

# Analytic Center Cutting-Plane Method

S. Boyd, L. Vandenberghe, and J. Skaf

April 25, 2018

## Contents

<b>1</b>	<b>Analytic center cutting-plane method</b>	<b>2</b>
<b>2</b>	<b>Computing the analytic center</b>	<b>3</b>
<b>3</b>	<b>Pruning constraints</b>	<b>5</b>
<b>4</b>	<b>Lower bound and stopping criterion</b>	<b>5</b>
<b>5</b>	<b>Convergence proof</b>	<b>7</b>
<b>6</b>	<b>Numerical examples</b>	<b>10</b>
6.1	Basic ACCPM . . . . .	10
6.2	ACCPM with constraint dropping . . . . .	10
6.3	Epigraph ACCPM . . . . .	13

In these notes we describe in more detail the analytic center cutting-plane method (ACCPM) for non-differentiable convex optimization, prove its convergence, and give some numerical examples. ACCPM was developed by Goffin and Vial [GV93] and analyzed by Nesterov [Nes95] and Atkinson and Vaidya [AV95].

These notes assume a basic familiarity with convex optimization (see [BV04]), cutting-plane methods (see the EE364b notes *Localization and Cutting-Plane Methods*), and subgradients (see the EE364b notes *Subgradients*).

## 1 Analytic center cutting-plane method

The basic ACCPM algorithm is:

*Analytic center cutting-plane method (ACCPM)*

**given** an initial polyhedron  $\mathcal{P}_0$  known to contain  $X$ .

$k := 0$ .

**repeat**

    Compute  $x^{(k+1)}$ , the analytic center of  $\mathcal{P}_k$ .

    Query the cutting-plane oracle at  $x^{(k+1)}$ .

    If the oracle determines that  $x^{(k+1)} \in X$ , quit.

    Else, add the returned cutting-plane inequality to  $\mathcal{P}$ .

$\mathcal{P}_{k+1} := \mathcal{P}_k \cap \{z \mid a^T z \leq b\}$

    If  $\mathcal{P}_{k+1} = \emptyset$ , quit.

$k := k + 1$ .

There are several variations on this basic algorithm. For example, at each step we can add multiple cuts, instead of just one. We can also prune or drop constraints, for example, after computing the analytic center of  $\mathcal{P}_k$ . Later we will see a simple but non-heuristic stopping criterion.

We can construct a cutting-plane  $a^T z \leq b$  at  $x^{(k)}$ , for the standard convex problem

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{subject to} && f_i(x) \leq 0, \quad i = 1, \dots, m, \end{aligned}$$

as follows. If  $x^{(k)}$  violates the  $i$ th constraint, *i.e.*,  $f_i(x^{(k)}) > 0$ , we can take

$$a = g_i, \quad b = g_i^T x^{(k)} - f_i(x^{(k)}), \tag{1}$$

where  $g_i \in \partial f_i(x^{(k)})$ . If  $x^{(k)}$  is feasible, we can take

$$a = g_0, \quad b = g_0^T x^{(k)} - f_0(x^{(k)}) + f_{\text{best}}^{(k)}, \tag{2}$$

where  $g_0 \in \partial f_0(x^{(k)})$ , and  $f_{\text{best}}^{(k)}$  is the best (lowest) objective value encountered for a feasible iterate.

## 2 Computing the analytic center

Each iteration of ACCPM requires computing the analytic center of a set of linear inequalities (and, possibly, determining whether the set of linear inequalities is feasible),

$$a_i^T x \leq b_i, \quad i = 1, \dots, m,$$

that define the current localization polyhedron  $\mathcal{P}$ . In this section we describe some methods that can be used to do this.

We note that the inequalities defined by  $a_i$  and  $b_i$ , as well as their number  $m$ , can change at each iteration of ACCPM, as we add new cutting-planes and possibly prune others. In addition, these inequalities can include some of the original inequalities that define  $\mathcal{P}_0$ .

To find the analytic center, we must solve the problem

$$\text{minimize } \Phi(x) = -\sum_{i=1}^m \log(b_i - a_i^T x). \quad (3)$$

This is an unconstrained problem, but the domain of the objective function is the open polyhedron

$$\mathbf{dom} \Phi = \{x \mid a_i^T x < b_i, \quad i = 1, \dots, m\},$$

*i.e.*, the interior of the polyhedron. Part of the challenge of computing the analytic center is that we are not given a point in the domain. One simple approach is to use a phase I optimization method (see [BV04, §11.4]) to find a point in  $\mathbf{dom} \Phi$  (or determine that  $\mathbf{dom} \Phi = \emptyset$ ). Once we find such a point, we can use a standard Newton method to compute the analytic center (see [BV04, §9.5]).

