## Complexity

**LP Feasibility Problem (LP feasibility)**
Given $A \in \mathbb{Z}^{m \times n}$, $b \in \mathbb{Z}^m$. Does there exist $x \in \mathbb{R}$ with $Ax = b$, $x \geq 0$?

Note that allowing $A, b$ to contain rational numbers does not make a difference, as we can multiply every number by a suitable large constant so that everything becomes integral but the feasible region does not change.

---

## The Bit Model

**Input size**

- The number of bits to represent a number $a \in \mathbb{Z}$ is

$$\lceil \log_2(|a|) \rceil + 1$$

- Let for an $m \times n$ matrix $M$, $L(M)$ denote the number of bits required to encode all the numbers in $M$.

$$\langle M \rangle := \sum_{i,j} \lceil \log_2(|m_{ij}|) + 1 \rceil$$

- In the following we assume that input matrices are encoded in a standard way, where each number is encoded in binary and then suitable separators are added in order to separate distinct number from each other.
- Then the input length is $L = \Theta(\langle A \rangle + \langle b \rangle)$.

---

- In the following we sometimes refer to $L := \langle A \rangle + \langle b \rangle$ as the input size (even though the real input size is something in $\Theta(\langle A \rangle + \langle b \rangle)$).
- In order to show that LP-decision is in NP we show that if there is a solution $x$ then there exists a small solution for which feasibility can be verified in polynomial time (polynomial in $L$).

---

Suppose that $Ax = b;\ x \geq 0$ is feasible.

Then there exists a basic feasible solution. This means a set $B$ of basic variables such that

$$x_B = A_B^{-1} b$$

and all other entries in $x$ are $0$.

In the following we show that this $x$ has small encoding length and we give an explicit bound on this length. So far we have only been handwaving and have said that we can compute $x$ via Gaussian elimination and it will be short...

## Size of a Basic Feasible Solution

**Lemma 2**

*Let $M \in \mathbb{Z}^{m\times m}$ be an invertible matrix and let $b \in \mathbb{Z}^m$. Further define $L = \langle M\rangle + \langle b\rangle + n\log_2 n$. Then a solution to $Mx = b$ has rational components $x_j$ of the form $\frac{D_j}{D}$, where $|D_j| \le 2^L$ and $|D| \le 2^L$.*

**Proof:**

Cramers rules says that we can compute $x_j$ as

$$x_j = \frac{\det(M_j)}{\det(M)}$$

where $M_j$ is the matrix obtained from $M$ by replacing the $j$-th column by the vector $b$.

---

## Bounding the Determinant

Let $X = A_B$. Then

$$\begin{aligned}|\det(X)| &= \left|\sum_{\pi\in S_n} \operatorname{sgn}(\pi) \prod_{1\le i\le n} X_{i\pi(i)}\right| \\ &\le \sum_{\pi\in S_n} \prod_{1\le i\le n} |X_{i\pi(i)}| \\ &\le n! \cdot 2^{\langle A\rangle+\langle b\rangle} \le 2^L \ .\end{aligned}$$

Analogously for $\det(M_j)$.

---

## Reducing LP-solving to LP decision.

Given an LP $\max\{c^Tx \mid Ax = b; x \ge 0\}$ do a binary search for the optimum solution

(Add constraint $c^Tx - \delta = M$; $\delta \ge 0$ or ($c^Tx \ge M$). Then checking for feasibility shows whether optimum solution is larger or smaller than $M$).

If the LP is feasible then the binary search finishes in at most

$$\log_2\left(\frac{2n2^{2L'}}{1/2^{L'}}\right) = \mathcal{O}(L')\ ,$$

as the range of the search is at most $-n2^{2L'},\dots,n2^{2L'}$ and the distance between two adjacent values is at least $\frac{1}{\det(A)} \ge \frac{1}{2^{L'}}$.

Here we use $L' = \langle A\rangle + \langle b\rangle + \langle c\rangle + n\log_2 n$ (it also includes the encoding size of $c$).

---

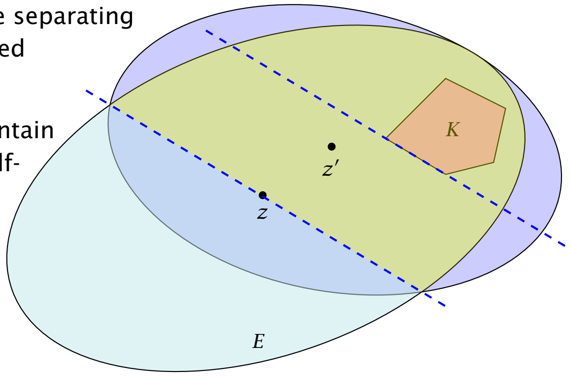
**How do we detect whether the LP is unbounded?**

Let $M_{\max} = n2^{2L'}$ be an upper bound on the objective value of a basic feasible solution.

