) \in \partial_1\phi(x_j,x_j)$ by convexity of $\phi(\cdot,x_j)$. Now by the subgradient inequality $g_j^\top d \leq \phi(x_j+d,x_j) - \phi(x_j,x_j)$ for all d and all j. Passing to the limit $j \to \infty$ and using axiom (M_3) gives $g^\top d \leq \limsup_{j\to\infty} \phi(x_j+d,x_j) - f(x) \leq \phi(x+d,x) - f(x)$ for every d, hence $g \in \partial_1\phi(x,x)$, which proves the claim.

Definition 2.5. A first-order model $\phi(\cdot,\cdot)$ of f on Ω is called strong if axioms (M_1), (M_3) and the following strong version of (M_2) are satisfied:

$(\widetilde{M_2})$ For every bounded set $B \subset \mathbb{R}^n$ there exists a constant $L > 0$ such that

$$f(y) - \phi(y,x) \leq L\|y-x\|^2 \tag{2.1}$$

for all $x \in B \cap \Omega, y \in B$.

Remark 2.6. Axiom $(\widetilde{M_2})$ could be written more suggestively as $f(y) - \phi(y,x) \leq \mathcal{O}(\|y - x\|^2)$ for $y - x \to 0$ uniformly on bounded sets.

Remark 2.7. Consider the case of a differentiable function $f : \mathbb{R}^n \to \mathbb{R}$. It seems natural to consider the first order Taylor expansion $\phi(y,x) = f(x) + \nabla f(x)^\top(y-x)$. Is this a model in the sense of Definition 2.1? The answer is no. While axioms (M_1) and (M_2) are satisfied, (M_3) is only satisfied when $\nabla f(x)$ is continuous.

Remark 2.8. Suppose f is of class C^1, then the standard model is the first-order Taylor expansion $\phi^\sharp(y,x) = f(x) + f^0(x;y-x) = f(x) + \nabla f(x)^\top d$. Is this model strong? The answer is no, because strongness requires an estimate of the form $f(y) - f(x) - \nabla f(x)^\top(y - x) \leq \mathcal{O}\left(\|y-x\|^2\right)$. A sufficient condition for this is $f \in C^{1,1}$, but strongness may fail for $f \in C^1 \setminus C^{1,1}$.

Remark 2.9. Consider the case of a composite function $f = g \circ F$, where $g : \mathbb{R}^m \to \mathbb{R}$ is convex and $F : \mathbb{R}^n \to \mathbb{R}^m$ is a mapping of class C^2. Then a first order model may be defined as

$$\phi(y,x) = g\left(F(x) + F'(x)(y-x)\right).$$

By Taylor's theorem $F(y) - [F(x) + F'(x)(y-x)] = \mathcal{O}\left(\|y-x\|^2\right)$ and by the fact that g is locally Lipschitz, we obtain an estimate of the form

$$|f(y) - \phi(y,x)| \leq L\|y-x\|^2,$$

which is even stronger than estimate (2.1), and implies all items in Definition 2.5.

Remark 2.10. The above example of a non-standard model is instructive, because we see that the first order model is far from unique. It may even happen that one of the models is strong, while another fails to be so. Consider for instance composite functions $f = \lambda_1 \circ F$, where $F : \mathbb{R}^n \to \mathbb{S}^m$ is class C^2 with values in the space $\mathbb{S}^m$ of $m \times m$ symmetric matrices, and where $\lambda_1 : \mathbb{S}^m \to \mathbb{R}$ is the maximum eigenvalue function. Then the nonstandard model $\phi(y, x) = \lambda_1 \left(F(x) + F'(x)(y - x)\right)$ is strong by Remark 2.9, but the standard model $\phi^\sharp(\cdot, x)$ is only strong at those x where $\lambda_1 \left(F(x)\right)$ has eigenvalue multiplicity 1.

Remark 2.11. A convex function f has the trivial strong model $\phi(y, x) = f(y)$. For short, f is its own strong model.

Remark 2.12. How about concave functions? Consider $-f$, where f is convex. Let $g \in \partial f(x)$, then the subgradient inequality gives $g^\top(y - x) \le f(y) - f(x)$ or what is the same

$$-f(y) \le -f(x) + g^\top(x - y) \le -f(x) + f^0(x; x - y) = -f(x) + (-f)^0(x; y - x) = \phi^\sharp(y, x).$$

That means the standard model is strong for concave functions. More generally, we have

Proposition 2.13. *Let f be locally Lipschitz. Suppose $-f$ is prox-regular at $\bar{x}$. Then the standard model of f is strong on a neighbourhood of $\bar{x}$.*

Proof. If $-f$ is prox-regular at $\bar{x}$ with respect to $\bar{g} \in \partial(-f)(\bar{x})$, then there exist $\epsilon > 0$ and $r > 0$ such that for all $x, x' \in B(\bar{x}, \epsilon)$ and all $g(x) \in \partial(-f)(x)$ with $\|g(x) - \bar{g}\| \le \epsilon$ one has

$$|f(x) - f(\bar{x})| < \epsilon \Rightarrow -f(x') \ge -f(x) + g(x)^\top(x' - x) - \tfrac{r}{2}\|x' - x\|^2.$$

That is the same as

$$\begin{aligned} f(x') &\le f(x) - g(x)^\top(x' - x) + \tfrac{r}{2}\|x' - x\|^2 \\ &\le f(x) + \sup_{-g \in \partial f(x)} (-g)^\top(x' - x) + \tfrac{r}{2}\|x' - x\|^2 = f(x) + f^0(x, x' - x) + \tfrac{r}{2}\|x' - x\|^2, \end{aligned}$$

which proves estimate (2.1) for f with constant $L = \frac{r}{2}$ on the ball $B(\bar{x}, \epsilon)$. □

Remark 2.14. Now let $f = f_1 - f_2$ where f_1 is convex and f_2 is prox-regular. Let $\phi_2^\sharp(\cdot, \cdot)$ be the standard model of $-f_2$, then in view of the previous remark

$$\phi(\cdot, x) = f_1(y) + \phi_2^\sharp(y, x) = f_1(y) - f_2(x) + (-f_2)^0(x; y - x)$$

is a natural candidate for a strong model of f. And it is indeed a strong model as soon as we can verify that it is a model. Axioms $(\widetilde{M}_2)$ and (M_3) being clear, what about axiom (M_1)? We have $\partial_1\phi(x, x) = \partial f_1(x) + \partial(-f_2)(x)$, which needs to be contained in $\partial f(x)$. Unfortunately, it is just the opposite inclusion which is always true, while $\partial f_1(x) + \partial(-f_2)(x) \subset \partial f(x)$ needs additional hypotheses, see e.g. [8]. Nonetheless, this shows that a natural candidate for a strong model for $f = f_1 - f_2$ is $\phi(y, x) = f_1 + \phi_2^\sharp(\cdot, x)$, where $\phi_2^\sharp$ is the standard model of $-f_2$.

Remark 2.15. Since $f(x) = |x|^{3/2}$ is convex, it is its own strong model $f = \phi(\cdot, x)$. But the standard model $\phi^\sharp(y, x) = |x|^{3/2} + \frac{3}{2}\text{sgn}(x)|x|^{1/2}(y - x)$ is not strong at $x = 0$. Indeed, at the origin we would have to find $L > 0$ such that $|y|^{3/2} \le \phi(y, 0) + L|y - 0|^2 = Ly^2$, but this fails for small y, no matter how large we choose L.

Now consider $f(x) = -|x|^{3/2}$, which is concave. The standard model is $\phi^\sharp(y, x) = -|x|^{3/2} - \frac{3}{2}\text{sgn}(x)|x|^{1/2}(y - x)$, which is strong by proposition 2.13. Notice that the differentiability properties of f and $-f$ are the same, and also is the standard model of $-|x|^{3/2}$ the negative of the standard model of $|x|^{3/2}$. Nonetheless, the second one is strong, the first one fails to be so.

Remark 2.16. Consider $f(x) = x^2 \sin(1/x)$ with $f(0) = 0$. At $x \neq 0$ the standard model is $\phi^\sharp(y,x) = f(x) + f'(x)(y-x)$. This formula is no longer correct at $x = 0$, where $\phi^\sharp(y,0) = f^0(0,y) = |y|$. This highlights the fact that f is not semismooth at $x = 0$, i.e. $\{f'(0)\} \neq \partial f(0)$.

Clearly estimate (2.1) holds at every $x \neq 0$ with some constant L_x, because f is of class C^2 around x. And at the origin we have $f(y) = y^2 \sin(1/y) \leq y^2 \leq f(0) + |y| = |y|$ for all $y \in [-1,1]$. In other words, estimate (2.1) holds again, even with $L_0 = 0$. Does this mean that the standard model is strong? The answer is no, and the reason is that the constants L_x explode as $x \to 0$. This is easily understood when we observe that $f''(x)$ is unbounded as $x \to 0$.

3 Working Model and Tangent Program

Let us now complete our local model by including second order information.

Definition 3.1. Let $\phi(\cdot,x)$ be a first order model of f at x. For a (not necessarily positive semidefinite) matrix $Q(x) = Q(x)^\top$ which is bounded on bounded sets of x, the function $\Phi(y,x) = \phi(y,x) + \frac{1}{2}(y-x)^\top Q(x)(y-x)$ is called a model of f at x.

Remark 3.2. We call $\frac{1}{2}(y-x)^\top Q(x)(y-x)$ the second order part of $\Phi(y,x)$. Why not allow more general second order expressions, say $\Psi(y,x)$ with $\partial_1\Psi(x,x) = 0$? Our choice is motivated by practical considerations, and will become clear as we go. Most of the theory would allow more general terms Ψ. However, solving the tangent program might then become impractical.

In our algorithm we do not work directly with $\phi(\cdot,x)$, but with an approximation $\phi_k(\cdot,x)$, which we update at each iteration k, and which is easier to manage than $\phi(\cdot,x)$.

Definition 3.3. We call $\phi_k(\cdot,x)$ a first-order working model of f at x if it is a convex function satisfying $\phi_k(x,x) = \phi(x,x) = f(x)$, $\partial_1\phi_k(x,x) \subset \partial_1\phi(x,x)$, and $\phi_k(\cdot,x) \leq \phi(\cdot,x)$. If $\Phi(y,x) = \phi(y,x) + \frac{1}{2}(y-x)^\top Q(x)(y-x)$ is a model, then $\Phi_k(y,x) = \phi_k(y,x) + \frac{1}{2}(y-x)^\top Q(x)(y-x)$ is called the corresponding working model.

Let $x \in \mathbb{R}^n$ be the current iterate. Suppose a working model $\Phi_k(y,x) = \phi_k(y,x) + \frac{1}{2}(y-x)^\top Q(x)(y-x)$ at counter k has been decided on. Fixing the so-called proximity control parameter $\tau_k > 0$, we solve the tangent program

$$\min_{y\in\mathbb{R}^n} \Phi_k(y,x) + \tfrac{\tau_k}{2}\|y-x\|^2. \tag{3.1}$$

Let y^{k+1} be a local solution of (3.1) in the sense that

$$0 \in \partial_1\phi_k(y^{k+1},x) + (Q(x) + \tau_k I)(y^{k+1}-x). \tag{3.2}$$

Notice that by monotonicity of the convex subdifferential y^{k+1} is unique as soon as $Q(x) + \tau_k I \succ 0$. We call y^{k+1} the trial step generated by the tangent program. The idea is that after some updates $k \to k+1$ the trial step y^{k+1} will improve over the current x and become the new iterate x^+.

Remark 3.4. In the convex case it is well-known that trust regions and proximity control can be seen as equivalent, cf. [15]. It turns out that the situation is the same even without convexity. Consider the trust region tangent program

$$\begin{array}{ll} \text{minimize} & \Phi_k(y,x) = \phi_k(y,x) + \frac{1}{2}(y-x)^\top Q(x)(y-x) \\ \text{subject to} & \|y-x\| \leq t_k \end{array} \tag{3.3}$$

where t_k is the trust region radius. Then solutions of (3.1) and (3.3) are in one-to-one correspondence in the sense that if y^{k+1} solves (3.1), then it is the local minimum of (3.3) for $t_k = \|x - y^{k+1}\|$. Conversely, if y^{k+1} is the local minimum of (3.3), and if the Lagrange multiplier for the trust region constraint associated with y^{k+1} is τ_k, then y^{k+1} solves (3.1) for that specific value of τ_k. Notice here that the usual argument to show that the Newton trust region program has no duality gap [31] carries over to our present case and explains why only τ-parameters with $Q(x) + \tau_k I \succeq 0$ are useful. The case $Q(x) + \tau_k I \succ 0$ is when the solution y^{k+1} of both programs is unique.

Let us get back to (3.1). The first question is what happens if the solution of (3.1) is $y^{k+1} = x$?

Lemma 3.5. *Suppose $y^{k+1} = x$. Then x is a critical point of f.*

Proof. By local optimality $0 \in \partial_1 \phi_k(y^{k+1}, x) + (Q(x) + \tau_k I)(y^{k+1} - x)$. Therefore $y^{k+1} = x$ implies $0 \in \partial_1 \phi_k(x, x)$. By axiom (M_1) we have $\partial_1 \phi_k(x, x) \subset \partial f(x)$, hence $0 \in \partial f(x)$ as claimed. □

In other words, unless x is already a solution of (1.1) in the sense of (1.2), the trial step y^{k+1} will *always* offer something new. Let us therefore assume that $0 \notin \partial f(x)$. In particular, in that case we have $\Phi_k(y^{k+1}, x) < \Phi_k(x, x) = f(x)$.

We call y^{k+1} a serious step if it is accepted as the new iterate, x^+, and a null step if it is rejected. In order to decide whether y^{k+1} is accepted or not, we introduce two constants $0 < \gamma < \Gamma < 1$ and compute the quotient

$$\rho_k = \frac{f(x) - f(y^{k+1})}{f(x) - \Phi_k(y^{k+1}, x)}, \tag{3.4}$$

which reflects the agreement between f and $\Phi_k(\cdot, x)$ at y^{k+1}. If the working model Φ_k is close to the true f, we expect $\rho_k \approx 1$. We say that agreement between f and $\Phi_k(\cdot, x)$ is good (at y^{k+1}) if $\rho_k > \Gamma$, where the reader might for instance imagine $\Gamma = \frac{3}{4}$. On the other hand we say that agreement between Φ_k and f is bad if $\rho_k < \gamma$, where the reader might imagine $\gamma = \frac{1}{4}$. Our strategy is now as follows. If $\rho_k \geq \gamma$, meaning that the trial step is *not bad*, we accept the trial step, and $x^+ = y^{k+1}$ becomes the new iterate. Otherwise y^{k+1} is rejected and ϕ_k has to be replaced by a better first order working model ϕ_{k+1} in order to obtain a better trial step y^{k+2} at the next sweep.

Notice that the bad case includes those trial steps where $\rho_k < 0$. Now since $0 \notin \partial_1 \Phi_k(x, x)$, it is always possible to decrease the value of $\Phi_k(y^{k+1}, x)$ below $\Phi_k(x, x) = f(x)$. In other words, the denominator in (3.4) is always > 0. Therefore, if $\rho_k < 0$, the numerator is < 0. This happens if the trial step y^{k+1} is not even a descent step of f.