A simple alternative is to use an infeasible start Newton method (see [BV04, §10.3]) to solve (3), starting from a point that is outside  $\mathbf{dom} \Phi$ , as suggested by Goffin and Sharifi-Mokhtarian in [GSM99]. We reformulate the problem as

$$\begin{aligned} &\text{minimize} && -\sum_{i=1}^m \log y_i \\ &\text{subject to} && y = b - Ax, \end{aligned} \quad (4)$$

with variables  $x$  and  $y$ . The infeasible start Newton method can be started from any  $x$  and any  $y \succ 0$ . We can, for example, take the initial  $x$  to be the previous point  $x_{\text{prev}}$ , and choose  $y$  as

$$y_i = \begin{cases} b_i - a_i^T x & \text{if } b_i - a_i^T x > 0 \\ 1 & \text{otherwise.} \end{cases}$$

In the basic form of ACCPM  $x_{\text{prev}}$  strictly satisfies all of the inequalities except the last one added; in this case,  $y_i = b_i - a_i^T x$  holds for all but one index.

We define the primal and dual residuals for (4) as

$$r_p = y + Ax - b, \quad r_d = \begin{bmatrix} A^T \nu \\ g + \nu \end{bmatrix}, \quad (5)$$

where  $g = -\mathbf{diag}(1/y_i)\mathbf{1}$  is the gradient of the objective. We also define  $r$  to be  $(r_d, r_p)$ . The Newton step at a point  $(x, y, \nu)$  is defined by the system of linear equations

$$\begin{bmatrix} 0 & 0 & A^T \\ 0 & H & I \\ A & I & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \nu \end{bmatrix} = - \begin{bmatrix} r_d \\ r_p \end{bmatrix},$$

where  $H = \mathbf{diag}(1/y_i^2)$  is the Hessian of the objective. We can solve this system by block elimination (see [BV04, §10.4]), using the expressions

$$\begin{aligned} \Delta x &= -(A^T H A)^{-1}(A^T g - A^T H r_p), \\ \Delta y &= -A \Delta x - r_p, \\ \Delta \nu &= -H \Delta y - g - \nu. \end{aligned} \tag{6}$$

We can compute  $\Delta x$  from the first equation in several ways. We can, for example, form  $A^T H A$ , then compute its Cholesky factorization, then carry out backward and forward substitution. Another option is to compute  $\Delta x$  by solving the least-squares problem

$$\Delta x = \operatorname{argmin}_z \left\| H^{1/2} A z - H^{1/2} r_p + H^{-1/2} g \right\|.$$

The infeasible start Newton method is:

*Infeasible start Newton method.*

**given** starting point  $x, y \succ 0$ , tolerance  $\epsilon > 0$ ,  $\alpha \in (0, 1/2)$ ,  $\beta \in (0, 1)$ .

$\nu := 0$ .

Compute residuals from (5).

**repeat**

1. Compute Newton step  $(\Delta x, \Delta y, \Delta \nu)$  using (6).

2. *Backtracking line search on  $\|r\|_2$ .*

$t := 1$ .

**while**  $y + t\Delta y \not\prec 0$ ,  $t := \beta t$ .

**while**  $\|r(x + t\Delta x, y + t\Delta y, \nu + t\Delta \nu)\|_2 > (1 - \alpha t)\|r(x, y, \nu)\|_2$ ,  $t := \beta t$ .

3. *Update.*  $x := x + t\Delta x$ ,  $y := y + t\Delta y$ ,  $\nu := \nu + t\Delta \nu$ .

**until**  $y = b - Ax$  and  $\|r(x, y, \nu)\|_2 \leq \epsilon$ .

This method works quite well, unless the polyhedron is empty (or, in practice, very small), in which case the algorithm does not converge. To guard against this, we fix a maximum number of iterations. Typical parameter values are  $\beta = 0.5$ ,  $\alpha = 0.01$ , with maximum iterations set to 50.

### 3 Pruning constraints

There is a simple method for ranking the relevance of the inequalities  $a_i^T x \leq b_i$ ,  $i = 1, \dots, m$  that define a polyhedron  $\mathcal{P}$ , once we have computed the analytic center  $x^*$ . Let

$$H = \nabla^2 \Phi(x^*) = \sum_{i=1}^m (b_i - a_i^T x^*)^{-2} a_i a_i^T.$$

Then the ellipsoid

$$\mathcal{E}_{\text{in}} = \{z \mid (z - x^*)^T H (z - x^*) \leq 1\}$$

lies inside  $\mathcal{P}$ , and the ellipsoid

$$\mathcal{E}_{\text{out}} = \{z \mid (z - x^*)^T H (z - x^*) \leq m^2\},$$

which is  $\mathcal{E}_{\text{in}}$  scaled by a factor  $m$  about its center, contains  $\mathcal{P}$ . Thus the ellipsoid  $\mathcal{E}_{\text{in}}$  at least grossly (within a factor  $m$ ) approximates the shape of  $\mathcal{P}$ .

This suggests the (ir)relevance measure

$$\eta_i = \frac{b_i - a_i^T x^*}{\|H^{-1/2} a_i\|} = \frac{b_i - a_i^T x^*}{\sqrt{a_i^T H^{-1} a_i}}$$

for the inequality  $a_i^T x \leq b_i$ . This factor is always at least one; if it is  $m$  or larger, then the inequality is certainly redundant.