We can add a constraint $c^Tx \ge M_{\max} + 1$ and check for feasibility.

# Ellipsoid Method

- Let $K$ be a convex set.
- Maintain ellipsoid $E$ that is guaranteed to contain $K$ provided that $K$ is non-empty.
- If center $z \in K$ STOP.
- Otw. find a hyperplane separating $K$ from $z$ (e.g. a violated constraint in the LP).
- Shift hyperplane to contain node $z$. $H$ denotes half-space that contains $K$.
- Compute (smallest) ellipsoid $E'$ that contains $E \cap H$.
- REPEAT



---

**Issues/Questions:**

- How do you choose the first Ellipsoid? What is its volume?
- How do you measure progress? By how much does the volume decrease in each iteration?
- When can you stop? What is the minimum volume of a non-empty polytop?

---

**Definition 3**

A mapping $f : \mathbb{R}^n \to \mathbb{R}^n$ with $f(x) = Lx + t$, where $L$ is an invertible matrix is called an affine transformation.

---

**Definition 4**

A ball in $\mathbb{R}^n$ with center $c$ and radius $r$ is given by

$$\begin{aligned} B(c,r) &= \{x \mid (x-c)^T(x-c) \le r^2\} \\ &= \{x \mid \sum_i (x-c)_i^2/r^2 \le 1\} \end{aligned}$$

$B(0,1)$ is called the unit ball.

**Definition 5**

An affine transformation of the unit ball is called an ellipsoid.

From $f(x) = Lx + t$ follows $x = L^{-1}(f(x) - t)$.

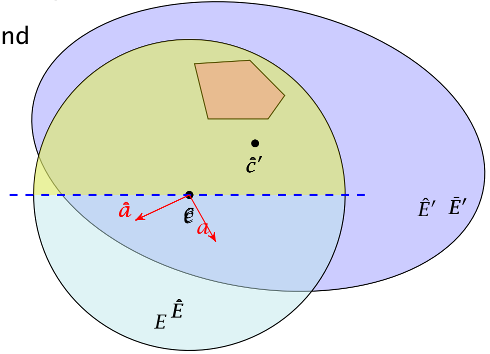
$$\begin{aligned} f(B(0,1)) &= \{f(x) \mid x \in B(0,1)\} \\ &= \{y \in \mathbb{R}^n \mid L^{-1}(y-t) \in B(0,1)\} \\ &= \{y \in \mathbb{R}^n \mid (y-t)^T {L^{-1}}^T L^{-1}(y-t) \le 1\} \\ &= \{y \in \mathbb{R}^n \mid (y-t)^T Q^{-1}(y-t) \le 1\} \end{aligned}$$

where $Q = LL^T$ is an invertible matrix.

---

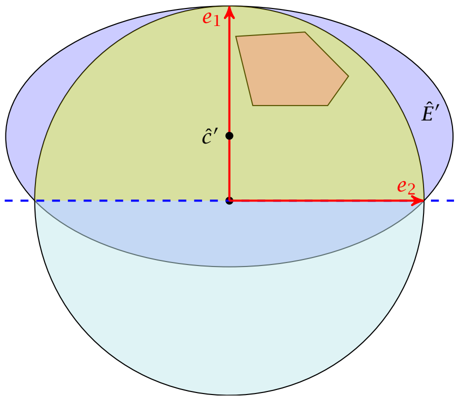
## How to Compute the New Ellipsoid

- Use $f^{-1}$ (recall that $f = Lx + t$ is the affine transformation of the unit ball) to rotate/distort the ellipsoid (back) into the unit ball.
- Use a rotation $R^{-1}$ to rotate the unit ball such that the normal vector of the halfspace is parallel to $e_1$.
- Compute the new center $\hat{c}'$ and the new matrix $\hat{Q}'$ for this simplified setting.
- Use the transformations $R$ and $f$ to get the new center $c'$ and the new matrix $Q'$ for the original ellipsoid $E$.