Suppose the trial step y^{k+1} is a null step. Then working model Φ_k was not entirely useful, and we need to improve it to do better at step $k + 1$. Since the quadratic term $\frac{1}{2}(y - x)^\top Q(x)(y - x)$ remains unchanged during the inner loop $k \to k + 1$, we have to improve $\phi_k(y, x)$, which is done through cutting planes and aggregation. But will this be sufficient, or will we have to increase τ_k? In order to decide we introduce another control parameter:

$$\widetilde{\rho}_k = \frac{f(x) - \Phi(y^{k+1}, x)}{f(x) - \Phi_k(y^{k+1}, x)},$$

which reflects the agreement between $\Phi(\cdot, x)$ and $\Phi_k(\cdot, x)$ at y^{k+1}. We fix a constant $\widetilde{\gamma}$ with $\gamma < \widetilde{\gamma} < 1$, which plays a role similar to γ. We say that $\Phi_k(y^{k+1}, x)$ is far from $\Phi(y^{k+1}, x)$ if $\widetilde{\rho}_k < \widetilde{\gamma}$.

Suppose $\rho_k < \gamma$, but $\widetilde{\rho}_k > \widetilde{\gamma}$ i.e. $\Phi_k(y^{k+1}, x)$ is far from $f(y^{k+1})$, but at the same time $\Phi_k(\cdot, x)$ is already close to $\Phi(\cdot, x)$ at y^{k+1}. Using aggregation and cutting planes *alone* would now drive $\Phi_{k+1}(y^{k+2}, x)$ even closer to $\Phi(y^{k+2}, x)$, but would not suffice to make progress. Namely, $\Phi(y^{k+1}, x)$ was too far from $f(y^{k+1})$, and this phenomenon is likely to persist if $\|y^{k+2} - x\| \approx \|y^{k+1} - x\|$. In order to force $\Phi(y^{k+2}, x)$ closer to $f(y^{k+2})$, we have to reduce the trust region radius, or dually, to tighten proximity control, which means increasing τ_k. This is the meaning of step 6 of the algorithm.

How to build the new ϕ_{k+1}? The first element to guarantee is exactness $\phi_{k+1}(x, x) = \phi(x, x) = f(x)$ and $\partial_1 \phi_{k+1}(x, x) \subset \partial f(x)$. To do this pick an element $g(x) \in \partial_1 \phi(x, x) \subset \partial f(x)$. We define the affine function $m(y) = f(x) + g(x)^\top (y - x)$ and assure that $\phi_{k+1}(y, x) \geq m(y)$ for every y, while $\phi_{k+1} \leq \phi$. Then $\partial_1 \phi_{k+1}(x, x) \subset \partial f(x)$ is assured. As the reader will notice, m does not depend on the iteration counter k, and we can keep $m(\cdot) \leq \phi_k(\cdot, x) \leq \phi(\cdot, x)$ at all k.

To make ϕ_{k+1} better than ϕ_k we use two more elements, referred to as cutting planes, and aggregation, and these are constructed iteratively. We start by explaining cutting planes.

Assume y^{k+1} is a null step. Pick $g_{k+1} \in \partial_1 \phi(y^{k+1}, x)$, then by convexity of $\phi(\cdot, x)$

$$g_{k+1}^\top (y - y^{k+1}) \leq \phi(y, x) - \phi(y^{k+1}, x),$$

or what is the same, $m_{k+1}(y) = \phi(y^{k+1}, x) + g_{k+1}^\top (y - y^{k+1})$ is an affine support function to $\phi(\cdot, x)$ at y^{k+1}. Putting $a_{k+1} = \phi(y^{k+1}, x) + g_{k+1}^\top (x - y^{k+1})$, we obviously have $m_{k+1}(y) = a_{k+1} + g_{k+1}^\top (y - x)$. The affine function $m_{k+1}(\cdot)$ is called the cutting plane.

Lemma 3.6. *Suppose the convex working model $\phi_{k+1}(\cdot, x)$ is such that $\phi_{k+1}(y, x) \geq m_{k+1}(y)$ for every $y \in \mathbb{R}^n$, i.e., the cutting plane m_{k+1} is an affine minorant of $\phi_{k+1}(\cdot, x)$. Then we have $\phi_{k+1}(y^{k+1}, x) = \phi(y^{k+1}, x)$, and $g_{k+1} \in \partial_1 \phi_{k+1}(y^{k+1}, x)$.*

Proof. A convex working model ϕ_{k+1} must satisfy $\phi_{k+1}(\cdot, x) \leq \phi(\cdot, x)$, so the best value ϕ_{k+1} can possibly attain at y^{k+1} is $\phi(y^{k+1}, x)$. Since $m_{k+1}(y^{k+1}) = \phi(y^{k+1}, x)$ and $m_{k+1} \leq \phi_k(\cdot, x)$, this value is indeed attained at y^{k+1}. Knowing that $\phi_{k+1}(y^{k+1}, x) = \phi(y^{k+1}, x)$, it is now clear that a subgradient of $\phi(\cdot, x)$ at y^{k+1} is also a subgradient of $\phi_{k+1}(\cdot, x)$ at y^{k+1}. □

The effect of making the cutting plane m_{k+1} an affine support function to the next convex model ϕ_{k+1} at y^{k+1} is that the unsuccessful trial step y^{k+1} is cut away at the next trial $k + 1$, paving the way for a better y^{k+2} to come.

Aggregation, which we explain next, is used to recycle some of the information stored in ϕ_k for the new working model ϕ_{k+1}. To be more precise, the optimality condition for program (3.1) implies $0 \in \partial_1 \phi_k(y^{k+1}, x) + (Q(x) + \tau_k I)(y^{k+1} - x)$. In other words,

$$g_{k+1}^* := (Q(x) + \tau_k I)(x - y^{k+1}) \in \partial_1 \phi_k(y^{k+1}, x). \tag{3.5}$$

That means $m_{k+1}^*(y) = \phi_k(y^{k+1}, x) + g_{k+1}^{*\top}(y - y^{k+1})$ is an affine support function to $\phi_k(\cdot, x)$ at y^{k+1}. We can also write it in the more convenient form $m_{k+1}^*(y) = a_{k+1}^* + g_{k+1}^{*\top}(y - x)$, where $a_{k+1}^* = \phi_k(y^{k+1}, x) + g_{k+1}^{*\top}(x - y^{k+1})$. We call m_{k+1}^* the aggregate plane.

Lemma 3.7. *Let m_{k+1}^* be the affine support function to $\phi_k(\cdot, x)$ at the trial step y^{k+1} selected by the optimality condition (3.5). If the new convex working model $\phi_{k+1}(\cdot, x)$ has m_{k+1}^* as an affine minorant, i.e., satisfies $\phi_{k+1}(y, x) \geq m_{k+1}^*(y)$ for all y, then we have $\phi_k(y^{k+1}, x) \leq \phi_{k+1}(y^{k+1}, x)$.*

Aggregation is a clever substitute for storing a full sequence of models $\phi_k \leq \phi_{k+1} \to \phi$ of increasing complexity. Naturally, the latter would turn out too expensive. Altogether we have identified the following list of conditions the convex working model has to satisfy

- **Exactness**. $m(\cdot) \leq \phi_{k+1}(\cdot, x) \leq \phi(\cdot, x)$.
- **Cutting plane**. $m_{k+1}(\cdot) \leq \phi_{k+1}(\cdot, x)$.
- **Aggregation**. $m^*_{k+1}(\cdot) \leq \phi_{k+1}(\cdot, x)$.

We will see that these conditions are sufficient to ensure convergence of our bundle method. The formal statement of our algorithm is given by the algorithm 1.

Remark 3.8. Notice that the most basic choice to assure these three conditions is $\phi_{k+1}(y, x) = \max\{m(y), m_{k+1}(y), m^*_{k+1}(y)\}$. These three affine functions are support functions to $\phi(\cdot, x)$, so $\phi_{k+1} \leq \phi$ is guaranteed.

It is instructive to specialize to the convex case $f = \phi(\cdot, x)$. With $Q(x) \equiv 0$ and $\phi_k(\cdot, x)$ the standard polyhedral working model of remark 3.8, we recover a classical bundle method. Since $\Phi(\cdot, x) = \phi(\cdot, x) = f$ we have $\widetilde{\rho}_k = \rho_k$, so $\rho_k < \gamma$ in step 5 of the algorithm always implies $\widetilde{\rho}_k = \rho_k < \gamma < \widetilde{\gamma}$ in step 6. In consequence τ_k is never increased, but could be decreased if $\rho_k > \Gamma$. This shows that in convex bundling τ could be frozen once and for all. This is significant because in the non-convex case it *cannot*. In those convex bundle methods where τ is *not* frozen the motivation is to give the user additional freedom in the management of τ, but this rests optional. In the non-convex case moving τ becomes mandatory and the freedom is significantly reduced.

Algorithm 1. Proximity control algorithm for (1.1).

```
Parameters: $0 < \gamma < \widetilde{\gamma} < \Gamma < 1$, and $0 < q < \infty$.
1: Initialize outer loop. Choose initial guess $x^1$ and an initial matrix $Q_1 = Q_1^\top$ with
   $-qI \preceq Q_1 \preceq qI$. Then fix memory control parameter $\tau_1^\sharp$ such that $Q_1 + \tau_1^\sharp I \succ 0$. Put
   $j = 1$.
2: Stopping test. At outer loop counter $j$, stop if $0 \in \partial f(x^j)$. Otherwise goto inner loop.
3: Initialize inner loop. Put inner loop counter $k = 1$ and initialize $\tau$-parameter using
   the memory element, i.e., $\tau_1 = \tau_j^\sharp$. Choose initial convex working model $\phi_1(\cdot, x^j)$, and
   let $\Phi_1(y, x^j) = \phi_1(y, x^j) + \frac{1}{2}(y - x^j)^\top Q_j (y - x^j)$.
4: Trial step generation. At inner loop counter $k$ solve tangent program

       $$\min_{y \in \mathbb{R}^n} \Phi_k(y, x^j) + \frac{\tau_k}{2}\|y - x^j\|^2.$$

   Solution is the new trial step $y^{k+1}$.
5: Acceptance test. Check whether

       $$\rho_k = \frac{f(x^j) - f(y^{k+1})}{f(x^j) - \Phi_k(y^{k+1}, x^j)} \geq \gamma.$$

   If this is the case put $x^{j+1} = y^{k+1}$ (serious step), quit inner loop and goto step 8. On
   the other hand, if this is not the case (null step) continue inner loop with step 6.
6: Update proximity parameter. Compute second control parameter
```

$$\widetilde{\rho}_k = \frac{f(x^j) - \Phi(y^{k+1}, x^j)}{f(x^j) - \Phi_k(y^{k+1}, x^j)}.$$

Then put

$$\tau_{k+1} = \begin{cases} \tau_k, & \text{if } \widetilde{\rho}_k < \widetilde{\gamma} \\ \\ 2\tau_k, & \text{if } \widetilde{\rho}_k \geq \widetilde{\gamma} \end{cases} \qquad \text{(bad)}$$

7: **Update working model**. Build new convex working model $\phi_{k+1}(\cdot, x^j)$ by respecting the three rules (exactness, cutting plane, aggregation) based on null step y^{k+1}. Then increase inner loop counter k and continue inner loop with step 4.

8: **Update Q_j and memory element**. Update matrix $Q_j \to Q_{j+1}$ respecting $Q_{j+1} = Q_{j+1}^\top$ and $-qI \preceq Q_{j+1} \preceq qI$. Then store new memory element

$$\tau_{j+1}^\sharp = \begin{cases} \tau_{k+1}, & \text{if } \gamma \leq \rho_k < \Gamma \qquad \text{(not bad)} \\ \\ \dfrac{\tau_{k+1}}{2}, & \text{if } \rho_k \geq \Gamma \qquad \text{(good)} \end{cases}$$

Increase $\tau_{j+1}^\sharp$ if necessary to ensure $Q_{j+1} + \tau_{j+1}^\sharp I \succ 0$. Increase outer loop counter j by 1 and loop back to step 2.

$\boxed{4}$ Analysis of the Inner Loop

In this section we prove that the inner loop terminates with a serious step after a finite number of updates $k \to k+1$. The current iterate x is fixed, and so is $Q := Q(x)$. We assume that $\phi(\cdot, x)$ satisfies axioms (M_1) and (M_2). Neither axiom (M_3) nor the strong version $(\widetilde{M_2})$ will be needed in this section.

Lemma 4.1. *Let x be the current iterate. Suppose the inner loop produces an infinite sequence of null steps y^{k+1}. Then there exists $k_0 \in \mathbb{N}$ such that $\widetilde{\rho}_k < \widetilde{\gamma}$ for all $k \geq k_0$.*

Proof. i) By assumption none of the trial steps is accepted, so that $\rho_k < \gamma$ for all $k \in \mathbb{N}$. Suppose contrary to the statement that there are infinitely many inner loop instances k where $\widetilde{\rho}_k \geq \widetilde{\gamma}$. According to step 6 of the algorithm this means that the doubling rule is applied infinitely often. Since the proximity parameter τ_k is never decreased in the inner loop, this implies $\tau_k \to \infty$.

ii) Let us prove that under these circumstances, $y^{k+1} \to x$. Recall that $g_{k+1}^* = (Q + \tau_k I)(x - y^{k+1}) \in \partial_1 \phi_k(y^{k+1}, x)$. By the subgradient inequality

$$g_{k+1}^{*\top}(x - y^{k+1}) \leq \phi_k(x, x) - \phi_k(y^{k+1}, x) \leq f(x) - m(y^{k+1}), \tag{4.1}$$

where we use $\phi_k(x, x) = f(x)$ and the fact that the exactness plane $m(\cdot)$ satisfies $m(y^{k+1}) \leq \phi_k(y^{k+1}, x)$. Since $m(y) = f(x) + g(x)^\top (y - x)$ for some $g(x) \in \partial f(x)$, the term on the right of (4.1) is $g(x)^\top (x - y^{k+1})$, and expanding the term on the left of (4.1) gives

$$(x - y^{k+1})^\top (Q + \tau_k I)(x - y^{k+1}) \leq g(x)^\top (x - y^{k+1}) \leq \|g(x)\| \|x - y^{k+1}\|. \tag{4.2}$$

Since $\tau_k \to \infty$, the term on the left of (4.2) behaves asymptotically like $\tau_k\|x - y^{k+1}\|^2$. Dividing (4.2) by $\|x - y^{k+1}\|$ therefore shows that $\tau_k\|x - y^{k+1}\|$ is bounded by $\|g(x)\|$. As $\tau_k \to \infty$, this could only mean $y^{k+1} \to x$.

iii) Let us use $y^{k+1} \to x$ and go back to formula (4.1). It is now clear from the passage to (4.2) that $\phi_k(x, x) - \phi_k(y^{k+1}, x) = f(x) - \phi_k(y^{k+1}, x)$ is sandwiched in between two terms with limit 0. This implies $\phi_k(y^{k+1}, x) \to f(x)$.

