10 Karmarkars Algorithm

- inequalities $Ax \le b$; $m \times n$ matrix A with rows a_i^T
- $P = \{x \mid Ax \le b\}$; $P^\circ := \{x \mid Ax < b\}$
- interior point algorithm: $x \in P^\circ$ throughout the algorithm
- for $x \in P^\circ$ define
$$s_i(x) := b_i - a_i^T x$$
as the slack of the i-th constraint

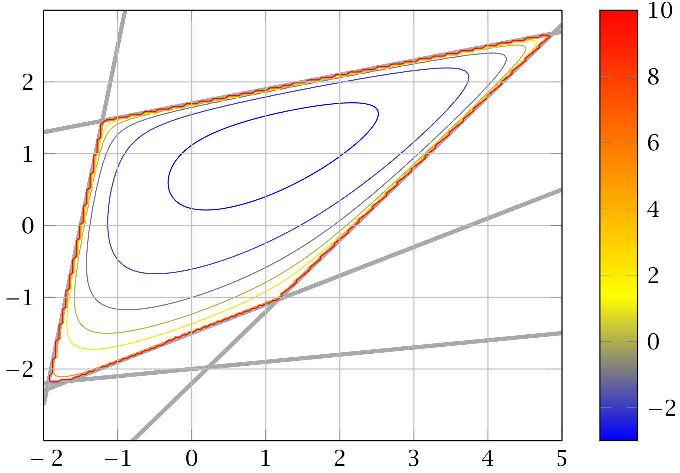
logarithmic barrier function:

$$\phi(x) = -\sum_{i=1}^{m} \ln(s_i(x))$$

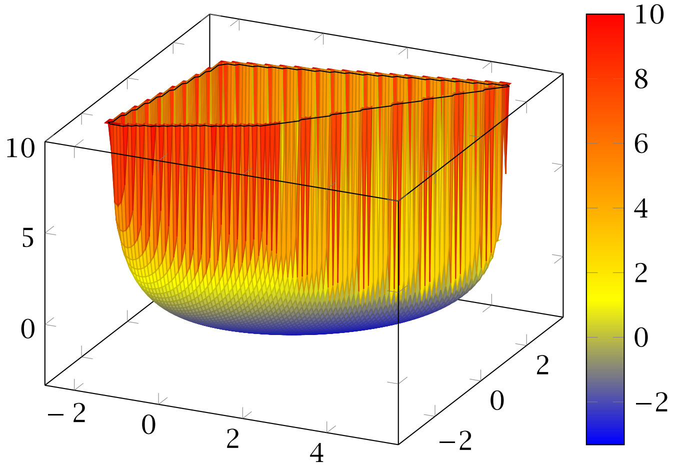
Penalty for point x; points close to the boundary have a very large penalty.

Throughout this section a_i denotes the i-th row as a column vector.

Penalty Function



Penalty Function



Gradient and Hessian

Taylor approximation:

$$\phi(x+\epsilon) \approx \phi(x) + \nabla\phi(x)^T\epsilon + \frac{1}{2}\epsilon^T\nabla^2\phi(x)\epsilon$$

Gradient:

$$\nabla\phi(x) = \sum_{i=1}^{m} \frac{1}{s_i(x)} \cdot a_i = A^T d_x$$

where $d_x^T = (1/s_1(x), \dots, 1/s_m(x))$. ($d_x$ vector of inverse slacks)

Hessian:

$$H_x := \nabla^2\phi(x) = \sum_{i=1}^{m} \frac{1}{s_i(x)^2} a_i a_i^T = A^T D_x^2 A$$

with $D_x = \mathrm{diag}(d_x)$.