---

## The Easy Case



- The new center lies on axis $x_1$. Hence, $\hat{c}' = te_1$ for $t > 0$.
- The vectors $e_1, e_2, \dots$ have to fulfill the ellipsoid constraint with equality. Hence $(e_i - \hat{c}')^T \hat{Q}'^{-1}(e_i - \hat{c}') = 1$.

---

## The Easy Case

- To obtain the matrix $\hat{Q}'^{-1}$ for our ellipsoid $\hat{E}'$ note that $\hat{E}'$ is axis-parallel.
- Let $a$ denote the radius along the $x_1$-axis and let $b$ denote the (common) radius for the other axes.
- The matrix

$$\hat{L}' = \begin{pmatrix} a & 0 & \dots & 0 \\ 0 & b & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & b \end{pmatrix}$$

maps the unit ball (via function $\hat{f}'(x) = \hat{L}'x$) to an axis-parallel ellipsoid with radius $a$ in direction $x_1$ and $b$ in all other directions.

## The Easy Case

- As $\hat{Q}' = \hat{L}'\hat{L}'^t$ the matrix $\hat{Q}'^{-1}$ is of the form

$$\hat{Q}'^{-1} = \begin{pmatrix} \frac{1}{a^2} & 0 & \dots & 0 \\ 0 & \frac{1}{b^2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{b^2} \end{pmatrix}$$

---

## The Easy Case

- $(e_1 - \hat{c}')^T \hat{Q}'^{-1} (e_1 - \hat{c}') = 1$ gives

$$\begin{pmatrix} 1-t \\ 0 \\ \vdots \\ 0 \end{pmatrix}^T \cdot \begin{pmatrix} \frac{1}{a^2} & 0 & \dots & 0 \\ 0 & \frac{1}{b^2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{b^2} \end{pmatrix} \cdot \begin{pmatrix} 1-t \\ 0 \\ \vdots \\ 0 \end{pmatrix} = 1$$

- This gives $(1-t)^2 = a^2$.

---

## The Easy Case

- For $i \neq 1$ the equation $(e_i - \hat{c}')^T \hat{Q}'^{-1} (e_i - \hat{c}') = 1$ looks like (here $i = 2$)

$$\begin{pmatrix} -t \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}^T \cdot \begin{pmatrix} \frac{1}{a^2} & 0 & \dots & 0 \\ 0 & \frac{1}{b^2} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \frac{1}{b^2} \end{pmatrix} \cdot \begin{pmatrix} -t \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = 1$$

- This gives $\frac{t^2}{a^2} + \frac{1}{b^2} = 1$, and hence

$$\frac{1}{b^2} = 1 - \frac{t^2}{a^2} = 1 - \frac{t^2}{(1-t)^2} = \frac{1-2t}{(1-t)^2}$$

---

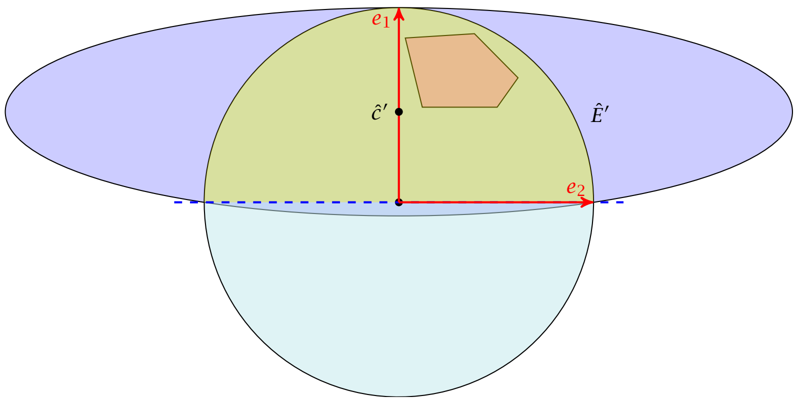
## Summary

So far we have

$$a = 1 - t \quad \text{and} \quad b = \frac{1-t}{\sqrt{1-2t}}$$

# The Easy Case

We still have many choices for $t$:



Choose $t$ such that the volume of $\hat{E}'$ is minimal!!!

---

# The Easy Case

We want to choose $t$ such that the volume of $\hat{E}'$ is minimal.