Keeping this in mind, let us use the subgradient inequality (4.2) again and subtract a term $\frac{1}{2}(x - y^{k+1})^\top Q(x - y^{k+1})$ from both sides. That gives the estimate

$$\tfrac{1}{2}(x - y^{k+1})^\top Q(x - y^{k+1}) + \tau_k\|x - y^{k+1}\|^2 \leq f(x) - \Phi_k(y^{k+1}, x).$$

For those k where $\tau_k > \|Q\|$, we clearly have

$$\|g^*_{k+1}\| \leq \|Q\|\|x - y^{k+1}\| + \tau_k\|x - y^{k+1}\| \leq 2\tau_k\|x - y^{k+1}\|.$$

Hence, for k with $\tau_k > \|Q\|$

$$\begin{aligned} \tfrac{1}{2}(x - y^{k+1})^\top Q(x - y^{k+1}) + \tau_k\|x - y^{k+1}\|^2 &\geq \tau_k\|x - y^{k+1}\|^2 - \tfrac{1}{2}\|Q\|\|x - y^{k+1}\|^2 \\ &\geq \tfrac{1}{2}\tau_k\|x - y^{k+1}\|^2 \geq \tfrac{1}{4}\|g^*_{k+1}\|\|x - y^{k+1}\|. \end{aligned}$$

Altogether, this gives

$$f(x) - \Phi_k(y^{k+1}, x) \geq \tfrac{1}{4}\|g^*_{k+1}\|\|x - y^{k+1}\|. \tag{4.3}$$

iv) Now let $\epsilon_k := \operatorname{dist}\big(g^*_{k+1}, \partial_1\phi(x, x)\big)$. We argue that $\epsilon_k \to 0$. Indeed, using the subgradient inequality at y^{k+1} in tandem with $\phi(\cdot, x) \geq \phi_k(\cdot, x)$, we have for all $y \in \mathbb{R}^n$

$$\phi(y, x) \geq \phi_k(y^{k+1}, x) + {g^*_{k+1}}^\top(y - y^{k+1}).$$

Since the subgradients g^*_{k+1} are bounded by part ii), there exists an infinite subsequence $\mathcal{N} \subset \mathbb{N}$ such that $g^*_{k+1} \to g^*$, $k \in \mathcal{N}$, for some g^*. Passing to the limit $k \in \mathcal{N}$ and using $y^{k+1} \to x$ and $\phi_k(y^{k+1}, x) \to f(x) = \phi(x, x)$, we have $\phi(y, x) \geq \phi(x, x) + g^{*\top}(y - x)$ for all y. Hence $g^* \in \partial_1\phi(x, x)$, which means $\epsilon_k = \operatorname{dist}(g^*_{k+1}, \partial_1\phi(x, x)) \leq \|g^*_{k+1} - g^*\| \to 0$, $k \in \mathcal{N}$, proving the claim.

v) Using the definition of ϵ_k, choose $\tilde{g}_{k+1} \in \partial_1\phi(x, x)$ such that $\|g^*_{k+1} - \tilde{g}_{k+1}\| = \epsilon_k$. Now observe that we enter the inner loop because $0 \notin \partial f(x)$, which by axiom (M_1) gives $0 \notin \partial_1\phi(x, x)$. That means $\operatorname{dist}(0, \partial_1\phi(x, x)) = \eta > 0$, so that $\|\tilde{g}_{k+1}\| \geq \eta > 0$ for all $k \in \mathcal{N}$. Hence $\|g^*_{k+1}\| \geq \eta - \epsilon_k > \frac{\eta}{2}$ for $k \in \mathcal{N}$ large enough, given that $\epsilon_k \to 0$. Going back with this to (4.3) we deduce

$$f(x) - \Phi_k(y^{k+1}, x) \geq \tfrac{\eta}{8}\|x - y^{k+1}\| \tag{4.4}$$

for $k \in \mathcal{N}$ large enough. Since $y^{k+1} \to x$, axiom (M_2) gives a sequence $\omega_k \to 0^+$ such that $f(y^{k+1}) - \phi(y^{k+1}, x) \leq \omega_k\|x - y^{k+1}\|$. Putting $\widetilde{\omega}_k := \omega_k + \|Q\|\|x - y^{k+1}\| \to 0$ implies

$$\begin{aligned} f(y^{k+1}) - \Phi(y^{k+1}, x) &\leq \omega_k\|x - y^{k+1}\| + \|Q\|\|x - y^{k+1}\|^2 \\ &\leq \widetilde{\omega}_k\|x - y^{k+1}\|. \end{aligned} \tag{4.5}$$

Now to conclude let us expand the test parameters as follows

$$\begin{aligned} \widetilde{\rho}_k &= \rho_k + \frac{f(y^{k+1}) - \Phi(y^{k+1}, x)}{f(x) - \Phi_k(y^{k+1}, x)} \\ &\leq \rho_k + \frac{\widetilde{\omega}_k\|x - y^{k+1}\|}{\eta/8\|x - y^{k+1}\|} = \rho_k + 8\widetilde{\omega}_k/\eta \qquad \text{(estimates (4.5) and (4.4))} \end{aligned}$$

Since $\widetilde{\omega}_k \to 0$, we have $\limsup_{k\to\infty} \widetilde{\rho}_k \leq \limsup_{k\to\infty} \rho_k \leq \gamma$, contradicting the fact that $\widetilde{\rho}_k \geq \widetilde{\gamma} > \gamma$ for infinitely many k. That proves the result. □

The technique of proof of the following result is essentially known, and in the case $Q = 0$ the results could be obtained from [10, Proposition 4.3], [15, Chapter XV] or Part II of [6]. For a recent extension to prox-regular f, see [14].

Lemma 4.2. *Let $0 \notin \partial f(x)$. Then the inner loop finds a serious iterate after a finite number of trial steps y^{k+1}.*

Proof. i) We proceed by contradiction and show that if the inner loop turns forever, we must have $0 \in \partial f(x)$.

From Lemma 4.1 we see that if the inner loop turns forever, there exists $k_0 \in \mathbb{N}$ such that $\rho_k < \gamma$ and $\widetilde{\rho}_k < \widetilde{\gamma}$ for all $k \geq k_0$. According to the update rule in step 8 of algorithm 1, this means the proximity control parameter is frozen from inner loop counter k_0 onwards: $\tau := \tau_k$ for $k \geq k_0$.

ii) We prove that the sequence of trial steps y^{k+1} is bounded. Using inequality (4.2) based on the subgradient inequality and on the support property of the exactness plane, we get the estimate

$$(x - y^{k+1})^\top (Q + \tau I)(x - y^{k+1}) \leq \|g(x)\| \|x - y^{k+1}\|.$$

Since the τ-parameter is frozen and $Q + \tau I \succ 0$, the expression on the left is the square $\|x - y^{k+1}\|^2_{Q+\tau I}$ of the Euclidean norm derived from $Q + \tau I$. Since both norms are equivalent, we deduce after dividing by $\|x - y^{k+1}\|$ that $\|x - y^{k+1}\|_{Q+\tau I} \leq C\|g(x)\|$ for some constant $C > 0$ and all k. This proves the claim.

Notice an important difference with Lemma 4.1. Since the τ-parameter is frozen, we cannot conclude at this stage that $y^{k+1} \to x$. Proving this will turn out considerably more complicated.

iii) Let us introduce the objective function of program (3.1) for $k \geq k_0$:

$$\psi_k(y, x) = \phi_k(y, x) + \tfrac{1}{2}(y - x)^\top (Q + \tau I)(y - x).$$

Let m^*_{k+1} be the aggregate plane, then as we know $\phi_k(y^{k+1}, x) = m^*_{k+1}(y^{k+1})$, and therefore also

$$\psi_k(y^{k+1}, x) = m^*_{k+1}(y^{k+1}) + \tfrac{1}{2}(y^{k+1} - x)^\top (Q + \tau I)(y^{k+1} - x).$$

We introduce the quadratic function $\psi^*_k(y, x) = m^*_{k+1}(y) + \frac{1}{2}(y - x)^\top (Q + \tau I)(y - x)$. Then

$$\psi_k(y^{k+1}, x) = \psi^*_k(y^{k+1}, x) \tag{4.6}$$

by what we have just seen. By the aggregate condition we have $m^*_{k+1}(y) \leq \phi_{k+1}(y, x)$, so that

$$\psi^*_k(y, x) \leq \psi_{k+1}(y, x). \tag{4.7}$$

Notice that $\nabla \psi^*_k(y, x) = \nabla m^*_{k+1}(y) + (Q + \tau I)(y - x) = g^*_{k+1} + (Q + \tau I)(y - x)$, so that $\nabla \psi^*_k(y^{k+1}, x) = 0$ by (3.5). We therefore have the relation

$$\psi^*_k(y, x) = \psi^*_k(y^{k+1}, x) + \tfrac{1}{2}(y - y^{k+1})^\top (Q + \tau I)(y - y^{k+1}), \tag{4.8}$$

which is obtained by Taylor expansion of $\psi^*_k(\cdot, x)$ at y^{k+1}. Notice again that step 8 of the algorithm assures $Q + \tau I \succ 0$, so that the quadratic expression defines the Euclidean norm $\| \cdot \|_{Q+\tau I}$.

iv) From the previous point iii) we now have

$$\begin{array}{rcll} \psi_k(y^{k+1},x) & \leq & \psi_k^*(y^{k+1},x) + \frac{1}{2}\|y^{k+2} - y^{k+1}\|^2_{Q+\tau I} & \text{(using (4.6))} \\ & = & \psi_k^*(y^{k+2},x) & \text{(using (4.8))} \\ & \leq & \psi_{k+1}(y^{k+2},x) & \text{(using (4.7))} \\ & \leq & \psi_{k+1}(x,x) & (y^{k+2} \text{ minimizer of } \psi_{k+1}) \\ & = & \phi_{k+1}(x,x) = f(x). & \end{array} \tag{4.9}$$

We deduce that the sequence $\psi_k(y^{k+1},x)$ is monotonically increasing and bounded above by $f(x)$. It therefore converges to some value $\psi^* \leq f(x)$.

Going back to (4.9) with this information shows that the term $\frac{1}{2}\|y^{k+2} - y^{k+1}\|^2_{Q+\tau I}$ is squeezed in between two convergent terms with the same limit, ψ^*, which implies $\frac{1}{2}\|y^{k+1} - y^{k+2}\|^2_{Q+\tau I} \to 0$. Consequently, $\|y^{k+1} - x\|^2_{Q+\tau I} - \|y^{k+2} - x\|^2_{Q+\tau I}$ also tends to 0, because the sequence of trial steps y^{k+1} is bounded by part ii).

Recalling $\phi_k(y,x) = \psi_k(y,x) - \frac{1}{2}\|y-x\|^2_{Q+\tau I}$, we deduce, using both convergence results, that

$$\begin{array}{c} \phi_{k+1}(y^{k+2},x) - \phi_k(y^{k+1},x) = \psi_{k+1}(y^{k+2},x) - \psi_k(y^{k+1},x) \\ -\frac{1}{2}\|y^{k+2} - x\|^2_{Q+\tau I} + \frac{1}{2}\|y^{k+1} - x\|^2_{Q+\tau I} \to 0. \end{array} \tag{4.10}$$

v) Recall that according to our algorithm the cutting plane m_{k+1} is an affine support function of $\phi_{k+1}(\cdot,x)$ at y^{k+1} (cf. Lemma 3.6). By the subgradient inequality this implies

$$g_{k+1}^\top(y - y^{k+1}) \leq \phi_{k+1}(y,x) - \phi_{k+1}(y^{k+1},x).$$

Since $\phi_{k+1}(y^{k+1},x) = \phi(y^{k+1},x)$, we have

$$\phi(y^{k+1},x) + g_{k+1}^\top(y - y^{k+1}) \leq \phi_{k+1}(y,x). \tag{4.11}$$

Now we estimate

$$\begin{array}{rcll} 0 & \leq & \phi(y^{k+1},x) - \phi_k(y^{k+1},x) & \\ & = & \phi(y^{k+1},x) + g_{k+1}^\top(y^{k+2} - y^{k+1}) - \phi_k(y^{k+1},x) - g_{k+1}^\top(y^{k+2} - y^{k+1}) & \\ & \leq & \phi_{k+1}(y^{k+2},x) - \phi_k(y^{k+1},x) + \|g_{k+1}\|\|y^{k+2} - y^{k+1}\| & \text{(using (4.11))} \end{array}$$

and this term converges to 0, because of (4.10), because the g_{k+1} are bounded, and because $y^{k+2} - y^{k+1} \to 0$ by part iv) above. Boundedness of the g_{k+1} follows from boundedness of the trial steps y^{k+1} shown in part ii). Indeed, $g_{k+1} \in \partial_1\phi(y^{k+1},x)$, and the subdifferential of $\phi(\cdot,x)$ is uniformly bounded on the bounded set $\{y^{k+1} : k \in \mathbb{N}\}$. We deduce that $\phi(y^{k+1},x) - \phi_k(y^{k+1},x) \to 0$. Obviously, that also gives $\Phi(y^{k+1},x) - \Phi_k(y^{k+1},x) \to 0$.

vi) We now proceed to prove $\Phi_k(y^{k+1},x) \to f(x)$, and then of course also $\Phi(y^{k+1},x) \to f(x)$. Assume this is not the case, then $\limsup_{k\to\infty} f(x) - \Phi_k(y^{k+1},x) =: \eta > 0$. Choose $\delta > 0$ such that $\delta < (1-\widetilde{\gamma})\eta$. It follows from v) above that there exists $k_1 \geq k_0$ such that

$$\Phi(y^{k+1},x) - \delta \leq \Phi_k(y^{k+1},x)$$

for all $k \geq k_1$. Using $\widetilde{\rho}_k \leq \widetilde{\gamma}$ for all $k \geq k_0$ then gives

$$\widetilde{\gamma}\left(\Phi_k(y^{k+1},x) - f(x)\right) \leq \Phi(y^{k+1},x) - f(x) \leq \Phi_k(y^{k+1},x) + \delta - f(x).$$

Passing to the limit implies $-\widetilde{\gamma}\eta \leq -\eta + \delta$, contradicting the choice of δ. This proves $\eta = 0$.

vii) Having shown $\Phi_k(y^{k+1},x) \to f(x)$, and therefore also $\Phi(y^{k+1},x) \to f(x)$, we now argue that $y^{k+1} \to x$. This follows indeed from the definition of ψ_k, because

$$\Phi_k(y^{k+1},x) \leq \psi_k(y^{k+1},x) = \Phi_k(y^{k+1},x) + \tfrac{\tau}{2}\|y^{k+1}-x\|^2 \leq \psi^* \leq f(x).$$

Since $\Phi_k(y^{k+1},x) \to f(x)$ by vi), we have indeed $\frac{\tau}{2}\|y^{k+1}-x\|^2 \to 0$ by a sandwich argument, which also proves *en passant* that $\psi^* = f(x)$ and $\phi_k(y^{k+1},x) \to f(x)$.

To finish the proof, let us now show $0 \in \partial f(x)$. Remember that by the necessary optimality condition for (3.1) we have $(Q+\tau I)(x-y^{k+1}) \in \partial_1\phi_k(y^{k+1},x)$. By the subgradient inequality,

$$\begin{aligned}(x-y^{k+1})^\top(Q+\tau I)(y-y^{k+1}) &\leq \phi_k(y,x)-\phi_k(y^{k+1},x)\\ &\leq \phi(y,x)-\phi_k(y^{k+1},x) \qquad \text{(using } \phi_k \leq \phi).\end{aligned}$$

Passing to the limit, observing $\|x-y^{k+1}\|^2_{Q+\tau I} \to 0$ and $\phi_k(y^{k+1},x) \to f(x) = \phi(x,x)$, we obtain

$$0 \leq \phi(y,x)-\phi(x,x)$$

for all $y \in \mathbb{R}^n$. This proves $0 \in \partial\phi_1(x,x)$, and since $\partial_1\phi(x,x) \subset \partial f(x)$ by (M_1), we have $0 \in \partial f(x)$. □

Remark 4.3. Our algorithm does not leave much freedom in the management of τ_k, but Lemmas 4.1 and 4.2 cover more general strategies. For instance, in case $\rho_k < \gamma$ and $\widetilde{\rho}_k \geq \widetilde{\gamma}$ it suffices to make sure that τ_k increases to infinity to ultimately force $\widetilde{\rho}_k < \widetilde{\gamma}$. This could of course be achieved by more general rules than just doubling. The proof of Lemma 4.1 easily adapts.