These factors (which are easily computed from the computations involved in the Newton method) can be used to decide which constraints to drop or prune. We simply drop constraints with the large values of  $\eta_i$ ; we keep constraints with smaller values. One typical scheme is to keep some fixed number  $N$  of constraints, where  $N$  is usually chosen to be between  $3n$  and  $5n$ . When this is done, the computational effort per iteration (*i.e.*, centering) does not grow as ACCPM proceeds, as it does when no pruning is done.

### 4 Lower bound and stopping criterion

In §7 of the notes *Localization and Cutting-Plane Methods* we described a general method for constructing a lower bound on the optimal value  $p^*$  of the convex problem

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{subject to} && f_1(x) \leq 0, \quad Cx \preceq d, \end{aligned} \tag{7}$$

assuming we have evaluated the value and a subgradient of its objective and constraint functions at some points. For notational simplicity we lump multiple constraints into one by forming the maximum of the constraint functions.

We will re-order the iterates so that at  $x^{(1)}, \dots, x^{(q)}$ , we have evaluated the value and a subgradient of the objective function  $f_0$ . This gives us the piecewise-linear underestimator  $\hat{f}_0$  of  $f_0$ , defined as

$$\hat{f}_0(z) = \max_{i=1, \dots, q} \left( f_0(x^{(i)}) + g_0^{(i)T} (z - x^{(i)}) \right) \leq f_0(z).$$

We assume that at  $x^{(q+1)}, \dots, x^{(k)}$ , we have evaluated the value and a subgradient of the constraint function  $f_1$ . This gives us the piecewise-linear underestimator  $\hat{f}_1$ , given by

$$\hat{f}_1(z) = \max_{i=q+1, \dots, k} \left( f_1(x^{(i)}) + g_1^{(i)T}(z - x^{(i)}) \right) \leq f_1(z).$$

Now we form the problem

$$\begin{aligned} & \text{minimize} && \hat{f}_0(x) \\ & \text{subject to} && \hat{f}_1(x) \leq 0, \quad Cx \preceq d, \end{aligned} \tag{8}$$

whose optimal value is a lower bound on  $p^*$ .

In ACCPM we can easily construct a lower bound on the problem (8), as a by-product of the analytic centering computation, which in turn gives a lower bound on the original problem (7). We first write (8) as the LP

$$\begin{aligned} & \text{minimize} && t \\ & \text{subject to} && f_0(x^{(i)}) + g_0^{(i)T}(x - x^{(i)}) \leq t, \quad i = 1, \dots, q \\ & && f_1(x^{(i)}) + g_1^{(i)T}(x - x^{(i)}) \leq 0, \quad i = q + 1, \dots, k \\ & && Cx \preceq d, \end{aligned} \tag{9}$$

with variables  $x$  and  $t$ . The dual problem is

$$\begin{aligned} & \text{maximize} && \sum_{i=1}^q \lambda_i (f_0(x^{(i)}) - g_0^{(i)T} x^{(i)}) + \sum_{i=q+1}^k \lambda_i (f_1(x^{(i)}) - g_1^{(i)T} x^{(i)}) - d^T \mu \\ & \text{subject to} && \sum_{i=1}^q \lambda_i g_0^{(i)} + \sum_{i=q+1}^k \lambda_i g_1^{(i)} + C^T \mu = 0 \\ & && \mu \succeq 0, \quad \lambda \succeq 0, \quad \sum_{i=1}^q \lambda_i = 1, \end{aligned} \tag{10}$$

with variables  $\lambda, \mu$ .

We can compute at modest cost a lower bound to the optimal value of (9), and hence to  $p^*$ , by finding a dual feasible point for (10).

The optimality condition for  $x^{(k+1)}$ , the analytic center of the inequalities

$$\begin{aligned} f_0(x^{(i)}) + g_0^{(i)T}(x - x^{(i)}) &\leq f_{\text{best}}^{(i)}, & i = 1, \dots, q, \\ f_1(x^{(i)}) + g_1^{(i)T}(x - x^{(i)}) &\leq 0, & i = q + 1, \dots, k, \\ c_i^T x &\leq d_i, & i = 1, \dots, m, \end{aligned}$$

is

$$\sum_{i=1}^q \frac{g_0^{(i)}}{f_{\text{best}}^{(i)} - f_0(x^{(i)}) - g_0^{(i)T}(x^{(k+1)} - x^{(i)})} + \sum_{i=q+1}^k \frac{g_1^{(i)}}{-f_1(x^{(i)}) - g_1^{(i)T}(x^{(k+1)} - x^{(i)})} + \sum_{i=1}^m \frac{c_i}{d_i - c_i^T x^{(k+1)}} = 0. \tag{11}$$

Let  $\tau_i = 1/(f_{\text{best}}^{(i)} - f_0(x^{(i)}) - g_0^{(i)T}(x^{(k+1)} - x^{(i)}))$  for  $i = 1, \dots, q$ . Comparing (11) with the equality constraint in (10), we can construct a dual feasible point by taking

$$\begin{aligned} \lambda_i &= \begin{cases} \tau_i / \mathbf{1}^T \tau & \text{for } i = 1, \dots, q \\ 1/(-f_1(x^{(i)}) - g_1^{(i)T}(x^{(k+1)} - x^{(i)}))(\mathbf{1}^T \tau) & \text{for } i = q + 1, \dots, k, \end{cases} \\ \mu_i &= 1/(d_i - c_i^T x^{(k+1)})(\mathbf{1}^T \tau) \quad i = 1, \dots, m. \end{aligned}$$