Proof for Gradient

$$
\begin{aligned}
\frac{\partial \phi(x)}{\partial x_i} &= \frac{\partial}{\partial x_i}\Big(-\sum_r \ln(s_r(x)) \Big) \\
&= -\sum_r \frac{\partial}{\partial x_i}\Big(\ln(s_r(x)) \Big) = -\sum_r \frac{1}{s_r(x)} \frac{\partial}{\partial x_i}\Big(s_r(x) \Big) \\
&= -\sum_r \frac{1}{s_r(x)} \frac{\partial}{\partial x_i}\Big(b_r - a_r^T x \Big) = \sum_r \frac{1}{s_r(x)} \frac{\partial}{\partial x_i}\Big(a_r^T x \Big) \\
&= \sum_r \frac{1}{s_r(x)} A_{ri}
\end{aligned}
$$

The i-th entry of the gradient vector is $\sum_r 1/s_r(x) \cdot A_{ri}$. This gives that the gradient is

$$
\nabla \phi(x) = \sum_r 1/s_r(x) a_r = A^T d_x
$$

Proof for Hessian

$$\frac{\partial}{\partial x_j}\left(\sum_r \frac{1}{s_r(x)} A_{ri}\right) = \sum_r A_{ri}\left(-\frac{1}{s_r(x)^2}\right) \cdot \frac{\partial}{\partial x_j}\left(s_r(x)\right)$$
$$= \sum_r A_{ri} \frac{1}{s_r(x)^2} A_{rj}$$

Note that $\sum_r A_{ri} A_{rj} = (A^T A)_{ij}$. Adding the additional factors $1/s_r(x)^2$ can be done with a diagonal matrix.

Hence the Hessian is

$$H_x = A^T D^2 A$$

Properties of the Hessian

H_x is positive semi-definite for $x \in P^\circ$

$$u^T H_x u = u^T A^T D_x^2 A u = \|D_x A u\|_2^2 \geq 0$$

This gives that $\phi(x)$ is convex.

If $\mathrm{rank}(A) = n$, H_x is positive definite for $x \in P^\circ$

$$u^T H_x u = \|D_x A u\|_2^2 > 0 \text{ for } u \neq 0$$

This gives that $\phi(x)$ is strictly convex.

$\|u\|_{H_x} := \sqrt{u^T H_x u}$ is a (semi-)norm; the unit ball w.r.t. this norm is an ellipsoid.

Dikin Ellipsoid

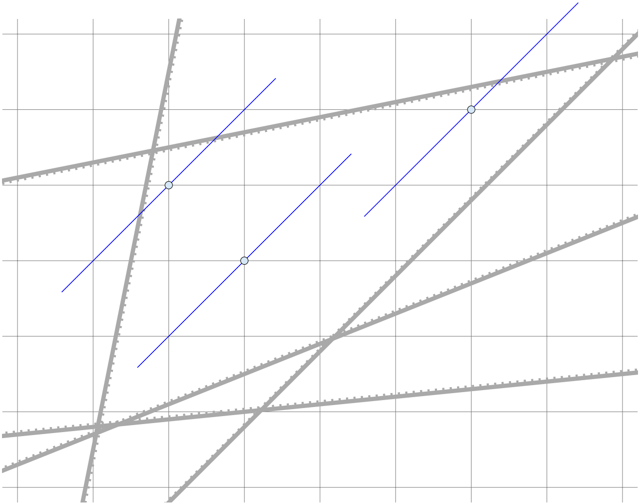
$$E_x = \{y \mid (y-x)^T H_x (y-x) \le 1\} = \{y \mid \|y-x\|_{H_x} \le 1\}$$

Points in E_x are feasible!!!

$$\begin{aligned}
(y-x)^T H_x (y-x) &= (y-x)^T A^T D_x^2 A (y-x) \\
&= \sum_{i=1}^{m} \frac{(a_i^T (y-x))^2}{s_i(x)^2} \\
&= \sum_{i=1}^{m} \frac{(\text{change of distance to } i\text{-th constraint going from } x \text{ to } y)^2}{(\text{distance of } x \text{ to } i\text{-th constraint})^2} \\
&\le 1
\end{aligned}$$

In order to become infeasible when going from x to y one of the terms in the sum would need to be larger than 1.