$\boxed{5}$ Convergence of the Outer Loop

All we have to do now is piece things together and show subsequence convergence of the sequence of serious steps x^j retained in the outer loop. Notice that the ideal model used at the serious iterates x^j is now

$$\Phi(y,x^j) = \phi(y,x^j) + \tfrac{1}{2}(y-x^j)^\top Q_j(y-x^j),$$

where $Q_j = Q(x^j)$ is updated in the outer loop (step 8 of the algorithm) and remains unchanged during the inner loop. That means, we allow the quadratic term to be updated only between serious steps. We assume that $Q_j + \tau_{k_j} I \succ 0$, which is guaranteed by starting the inner loop with a τ parameter having this property (step 8). Since the τ-parameter is never decreased in the inner loop, this property remains valid at all instances k of the inner loop. We now have the following

Theorem 5.1. *Let f be a locally Lipschitz function, and let $x^1 \in \mathbb{R}^n$ be such that the level set $\{x \in \mathbb{R}^n : f(x) \leq f(x^1)\}$ is bounded. Suppose f has a strong model $\phi(\cdot,x)$. Then every accumulation point of the sequence of serious steps x^j generated by the algorithm is a critical point of f.*

Proof. i) From the analysis in section 4 we know that the inner loop always ends after a finite number of steps k with a new x^+ satisfying the acceptance test, unless we have finite termination due to $0 \in \partial f(x)$. Let us exclude this case for the following. Let x^j denote the infinite sequence of serious steps, satisfying the acceptance test in step 5 of the algorithm, and let $\bar{x}$ be an accumulation point of x^j. We assume that at outer loop counter j the inner loop finds a serious step at inner counter $k = k_j$. In other words, $y^{k_j+1} = x^{j+1}$ passes the acceptance test in step 5 of the algorithm and becomes a serious iterate, while the y^{k+1} with $k < k_j$ are null steps. That means

$$f(x^j) - f(x^{j+1}) \geq \gamma \left(f(x^j) - \Phi_{k_j}(x^{j+1}, x^j)\right). \tag{5.1}$$

Now recall that $\tau_{k_j}\left(x^j - x^{j+1}\right) \in \partial_1 \Phi_{k_j}(x^{j+1}, x^j)$ due to (3.5), or what is the same, $(Q_j + \tau_{k_j} I)(x^j - x^{j+1}) \in \partial_1 \phi_{k_j}(x^{j+1}, x^j)$. The subgradient inequality for $\phi_{k_j}(\cdot, x^j)$ at x^{j+1} therefore gives

$$\begin{array}{rcl}\left(x^j - x^{j+1}\right)^\top (Q_j + \tau_{k_j} I)(x^j - x^{j+1}) & \leq & \phi_{k_j}(x^j, x^j) - \phi_{k_j}(x^{j+1}, x^j) \\ & = & f(x^j) - \phi_{k_j}(x^{j+1}, x^j),\end{array}$$

using $\phi_{k_j}(x^j, x^j) = f(x^j)$. With $\Phi_k(y, x^j) = \phi_k(y, x^j) + \frac{1}{2}(y - x^j)^\top Q_j (y - x^j)$ we therefore have

$$\tfrac{1}{2}\|x^{j+1} - x^j\|^2_{Q_j + \tau_{k_j} I} \leq f(x^j) - \Phi_{k_j}(x^{j+1}, x^j) \leq \gamma^{-1}\left(f(x^j) - f(x^{j+1})\right),$$

using (5.1). Now, recall that $f(x^j) \geq f(x^{j+1})$ because our method is of descent type in the serious steps. Since $\bar{x}$ is an accumulation point of the sequence x^j, $f(\bar{x})$ is an accumulation point of $f(x^j)$, and by monotonicity, we deduce $f(x^j) \to f(\bar{x})$, $(j \to \infty)$. Hence $f(x^j) - f(x^{j+1}) \to 0$ and then also $\|x^{j+1} - x^j\|^2_{Q_j + \tau_{k_j} I} \to 0$ as $j \to \infty$.

ii) Let us now prove that this implies $g_j := (Q_j + \tau_{k_j} I)\left(x^j - x^{j+1}\right) \to 0$, $(j \to \infty)$. Assume on the contrary that there exists an infinite subset $\mathcal{N}$ of $\mathbb{N}$ and some $\mu > 0$ such that $\|g_j\| = \|(Q_j + \tau_{k_j} I)(x^j - x^{j+1})\| \geq \mu > 0$ for every $j \in \mathcal{N}$.

We first notice that under this assumption the τ_{k_j}, $j \in \mathcal{N}$, must be unbounded. Indeed, assume on the contrary that the τ_{k_j}, $j \in \mathcal{N}$, are bounded. By boundedness of Q_j, and boundedness of the serious steps, there exists then an infinite subsequence $\mathcal{N}'$ of $\mathcal{N}$ such that Q_j, τ_{k_j} and $x^j - x^{j+1}$ converge respectively to $\bar{Q}$, $\bar{\tau}$ and $\delta\bar{x}$ as $j \in \mathcal{N}'$. This implies that the corresponding subsequence of g_j converges to $(\bar{Q} + \bar{\tau} I)\delta\bar{x}$, where $\|(\bar{Q} + \bar{\tau} I)\delta\bar{x}\| \geq \mu > 0$. Similarly, $(x^j - x^{j+1})^\top (Q_j + \tau_{k_j} I)(x^j - x^{j+1}) \to \delta\bar{x}^\top (\bar{Q} + \bar{\tau} I)\delta\bar{x}$. By part i) of the proof we have $g_j^\top (x^{j+1} - x^j) = \|x^{j+1} - x^j\|^2_{Q_j + \tau_{k_j} I} \to 0$, which means $\delta\bar{x}^\top (\bar{Q} + \bar{\tau} I)\delta\bar{x} = 0$. Since $\bar{Q} + \bar{\tau} I$ is symmetric and $\bar{Q} + \bar{\tau} I \succeq 0$, we deduce $(\bar{Q} + \bar{\tau} I)\delta\bar{x} = 0$, contradicting $\|(\bar{Q} + \bar{\tau} I)\delta\bar{x}\| \geq \mu > 0$. This argument proves that the τ_{k_j}, $j \in \mathcal{N}$, are unbounded.

iii) We now argue that there exists yet another infinite subsequence $\mathcal{N}'$ of $\mathbb{N}$ with $\tau_{k_j} \to \infty$, $(j \in \mathcal{N}')$, such that in addition for each $j \in \mathcal{N}'$, the doubling rule in step 6 of the algorithm was applied at least once before the step $x^{j+1} = y^{k_j+1}$ was accepted. Indeed, to construct $\mathcal{N}'$ we let, for every $j \in \mathcal{N}$, $j' \leq j$ be that outer-loop instant where the τ-parameter was increased for the last time before j, and we put $\mathcal{N}' := \{j' : j \in \mathcal{N}\}$. It is possible that $j' = j$, but in general we may have $j' < j$, and we only know that

$$2\tau_{k_{j'-1}} \leq \tau_{k'_j} \text{ and } \tau_{k_{j'}} \geq \tau_{k_{j'+1}} \geq \cdots \geq \tau_{k_j}.$$

Since $\tau_{k_j} \to \infty$, $j \in \mathcal{N}$, this assures $\tau_{k_{j'}} \to \infty$, $j' \in \mathcal{N}'$. Since the doubling rule was applied at least once at the outer-loop counter j', the set $\mathcal{N}'$ is as claimed.

Let us say that for $j \in \mathcal{N}'$ the doubling rule was applied for the last time at stage $\tau_{k_j - \nu_j}$ for some $\nu_j \geq 1$. That means, $\tau_{k_j - \nu_j + 1} = 2\tau_{k_j - \nu_j}$, while the τ-parameter remained unchanged during the following inner steps before acceptance:

$$\tau_{k_j} = \tau_{k_j - 1} = \cdots = \tau_{k_j - \nu_j + 1} = 2\tau_{k_j - \nu_j}. \tag{5.2}$$

Now recall that in step 6 of the algorithm we have $\rho_k < \gamma$ and $\widetilde{\rho}_k \geq \widetilde{\gamma}$ for those k where the trial step was not accepted and the doubling rule was applied. Since this is the case at stage $k_j - \nu_j$ we have

$$\rho_{k_j-\nu_j} = \frac{f(x^j) - f(y^{k_j-\nu_j+1})}{f(x^j) - \Phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j)} < \gamma$$

and

$$\widetilde{\rho}_{k_j-\nu_j} = \frac{f(x^j) - \Phi(y^{k_j-\nu_j+1}, x^j)}{f(x^j) - \Phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j)} \geq \widetilde{\gamma}.$$

By (5.2) we now have $(Q_j + \frac{1}{2}\tau_{k_j} I)\left(x^j - y^{k_j-\nu_j+1}\right) \in \partial_1 \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j)$. Using the subgradient inequality for $\phi_{k_j-\nu_j}(\cdot, x^j)$ at $y^{k_j-\nu_j+1}$ and $\phi_{k_j-\nu_j}(x^j, x^j) = f(x^j)$, we obtain

$$\begin{aligned}\left(x^j - y^{k_j-\nu_j+1}\right)^\top (Q_j + \tfrac{1}{2}\tau_{k_j} I)\left(x^j - y^{k_j-\nu_j+1}\right) &\leq \phi_{k_j-\nu_j}(x^j, x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j) \\ &= f(x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j),\end{aligned}$$

which on subtracting $\frac{1}{2}(x^j - y^{k_j-\nu_j+1})^\top Q_j (x^j - y^{k_j-\nu_j+1})$ from both sides becomes

$$\tfrac{1}{2}(x^j - y^{k_j-\nu_j+1})^\top (Q_j + \tau_{k_j} I)(x^j - y^{k_j-\nu_j+1}) \leq f(x^j) - \Phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j).$$

Using $\|x^j - y^{k_j-\nu_j+1}\|^2_{Q_j+\tau_{k_j} I} \geq (\tau_{k_j} - \|Q_j\|)\|x^j - y^{k_j-\nu_j+1}\|^2$, this could also be written as

$$\frac{(\tau_{k_j} - \|Q_j\|)\|x^j - y^{k_j-\nu_j+1}\|^2}{f(x^j) - \Phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j)} \leq 2. \tag{5.3}$$

Now, substituting (5.3) into thē expression for $\widetilde{\rho}_{k_j-\nu_j}$ and expanding gives

$$\begin{aligned}\widetilde{\rho}_{k_j-\nu_j} &= \rho_{k_j-\nu_j} + \frac{f(y^{k_j-\nu_j+1}) - \Phi(y^{k_j-\nu_j+1}, x^j)}{f(x^j) - \Phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j)} \\ &\leq \rho_{k_j-\nu_j} + \frac{(L + \|Q_j\|)\|x^j - y^{k_j-\nu_j+1}\|^2}{f(x^j) - \Phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j)} \qquad \text{(using (2.1))} \\ &\leq \rho_{k_j-\nu_j} + 2\frac{L + \|Q_j\|}{\tau_{k_j} - \|Q_j\|} \qquad \text{(using (5.3)).}\end{aligned}$$

Here estimate (2.1) is applied to the set $B = \{x^j, y^{k_j-\nu_j+1} : j \in \mathcal{N}'\}$, which as we now argue is bounded. Indeed, to see this observe that $\tau_{k_j-\nu_j} = \frac{1}{2}\tau_{k_j} \to \infty$ as $j \in \mathcal{N}'$. Applying the subgradient inequality to $\widetilde{g}_j = (Q_j + \tau_{k_j-\nu_j} I)(x^j - y^{k_j-\nu_j+1}) \in \partial_1 \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j)$ gives

$$\begin{aligned}(x^j - y^{k_j-\nu_j+1})^\top (Q_j + \tau_{k_j-\nu_j+1} I)(x^j - y^{k_j-\nu_j+1}) &\leq \phi_{k_j-\nu_j}(x^j, x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j) \\ &\leq f(x^j) - m_j(y^{k_j-\nu_j+1}) = g(x^j)^\top (x^j - y^{k_j-\nu_j+1}) \\ &\leq \|g(x^j)\|\|x^j - y^{k_j-\nu_j+1}\|.\end{aligned}$$

Here $m_j(\cdot)$ is the exactness plane at x^j. By (5.2) we have $\tau_{k_j-\nu_j+1} \to \infty$, so using boundedness of the x^j and boundedness of the Q_j, we deduce (via the argument already employed in the proof of Lemma 4.1) that the $y^{k_j-\nu_j+1}$ are bounded.

Going back to the above estimate involving $\widetilde{\rho}_{k_j-\nu_j}$ and $\rho_{k_j-\nu_j}$, notice that $\rho_{k_j-\nu_j} < \gamma$ and $(L + \|Q_j\|)/(\tau_{k_j} - \|Q_j\|) \to 0$ imply $\limsup_{j\to\infty} \widetilde{\rho}_{k_j-\nu_j} \leq \gamma$ in that estimate, contradicting $\widetilde{\rho}_{k_j-\nu_j} \geq \widetilde{\gamma} > \gamma$ for the infinitely many $j \in \mathcal{N}'$. That proves $g_j \to 0$, $j \to \infty$.

iv) Having shown that $g_j := (Q_j + \tau_{k_j} I)(x^j - x^{j+1}) \to 0$, $(j \to \infty)$, let us argue that every accumulation point $\bar{x}$ of the sequence x^j of serious steps must be critical. Notice that since $\{x \in \mathbb{R}^n : f(x) \leq f(x^1)\}$ is bounded by hypothesis, and since our algorithm is of descent type in the serious steps, the sequence x^j is bounded.

Since g_j is a subgradient of $\phi_{k_j}(\cdot, x^j)$ at $x^{j+1} = y^{k_j+1}$, we have for every test vector h:

$$\begin{array}{rcl} g_j^\top h & \leq & \phi_{k_j}(x^{j+1} + h, x^j) - \phi_{k_j}(x^{j+1}, x^j) \\ & \leq & \phi(x^{j+1} + h, x^j) - \phi_{k_j}(x^{j+1}, x^j) \qquad (\text{using } \phi_{k_j} \leq \phi). \end{array}$$

Now we use the fact that $y^{k_j+1} = x^{j+1}$ was accepted in step 5 of the algorithm, which means

$$\gamma^{-1}\left(f(x^j) - f(x^{j+1})\right) \geq f(x^j) - \Phi_{k_j}(x^{j+1}, x^j).$$

Combining these two estimates for a fixed test vector h gives:

$$\begin{array}{rcl} g_j^\top h & \leq & \phi(x^{j+1} + h, x^j) - f(x^j) + f(x^j) - \phi_{k_j}(x^{j+1}, x^j) \\ & = & \phi(x^{j+1} + h, x^j) - f(x^j) + f(x^j) - \Phi_{k_j}(x^{j+1}, x^j) + \frac{1}{2}(x^j - x^{j+1})^\top Q_j (x^j - x^{j+1}) \\ & \leq & \phi(x^{j+1} + h, x^j) - f(x^j) + \gamma^{-1}\left(f(x^j) - f(x^{j+1})\right) + \frac{1}{2}(x^j - x^{j+1})^\top Q_j (x^j - x^{j+1}) \\ & = & \phi(x^{j+1} + h, x^j) - f(x^j) + \gamma^{-1}\left(f(x^j) - f(x^{j+1})\right) \\ & & \qquad + \frac{1}{2}(x^j - x^{j+1})^\top (Q_j + \tau_{k_j} I)(x^j - x^{j+1}) - \frac{\tau_{k_j}}{2}\|x^j - x^{j+1}\|^2 \\ & \leq & \phi(x^{j+1} + h, x^j) - f(x^j) + \gamma^{-1}\left(f(x^j) - f(x^{j+1})\right) \\ & & \qquad + \frac{1}{2}(x^j - x^{j+1})^\top (Q_j + \tau_{k_j} I)(x^j - x^{j+1}). \end{array}$$

Now fix $h' \in \mathbb{R}^n$. Plugging $h = x^j - x^{j+1} + h'$ in the above estimate gives

$$\tfrac{1}{2}\|x^j - x^{j+1}\|^2_{Q_j + \tau_{k_j} I} + g_j^\top h' \leq \phi(x^j + h', x^j) - f(x^j) + \gamma^{-1}\left(f(x^j) - f(x^{j+1})\right).$$

Passing to the limit $j \in \mathcal{N}$ and using, in the order named, $\|x^j - x^{j+1}\|^2_{Q_j + \tau_{k_j} I} \to 0$, $g_j \to 0$, $x^j \to \bar{x}$, $f(x^j) \to f(\bar{x}) = \phi(\bar{x}, \bar{x})$ and $f(x^j) - f(x^{j+1}) \to 0$, we obtain:

$$0 \leq \phi(\bar{x} + h', \bar{x}) - \phi(\bar{x}, \bar{x}). \tag{5.4}$$

Here the rightmost term $f(x^j) - f(x^{j+1}) \to 0$ converges by monotonicity, while convergence of the leftmost term was shown in part ii). Notice that estimate (5.4) is the only place in the whole demonstration where axiom (M_3) is used, because we need $\limsup_{j \in \mathcal{N}} \phi(x^j + h', x^j) \leq \phi(\bar{x} + h', \bar{x})$.