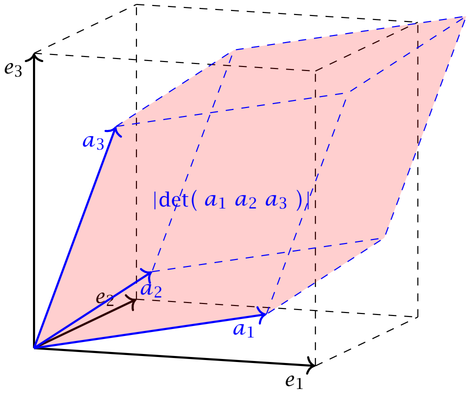
**Lemma 6**

*Let $L$ be an affine transformation and $K \subseteq \mathbb{R}^n$. Then*

$$\mathrm{vol}(L(K)) = |\det(L)| \cdot \mathrm{vol}(K) \enspace .$$

---

# n-dimensional volume



---

# The Easy Case

- We want to choose $t$ such that the volume of $\hat{E}'$ is minimal.

$$\mathrm{vol}(\hat{E}') = \mathrm{vol}(B(0,1)) \cdot |\det(\hat{L}')| \enspace ,$$

- Recall that

$$\hat{L}' = \begin{pmatrix} a & 0 & \dots & 0 \\ 0 & b & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & b \end{pmatrix}$$

- Note that $a$ and $b$ in the above equations depend on $t$, by the previous equations.

# The Easy Case

$$\begin{aligned}\mathrm{vol}(\hat{E}') &= \mathrm{vol}(B(0,1)) \cdot |\det(\hat{L}')| \\ &= \mathrm{vol}(B(0,1)) \cdot ab^{n-1} \\ &= \mathrm{vol}(B(0,1)) \cdot (1-t) \cdot \left(\frac{1-t}{\sqrt{1-2t}}\right)^{n-1} \\ &= \mathrm{vol}(B(0,1)) \cdot \frac{(1-t)^n}{(\sqrt{1-2t})^{n-1}}\end{aligned}$$

We use the shortcut $\Phi := \mathrm{vol}(B(0,1))$.

---

# The Easy Case

$$\begin{aligned}\frac{\mathrm{d}\,\mathrm{vol}(\hat{E}')}{\mathrm{d}\,t} &= \frac{\mathrm{d}}{\mathrm{d}\,t}\left(\Phi \frac{(1-t)^n}{(\sqrt{1-2t})^{n-1}}\right) \\ &= \frac{\Phi}{N^2} \cdot \Bigg( \underbrace{(-1) \cdot n\cancel{(1-t)^{n-1}}}_{\text{derivative of numerator}} \cdot \underbrace{\cancelto{1-2t}{(\sqrt{1-2t})^{n-1}}}_{\text{denominator}} \\ &\quad \cancel{-}\underbrace{(n-1)\cancel{(\sqrt{1-2t})^{n-2}}}_{\text{outer derivative}} \cdot \underbrace{\frac{1}{\cancel{2}\cancel{\sqrt{1-2t}}} \cdot \cancel{(-2)}}_{\text{inner derivative}} \cdot \underbrace{\cancelto{1-t}{(1-t)^n}}_{\text{numerator}} \Bigg) \\ &= \frac{\Phi}{N^2} \cdot (\sqrt{1-2t})^{n-3} \cdot (1-t)^{n-1} \\ &\quad \cdot \Big( (n-1)(1-t) - n(1-2t) \Big) \\ &= \frac{\Phi}{N^2} \cdot (\sqrt{1-2t})^{n-3} \cdot (1-t)^{n-1} \cdot \Big( (n+1)t - 1 \Big)\end{aligned}$$

$N$ = denominator

---

# The Easy Case

- We obtain the minimum for $t = \frac{1}{n+1}$.
- For this value we obtain

$$a = 1 - t = \frac{n}{n+1} \text{ and } b = \frac{1-t}{\sqrt{1-2t}} = \frac{n}{\sqrt{n^2-1}}$$

To see the equation for $b$, observe that

$$b^2 = \frac{(1-t)^2}{1-2t} = \frac{(1-\frac{1}{n+1})^2}{1-\frac{2}{n+1}} = \frac{(\frac{n}{n+1})^2}{\frac{n-1}{n+1}} = \frac{n^2}{n^2-1}$$

---

# The Easy Case

Let $\gamma_n = \frac{\mathrm{vol}(\hat{E}')}{\mathrm{vol}(B(0,1))} = ab^{n-1}$ be the ratio by which the volume changes:

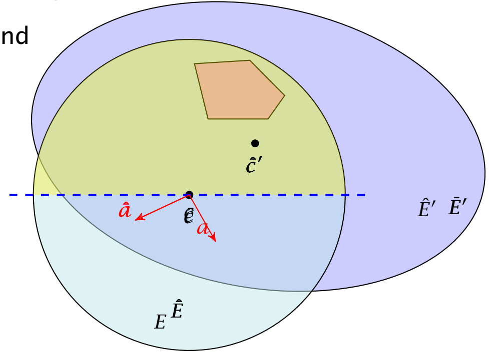
$$\begin{aligned}\gamma_n^2 &= \left(\frac{n}{n+1}\right)^2 \left(\frac{n^2}{n^2-1}\right)^{n-1} \\ &= \left(1 - \frac{1}{n+1}\right)^2 \left(1 + \frac{1}{(n-1)(n+1)}\right)^{n-1} \\ &\le e^{-2\frac{1}{n+1}} \cdot e^{\frac{1}{n+1}} \\ &= e^{-\frac{1}{n+1}}\end{aligned}$$

where we used $(1+x)^a \le e^{ax}$ for $x \in \mathbb{R}$ and $a > 0$.

This gives $\gamma_n \le e^{-\frac{1}{2(n+1)}}$.

# How to Compute the New Ellipsoid

- Use $f^{-1}$ (recall that $f = Lx + t$ is the affine transformation of the unit ball) to rotate/distort the ellipsoid (back) into the unit ball.
- Use a rotation $R^{-1}$ to rotate the unit ball such that the normal vector of the halfspace is parallel to $e_1$.
- Compute the new center $\hat{c}'$ and the new matrix $\hat{Q}'$ for this simplified setting.
- Use the transformations $R$ and $f$ to get the new center $c'$ and the new matrix $Q'$ for the original ellipsoid $E$.



---

**Our progress is the same:**

$$e^{-\frac{1}{2(n+1)}} \geq \frac{\mathrm{vol}(\hat{E}')}{\mathrm{vol}(B(0,1))} = \frac{\mathrm{vol}(\hat{E}')}{\mathrm{vol}(\hat{E})} = \frac{\mathrm{vol}(R(\hat{E}'))}{\mathrm{vol}(R(\hat{E}))}$$
$$= \frac{\mathrm{vol}(\bar{E}')}{\mathrm{vol}(\bar{E})} = \frac{\mathrm{vol}(f(\bar{E}'))}{\mathrm{vol}(f(\bar{E}))} = \frac{\mathrm{vol}(E')}{\mathrm{vol}(E)}$$

Here it is important that mapping a set with affine function $f(x) = Lx + t$ changes the volume by factor $\det(L)$.

---

# The Ellipsoid Algorithm

**How to Compute The New Parameters?**

The transformation function of the (old) ellipsoid: $f(x) = Lx + c$;

The halfspace to be intersected: $H = \{x \mid a^T(x - c) \leq 0\}$;

$$\begin{aligned} f^{-1}(H) &= \{f^{-1}(x) \mid a^T(x - c) \leq 0\} \\ &= \{f^{-1}(f(y)) \mid a^T(f(y) - c) \leq 0\} \\ &= \{y \mid a^T(f(y) - c) \leq 0\} \\ &= \{y \mid a^T(Ly + c - c) \leq 0\} \\ &= \{y \mid (a^TL)y \leq 0\} \end{aligned}$$

This means $\bar{a} = L^Ta$.

The center $\bar{c}$ is of course at the origin.

---

# The Ellipsoid Algorithm

After rotating back (applying $R^{-1}$) the normal vector of the halfspace points in negative $x_1$-direction. Hence,

$$R^{-1}\Big(\frac{L^Ta}{\|L^Ta\|}\Big) = -e_1 \quad \Rightarrow \quad -\frac{L^Ta}{\|L^Ta\|} = R \cdot e_1$$

Hence,

$$\bar{c}' = R \cdot \hat{c}' = R \cdot \frac{1}{n+1} e_1 = -\frac{1}{n+1} \frac{L^Ta}{\|L^Ta\|}$$

$$\begin{aligned} c' &= f(\bar{c}') = L \cdot \bar{c}' + c \\ &= -\frac{1}{n+1} L \frac{L^Ta}{\|L^Ta\|} + c \\ &= c - \frac{1}{n+1} \frac{Qa}{\sqrt{a^TQa}} \end{aligned}$$

For computing the matrix $Q'$ of the new ellipsoid we assume in the following that $\hat{E}', \bar{E}'$ and $E'$ refer to the ellispoids centered in the origin.