Dikin Ellipsoids



Analytic Center

$$x_{\mathrm{ac}} := \arg\min_{x \in P^\circ} \phi(x)$$

- x_{ac} is solution to

$$\nabla \phi(x) = \sum_{i=1}^{m} \frac{1}{s_i(x)} a_i = 0$$

- depends on the description of the polytope
- x_{ac} exists and is unique iff P° is nonempty and bounded

Central Path

In the following we assume that the LP and its dual are strictly feasible and that $\mathrm{rank}(A) = n$.

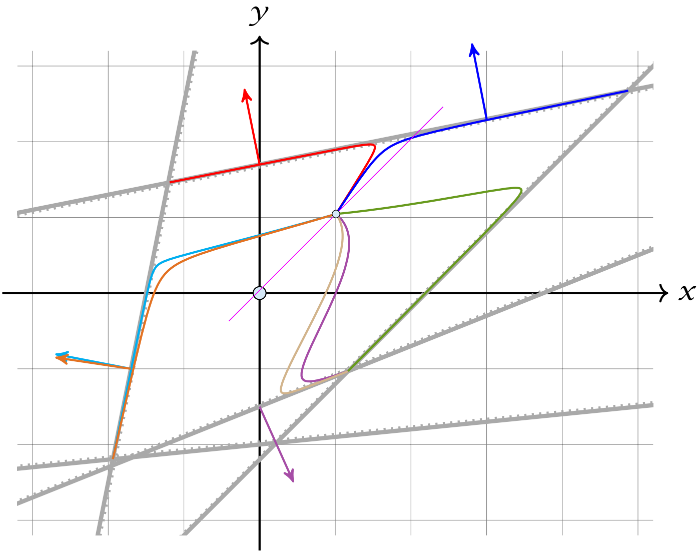
Central Path:
Set of points $\{x^*(t) \mid t > 0\}$ with

$$x^*(t) = \mathrm{argmin}_x \{tc^T x + \phi(x)\}$$

- $t = 0$: analytic center
- $t = \infty$: optimum solution

$x^*(t)$ exists and is unique for all $t \geq 0$.

Different Central Paths



Central Path

Intuitive Idea:
Find point on central path for large value of t. Should be close to optimum solution.

Questions:

- Is this really true? How large a t do we need?
- How do we find corresponding point $x^*(t)$ on central path?

The Dual

primal-dual pair:

$$\begin{array}{ll} \text{min} & c^T x \\ \text{s.t.} & Ax \le b \end{array}$$

$$\begin{array}{ll} \text{max} & -b^T z \\ \text{s.t.} & A^T z + c = 0 \\ & z \ge 0 \end{array}$$

Assumptions

- primal and dual problems are strictly feasible;
- $\text{rank}(A) = n$.

Note that the right LP in standard form is equal to $\max\{-b^T y \mid -A^T y = c, x \ge 0\}$. The dual of this is $\min\{c^T x \mid -Ax \ge -b\}$ (variables x are unrestricted).

Force Field Interpretation

Point $x^*(t)$ on central path is solution to $tc + \nabla\phi(x) = 0$

- We can view each constraint as generating a repelling force. The combination of these forces is represented by $\nabla\phi(x)$.
- In addition there is a force tc pulling us towards the optimum solution.

The "gravitational force" actually pulls us in direction $-\nabla\Phi(x)$. We are minimizing, hence, optimizing in direction $-c$.

How large should t be?

Point $x^*(t)$ on central path is solution to $tc + \nabla\phi(x) = 0$.