Now the test vector h' in (5.4) is arbitrary, which shows $0 \in \partial_1 \phi(\bar{x}, \bar{x})$. By axiom (M_1) we have $0 \in \partial f(\bar{x})$. This completes the proof. □

Remark 5.2. The now standard way to prove convergence of bundle methods in the convex case is the anchor technique of Kiwiel [16], see also [27]. The complicated argument above becomes necessary due to the absence of convexity.

6 Convergence Without Strongness

In this section we show that the strongness assumption $(\widetilde{M}_2)$ can be substantially weakened if the algorithm is slightly modified. What is needed is an additional safeguard rule (S) against exceedingly large τ-parameters. To begin, we need the following

Definition 6.1. The first-order model ϕ of f is called strict if (M_1), (M_3) and the following strict version of axiom (M_2) are satisfied:

$(\widehat{M_2})$ For every bounded set B and every $\epsilon > 0$ there exists $\delta > 0$ such that for all $x, y \in B$, $\|x - y\| \leq \delta$ implies $f(y) - \phi(y, x) \leq \epsilon\|y - x\|$.

Remark 6.2. Condition $(\widehat{M_2})$ could also be written more suggestively as $f(y) - \phi(y, x) \leq \mathrm{o}(\|y - x\|)$, as $y - x \to 0$ uniformly on bounded sets. Yet another practical way to write $(\widehat{M_2})$ is as follows: For every x and sequences $x_j \to x$ and $y_j \to x$ there exists a sequence $\epsilon_j \to 0^+$ such that $f(y_j) \leq \phi(y_j, x_j) + \epsilon_j\|y_j - x_j\|$.

Remark 6.3. It is clear that every strong model is strict, and that every strict model is a model. None of these implications is reversible.

Remark 6.4. Let $f \in C^1$, then the standard model is $\phi^\sharp(y, x) = f(x) + \nabla f(x)^\top(y - x)$. Notice that $\phi^\sharp$ is strict because f is strictly differentiable. Since $\phi^\sharp$ need not be strong for $f \in C^1 \setminus C^{1,1}$, it follows that strict models need not be strong.

Remark 6.5. We show by way of an example that a model need not be strict. We take $f(x) = x^2 \sin(1/x)$ on the real line, where $f(0) = 0$. Notice that f is differentiable everywhere, but $\partial f(0) = [-1, 1]$, even though $f'(0) = 0$. We show that the standard model of f is not strict.

Observe that at $x \neq 0$ we have

$$\phi^\sharp(y, x) = x^2 \sin(1/x) + (2x \sin(1/x) - \cos(1/x))\,(y - x).$$

Therefore writing $f(y) = \phi^\sharp(y, \bar{x}) + \epsilon(y, x)(y - x)$ we find that the o-term is

$$\epsilon(y, x) = \frac{y^2 \sin(1/y) - x^2 \sin(1/x)}{y - x} - 2x \sin(1/x) + \cos(1/x).$$

If the standard model was to be strict on a bounded interval B containing 0, we would have to have $\epsilon(y, x) \to 0$ as $x, y \to 0$. But this is not the case. Take $x_k = 1/k\pi \to 0$ as $k \to \infty$ and $y_\ell = (\ell\pi + \pi/2)^{-1} \to 0$ as $\ell \to \infty$. Then $\sin(1/y_\ell) = \pm 1$ and $\sin(1/x_k) = 0$, while $\cos(1/x_k) = \pm 1$. Then $\epsilon(y_\ell, x_k) = \big(\pm y_\ell^2/(y_\ell - x_k)\big) \pm 1 = \pm(1/\pi)\frac{k+\pi(k\ell-\ell^2-\ell+\frac{k}{2}-\frac{1}{4})}{k\ell+\frac{k}{2}-\ell^2-\ell-\frac{1}{4}}$. Letting $k = \ell \to \infty$ gives $\epsilon(y_k, x_k) \to \mp(2 - \pi)/\pi \neq 0$.

As we shall see, axiom $(\widehat{M_2})$ allows a finer analysis of the mechanism in the proof of Theorem 5.1. In order to study this, let us introduce the following terminology. A subsequence x^j, $j \in \mathcal{N}$ of the sequence of serious iterates is called of type I if τ_j is bounded. It is called of type II if τ_j is not bounded. We say that $j \in \mathcal{N}$ is of type IIa if $\tau_j \to \infty$ as $j \in \mathcal{N}$, but for every $j \in \mathcal{N}$ the τ-parameter was not increased during the jth run of the inner loop. The sequence is called of type IIb if $\tau_j \to \infty$, $j \in \mathcal{N}$, and there exists an infinite subset $\mathcal{N}'$ of $\mathcal{N}$ such that for every $j' \in \mathcal{N}'$, the τ-parameter was increased at least once during the inner loop j'.

With these preparations we are ready for

Theorem 6.6. *Let f have a strict model ϕ and suppose the level set $\{x \in \mathbb{R}^n : f(x) \leq f(x^1)\}$ is bounded. Suppose $\bar{x}$ is the accumulation point of a sub-sequence of type I or of type IIb of the sequence x^j of serious iterates. Then $0 \in \partial f(\bar{x})$.*

Proof. The case of a sequence of type I where τ_{k_j} is bounded poses no problem, so suppose the sequence $j \in J$ is of type IIb and has $\tau_{k_j} \to \infty$, and $x^j \to \bar{x}$. We have to show that $\bar{x}$ is critical. Since the sequence is of type IIb, we are now in the situation of part iii) of the proof of Theorem 5.1, but have to manage without estimate (2.1), because we do not assume that our model is strong. We consider $\rho_{k_j-\nu_j} < \gamma$ and $\widetilde{\rho}_{k_j-\nu_j} \geq \widetilde{\gamma}$ as in that proof. In particular, the null step $y^{k_j-\nu_j+1}$ will again play a decisive role.

Notice first that as $\tau_{k_j} \to \infty$ and $\tau_{k_j} = 2\tau_{k_j-\nu_j}$, boundedness of the subgradients $\widetilde{g}_j := (Q_j + \frac{1}{2}\tau_{k_j} I)(x^j - y^{k_j-\nu_j+1}) \in \partial_1 \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j)$ shows $y^{k_j-\nu_j+1} \to \bar{x}$. Suppose there exists a subsequence J' of J such that $\|\widetilde{g}_{j'}\| \to 0$, $j' \in J'$. Then for a test vector h and $j \in J'$:

$$\begin{aligned} \widetilde{g}_j^\top h &\leq \phi_{k_j-\nu_j}(y^{k_j+\nu_j+1} + h, x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j) \\ &\leq \phi(y^{k_j+\nu_j+1} + h, x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j). \end{aligned} \tag{6.1}$$

Now we use the fact that $\widetilde{\rho}_{k_j-\nu_j} \geq \widetilde{\gamma}$, then

$$f(x^j) - \Phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j) \leq \widetilde{\gamma}^{-1}\left(f(x^j) - \Phi(y^{k_j-\nu_j+1}, x^j)\right).$$

Adding $\frac{1}{2}(y^{k_j-\nu_j+1} - x^j)^\top Q_j (y^{k_j-\nu_j+1} - x^j)$ on both sides gives

$$\begin{aligned} f(x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j) \leq& \widetilde{\gamma}^{-1}\left(f(x^j) - \Phi(y^{k_j-\nu_j+1}, x^j)\right) \\ &+ \tfrac{1}{2}(y^{k_j-\nu_j+1} - x^j)^\top Q_j (y^{k_j-\nu_j+1} - x^j). \end{aligned}$$

Combining this and estimate (6.1) gives

$$\begin{aligned} \widetilde{g}_j^\top h \leq \phi(y^{k_j-\nu_j+1} + \bar{h}, x^j) - f(x^j) + \widetilde{\gamma}^{-1}\left(f(x^j) - \Phi(y^{k_j-\nu_j+1}, x^j)\right) \\ + \tfrac{1}{2}(y^{k_j-\nu_j+1} - x^j)^\top Q_j (y^{k_j-\nu_j+1} - x^j). \end{aligned}$$

As we have seen $y^{k_j-\nu_j+1} - x^j \to 0$, hence the rightmost term converges to 0 by boundedness of Q_j. Moreover, we claim that $\lim f(x^j) - \Phi(y^{k_j-\nu_j+1}, x^j) = 0$, so the term $\widetilde{\gamma}^{-1}(\dots)$ on the right hand side converges to 0. Indeed since $y^{k_j-\nu_j+1} - x^j \to 0$ and $x^j \to \bar{x}$, axiom (M_3) gives $\limsup \phi(y^{k_j-\nu_j+1}, x^j) \leq \phi(\bar{x}, \bar{x}) = f(\bar{x})$. Since ϕ is strict, axiom $(\widehat{M_2})$ gives $\epsilon_j \to 0$ such that

$$f(y^{k_j-\nu_j+1}) - \phi(y^{k_j-\nu_j+1}, x^j) \leq \epsilon_j \|y^{k_j-\nu_j+1} - x^j\|. \tag{6.2}$$

Passing to the limit in (6.2) then gives $\liminf \phi(y^{k_j-\nu_j+1}, x^j) \geq f(\bar{x})$, so the two estimates together show $f(x^j) - \phi(y^{k_j-\nu_j+1}, x^j) \to 0$. Since the quadratic term converges to 0, we deduce $f(x^j) - \Phi(y^{k_j-\nu_j+1}, x^j) \to 0$. Going back with this information to the above subgradient inequality and passing to the limit shows

$$0 \leq \phi(\bar{x} + h, \bar{x}) - f(\bar{x}) = \phi(\bar{x} + h, \bar{x}) - \phi(\bar{x}, \bar{x}),$$

where we use axiom (M_3) for the first term on the right hand side. This proves $0 \in \partial_1 \phi(\bar{x}, \bar{x})$, because h was arbitrary. As a consequence of this argument, we now see that the case of a subsequence $j \in J'$ where $\widetilde{g}_j \to 0$ was in fact the easier one to deal with. To settle the difficult case, let us now assume that $\|\widetilde{g}_j\| \geq \eta$ for some $\eta > 0$ and all $j \in J$.

We argue that with $\|\widetilde{g}_j\| \geq \eta$ there exists $\theta > 0$ such that

$$f(x^j) - \Phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j) \geq \theta \|y^{k_j-\nu_j+1} - x^j\| \tag{6.3}$$

for all $j \in J$ sufficiently large. Indeed, by the subgradient inequality we have

$$\begin{aligned} \widetilde{g}_j^\top (x^j - y^{k_j-\nu_j+1}) &\leq \phi_{k_j-\nu_j}(x^j, x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j) \\ &= f(x^j) - \phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j). \end{aligned}$$

Subtracting $\frac{1}{2}(y^{k_j-\nu_j+1} - x^j)^\top Q_j (y^{k_j-\nu_j+1} - x^j)$ from both sides gives

$$\tfrac{1}{2}(y^{k_j-\nu_j+1} - x^j)^\top (Q_j + \tau_{k_j} I)(y^{k_j-\nu_j+1} - x^j) \leq f(x^j) - \Phi_{k_j-\nu_j}(y^{k_j-\nu_j+1}, x^j).$$

Now as $\tau_{k_j} \to \infty$, we have $\frac{1}{4}\|\widetilde{g}_j\|\|y^{k_j-\nu_j+1} - x^j\| \leq \frac{1}{2}(y^{k_j-\nu_j+1} - x^j)^\top (Q_j + \tau_{k_j} I)(y^{k_j-\nu_j+1} - x^j)$ for j large enough, which proves formula (6.3) with $\theta = \frac{1}{4}\eta$.

Next using (6.2) and subtracting the usual $\frac{1}{2}(y^{k_j-\nu_j+1} - x^j)^\top Q_j (y^{k_j-\nu_j+1} - x^j)$ from both sides gives

$$f(x^j) - \Phi(y^{k_j-\nu_j+1}, x^j) \leq \widetilde{\epsilon}_j \|y^{k_j-\nu_j+1} - x^j\|, \tag{6.4}$$

where $\widetilde{\epsilon}_j := \epsilon_j + \frac{1}{2}\|Q_j\|\|y^{k_j-\nu_j+1} - x^j\| \to 0$. Combining (6.3) and (6.4) gives the estimate

$$\widetilde{\rho}_{k_j-\nu_j} \leq \rho_{k_j-\nu_j} + \frac{\widetilde{\epsilon}_j \|y^{k_j-\nu_j+1} - x^j\|}{\theta \|y^{k_j-\nu_j+1} - x^j\|}$$

which as in the proof of Theorem 5.1 shows $\limsup \widetilde{\rho}_{k_j-\nu_j} \leq \limsup \rho_{k_j-\nu_j} \leq \gamma$, contradicting $\widetilde{\rho}_{k_j-\nu_j} \geq \widetilde{\gamma} > \gamma$ for the infinitely many $j \in J'$. That completes the proof. □

The conclusion of Theorem 6.6 is weaker than that of Theorem 5.1, because the sequence of x^j which admits subsequence convergence cannot be identified algorithmically. This can be remedied by adding a safeguard rule to the algorithm.

Corollary 6.7. *Suppose f has a strict model ϕ and let the level set $\{x \in \mathbb{R}^n : f(x) \leq f(x^1)\}$ be bounded. Suppose further that we include the following safeguard rule (S) against large proximity control parameters in step 8 of algorithm 1:*

(S) If $\tau_j^\sharp > T$, then set $\tau_j^\sharp = T$,

where $T > q$ is some large constant. Then every accumulation point $\bar{x}$ of the sequence x^j of serious iterates is critical.

Proof. By Theorem 6.6 it suffices to show that no sequences of type IIa occur. This is indeed ensured by adding the safeguard rule (S) to the algorithm. Suppose contrary to what is claimed that x^j $(j \in J)$ is a sequence of type IIa. Let $j \in J$. As soon as the jth inner loop begins, the τ parameter is set to $\tau_j^\sharp$, the memory element of the previous sweep. But the sequence is of type IIa, so the τ-parameter is never increased during the jth inner loop. Since the jth inner loop ends with τ_{k_j}, we have $\tau_j^\sharp = \tau_{k_j}$. Since $\tau_{k_j} \to \infty$, $j \in J$, we deduce $\tau_j^\sharp \to \infty$, $j \in J$, which contradicts rule (S). □

Remark 6.8. Notice that $T > q$ is necessary so that (S) is not in conflict with the rule $Q_j + \tau_j^\sharp I \succ 0$ in step 8.