Using these values of  $\lambda$  and  $\mu$ , we conclude that

$$p^* \geq l^{(k+1)},$$

where  $l^{(k+1)} = \sum_{i=1}^q \lambda_i (f_0(x^{(i)}) - g_0^{(i)T} x^{(i)}) + \sum_{i=q+1}^k \lambda_i (f_1(x^{(i)}) - g_1^{(i)T} x^{(i)}) - d^T \mu$ .

Let  $l_{\text{best}}^{(k)}$  be the best lower bound found after  $k$  iterations. The ACCPM algorithm can be stopped once the gap  $f_{\text{best}}^{(k)} - l_{\text{best}}^{(k)}$  is less than a desired value  $\epsilon > 0$ . This guarantees that  $x^{(k)}$  is, at most,  $\epsilon$ -suboptimal.

## 5 Convergence proof

In this section we give a convergence proof for ACCPM, adapted from Ye [Ye97, Chap. 6]. We take the initial polyhedron as the unit box, centered at the origin, with unit length sides, *i.e.*, the initial set of linear inequalities is

$$-(1/2)\mathbf{1} \preceq z \preceq (1/2)\mathbf{1},$$

so the first analytic center is  $x^{(1)} = 0$ . We assume the target set  $X$  contains a ball with radius  $r < 1/2$ , and show that the number of iterations is no more than a constant times  $n^2/r^2$ .

Assuming the algorithm has not terminated, the set of inequalities after  $k$  iterations is

$$-(1/2)\mathbf{1} \preceq z \preceq (1/2)\mathbf{1}, \quad a_i^T z \leq b_i, \quad i = 1, \dots, k. \quad (12)$$

We assume the cuts are neutral, so  $b_i = a_i^T x^{(i)}$  for  $i = 1, \dots, k$ . Without loss of generality we normalize the vectors  $a_i$  so that  $\|a_i\|_2 = 1$ . We will let  $\phi_k : \mathbf{R}^n \rightarrow \mathbf{R}$  be the logarithmic barrier function associated with the inequalities (12),

$$\phi_k(z) = -\sum_{i=1}^n \log(1/2 + z_i) - \sum_{i=1}^n \log(1/2 - z_i) - \sum_{i=1}^k \log(b_i - a_i^T z).$$

The iterate  $x^{(k+1)}$  is the minimizer of this logarithmic barrier function.

Since the algorithm has not terminated, the polyhedron  $\mathcal{P}_k$  defined by (12) still contains the target set  $X$ , and hence also a ball with radius  $r$  and (unknown) center  $x_c$ . We have  $(-1/2 + r)\mathbf{1} \preceq x_c \preceq (1/2 - r)\mathbf{1}$ , and the slacks of the inequalities  $a_i^T z \leq b_i$  evaluated at  $x_c$  also exceed  $r$ :

$$b_i - \sup_{\|v\|_2 \leq 1} a_i^T (x_c + rv) = b_i - a_i^T x_c - r\|a_i\|_2 = b_i - a_i^T x_c - r \geq 0.$$

Therefore  $\phi_k(x_c) \leq -(2n + k) \log r$  and, since  $x^{(k)}$  is the minimizer of  $\phi_k$ ,

$$\phi_k(x^{(k)}) = \inf_z \phi_k(z) \leq \phi_k(x_c) \leq (2n + k) \log(1/r). \quad (13)$$

We can also derive a lower bound on  $\phi_k(x^{(k)})$  by noting that the functions  $\phi_j$  are self-concordant for  $j = 1, \dots, k$ . Using the inequality (9.48), [BV04, p.502], we have

$$\phi_j(x) \geq \phi_j(x^{(j)}) + \sqrt{(x - x^{(j)})^T H_j (x - x^{(j)})} - \log(1 + \sqrt{(x - x^{(j)})^T H_j (x - x^{(j)})})$$

for all  $x \in \mathbf{dom} \phi_j$ , where  $H_j$  is the Hessian of  $\phi_j$  at  $x^{(j)}$ . If we apply this inequality to  $\phi_{k-1}$  we obtain

$$\begin{aligned} \phi_k(x^{(k)}) &= \inf_x \phi_k(x) \\ &= \inf_x \left( \phi_{k-1}(x) - \log(-a_k^T(x - x^{(k-1)})) \right) \\ &\geq \inf_v \left( \phi_{k-1}(x^{(k-1)}) + \sqrt{v^T H_{k-1} v} - \log(1 + \sqrt{v^T H_{k-1} v}) - \log(-a_k^T v) \right). \end{aligned}$$