This means

$$tc + \sum_{i=1}^{m} \frac{1}{s_i(x^*(t))} a_i = 0$$

or

$$c + \sum_{i=1}^{m} z_i^*(t) a_i = 0 \quad \text{with} \quad z_i^*(t) = \frac{1}{t s_i(x^*(t))}$$

- $z^*(t)$ is strictly dual feasible: ($A^T z^* + c = 0$; $z^* > 0$)
- duality gap between $x := x^*(t)$ and $z := z^*(t)$ is

$$c^T x + b^T z = (b - Ax)^T z = \frac{m}{t}$$

- if gap is less than $1/2^{\Omega(L)}$ we can snap to optimum point

How to find $x^*(t)$

First idea:

- start somewhere in the polytope
- use iterative method (Newtons method) to minimize $f_t(x) := tc^T x + \phi(x)$

Newton Method

Quadratic approximation of f_t

$$f_t(x+\epsilon) \approx f_t(x) + \nabla f_t(x)^T \epsilon + \frac{1}{2}\epsilon^T H_{f_t}(x)\,\epsilon$$

Suppose this were exact:

$$f_t(x+\epsilon) = f_t(x) + \nabla f_t(x)^T \epsilon + \frac{1}{2}\epsilon^T H_{f_t}(x)\,\epsilon$$

Then gradient is given by:

$$\nabla f_t(x+\epsilon) = \nabla f_t(x) + H_{f_t}(x) \cdot \epsilon$$

Note that for the one-dimensional case $g(\epsilon) = f(x) + f'(x)\epsilon + \frac{1}{2}f''(x)\epsilon^2$, then $g'(\epsilon) = f'(x) + f''(x)\epsilon$.

Newton Method

Observe that $H_{f_t}(x) = H(x)$, where $H(x)$ is the Hessian for the function $\phi(x)$ (adding a linear term like tc^Tx does not affect the Hessian).

Also $\nabla f_t(x) = tc + \nabla\phi(x)$.

We want to move to a point where this gradient is 0:

Newton Step at $x \in P^\circ$

$$\begin{aligned}\Delta x_{\text{nt}} &= -H_{f_t}^{-1}(x)\nabla f_t(x)\\ &= -H_{f_t}^{-1}(x)(tc + \nabla\phi(x))\\ &= -(A^TD_x^2A)^{-1}(tc + A^Td_x)\end{aligned}$$

Newton Iteration:

$$x := x + \Delta x_{\text{nt}}$$

Measuring Progress of Newton Step

Newton decrement:

$$\begin{aligned}\lambda_t(x) &= \|D_x A \Delta x_{\text{nt}}\| \\ &= \|\Delta x_{\text{nt}}\|_{H_x}\end{aligned}$$

Square of Newton decrement is linear estimate of reduction if we do a Newton step:

$$-\lambda_t(x)^2 = \nabla f_t(x)^T \Delta x_{\text{nt}}$$

- $\lambda_t(x) = 0$ iff $x = x^*(t)$
- $\lambda_t(x)$ is measure of proximity of x to $x^*(t)$

Recall that Δx_{nt} fulfills $-H(x)\Delta x_{\text{nt}} = \nabla f_t(x)$.

Convergence of Newtons Method

Theorem 55
If $\lambda_t(x) < 1$ *then*

- $x_+ := x + \Delta x_{nt} \in P^\circ$ *(new point feasible)*
- $\lambda_t(x_+) \leq \lambda_t(x)^2$

This means we have quadratic convergence. Very fast.

Convergence of Newtons Method

feasibility:

- $\lambda_t(x) = \|\Delta x_{\mathsf{nt}}\|_{H_x} < 1$; hence x_+ lies in the Dikin ellipsoid around x.

Convergence of Newtons Method

bound on $\lambda_t(x^+)$:
we use $D := D_x = \mathrm{diag}(d_x)$ and $D_+ := D_{x^+} = \mathrm{diag}(d_{x^+})$

$$\begin{aligned}\lambda_t(x^+)^2 &= \|D_+ A\Delta x_{\mathsf{nt}}^+\|^2 \\ &\le \|D_+ A\Delta x_{\mathsf{nt}}^+\|^2 + \|D_+ A\Delta x_{\mathsf{nt}}^+ + (I - D_+^{-1}D)DA\Delta x_{\mathsf{nt}}\|^2 \\ &= \|(I - D_+^{-1}D)DA\Delta x_{\mathsf{nt}}\|^2\end{aligned}$$

To see the last equality we use Pythagoras

$$\|a\|^2 + \|a+b\|^2 = \|b\|^2$$

if $a^T(a+b) = 0$.