---

Recall that

$$\hat{Q}' = \begin{pmatrix} a^2 & 0 & \dots & 0 \\ 0 & b^2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & b^2 \end{pmatrix}$$

This gives

$$\hat{Q}' = \frac{n^2}{n^2-1}\Big(I - \frac{2}{n+1}e_1e_1^T\Big)$$

Note that $e_1e_1^T$ is a matrix $M$ that has $M_{11} = 1$ and all other entries equal to $0$.

because for $a^2 = {}^{n^2}/_{(n+1)^2}$ and $b^2 = {}^{n^2}/_{n^2-1}$

$$\begin{aligned} b^2 - b^2\frac{2}{n+1} &= \frac{n^2}{n^2-1} - \frac{2n^2}{(n-1)(n+1)^2} \\ &= \frac{n^2(n+1) - 2n^2}{(n-1)(n+1)^2} = \frac{n^2(n-1)}{(n-1)(n+1)^2} = a^2 \end{aligned}$$

---

# 9 The Ellipsoid Algorithm

$$\begin{aligned} \bar{E}' &= R(\hat{E}') \\ &= \{R(x) \mid x^T\hat{Q}'^{-1}x \le 1\} \\ &= \{y \mid (R^{-1}y)^T\hat{Q}'^{-1}R^{-1}y \le 1\} \\ &= \{y \mid y^T(R^T)^{-1}\hat{Q}'^{-1}R^{-1}y \le 1\} \\ &= \{y \mid y^T(\underbrace{R\hat{Q}'R^T}_{\bar{Q}'})^{-1}y \le 1\} \end{aligned}$$

---

# 9 The Ellipsoid Algorithm

Hence,

$$\begin{aligned} \bar{Q}' &= R\hat{Q}'R^T \\ &= R \cdot \frac{n^2}{n^2-1}\Big(I - \frac{2}{n+1}e_1e_1^T\Big) \cdot R^T \\ &= \frac{n^2}{n^2-1}\Big(R \cdot R^T - \frac{2}{n+1}(Re_1)(Re_1)^T\Big) \\ &= \frac{n^2}{n^2-1}\Big(I - \frac{2}{n+1}\frac{L^Taa^TL}{\|L^Ta\|^2}\Big) \end{aligned}$$

Here we used the equation for $Re_1$ proved before, and the fact that $RR^T = I$, which holds for any rotation matrix. To see this observe that the length of a rotated vector $x$ should not change, i.e.,

$$x^TIx = (Rx)^T(Rx) = x^T(R^TR)x$$

which means $x^T(I - R^TR)x = 0$ for every vector $x$. It is easy to see that this can only be fulfilled if $I - R^TR = 0$.

# 9 The Ellipsoid Algorithm

$$\begin{aligned} E' &= L(\bar{E}') \\ &= \{L(x) \mid x^T \bar{Q}'^{-1} x \le 1\} \\ &= \{y \mid (L^{-1}y)^T \bar{Q}'^{-1} L^{-1} y \le 1\} \\ &= \{y \mid y^T (L^T)^{-1} \bar{Q}'^{-1} L^{-1} y \le 1\} \\ &= \{y \mid y^T (\underbrace{L\bar{Q}'L^T}_{Q'})^{-1} y \le 1\} \end{aligned}$$

---

# 9 The Ellipsoid Algorithm

Hence,

$$\begin{aligned} Q' &= L\bar{Q}'L^T \\ &= L \cdot \frac{n^2}{n^2-1}\Big(I - \frac{2}{n+1}\frac{L^T a a^T L}{a^T Q a}\Big) \cdot L^T \\ &= \frac{n^2}{n^2-1}\Big(Q - \frac{2}{n+1}\frac{Q a a^T Q}{a^T Q a}\Big) \end{aligned}$$

---

# Incomplete Algorithm

**Algorithm 1** ellipsoid-algorithm

```
input: point $c \in \mathbb{R}^n$, convex set $K \subseteq \mathbb{R}^n$
output: point $x \in K$ or "$K$ is empty"
$Q \leftarrow$ ???
repeat
    if $c \in K$ then return $c$
    else
        choose a violated hyperplane $a$
        $c \leftarrow c - \frac{1}{n+1}\frac{Qa}{\sqrt{a^T Q a}}$
        $Q \leftarrow \frac{n^2}{n^2-1}\Big(Q - \frac{2}{n+1}\frac{Q a a^T Q}{a^T Q a}\Big)$
    endif
until ???
return "$K$ is empty"
```

---

# Repeat: Size of basic solutions

**Lemma 7**
*Let $P = \{x \in \mathbb{R}^n \mid Ax \le b\}$ be a bounded polyhedron. Let $\langle a_{\max}\rangle$ be the maximum encoding length of an entry in $A, b$. Then every entry $x_j$ in a basic solution fulfills $|x_j| = \frac{D_j}{D}$ with $D_j, D \le 2^{2n\langle a_{\max}\rangle + 2n\log_2 n}$.*

In the following we use $\delta := 2^{2n\langle a_{\max}\rangle + 2n\log_2 n}$.