By setting the gradient of the righthand side equal to zero, we find that it is minimized at

$$\hat{v} = -\frac{1 + \sqrt{5}}{2\sqrt{a_k^T H_{k-1}^{-1} a_k}} H_{k-1}^{-1} a_k,$$

which yields

$$\begin{aligned} \phi_k(x^{(k)}) &\geq \phi_{k-1}(x^{(k-1)}) + \sqrt{\hat{v}^T H_{k-1} \hat{v}} - \log(1 + \sqrt{\hat{v}^T H_{k-1} \hat{v}}) - \log(-a_k^T \hat{v}) \\ &= \phi_{(k-1)}(x^{(k-1)}) + 0.1744 - \frac{1}{2} \log(a_k^T H_{k-1}^{-1} a_k) \\ &\geq 0.1744k - \frac{1}{2} \sum_{i=1}^k \log(a_i^T H_{i-1}^{-1} a_i) + 2n \log 2 \\ &\geq 0.1744k - \frac{k}{2} \log \left( \frac{1}{k} \sum_{i=1}^k a_i^T H_{i-1}^{-1} a_i \right) + 2n \log 2 \\ &\geq -\frac{k}{2} \log \left( \frac{1}{k} \sum_{i=1}^k a_i^T H_{i-1}^{-1} a_i \right) + 2n \log 2 \end{aligned} \tag{14}$$

because  $\phi_0(x^{(0)}) = 2n \log 2$ . We can further bound the second term on the righthand side by noting that

$$H_i = 4 \mathbf{diag}(\mathbf{1} + 2x^{(i)})^{-2} + 4 \mathbf{diag}(\mathbf{1} - 2x^{(i)})^{-2} + \sum_{j=1}^i \frac{1}{(b_j - a_j^T x^{(i)})^2} a_j a_j^T \succeq I + \frac{1}{n} \sum_{j=1}^i a_j a_j^T$$

because  $-(1/2)\mathbf{1} \prec x^{(i)} \prec (1/2)\mathbf{1}$  and

$$b_i - a_i^T x^{(k)} = a_i^T (x^{(i-1)} - x^{(k)}) \leq \|a_i\|_2 \|x^{(i-1)} - x^{(k)}\|_2 \leq \sqrt{n}.$$

Define  $B_0 = I$  and  $B_i = I + (1/n) \sum_{j=1}^i a_j a_j^T$  for  $i \geq 1$ . Then

$$n \log(1 + k/n^2) = n \log(\mathbf{Tr} B_k/n) \geq \log \det B_k$$



$$\begin{aligned}
&= \log \det B_{k-1} + \log\left(1 + \frac{1}{n} a_k^T B_{k-1}^{-1} a_k\right) \\
&\geq \log \det B_{k-1} + \frac{1}{2n} a_k^T B_{k-1}^{-1} a_k \\
&\geq \frac{1}{2n} \sum_{i=1}^k a_i^T B_{i-1}^{-1} a_i.
\end{aligned}$$

(The second inequality follows from the fact that  $a_k^T B_{k-1}^{-1} a_k \leq 1$ , and  $\log(1+x) \geq (\log 2)x \geq x/2$  for  $0 \leq x \leq 1$ .) Therefore

$$\sum_{i=1}^k a_i^T H_{i-1}^{-1} a_i \leq \sum_{i=1}^k a_i^T B_{i-1}^{-1} a_i \leq 2n^2 \log\left(1 + \frac{k}{n^2}\right),$$

and we can simplify (14) as

$$\begin{aligned}
\phi_k(x^{(k)}) &\geq -\frac{k}{2} \log\left(2 \frac{\log(1+k/n^2)}{k/n^2}\right) + 2n \log 2 \\
&= (2n - \frac{k}{2}) \log 2 + \frac{k}{2} \log\left(\frac{k/n^2}{\log(1+k/n^2)}\right). \tag{15}
\end{aligned}$$

Combining this lower bound with the upper bound (13) we find

$$-\frac{k}{2} \log 2 + \frac{k}{2} \log\left(\frac{k/n^2}{\log(1+k/n^2)}\right) \leq k \log(1/r) + 2n \log(1/(2r)). \tag{16}$$

From this it is clear that the algorithm terminates after a finite  $k$ : since the ratio  $(k/n^2)/\log(1+k/n^2)$  goes to infinity, the left hand side grows faster than linearly as  $k$  increases.

We can derive an explicit bound on  $k$  as follows. Let  $\alpha(r)$  be the solution of the nonlinear equation

$$\alpha / \log(1 + \alpha) = 1/(2r^4).$$

Suppose  $k > \max\{2n, n^2\alpha(r)\}$ . Then we have a contradiction in (16):

$$k \log(1/(2r^2)) \leq -\frac{k}{2} \log \sqrt{2} + \frac{k}{2} \log\left(\frac{k/n^2}{\log(1+k/n^2)}\right) \leq k \log(1/r) + 2n \log(1/(2r)),$$

*i.e.*,  $k \log(1/(2r)) \leq 2n \log(1/(2r))$ . We conclude that

$$\max\{2n, n^2\alpha(r)\}$$

is an upper bound on the number of iterations.