Convergence of Newtons Method

$$
\begin{aligned}
DA\Delta x_{\text{nt}} &= DA(x^+ - x) \\
&= D(b - Ax - (b - Ax^+)) \\
&= D(D^{-1}\vec{1} - D_+^{-1}\vec{1}) \\
&= (I - D_+^{-1}D)\vec{1}
\end{aligned}
$$

$$
\begin{aligned}
&a^T(a+b) \\
&\quad = \Delta x_{\text{nt}}^{+T} A^T D_+ \Big(D_+ A \Delta x_{\text{nt}}^+ + (I - D_+^{-1}D) DA\Delta x_{\text{nt}} \Big) \\
&\quad = \Delta x_{\text{nt}}^{+T} \Big(A^T D_+^2 A \Delta x_{\text{nt}}^+ - A^T D^2 A \Delta x_{\text{nt}} + A^T D_+ D A \Delta x_{\text{nt}} \Big) \\
&\quad = \Delta x_{\text{nt}}^{+T} \Big(H_+ \Delta x_{\text{nt}}^+ - H \Delta x_{\text{nt}} + A^T D_+ \vec{1} - A^T D \vec{1} \Big) \\
&\quad = \Delta x_{\text{nt}}^{+T} \Big(-\nabla f_t(x^+) + \nabla f_t(x) + \nabla \phi(x^+) - \nabla \phi(x) \Big) \\
&\quad = 0
\end{aligned}
$$

Convergence of Newtons Method

bound on $\lambda_t(x^+)$:
we use $D := D_x = \mathrm{diag}(d_x)$ and $D_+ := D_{x^+} = \mathrm{diag}(d_{x^+})$

$$
\begin{aligned}
\lambda_t(x^+)^2 &= \|D_+ A\Delta x_{\mathrm{nt}}^+\|^2 \\
&\le \|D_+ A\Delta x_{\mathrm{nt}}^+\|^2 + \|D_+ A\Delta x_{\mathrm{nt}}^+ + (I - D_+^{-1}D)DA\Delta x_{\mathrm{nt}}\|^2 \\
&= \|(I - D_+^{-1}D)DA\Delta x_{\mathrm{nt}}\|^2 \\
&= \|(I - D_+^{-1}D)^2\vec{1}\|^2 \\
&\le \|(I - D_+^{-1}D)\vec{1}\|^4 \\
&= \|DA\Delta x_{\mathrm{nt}}\|^4 \\
&= \lambda_t(x)^4
\end{aligned}
$$

The second inequality follows from $\sum_i y_i^4 \le \left(\sum_i y_i^2\right)^2$

If $\lambda_t(x)$ is large we do not have a guarantee.

Try to avoid this case!!!

Path-following Methods

Try to slowly travel along the central path.

Algorithm 1 PathFollowing

1: start at analytic center

2: **while** solution not good enough **do**

3: make step to improve objective function

4: recenter to return to central path

Short Step Barrier Method

simplifying assumptions:

- a first central point $x^*(t_0)$ is given
- $x^*(t)$ is computed exactly in each iteration

ϵ is approximation we are aiming for

start at $t = t_0$, repeat until $m/t \leq \epsilon$

- compute $x^*(\mu t)$ using Newton starting from $x^*(t)$
- $t := \mu t$

where $\mu = 1 + 1/(2\sqrt{m})$

Short Step Barrier Method

gradient of f_{t^+} at ($x = x^*(t)$)

$$\begin{aligned}\nabla f_{t^+}(x) &= \nabla f_t(x) + (\mu - 1)tc \\ &= -(\mu - 1)A^T D_x \vec{1}\end{aligned}$$

This holds because $0 = \nabla f_t(x) = tc + A^T D_x \vec{1}$.