Note that here we have $P = \{x \mid Ax \le b\}$. The previous lemmas we had about the size of feasible solutions were slightly different as they were for different polytopes.

## Repeat: Size of basic solutions

**Proof:**
Let $\bar{A} = \left[A \; -A \; I_m\right]$, $b$, be the matrix and right-hand vector after transforming the system to standard form.

The determinant of the matrices $\bar{A}_B$ and $\bar{M}_j$ (matrix obt. when replacing the $j$-th column of $\bar{A}_B$ by $b$) can become at most

$$\begin{aligned}\det(\bar{A}_B), \det(\bar{M}_j) &\le \|\vec{\ell}_{\max}\|^{2n} \\ &\le (\sqrt{2n} \cdot 2^{\langle a_{\max}\rangle})^{2n} \le 2^{2n\langle a_{\max}\rangle + 2n\log_2 n} \;,\end{aligned}$$

where $\vec{\ell}_{\max}$ is the longest column-vector that can be obtained after deleting all but $2n$ rows and columns from $\bar{A}$.

This holds because columns from $I_m$ selected when going from $\bar{A}$ to $\bar{A}_B$ do not increase the determinant. Only the at most $2n$ columns from matrices $A$ and $-A$ that $\bar{A}$ consists of contribute.

---

## How do we find the first ellipsoid?

For feasibility checking we can assume that the polytop $P$ is bounded; it is sufficient to consider basic solutions.

Every entry $x_i$ in a basic solution fulfills $|x_i| \le \delta$.

Hence, $P$ is contained in the cube $-\delta \le x_i \le \delta$.

A vector in this cube has at most distance $R := \sqrt{n}\delta$ from the origin.

Starting with the ball $E_0 := B(0, R)$ ensures that $P$ is completely contained in the initial ellipsoid. This ellipsoid has volume at most $R^n \operatorname{vol}(B(0,1)) \le (n\delta)^n \operatorname{vol}(B(0,1))$.

---

## When can we terminate?

Let $P := \{x \mid Ax \le b\}$ with $A \in \mathbb{Z}$ and $b \in \mathbb{Z}$ be a bounded polytop. Let $\langle a_{\max}\rangle$ be the encoding length of the largest entry in $A$ or $b$.

Consider the following polyhedron

$$P_\lambda := \left\{ x \mid Ax \le b + \frac{1}{\lambda}\begin{pmatrix}1\\ \vdots\\ 1\end{pmatrix} \right\} \;,$$

where $\lambda = \delta^2 + 1$.

---

**Lemma 8**
*$P_\lambda$ is feasible if and only if $P$ is feasible.*

$\Longleftarrow$: obvious!

$\Rightarrow$:

Consider the polyhedrons

$$\bar{P} = \left\{x \mid \left[A\ -A\ I_m\right]x = b; x \ge 0\right\}$$

and

$$\bar{P}_\lambda = \left\{x \mid \left[A\ -A\ I_m\right]x = b + \frac{1}{\lambda}\begin{pmatrix}1\\ \vdots\\ 1\end{pmatrix}; x \ge 0\right\} .$$

$P$ is feasible if and only if $\bar{P}$ is feasible, and $P_\lambda$ feasible if and only if $\bar{P}_\lambda$ feasible.

$\bar{P}_\lambda$ is bounded since $P_\lambda$ and $P$ are bounded.

Let $\bar{A} = \left[A\ -A\ I_m\right]$.

$\bar{P}_\lambda$ feasible implies that there is a basic feasible solution represented by

$$x_B = \bar{A}_B^{-1}b + \frac{1}{\lambda}\bar{A}_B^{-1}\begin{pmatrix}1\\ \vdots\\ 1\end{pmatrix}$$

(The other $x$-values are zero)

The only reason that this basic feasible solution is not feasible for $\bar{P}$ is that one of the basic variables becomes negative.