Remark 6.9. The safeguard rule (S) should not be confused with a restart mechanism, where τ is re-set as soon as it gets too large numerically. This may be seen from the fact that (S) does *not* prevent the τ-parameter to go to infinity in the outer loop. This may still happen if there is a sub-sequence of type IIb. On the other hand, from a practical point of view, rule (S) is certainly hard to distinguish from such a restart procedure.

$\boxed{7}$ Applications

In order to illustrate our method, which is fairly abstract, let us discuss some applications.

Example 7.1. Steepest descent. Let $f \in C^1$, then as we know $\phi^\sharp(y,x) = f(x) + \nabla f(x)^\top (y-x)$. Put $Q(x) = 0$. We let $\phi_k = \phi$ for every k, then the only parameter which changes in the inner loop is τ_k, and the inner loop becomes in fact a linesearch. Namely, the tangent program is

$$\min_{y\in\mathbb{R}^n} f(x) + \nabla f(x)^\top (y-x) + \tfrac{\tau_k}{2}\|y-x\|^2,$$

so the trial step is $y^{k+1} = x - \tau_k^{-1}\nabla f(x)$, which we accept as soon as

$$\begin{array}{rcl} f(x) - f(y^{k+1}) & \geq & \gamma\left(f(x) - [f(x) + \nabla f(x)^\top (y^{k+1}-x)]\right) \\ & = & -\gamma \nabla f(x)^\top (y^{k+1}-x) = \gamma\tau_k^{-1}\nabla f(x)^\top \nabla f(x). \end{array}$$

This is just steepest descent with steps of size τ_k^{-1}, where the step has to satisfy the Armijo condition with $0 < \gamma < 1$. The doubling rule in step 6 of the algorithm becomes the backtracking strategy where steplength is halved.

It follows from Theorem 5.1 that the steepest descent method with memorized stepsize converges if $f \in C^{1,1}$. Here memorized means that we start the $(j+1)$th linesearch with exactly that steplength which was accepted at iteration j, or with the doubled steplength if $\rho > \Gamma$.

On the other hand, when only $f \in C^1$, then in order to converge the steepest descent method needs the safeguard rule (S). In terms of the linesearch this means the initial steplength at $j+1$ has to be reset to T^{-1} if the step accepted at j was smaller than T^{-1}. When $f \in C^1$ the steepest descent method converges if the linesearch is memoryless and starts each time at $t = 1$.

Example 7.2. Newton's method. Now let $f \in C^2$. Let x be a serious iterate. We choose $Q(x) = \nabla^2 f(x)$. Taking again $\phi_k = \phi$, the tangent program is

$$\min_{y\in\mathbb{R}^n} f(x) + \nabla f(x)^\top (y-x) + \tfrac{1}{2}(y-x)^\top (\nabla^2 f(x) + \tau_k I)(y-x)$$

so we have

$$y^{k+1} = x - (\nabla^2 f(x) + \tau_k I)^{-1}\nabla f(x).$$

This is a damped Newton step. For large τ_k we are essentially doing a steepest descent step, and multiplying τ_k by 2 corresponds to backtracking. On the other hand, small τ_k brings us close to doing the Newton step.

However we now see a weakness of the dual approach since τ_k, the Lagrange multiplier of the trust region program, is non-zero. But in the case of quadratic convergence of Newton's method, we expect the trust region constraint to be inactive in the end, a case excluded by choosing $\tau_k > 0$.

For all that, can this method still converge superlinearly? The answer is yes. Assume we are in the neighbourhood of a minimum where the sufficient second order optimality condition is satisfied: $Q_j = \nabla^2 f(x^j) \succeq \epsilon I \succ 0$ for all j. According to the update rule in step 8 of the algorithm, the memory parameter $\tau_{j+1}^\sharp$ need not be adjusted to force $Q_{j+1} + \tau_{j+1}^\sharp I \succ 0$. That means, the halving rule is applied as soon as $\rho_k \geq \Gamma$. But we know that ρ_{k_j} will be close to 1 in the convergence zone of Newton's method, so we get $\rho_{k_j} \geq \Gamma$ in the neighbourhood of the qualified minimum. From some counter j_0 onwards, we therefore have $\tau_j = 2^{-(j-j_0)}\tau_{j_0}$. The remaining question is whether the method $x^{j+1} = x^j - \left(\nabla^2 f(x^j) + 2^{-j}\tau_0 I\right)^{-1} \nabla f(x^j)$ converges superlinearly. This is indeed the case by the Broyden-Moré theorem [11, Thm. 8.2.4], so our method converges locally superlinearly if f is of class C^2.

Example 7.3. Piecewise quadratic model. In some applications it may be interesting to use a piecewise quadratic model $\Phi(y,x)$. This is for instance the case if f itself is piecewise of class C^2, when the domains of smoothness are explicitly known, and gradient and Hessians are available. Lukšan and Vlček [23] use such an approach, but in contrast with our method they use a line search scheme. Approximations of the Hessian are computed by a finite difference process. Their convergence proof uses a hypothesis, which in our case amounts to boundedness of $Q_j + \tau_k I$, and which would be artificial.

8 Practical Aspects: Solving the Tangent Program

A crucial point which we have not touched upon yet is whether our approach is practical. For this to be true we have to assure that solving the tangent program (3.1) is considerably easier than solving the original problem. Naturally, this point can only be clarified by looking at important classes of examples.

8.1 Polyhedral models

Let us start by looking at polyhedral working models, the most common case. In fact, all traditional bundle approaches are based on polyhedral models, because if nothing specific is known about f or ϕ, all we can do is generate finitely many cutting planes and aggregate planes. At a fixed instant k of the inner loop, the tangent program has then the form

$$\min_{y\in\mathbb{R}^n} \max_{i=1,\dots,p} a_i + g_i^\top(y-x) + \tfrac{1}{2}(y-x)^\top(Q+\tau I)(y-x).$$

This can easily be transformed into a quadratic program, which is even convex because $Q+\tau I \succ 0$. It takes the form

$$\begin{array}{ll} \text{minimize} & t + \frac{1}{2}(y-x)^\top(Q+\tau I)(y-x) \\ \text{subject to} & a_i + g_i^\top(y-x) \le t,\ i=1,\dots,p \end{array}$$

with decision variable $(t,y)\in\mathbb{R}^{n+1}$. This may be solved efficiently with standard software tools for convex quadratic programming.

If the solution y^{k+1} of the tangent QP is a null step, the current working model is improved using cutting planes and aggregation. For a polyhedral model the aggregate plane is of the form

$$m_{k+1}^*(y) = a_{k+1}^\star + {g_{k+1}^\star}^\top(y-x) = \sum_{i\in I}\lambda_i\left[a_i + g_i^\top(y-x)\right],$$

where $I = \{i\in\{1,\dots,p\} : a_i + g_i^\top(y^{k+1}-x) = \max_{i=1,\dots,p} a_i + g_i^\top(y^{k+1}-x)\}$ is the set of active indices, and $\lambda_i \ge 0$, $\sum_{i\in I}\lambda_i = 1$.

8.2 Eigenvalue optimization

Let us next examine eigenvalue optimization, a case of practical interest. Here $f = \lambda_1 \circ F$. Assuming that the strong model $\phi(y,x) = \lambda_1\left(F(x) + F'(x)(y-x)\right)$ is used, the working model may include non-polyhedral parts.

Since f and ϕ are both composite functions of the maximum eigenvalue function, subgradients of f and $\phi(\cdot, x)$ are obtained by a chain rule

$$\partial f(x) = F'(x)^\star \partial\lambda_1 \left(F(x)\right),$$

$$\partial_1\phi(y,x) = F'(x)^\star \partial\lambda_1 \left(F(x) + F'(x)(y-x)\right).$$

Now as is well-known, for $X \in \mathbb{S}^m$,

$$\partial\lambda_1(X) = \{QYQ^\top \in \mathbb{S}^m : Y \succeq 0, \mathrm{Tr}(Y) = 1, Y \in \mathbb{S}^r\},$$

where r is the eigenvalue multiplicity of $\lambda_1(X)$, and where the r columns of the orthogonal matrix Q form a basis of the associated eigenspace. Therefore exactness and cutting planes lead to pairs (a, g), where the subgradients are of the form

$$g = F'(x)^\star G,\ G = QYQ^\top, Y \succeq 0, \mathrm{Tr}(Y) = 1$$

with

$$G \bullet [F(x) + F'(x)(y-x)] = \lambda_1(F(x) + F'(x)(y-x)),$$

and where

$$a = G \bullet F(x) = Y \bullet [Q^\top F(x) Q].$$

Assume we have a family $(a_i(Y_i), g_i(Y_i))$, $i = 1, \dots, p$ of such pairs, stemming from certain y^i, indexed by the finitely many y^i and the infinite set of constraints $Y_i \succeq 0$, $\mathrm{Tr}(Y_i) = 1$. Assume that the aggregate plane is (a_0, g_0), which we add to the list. Let $\mathcal{G}$ be the set of these pairs, then the tangent program is

$$\min_{y\in\mathbb{R}^n} \max_{(a,g)\in\mathrm{co}(\mathcal{G})} a + g^\top(y-x) + \tfrac{1}{2}(y-x)^\top(Q+\tau I)(y-x).$$

Using Fenchel duality, we may switch the min and the max operator. The then inner minimum may be computed explicitly and yields $y - x = -(Q+\tau I)^{-1}g$. Substituting this back gives the dual form of the tangent program

$$\max\{a - \tfrac{1}{2}g^\top(Q+\tau I)^{-1}g : (a,g) \in \mathrm{co}(\mathcal{G})\}.$$

Going back to the way the elements of $\mathcal{G}$ are constructed, we have to solve the program

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{i=0}^{p} \mu_i \left(a_i(Y_i) - \tfrac{1}{2}g_i(Y_i)^\top(Q+\tau I)^{-1}g_i(Y_i)\right) \\ \text{subject to} & \mu_i \geq 0,\ \displaystyle\sum_{i=0}^{p}\mu_i = 1 \\ & Y_i \succeq 0,\ \mathrm{Tr}(Y_i) = 1 \end{array}$$

Introducing the variable s, we can replace $g^\top(Q+\tau I)^{-1}g \leq s$ by the LMI $\begin{bmatrix} Q+\tau I & g \\ g^\top & s \end{bmatrix} \succeq 0$. Introducing $Z_i = \mu_i Y_i$, we finally get the semidefinite program

$$\begin{array}{ll} \text{maximize} & \displaystyle\sum_{i=0}^{p} a_i(Z_i) - s_i \\ \text{subject to} & \begin{bmatrix} Q+\tau I & g_i(Z_i) \\ g_i(Z_i)^\top & s_i \end{bmatrix} \succeq 0, Z_i \succeq 0, i = 0, \dots, p \\ & a_i(Z_i) = Z_i \bullet [Q_i^\top F(x) Q_i], g_i(Z_i) = F'(x)^\star[Q_i Z_i Q_i^\top], \\ & \displaystyle\sum_{i=0}^{p} \mathrm{Tr}(Z_i) = 1 \end{array}$$

with decision variable (s, Z). The trial step y^{k+1} is then obtained by the return formula $y^{k+1} = x - (Q + \tau I)^{-1} \sum_{i=0}^{p} g_i(Z_i)$ where $(s_0, \dots, s_p, Z_0, \dots, Z_p)$ is the dual optimal solution and $\sum_{i=0}^{p} g_i(Z_i) = g^\star_{k+1}$ the aggregate subgradient. Notice that these tangent SDPs are usually of small size, and could be solved by existing SDP solvers.

$\boxed{9}$ Practical Aspects: Recycling Cutting Planes

When a new inner loop starts in step 3 of the algorithm, a new working model $\phi_1(\cdot, x^{j+1})$ is formed at the new serious iterate x^{j+1}. In the convex case this model does not start from scratch, because one can recycle some of the affine support planes (cutting planes or aggregate planes) from x^j, so that $\phi_1(\cdot, x^{j+1})$ starts close to where $\phi_{k_j}(\cdot, x^j)$ ended. This happens naturally because these planes are affine support functions of f and remain meaningful as we go from x^j to x^{j+1}.

This changes significantly when f is nonconvex. It is then a priori impossible to use planes $m_j(\cdot)$ from x^j at the new x^{j+1}, because it is not even clear whether $m_j(x^{j+1}) \leq f(x^{j+1})$. This puts some doubt on our strategy to memorize the τ-parameter via $\tau_j^\sharp$, because the latter presumes some sort of continuity in the working model. Fortunately, in many cases recycling of planes between serious steps $x^j \to x^{j+1}$ is still possible.

Consider for example $f = h \circ F$, with h convex and F of class C^1. Suppose $m_j(y) = a_j + g_j^\top (y - x^j)$ is one of the planes used at x^j. That means $a_j \leq \phi(x^j, x^j) = f(x^j)$. Can we recycle m_j at x^{j+1} so that it contributes reasonably to the working model at x^{j+1}? According to the chain rule we know that $g_j = F'(x^j)^* \widetilde{g}_j$ for some subgradient $\widetilde{g}_j \in \partial h\left(F(x^j)\right)$. Since h is convex, $\widetilde{g}_j$ is still useful even though we pass from x^j to x^{j+1}. We therefore put $g_{j+1} := F'(x^{j+1})^* \widetilde{g}_j$, and we build the plane $m_{j+1}(y) = a_{j+1} + g_{j+1}^\top (y - x^{j+1})$, where $a_{j+1} = f(x^{j+1})$. The procedure does not interfere with our convergence analysis, so it is up to the user to do this in a sophisticated way if the particular structure of the application can be exploited.

A general way to recycle planes which applies without any specific structure of f is downshifting. Suppose $m_j(y) = a + g^\top (y - x^j)$ is a plane used at x^j. If the new serious iterate x^{j+1} arrives, then compute the shift $s = [f(x^{j+1}) - m_j(x^{j+1})]_+ + c\|x^j - x^{j+1}\|^2$, where $c > 0$ is some small fixed constant. Now recycle the plane under the new guise $m_{j+1}(y) = m_j(y) - s$. See e.g. [29,33] for convergence results with downshifted planes.

$\boxed{10}$ Numerical Tests

We now test our method on several examples in eigenvalue optimization and automatic control.

Example 10.1. Max-quad-function. Let us start with an academic example, the minimization of the maximum of a finite number of quadratic functions:

$$\min_{x \in \mathbb{R}^n} f(x), \quad f(x) := \max\{q_i(x) : i = 1, \dots, r\} \text{ with } q_i(x) = \tfrac{1}{2} x^\top A_i x + b_i^\top x,$$

where $A_i \in \mathbb{S}^n$ and $b_i \in \mathbb{R}^n$, $i = 1, \dots, r$. Our goal in this example is to get some hints on how to build the second order part of our model, i.e., how to choose the symmetric matrix $Q(x)$. To do this we assume $\phi = \phi_k$.

We introduce a polyhedral convex model ϕ of f at a given point $x \in \mathbb{R}^n$ defined by:

$$\phi(y,x) = \max\left\{q_i(x) + \nabla q_i(x)^\top (y-x) : i = 1,\dots,r\right\}. \tag{10.1}$$

We can easily check that ϕ is a strong first order model in the sense of definition 2.5 and that $\partial_1\phi(x,x) = \partial f(x)$. We replace program $\min\limits_{x\in\mathbb{R}^n} f(x)$ by the constrained program

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & q_i(x) \leq t,\ i = 1,\dots,r \end{array} \tag{10.2}$$

so that it makes perfect sense to apply SQP. The tangent QP for (10.2) in the SQP formalism is

$$\begin{array}{ll} \text{minimize} & \delta t + \frac{1}{2}\delta x^\top \nabla^2_{xx} L(x,t,\lambda)\delta x \\ \text{subject to} & q_i(x) + \nabla q_i(x)^\top \delta x \leq t + \delta t,\ i = 1,\dots,r. \end{array} \tag{10.3}$$

Here the matrix $\nabla^2_{xx}L(x,t,\lambda)$ is the (x,x)-block of the Hessian of the Lagrangian $L(x,t;\lambda) = t + \sum_{i=1}^r \lambda_i\left(q_i(x) - t\right)$ of (10.2). In other words: $\nabla^2_{xx}L(x,t;\lambda) = \sum\limits_{i=1}^r \lambda_i \nabla^2 q_i(x) = \sum\limits_{i=1}^r \lambda_i A_i$.