The Newton decrement is

$$\begin{aligned}\lambda_{t^+}(x)^2 &= \nabla f_{t^+}(x)^T H^{-1} \nabla f_{t^+}(x) \\ &= (\mu - 1)^2 \vec{1}^T B (B^T B)^{-1} B^T \vec{1} \qquad B = D_x^T A \\ &\le (\mu - 1)^2 m \\ &= 1/4\end{aligned}$$

This means we are in the range of quadratic convergence!!!

Number of Iterations

the number of Newton iterations per outer iteration is very small; in practise only 1 or 2

Number of outer iterations:

We need $t_k = \mu^k t_0 \geq m/\epsilon$. This holds when

$$k \geq \frac{\log(m/(\epsilon t_0))}{\log(\mu)}$$

We get a bound of

$$\mathcal{O}\left(\sqrt{m}\log\frac{m}{\epsilon t_0}\right)$$

We show how to get a starting point with $t_0 = 1/2^L$. Together with $\epsilon \approx 2^{-L}$ we get $\mathcal{O}(L\sqrt{m})$ iterations.

Explanation for previous slide

$P = B(B^TB)^{-1}B^T$ is a symmetric real-valued matrix; it has n linearly independent Eigenvectors. Since it is a projection matrix ($P^2 = P$) it can only have Eigenvalues 0 and 1 (because the Eigenvalues of P^2 are λ_i^2, where λ_i is Eigenvalue of P).

The expression

$$\max_v \frac{v^TPv}{v^Tv}$$

gives the largest Eigenvalue for P. Hence, $\vec{1}^TP\vec{1} \leq \vec{1}^T\vec{1} = m$

Damped Newton Method

We assume that the polytope (not just the LP) is bounded. Then $Av \leq 0$ is not possible.

For $x \in P^\circ$ and direction $v \neq 0$ define

$$\sigma_x(v) := \max_i \frac{a_i^T v}{s_i(x)}$$

$a_i^T v$ is the change on the left hand side of the i-th constraint when moving in direction of v.

If $\sigma_x(v) > 1$ then for one coordinate this change is larger than the slack in the constraint at position x.

By downscaling v we can ensure to stay in the polytope.

Observation:

$$x + \alpha v \in P \qquad \text{for } \alpha \in \{0, 1/\sigma_x(v)\}$$

Damped Newton Method

Suppose that we move from x to $x + \alpha v$. The linear estimate says that $f_t(x)$ should change by $\nabla f_t(x)^T \alpha v$.

The following argument shows that f_t is well behaved. For small α the reduction of $f_t(x)$ is close to linear estimate.

$$f_t(x + \alpha v) - f_t(x) = tc^T \alpha v + \phi(x + \alpha v) - \phi(x)$$

$$\begin{aligned} \phi(x + \alpha v) - \phi(x) &= -\sum_i \log(s_i(x + \alpha v)) + \sum_i \log(s_i(x)) \\ &= -\sum_i \log(s_i(x + \alpha v)/s_i(x)) \\ &= -\sum_i \log(1 - a_i^T \alpha v / s_i(x)) \end{aligned}$$

$$s_i(x + \alpha v) = b_i - a_i^T x - a_i^T \alpha v = s_i(x) - a_i^T \alpha v$$

Damped Newton Method

$\nabla f_t(x)^T \alpha v$
$= \left(tc^T + \sum_i a_i^T / s_i(x)\right) \alpha v$
$= tc^T \alpha v + \sum_i \alpha w_i$

Define $w_i = a_i^T v / s_i(x)$ and $\sigma = \max_i w_i$. Then

Note that $\|w\| = \|v\|_{H_x}$.