Hence, there exists $i$ with

$$(\bar{A}_B^{-1}b)_i < 0 \le (\bar{A}_B^{-1}b)_i + \frac{1}{\lambda}(\bar{A}_B^{-1}\vec{1})_i$$

By Cramers rule we get

$$(\bar{A}_B^{-1}b)_i < 0 \quad \Longrightarrow \quad (\bar{A}_B^{-1}b)_i \le -\frac{1}{\det(\bar{A}_B)}$$

and

$$(\bar{A}_B^{-1}\vec{1})_i \le \det(\bar{M}_j) \ ,$$

where $\bar{M}_j$ is obtained by replacing the $j$-th column of $\bar{A}_B$ by $\vec{1}$.

However, we showed that the determinants of $\bar{A}_B$ and $\bar{M}_j$ can become at most $\delta$.

Since, we chose $\lambda = \delta^2 + 1$ this gives a contradiction.

**Lemma 9**

*If $P_\lambda$ is feasible then it contains a ball of radius $r := 1/\delta^3$. This has a volume of at least $r^n \mathrm{vol}(B(0,1)) = \frac{1}{\delta^{3n}}\mathrm{vol}(B(0,1))$.*

**Proof:**

If $P_\lambda$ feasible then also $P$. Let $x$ be feasible for $P$.
This means $Ax \le b$.

Let $\vec{\ell}$ with $\|\vec{\ell}\| \le r$. Then

$$\begin{aligned}(A(x+\vec{\ell}))_i &= (Ax)_i + (A\vec{\ell})_i \le b_i + \vec{a}_i^T\vec{\ell}\\ &\le b_i + \|\vec{a}_i\| \cdot \|\vec{\ell}\| \le b_i + \sqrt{n} \cdot 2^{\langle a_{\max}\rangle} \cdot r\\ &\le b_i + \frac{\sqrt{n} \cdot 2^{\langle a_{\max}\rangle}}{\delta^3} \le b_i + \frac{1}{\delta^2+1} \le b_i + \frac{1}{\lambda}\end{aligned}$$

Hence, $x + \vec{\ell}$ is feasible for $P_\lambda$ which proves the lemma.

How many iterations do we need until the volume becomes too small?

$$e^{-\frac{i}{2(n+1)}} \cdot \mathrm{vol}(B(0,R)) < \mathrm{vol}(B(0,r))$$

Hence,

$$\begin{aligned} i &> 2(n+1)\ln\left(\frac{\mathrm{vol}(B(0,R))}{\mathrm{vol}(B(0,r))}\right) \\ &= 2(n+1)\ln\left(n^n\delta^n \cdot \delta^{3n}\right) \\ &= 8n(n+1)\ln(\delta) + 2(n+1)n\ln(n) \\ &= \mathcal{O}(\mathrm{poly}(n, \langle a_{\max}\rangle)) \end{aligned}$$

---

**Algorithm 1** ellipsoid-algorithm

```
input: point $c \in \mathbb{R}^n$, convex set $K \subseteq \mathbb{R}^n$, radii $R$ and $r$
          with $K \subseteq B(c,R)$, and $B(x,r) \subseteq K$ for some $x$
output: point $x \in K$ or "$K$ is empty"
$Q \leftarrow \mathrm{diag}(R^2,\dots,R^2)$ // i.e., $L = \mathrm{diag}(R,\dots,R)$
repeat
    if $c \in K$ then return $c$
    else
        choose a violated hyperplane $a$
        $c \leftarrow c - \frac{1}{n+1}\frac{Qa}{\sqrt{a^TQa}}$
        $Q \leftarrow \frac{n^2}{n^2-1}\left(Q - \frac{2}{n+1}\frac{Qaa^TQ}{a^TQa}\right)$
    endif
until $\det(Q) \le r^{2n}$ // i.e., $\det(L) \le r^n$
return "$K$ is empty"
```

---

**Separation Oracle:**

Let $K \subseteq \mathbb{R}^n$ be a convex set. A separation oracle for $K$ is an algorithm $A$ that gets as input a point $x \in \mathbb{R}^n$ and either

- certifies that $x \in K$,
- or finds a hyperplane separating $x$ from $K$.

We will usually assume that $A$ is a polynomial-time algorithm.

In order to find a point in $K$ we need

- a guarantee that a ball of radius $r$ is contained in $K$,
- an initial ball $B(c,R)$ with radius $R$ that contains $K$,
- a separation oracle for $K$.

The Ellipsoid algorithm requires $\mathcal{O}(\mathrm{poly}(n) \cdot \log(R/r))$ iterations. Each iteration is polytime for a polynomial-time Separation oracle.