Let us see whether we can get back from (10.3) to the tangent program (3.1). Writing the constraints in (10.3) as $\max_{i=1,\dots,r} q_i(x) + \nabla q_i(x)^\top \delta x \leq t + \delta t$, we can eliminate δt from the tangent program and write it as

$$\min_{\delta x\in\mathbb{R}^n} \max_{i=1,\dots,r} \left(q_i(x) + \nabla q_i(x)^\top \delta x\right) - t + \tfrac{1}{2}\delta x^\top \nabla^2_{xx} L(x,t,\lambda)\delta x.$$

Since $t = \max_{i=1,\dots,r} q_i(x)$ is constant, this is equivalent to (3.1) (with $\phi = \phi_k$) if we use the strong model (10.1), and if $Q(x) = \nabla^2_{xx}L(x,t,\lambda)$. The latter does not depend on t, but on λ, so that Lagrange multiplier estimates λ are required. Can λ be related to the nonsmooth context? Yes it can, namely, if $y^{k+1} = x^+$ is a serious step, then the last aggregate subgradient is $g^*_{k+1} = \left(Q(x) + \tau_k I\right)(x - x^+)$, so $g^*_{k+1} = \sum_{i=1}^r \lambda_i^+ \nabla q_i(x)$, where λ_i^+ are a convex combination involving only indices i where the maximum $\phi(x^+,x)$ is attained. In other words, the coefficients $\dots,\lambda_i^-,\lambda_i,\lambda_i^+,\dots$ are in principle available if the latest aggregate subgradients before each serious step are stored.

Our goal is to get some information as to what to do in the case of a genuinely nonsmooth function, where we cannot use $\nabla^2_{xx}L = Q(x)$ directly. What we propose to do is let $Q(x)$ be an approximation for $\nabla^2_{xx}L$, maintained e.g. by an SR1 update. For this we have to find an appropriate secant equation for $\nabla^2_{xx}L$.

Taylor expansion at x^+ gives $\nabla_x L(x,t;\lambda^+) \approx \nabla_x L(x^+,t;\lambda^+) + \nabla^2_{xx}L(x^+,t;\lambda^+)(x - x^+)$, so that we get the secant equation

$$Q(x)\delta x = \sum_{i=1}^r \lambda_i^+ \left[\nabla q_i(x^+) - \nabla q_i(x)\right].$$

Here the term $\sum_{i=1}^r \lambda_i^+ \nabla q_i(x)$ is the aggregate subgradient in our proximity control algorithm, which is available. The question is how to get hold of $\sum_{i=1}^r \lambda_i^+ \nabla q_i(x^+)$, which is not an aggregate subgradient. This may in principle be done in the following way.

Let y^{k+1} and $\lambda^{(k+1)}$ respectively be the solution and the corresponding Lagrange multiplier given by the tangent program after k iterations of the inner loop. In other words, trial steps $y^1, y^2, \dots, y^k$ have been rejected (null steps), while $y^{k+1} = x^+$ is the serious step. Then the aggregate subgradient approximating $\sum_{i=1}^r \lambda_i^+ \nabla q_i(x)$ is given by:

$$g^*_{k+1} = (Q(x) + \tau_k I)(x - x^+) \in \partial_1 \phi_k(x^+, x). \tag{10.4}$$

Now introducing the multiplier $\lambda^{(k+1)}$, it can also be expressed as $g^*_{k+1} = \sum_i \lambda_i^{(k+1)} g_i$, where the subgradients g_i may have three different origins: the exactness rule, the cutting plane and aggregation. We denote E and CP the (finite) sets of indices of subgradients chosen during the exactness and the cutting planes stages. Then

$$g^*_{k+1} = \sum_{i \in E} \lambda_i^{(k+1)} \nabla q_i(x) + \lambda_{icp}^{(k+1)} \nabla q_{icp}(x) + \lambda_{iAgg}^{(k+1)} g^\star_k,$$

where $g^\star_k = (Q(x) + \tau_{k-1} I)(x - y^k)$ is the second to last aggregate subgradient computed in the inner loop. Our idea to estimate $\sum_i \lambda_i^+ \nabla q_i(x^+)$ is to re-evaluate this expression at x^+. The main difficulty is clearly how to re-evaluate the aggregate subgradient $g^\star_k$ at x^+. This could be done recursively in the inner loop counter k by proving that g^*_{k+1} is of the form:

$$g^*_{k+1} = \sum_{i \in E} \widetilde{\lambda}_i^{(k+1)} \nabla q_i(x) + \sum_{i \in CP} \widetilde{\lambda}_i^{(k+1)} \nabla q_i(x)$$

where the new multiplier $\widetilde{\lambda}_i^{(k+1)}$ is given by:

$$\widetilde{\lambda}^{(1)} = \lambda^{(1)}, \quad \widetilde{\lambda}^{(k+1)} = \left[\begin{array}{c|c|c} I_{n_{exact}} & 0 & \\ \cline{1-2} 0 & 0 & \widetilde{\lambda}^{(k)} \\ \hline 0 & 1 & 0 \end{array} \right] \lambda^{(k+1)}. \tag{10.5}$$

Consequently the term $\sum_i \lambda_i^+ \nabla q_i(x^+)$ could now be estimated by:

$$g^+ = \sum_{i \in E} \widetilde{\lambda}_i^{(k+1)} \nabla q_i(x^+) + \sum_{i \in CP} \widetilde{\lambda}_i^{(k+1)} \nabla q_i(x^+). \tag{10.6}$$

Altogether the information stored at each iteration of the inner loop is the set of indices selected both by the exactness and the cutting plane processes and the modified Lagrange multiplier (see relation (10.5)). Then the SR1 update is performed the following way:

$$\begin{array}{rcl} s &=& x^+ - x \\ y &=& G^+ \widetilde{\lambda}^+ - (Q(x) + \tau_k I)(x - x^+) \\ && \text{where the columns of } G^+ \text{ are given by the subgradients computed by} \\ && \text{the exactness rule and those defining the successive cutting planes} \\ Q(x^+) &=& Q(x) + \dfrac{(y - Q(x)s)(y - Q(x)s)^\top}{(y - Q(x)s)^\top s} \end{array}$$

Numerical results. The following numerical stopping criteria have been used. Let $\varepsilon > 0$ be a tolerance parameter. We first check criticality $0 \in \partial f(x)$ by computing:

$$\inf\{\|h\| : h \in \partial f(x)\} < \varepsilon. \tag{10.7}$$

Two additional tests are implemented to avoid pointless computational efforts during the final phase, where iterates make minor progress. We compare the progress of the local model around the current iterate, and we evaluate the relative step length to the optimization variable gains:

$$f(x) - f(x^+) < \varepsilon(|f(x)| + 1), \text{ and } \left\|x - x^+\right\| < \varepsilon(\|x\| + 1). \tag{10.8}$$

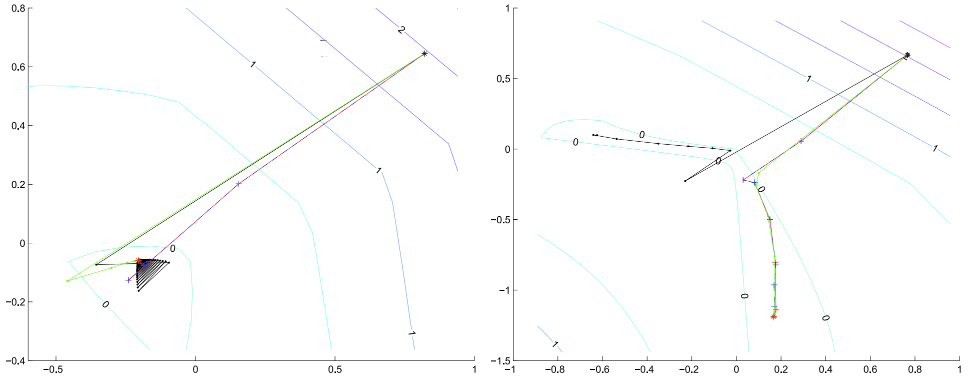
The algorithm is stopped if either (10.7) or both criteria in (10.8) are satisfied.

We performed experiments with a max-quad-function where the first order and several second order approaches are compared, including the approximation outlined above. Fig. 1 presents the results of our trust region algorithm 1 for the four different models of the objective f of the form:

$$\Phi(y,x) = \phi(y,x) + \tfrac{1}{2}(y-x)^\top Q(x)(y-x)$$

where:

- (in black dash-dotted line) $Q(x) = 0$: we use the first order model ϕ without any second order information.
- (in green dashed line) $Q(x)$ is the true Hessian of the Lagrangian of SQP, computed as the convex combination of the matrices A_i and whose coefficients λ_i^+ are given by the aggregation process: $Q(x) = \sum_{i=1}^r \lambda_i^+ A_i$.
- (in red dash-crossed line) $Q(x)$ is an approximation of the Hessian of the Lagrangian involved in a SQP approach, computed using the Symmetric Rank One (SR1) update.
- (in blue dashed line) $Q(x)$ is the approximation of the Hessian of the Lagrangian involved in a SQP approach, computed using formula (10.4), (10.5) and (10.6) to perform the SR1 update. Only this approximation would be available in a truly nonsmooth application.



	Convex ex. (left fig.)		Nonconvex ex. (right fig.)	
	$f(x^\star)$	Iter	$f(x^\star)$	Iter
TR without 2nd order term Q	-0.03527	50	-0.031478	10
TR & SR1 update	-0.03527	10	-0.104917	9
TR & SR1 approx	-0.03527	10	-0.104917	11
TR & Q=Hessian of Lagrangian	-0.03527	27	-0.104917	9

Figure 1: Performance of the proximity control algorithm for the quadratic min-max problem. We compare four different models of $Q(x)$ on two randomly generated examples in dimension 2 with $r = 10$ quadratics. The case on the left is convex, the case on the right non-convex.

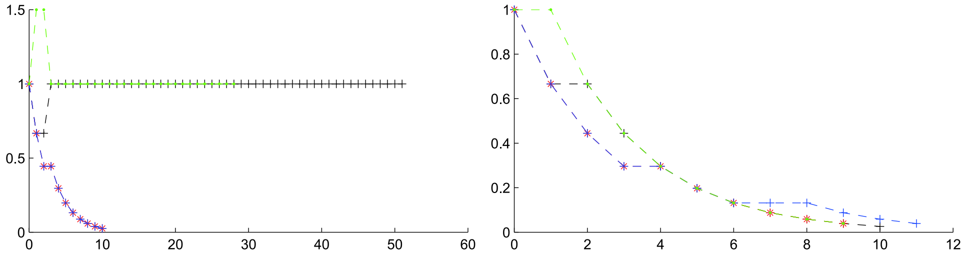


Figure 2: Behaviour of the proximity control parameter τ_k along serious iterations for the two examples presented on Fig. 1.

As expected, in the convex case (see Fig. 1 left), we observe an acceleration of the convergence with second order information Q, compared to the case where $Q = 0$. Using a second order model in the sense of definition 2.5 improves the approximation of the objective function f by the model Φ and reduces zigzagging.

Table 1 suggests that this might be particularly significant in high dimension and for nonconvex problems. In addition, in the nonconvex case, as illustrated by Fig. 1 right, we observe that, unlike the first order method, the three second order methods converge towards the global minimum of the nonconvex problem.

	$n = 2,\ m = 0$		$n = 2,\ m = 557$		$n = 10,\ m = 0$	
	$f(x^\star)$	Iter.	$f(x^\star)$	Iter.	$f(x^\star)$	Iter.
Matlab *fminimax*	-2.78939e-04	-	-8.42395e-06	-	Time out	-
TR & $Q = 0$	-2.78971e-04	185	-1.03922e-05	48	-1.29756e-02	30
TR & SR1 update	-2.78971e-04	13	-1.05374e-05	7	-1.29756e-02	17
TR & SR1 approx	-2.78971e-04	10	-1.05374e-05	7	-1.29756e-02	17
TR & Q Hessian	-2.78971e-04	20	-1.05395e-05	11	-1.29756e-02	15

Table 1: Proximity control algorithm 1 applied to n-dimensional quadratic min-max examples with $r = 1000$ quadratics and m negative semidefinite matrices A_i. The numerical data $(A_i,\ b_i,\ x_0)$ are randomly generated.

Our last remark concerns the behaviour of the proximity control parameter τ_k along serious iterations. Recall that a decreasing proximity parameter τ_k means a good agreement between the objective f and the current working model $\Phi_k(\cdot, x)$. In particular, in the convex case (Figure 2 left), introducing second order terms via a SR1 update improves approximations and reduces the number of steps computed by our algorithm.

Example 10.2. H_∞-controller synthesis was one of the motivating examples to develop our model based bundling approach. Here the objective function is of the form

$$f(x) = \max_{\omega \in [0,\infty]} \sigma_1 \left(T(x, j\omega) \right), \tag{10.9}$$

where the unknown variable $x = \operatorname{vec}(K)$ represents the gains of the feedback controller $K \in \mathbb{R}^{n_y \times n_u}$ to be designed (n_y the number of measured outputs of the system, n_u the number of controlled inputs), and $s \mapsto T(x, s)$ is a stable closed loop transfer operator, depending rationally on x, mapping into a space $\mathbb{C}^{n_z \times n_w}$ of matrices. The objective f is the H_∞ norm of the transfer operator $T(x,\cdot)$ as a function of $x = \operatorname{vec}(K)$ and as such is subject to two sources of nonsmoothness, the infinite max-operator, and the maximum singular value function σ_1, which is convex but generally nonsmooth. Non-convexity of f springs from the nonlinearity of T.

Most H_∞ synthesis techniques in the literature are based on the Kalman-Yakubovitch-Popov Lemma and amount to solving bilinear matrix inequalities (BMIs). Here numerical problems arise from the strong disparity between controller gains and Lyapunov variables. Our proximity control algorithm allows to perform H_∞ synthesis from the frequency domain point of view, i.e., by minimizing (10.9) directly. Since (10.9) depends only on controller variables K, (that is, on $x = \operatorname{vec}(K)$), we do not have to identify Lyapunov variables, which is a great advantage when large plants are considered. In exchange, a difficult semi-infinite nonsmooth and non-convex optimization program (10.9) has to be solved.

In our experiments we evaluate the proximity control algorithm on a variety of H_∞ case studies and compare with older experiments obtained with the line-search based method of [1,4].

Local model. The objective function to be minimized is $f(x) = \sup_{\omega\in[0,\infty]} \sigma_1(T(x,j\omega)) = \|T(x,\cdot)\|_\infty$. The strong first-order local model we use is:

$$\begin{aligned} \phi(y,x) \quad &:= \quad \sup_{\omega\in[0,\infty]} \sigma_1\left(T(x,j\omega) + T'(x,j\omega)(y-x)\right) \\ &= \quad \left\|T(x,\cdot) + T'(x,\cdot)(y-x)\right\|_\infty , \end{aligned} \tag{10.10}$$

where $T'(x,\cdot)$ is the derivative of $T(x,\cdot)$ with respect to x, and where $\|G(\cdot)\|_\infty$ stands for the H_∞ norm of a transfer matrix operator $G(s)$. Expressions of $T'(x,\cdot)$ in terms of the plant state space data can be found in [4]. Strongness of this model follows from remark 2.9. In these tests, no second order term $Q(x)$ was used.