$$
\begin{aligned}
f_t(x + \alpha v) &- f_t(x) - \nabla f_t(x)^T \alpha v \\
&= -\sum\nolimits_i (\alpha w_i + \log(1 - \alpha w_i)) \\
&\le -\sum_{w_i > 0} (\alpha w_i + \log(1 - \alpha w_i)) + \sum_{w_i \le 0} \frac{\alpha^2 w_i^2}{2} \\
&\le -\sum_{w_i > 0} \frac{w_i^2}{\sigma^2} \Big(\alpha\sigma + \log(1 - \alpha\sigma)\Big) + \frac{(\alpha\sigma)^2}{2} \sum_{w_i \le 0} \frac{w_i^2}{\sigma^2}
\end{aligned}
$$

For $|x| < 1$, $x \le 0$:

$$x + \log(1 - x) = -\frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} - \cdots \ge -\frac{x^2}{2} = -\frac{y^2}{2}\frac{x^2}{y^2}$$

For $|x| < 1$, $0 < x \le y$:

$$
\begin{aligned}
x + \log(1 - x) &= -\frac{x^2}{2} - \frac{x^3}{3} - \frac{x^4}{4} - \cdots = \frac{x^2}{y^2}\left(-\frac{y^2}{2} - \frac{y^2 x}{3} - \frac{y^2 x^2}{4} - \ldots\right) \\
&\ge \frac{x^2}{y^2}\left(-\frac{y^2}{2} - \frac{y^3}{3} - \frac{y^4}{4} - \ldots\right) = \frac{x^2}{y^2}(y + \log(1 - y))
\end{aligned}
$$

Damped Newton Method

For $x \geq 0$
$$\frac{x^2}{2} \leq \frac{x^2}{2} + \frac{x^3}{3} + \frac{x^4}{4} + \cdots = -\Big(x + \log(1-x)\Big)$$

$$\begin{aligned} &\leq -\sum_i \frac{w_i^2}{\sigma^2}\Big(\alpha\sigma + \log(1-\alpha\sigma)\Big) \\ &= -\frac{1}{\sigma^2}\|v\|_{H_x}^2\Big(\alpha\sigma + \log(1-\alpha\sigma)\Big) \end{aligned}$$

Damped Newton Iteration:

In a damped Newton step we choose

$$x_+ = x + \frac{1}{1+\sigma_x(\Delta x_{\text{nt}})}\Delta x_{\text{nt}}$$

This means that in the above expressions we choose $\alpha = \frac{1}{1+\sigma}$ and $v = \Delta x_{\text{nt}}$. Note that it wouldn't make sense to choose α larger than 1 as this would mean that our real target $(x + \Delta x_{\text{nt}})$ is inside the polytope but we overshoot and go further than this target.

Damped Newton Method

Theorem:
In a damped Newton step the cost decreases by at least

$$\lambda_t(x) - \log(1 + \lambda_t(x))$$

Proof: The decrease in cost is

$$-\alpha \nabla f_t(x)^T v + \frac{1}{\sigma^2}\|v\|_{H_x}^2 (\alpha\sigma + \log(1 - \alpha\sigma))$$

Choosing $\alpha = \frac{1}{1+\sigma}$ and $v = \Delta x_{\text{nt}}$ gives

$$\frac{1}{1+\sigma}\lambda_t(x)^2 + \frac{\lambda_t(x)^2}{\sigma^2}\left(\frac{\sigma}{1+\sigma} + \log\left(1 - \frac{\sigma}{1+\sigma}\right)\right)$$
$$= \frac{\lambda_t(x)^2}{\sigma^2}\left(\sigma - \log(1+\sigma)\right)$$

With $v = \Delta x_{\text{nt}}$ we have $\|w\|_2 = \|v\|_{H_x} = \lambda_t(x)$; further recall that $\sigma = \|w\|_\infty$; hence $\sigma \leq \lambda_t(x)$.