## 6 Numerical examples

We consider the problem of minimizing a piecewise linear function:

$$\text{minimize } f(x) = \max_{i=1,\dots,m} (a_i^T x + b_i),$$

with variable  $x \in \mathbf{R}^n$ . The particular problem instance we use to illustrate the different methods has  $n = 20$  variables and  $m = 100$  terms, with problem data  $a_i$  and  $b_i$  generated from a unit normal distribution. Its optimal value (which is easily computed via linear programming) is  $f^* \approx 1.1$ .

### 6.1 Basic ACCPM

We use the basic ACCPM algorithm described in §1, with the infeasible start Newton method used to carry out the centering steps. We take  $\mathcal{P}_0$  to be the unit box  $\{z \mid \|z\|_\infty \leq 1\}$ . We keep track of  $f_{\text{best}}$ , the best objective value found, and use this to generate deep objective cuts. Figure 1 shows convergence of  $f^{(k)} - f^*$ , which is nearly linear (on a semi-log scale).

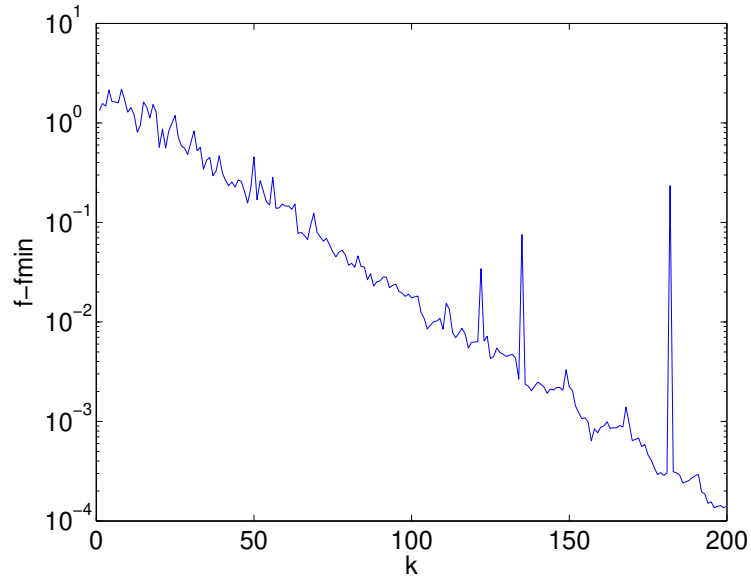
Figure 2 shows the convergence of the true suboptimality  $f_{\text{best}}^{(k)} - f^*$  (which is not available as the algorithm is running), along with the upper bound on suboptimality  $f_{\text{best}}^{(k)} - l_{\text{best}}^{(k)}$  (which is available as the algorithm runs).

Figure 3 shows  $f_{\text{best}}^{(k)} - f^*$  versus the cumulative number of Newton steps required by the infeasible start Newton method in the centering steps. This plots shows that around 10 Newton steps are needed, on average, to carry out the centering. We can see that as the algorithm progresses (and  $\mathcal{P}^{(k)}$  gets small), there is a small increase in the number of Newton steps required to achieve the same factor increase in accuracy.

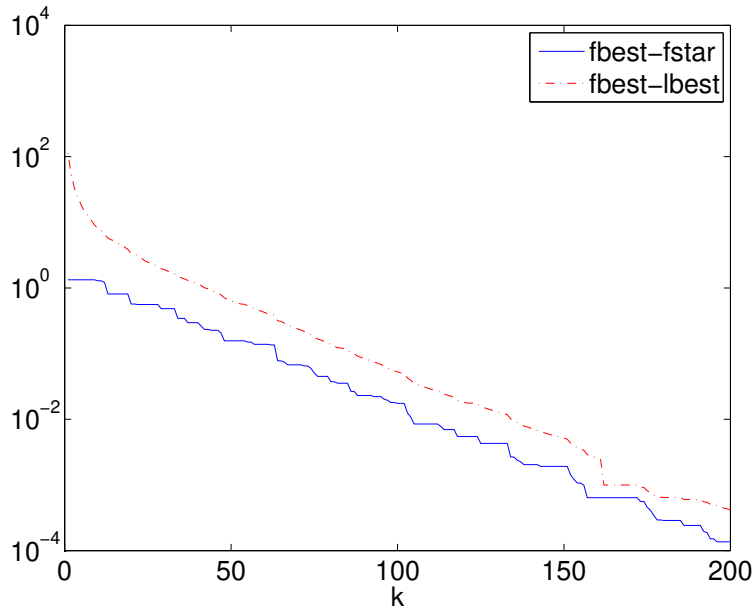
### 6.2 ACCPM with constraint dropping

We illustrate ACCPM with constraint dropping, keeping at most  $N = 3n$  constraints, using the constraint dropping scheme described in §3. Figure 4 shows convergence of  $f^{(k)} - f^*$  with and without constraint dropping. The plots show that keeping only  $3n$  constraints has almost no effect on the progress of the algorithm, as measured in iterations. At  $k = 200$  iterations, the pruned polyhedron has 60 constraints, whereas the unpruned polyhedron has 240 constraints.