Computing subgradients. The following result explains how to compute subgradients of σ_1.

Lemma 10.3. *Let $A \in \mathbb{C}^{m\times n}$ and suppose $\sigma_1(A)$ has multiplicity $r \geq 1$ (with $r \leq \min\{m,n\}$). Let $U_1 \in \mathbb{C}^{m\times r}$ and $V_1 \in \mathbb{C}^{n\times r}$ be such that*

$$U_1^H A V_1 = \sigma_1(A) I_r. \tag{10.11}$$

Then

$$\partial\sigma_1(A) = \left\{V_1 Y U_1^H \ : \ Y \in \mathscr{S}_r\right\}, \tag{10.12}$$

where $\mathscr{S}_r = \{A \in \mathbb{C}^{r\times r} \ : \ A = A^H, 0 \preceq A \preceq I_r, \operatorname{Tr}(A) = 1\}$. We have the following useful equivalent formulations

$$\partial\sigma_1(A) = \left\{\frac{A^H U_1 Y U_1^H}{\sigma_1(A)} \ : \ Y \in \mathscr{S}_r\right\} = \left\{\frac{V_1 Y V_1^H A^H}{\sigma_1(A)} \ : \ Y \in \mathscr{S}_r\right\}.$$

The first-order model ϕ has a convenient structure, because (10.10) shows that $\phi(y,x)$ is the H_∞ norm of the transfer function $s \mapsto T(x,s) + T'(x,s)(y-x)$. In consequence, the function value of ϕ and its subgradients can be computed using the same H_∞ norm algorithm of [7] used already to compute objective value $f(x)$ and subgradients at x. This requires computing the adjoint of the operator $T'(x,j\omega)$, which is somewhat technical, so we point the reader to [4] for details.

Notice however that computing $\phi(y,x)$ is more expensive than computing $f(y)$. Indeed, since $s \mapsto T(x,s)+T'(x,s)(y-x)$ is built from a parallel connection of the transfer functions $T(x,\cdot)$ and $T'(x,\cdot)(y-x)$, it has as many states as both transfer functions taken together. While $T(x,\cdot)$ has n_x states, $T'(x,\cdot)(y-x)$ has even $2 \times n_x$ states, being itself a serial connection between two transfer function with n_x states. Hence, $T(x,\cdot)+T'(x,\cdot)(y-x)$ has $3 \times n_x$ states, and computing $\phi(y,x)$ is therefore more expensive than computing $f(x)$. As a consequence, the inner loop may significantly slowdown the algorithm. For small to medium size problems, this is not really felt, but for very large system the cost of $\phi(y,x)$ becomes dominant. This is the price to be paid for its better performance of the trust region method compared to line-search based method.

Implementation and stopping criteria. The method has been implemented in Matlab, and all numerical experiments have been performed on a 2Ghz Linux computer.

As in the previous example, we stop the algorithm if either (10.7) or the two conditions in (10.8) are satisfied. In the line-search method [4] a different stopping criterion based on a measure of criticality θ was used. The algorithm was stopped if $|\theta| < \varepsilon_\theta$. For comparison, the optimality function θ is also computed here, but *a posteriori* to measure criticality of the controller obtained by the proximity control method. We have chosen $\varepsilon_\theta = 1e-5$ for all numerical tests.

In a practical implementation it is necessary to add yet another stopping test to (10.7) and (10.8) used in the outer loop. We need a mechanism to detect convergence in the inner loop, because even though theoretically we only enter the inner loop when $0 \notin \partial f(x^j)$, it may happen that x^j is near optimal. This will typically lead to a lengthy inner loop having a hard time to find x^{j+1}, which when found will give only marginal progress over x^j. The inner loop is therefore stopped if the descent of the working model is too small, i.e.

$$f(x) - \phi_k(y^{k+1},x) < \varepsilon_2,$$

where the parameter ε_2 is chosen very small in order to stop only when the algorithm gets stuck in doing null steps. We have set $\varepsilon_2 = 0.01 \times \varepsilon$ for our tests.

Results. H_∞ synthesis was performed on four models from the COMPL$_e$IB library [19] : aircraft AC2, helicopter HE4, aircraft AC14, and distillation column BDT2. More details about these models can be found in [20]. Table 2 shows the sizes ranging from small to large. For their numerical experiments, static feedback controllers were synthesized to compare with proximity control and line-search based methods.

Model	n_x	n_z	n_w	n_y	n_u
AC2	5	5	3	3	3
HE4	8	12	8	6	4
AC14	40	11	4	4	3
BDT2	82	4	2	4	4

Table 2: Sizes of the models treated in numerical experiments

Table 3 presents results for the four models with both linesearch (LS) and proximity control (PC). For each experiments, the value of the H_∞ norm γ_∞, the number of outer and inner iterations, the mean CPU time for outer and inner iterations, the total CPU, and the criticality measure $|\theta|$ are reported. We can see that our proximity control method achieves better gains, except for AC2, where both methods find the same local minimum. In exchange, PC needs generally more iterations and takes more time than the LS approach. To compare the speed of the methods, we have reported the CPU times of LS and PC needed to reach the maximum value of the two γ_∞ values obtained by each algorithm.

- PC is faster on the HE4 model: it reaches the value 34.07 in 1.34 seconds within 16 serious step, whereas LS needs 1.5 seconds and 51 iterations. Our algorithm being a first order approach, minimization sometimes stalls when approaching the local minimum. Indeed, PC needs 19.85 seconds to perform 90% of the whole optimization, whereas 85.4% of CPU time are used for the last 10%.

- LS is faster for AC14, where it reaches the optimal value 104.43 in 4.3 seconds after 62 iterations, whereas PC needs 10 seconds and 37 serious steps to reach the same value.

- For BDT2, PC reaches the same value as LS in 22 seconds and 22 serious steps, which is faster than LS.

It can be observed in Table 3 that the LS optimization stops earlier than PC, often with a quite large value of $|\theta|$. This strange behaviour can be explained by the following fact: linesearch is not suited to handle situations, where the two first singular values of the transfer function coalesce. This is explained in [4], where for numerical simplification the authors make the assumption that the maximal singular value is simple. Figure 10.2 shows clearly that this is not the case for last step of the linesearch method on the AC14 example. This explain why LS stops prematurely. Moreover, it should be highlighted that the criticality measure θ has no meaning when the hypotheses are not satisfied. The same phenomenon was observed for the HE4 model. Again LS stops prematurely when the maximal singular value becomes non-simple at some frequencies. On both HE4 and AC14 example, PC continued to optimize even when a multiple singular-value was encountered.

Method	γ_∞	it	it$_{in}$	CPU	CPU$_{in}$	Time	$\lvert\theta\rvert$
HE4 - LS	34.07450	51	-	2.85213e-02	-	1.5	9.0
HE4 - PC	23.73891	641	2697	1.72951e-02	4.62476e-02	136	4.8e-3
AC2 - LS	0.11149	43	-	3.32525e-02	-	1.5	4.1e-6
AC2 - PC	0.11149	31	265	1.22026e-02	3.27288e-02	9	1.4e-6
AC14 - LS	104.43078	62	-	6.92522e-02	-	4.3	1.0
AC14 - PC	103.43986	134	494	3.47064e-02	2.32865e-01	120	1.6e+2
BDT2 - LS	0.82873	71	-	3.87617e-01	-	27.5	3.1e-4
BDT2 - PC	0.67687	854	1533	1.35960e-01	1.54915e+00	2490	1.2e-4

Table 3: Results of H_∞ synthesis on four models from [19], using linesearch (LS) and proximity control (PC) algorithms. For each experiment, the optimal value of the H_∞ norm γ_∞, the number of outer and inner iterations, the mean CPU times for serious steps and null steps, and the total CPU and criticality measure $|\theta|$ are given.

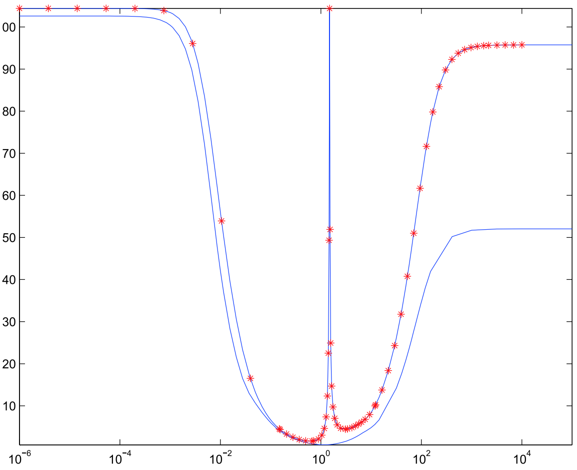


Figure 3: Frequency plot $\omega \mapsto \sigma_i\left(T(x, j\omega)\right)$ of the first two singular values $i = 1, 2$ at the last step x before acceptance of the LS method on AC14 shows coalescence $\sigma_1 \approx \sigma_2$ on a relatively large low frequency band.

Acknowledgement

This research was supported by grants from Agence Nationale de la Recherche (ANR) under contract *Guidage*, by *Fondation de Recherche pour l'Aéronautique et l'Espace* (FNRAE) under contract *Survol*, and by Agence Nationale de la Recherche (ANR) under contract *Controvert*.

References

[1] P. Apkarian and D. Noll, Nonsmooth optimization for multidisk H_∞ synthesis, *European J. Control* 12 (2006) 229–244.

[2] P. Apkarian and D. Noll, Controller design via nonsmooth multi-directional search, *SIAM J. Control and Optim.* 44 (2006) 1923–1949.

[3] P. Apkarian and D. Noll, IQC analysis and synthesis via nonsmooth optimization, *Systems and Control Letters* 55 (2006) 971–981.

[4] P. Apkarian and D. Noll, Nonsmooth H_∞ synthesis. *IEEE Trans. Autom. Control* 51 (2006) 71–86.

[5] V. Bompart, D. Noll and P. Apkarian, Second-order nonsmooth optimization of the H_∞ norm, *Numerische Mathematik* 107 (2007) 433–454.

[6] F. Bonnans, J.-C. Gilbert, C. Lemaréchal and C. Sagastizábal, *Numerical Optimization*, Theoretical and Practical Aspects, Springer Verlag, Berlin, 2006.

[7] S. Boyd and V. Balakrishnan, A regularity result for the singular values of a transfer matrix and a quadratically convergent algorithm for computing its L_∞-norm, *Systems and Control Letters* 15 (1990) 1–7.

[8] F. Clarke, *Optimization and Nonsmooth Analysis*, Canadian Math. Society Series, John Wiley & Sons, New York, 1983.

[9] A.R. Conn, N.I.M. Gould and P.L. Toint, *Trust-Region Methods.* MPS/SIAM Series On Optimization, Philadelphia, PA, USA, 2000.

[10] R. Correa and C. Lemaréchal, Convergence of some algorithms for convex minimization. *Math. Program.* 62 (1993) 261–275.

[11] J.E. Dennis Jr. and R.B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice Hall Series in Computational Mathematics, 1983.

[12] A. Fuduli, M. Gaudioso and G. Giallombardo, A DC piecewise affine model and a bundling technique in nonconvex nonsmooth optimization, *Optimization Method and Software* 19 (2004) 89–102.

[13] A. Fuduli, M. Gaudioso and G. Giallombardo, Minimizing nonconvex nonsmooth functions via cutting planes and proximity control, *SIAM J. Optim.* 14 (2004) 743–756.

[14] W.L. Hare and C. Sagastizábal, Computing proximal points of nonconvex functions, *Math. Program.* Series B. Published electronically may 2007.

[15] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms, vol I: and vol II: Advanced Theory and Bundle Methods*, vol. 306 of Grundlehren der mathematischen Wissenschaften, Springer Verlag, New York, Heidelberg, Berlin, 1993.

[16] K.C. Kiwiel, Proximity control in bundle methods for convex nondifferentiable optimization, Math. Program. 46 (1990) 105–122.

[17] K.C. Kiwiel, *Methods of Descent for Nondifferentiable Optimization*, Springer Lect. Notes in Math., vol. 1133, Springer Verlag, 1985.

[18] K.C. Kiwiel, An aggregate subgradient method for nonsmooth convex minimization, *Math. Program.* 27 (1983) 320–341.

[19] F. Leibfritz. COMPL_eIB, COnstraint Matrix-optimization Problem LIbrary - a collection of test examples for nonlinear semidefinite programs, control system design and related problems. Technical report, Universität Trier, 2003.

[20] F. Leibfritz, W. Lipinski. Description of the benchmark examples in COMPL$_e$IB Technical report, Universität Trier, 2003.

[21] C. Lemaréchal, *Nondifferentiable Optimization.* Chapter VII in: Handbooks in Operations Research and Management Sciences, vol. 1, 1989.

[22] C. Lemaréchal, Lagrangian relaxation, in *Computational and Combinatorial Optimization*, M. Junger, D. Naddef (eds.), 2001, pp. 115–169.

[23] L. Lukšan and J. Vlček, *A bundle-Newton method for nonsmooth unconstrained minimization*, *Math. Program.* 83 (1998) 373–391.

[24] D. Noll and P. Apkarian, Spectral bundle method for nonconvex maximum eigenvalue functions: first-order methods, *Math. Program.* Series B, 104 (2005) 701–727.

[25] D. Noll and P. Apkarian, Spectral bundle method for nonconvex maximum eigenvalue functions: second-order methods, *Math. Program.* Series B, 104 (2005) 729–747.

[26] E. Polak, On the mathematical foundations of nondifferentiable optimization in engineering design, *SIAM Review* 29 (1987) 21–89.

[27] A. Ruszczyński, *Nonlinear optimization*, Princeton University Press, 2006.

[28] H. Schramm, Eine Kombination von Bundle- und Trust-Region-Verfahren zur Lösung nicht-differenzierbarer Optimierungsprobleme. Bayreuther Mathematische Schriften, Heft 30, Bayreuth 1989.

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

[30] N. Shor, *Minimization Methods for Nondifferentiable Functions.* Springer Verlag, 1985.

[31] R.J. Stern and H. Wolkowicz, Indefinite trust region programs and nonsymmetric eigenvalue perturbations, *SIAM J. Optim.* 5 (1995) 286–313.

[32] P. Wolfe, A method of conjugate subgradients for minimizing nondifferentiable functions, in *Nondifferentiable Optimization*, M.L. Balinski, P. Wolfe (eds.), Math. Programming Studies, vol. 3. North-Holland, 1975, pp. 145–173.

[33] J. Zowe, The BT-Algorithm for minimizing a nonsmooth functional subject to linear constraints, in *Nonsmooth Optimization and Related Topics* , F. H. Clarke, V. F. Demyanov, F. Gianessi (eds.), Plenum Press, 1989.

Manuscript received 26 October 2007
revised 27 April 2008
accepted for publication 25 May 2008

Dominikus Noll
Université Paul Sabatier, Institut de Mathématiques
118 route de Narbonne, 31062 Toulouse, France
E-mail address: `dominikus.noll@wanadoo.fr`

Olivier Prot
Université Paul Sabatier, Institut de Mathématiques
118 route de Narbonne, 31062 Toulouse, France
E-mail address: `oprot@mip.ups-tlse.fr`

Aude Rondepierre
Université Paul Sabatier, Institut de Mathématiques
118 route de Narbonne, 31062 Toulouse, France
E-mail address: `aude.rondepierre@math.univ-toulouse.fr`