The first inequality follows since the function $\frac{1}{x^2}(x - \log(1+x))$ is monotonically decreasing.

$$\geq \lambda_t(x) - \log(1 + \lambda_t(x))$$
$$\geq 0.09$$

for $\lambda_t(x) \geq 0.5$

Centering Algorithm:

Input: precision δ; starting point x

1. compute Δx_{nt} and $\lambda_t(x)$
2. if $\lambda_t(x) \leq \delta$ return x
3. set $x := x + \alpha\Delta x_{\text{nt}}$ with

$$\alpha = \begin{cases} \frac{1}{1+\sigma_x(\Delta x_{\text{nt}})} & \lambda_t \geq 1/2 \\ 1 & \text{otw.} \end{cases}$$

Centering

Lemma 56
The centering algorithm starting at x_0 reaches a point with $\lambda_t(x) \leq \delta$ after

$$\frac{f_t(x_0) - \min_y f_t(y)}{0.09} + \mathcal{O}(\log\log(1/\delta))$$

iterations.

This can be very, very slow...

How to get close to analytic center?

Let $P = \{Ax \leq b\}$ be our (feasible) polyhedron, and x_0 a feasible point.

We change $b \to b + \frac{1}{\lambda} \cdot \vec{1}$, where $L = \langle A\rangle + \langle b\rangle + \langle c\rangle$ (encoding length) and $\lambda = 2^{2L}$. Recall that a basis is feasible in the old LP iff it is feasible in the new LP.

Lemma [without proof]
The inverse of a matrix M can be represented with rational numbers that have denominators $z_{ij} = \det(M)$.

For two basis solutions x_B, $x_{\bar{B}}$, the cost-difference $c^T x_B - c^T x_{\bar{B}}$ can be represented by a rational number that has denominator $z = \det(A_B) \cdot \det(A_{\bar{B}})$.

This means that in the perturbed LP it is sufficient to decrease the duality gap to $1/2^{4L}$ (i.e., $t \approx 2^{4L}$). This means the previous analysis essentially also works for the perturbed LP.

For a point x from the polytope (not necessarily BFS) the objective value $\bar{c}^T x$ is at most $n 2^M 2^L$, where $M \le L$ is the encoding length of the largest entry in $\bar{c}$.

How to get close to analytic center?

Note that an entry in $\hat{c}$ fulfills $|\hat{c}_i| \leq 2^{2L}$. This holds since the slack in every constraint at x_0 is at least $\lambda = 1/2^{2L}$, and the gradient is the vector of inverse slacks.

Start at x_0.

Choose $\hat{c} := -\nabla\phi(x)$.

$x_0 = x^*(1)$ is point on central path for $\hat{c}$ and $t = 1$.

You can travel the central path in both directions. Go towards 0 until $t \approx 1/2^{\Omega(L)}$. This requires $O(\sqrt{m}L)$ outer iterations.

Let $x_{\hat{c}}$ denote this point.

Let x_c denote the point that minimizes

$$t \cdot c^T x + \phi(x)$$

(i.e., same value for t but different c, hence, different central path).

How to get close to analytic center?

Clearly,

$$t \cdot \hat{c}^T x_{\hat{c}} + \phi(x_{\hat{c}}) \le t \cdot \hat{c}^T x_c + \phi(x_c)$$

The difference between $f_t(x_{\hat{c}})$ and $f_t(x_c)$ is

$$\begin{aligned} tc^T x_{\hat{c}} + \phi(x_{\hat{c}}) - tc^T x_c - \phi(x_c) & \\ & \le t(c^T x_{\hat{c}} + \hat{c}^T x_c - \hat{c}^T x_{\hat{c}} - c^T x_c) \\ & \le 4tn2^{3L} \end{aligned}$$

For $t = 1/2^{\Omega(L)}$ the last term becomes constant. Hence, using damped Newton we can move from $x_{\hat{c}}$ to x_c quickly.

In total for this analysis we require $\mathcal{O}(\sqrt{m}L)$ outer iterations for the whole algorithm.

One iteration can be implemented in $\tilde{\mathcal{O}}(m^3)$ time.