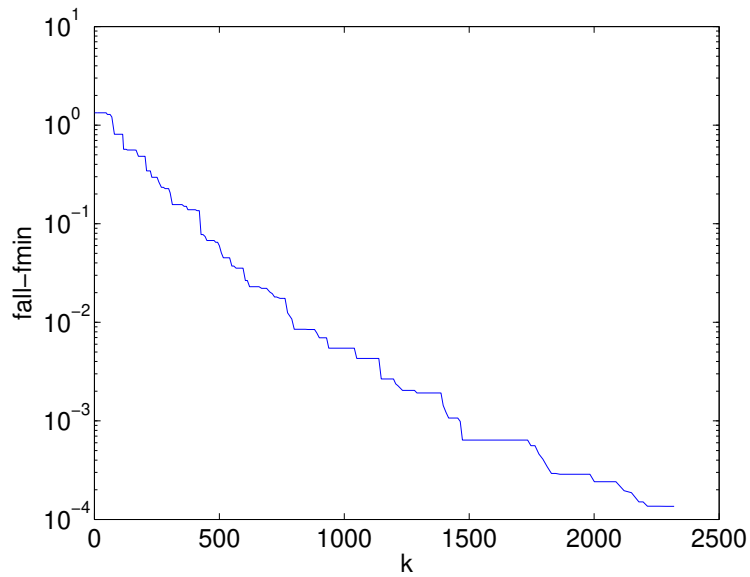
The number of flops per Newton step is, to first order,  $m_k n^2$ , where  $m_k$  is the number of constraints at iteration  $k$ , so the total flop count of iteration  $k$  can be estimated as  $N_k m_k n^2$ , where  $N_k$  is the number of Newton steps required in iteration  $k$ . Figure 5 shows the convergence of  $f_{\text{best}}^{(k)} - f^*$  versus the (estimated) cumulative flop count.



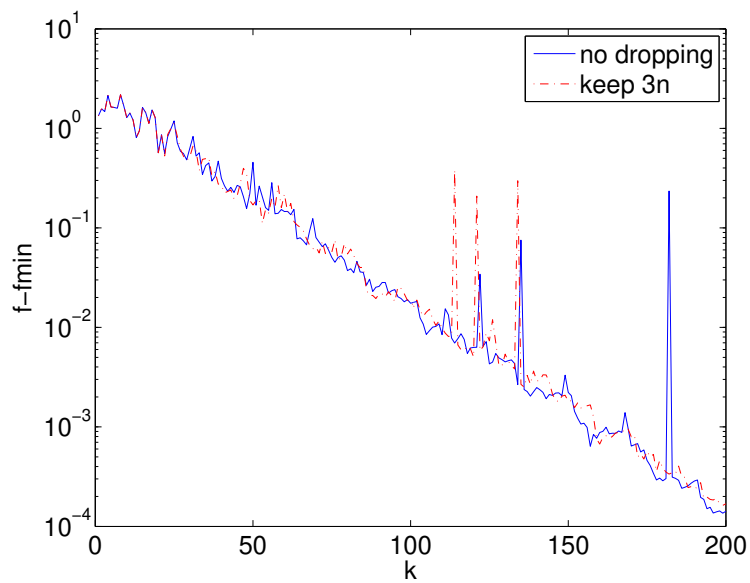
**Figure 1:** The value of  $f^{(k)} - f^*$  versus iteration number  $k$ , for the basic ACCPM.



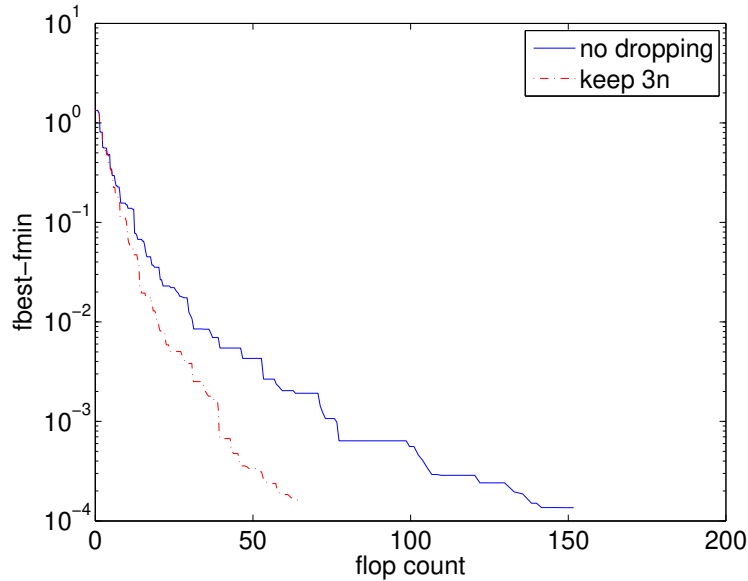
**Figure 2:** The value of  $f_{\text{best}}^{(k)} - f^*$  (in blue) and the value of  $f_{\text{best}}^{(k)} - l_{\text{best}}^{(k)}$  (in red) versus iteration number  $k$  for the basic ACCPM.



**Figure 3:** The value of  $f_{\text{best}}^{(k)} - f^*$  versus the cumulative number of Newton steps, for the basic ACCPM.



**Figure 4:** The value of  $f^{(k)} - f^*$  versus iteration number  $k$  in the case where all constraints are kept (blue) and only  $3n$  constraints are kept (red).

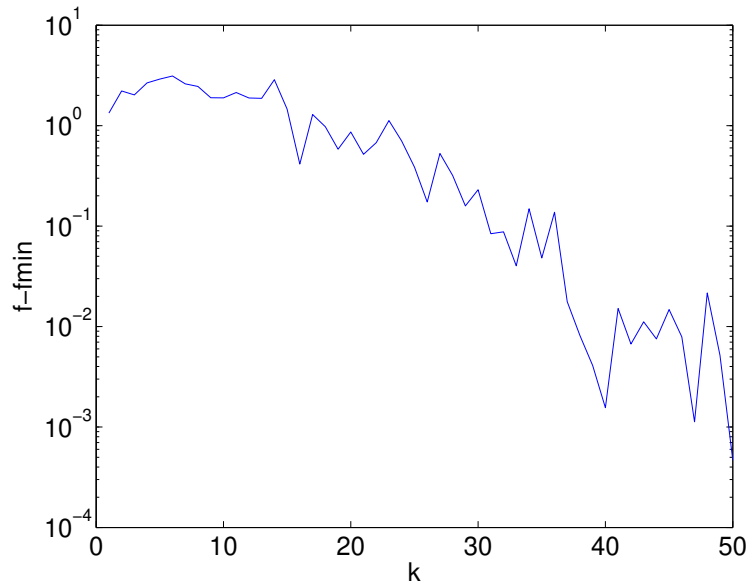


**Figure 5:** The value of  $f_{\text{best}}^{(k)} - f^*$  versus estimated cumulative flop count in the case where all constraints are kept (blue) and only  $3n$  constraints are kept (red).

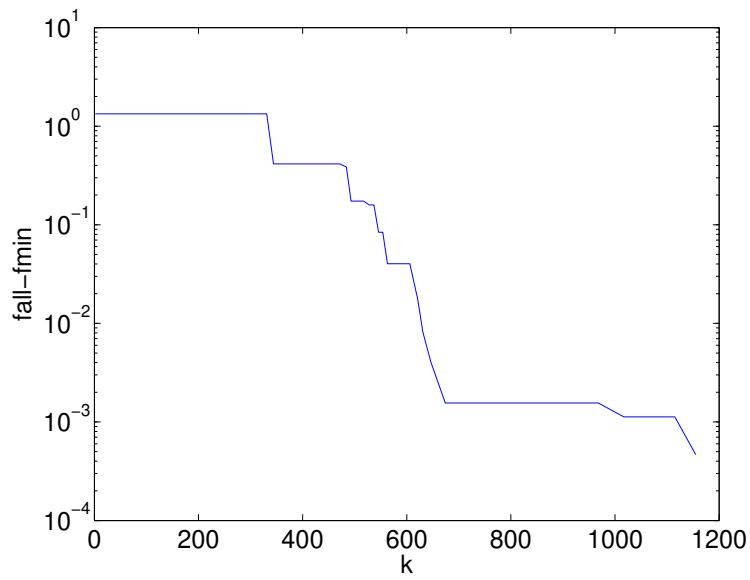
### 6.3 Epigraph ACCPM

Figure 6 shows convergence of  $f^{(k)} - f^*$  for the epigraph ACCPM. (See §6 of the EE364b notes *Localization and Cutting-Plane Methods* for details.) Epigraph ACCPM requires only 50 iterations to reach the same accuracy that was reached by basic ACCPM in 200 iterations.

The convergence of  $f_{\text{best}}^{(k)} - f^*$  versus the cumulative number of Newton steps is shown in figure 7. We see that in epigraph ACCPM the average number of Newton steps per iteration is a bit higher than for basic ACCPM, but a substantial advantage remains.



**Figure 6:** The value of  $f^{(k)} - f^*$  versus iteration number  $k$ , for epigraph ACCPM.



**Figure 7:** The value of  $f_{\text{best}}^{(k)} - f^*$  versus the cumulative number of Newton steps, for epigraph ACCPM.

## References

- [AV95] D. Atkinson and P. Vaidya. A cutting plane algorithm for convex programming that uses analytic centers. *Mathematical Programming*, 69:1–43, 1995.
- [BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [GSM99] J. Goffin and F. Sharifi-Mokhtarian. Using the primal-dual infeasible newton method in the analytic center method for problems defined by deep cutting planes. *Journal of Optimization Theory and Applications*, 101:35–58, April 1999.
- [GV93] J. Goffin and J. Vial. On the computation of weighted analytic centers and dual ellipsoids with the projective algorithm. *Mathematical Programming*, 60:81–92, 1993.
- [Nes95] Y. Nesterov. Cutting-plane algorithms from analytic centers: efficiency estimates. *Mathematical Programming*, 69:149–176, 1995.
- [Ye97] Yinyu Ye. Complexity analysis of the analytic center cutting plane method that uses multiple cuts. *Mathematical Programming*, 78:85–104, 1997.