(from last time)

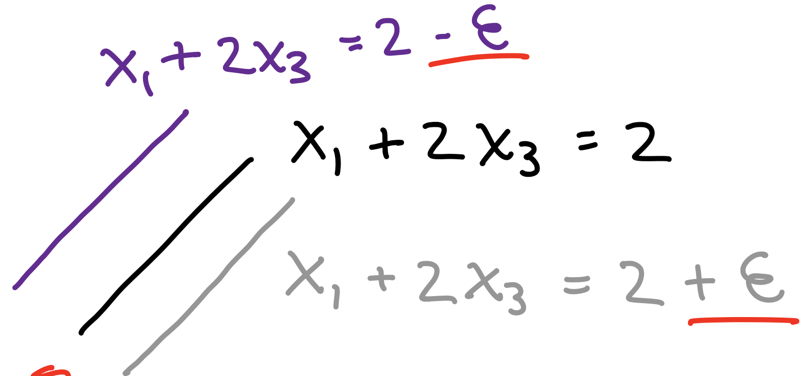
$$\begin{pmatrix}0\\0\\0\end{pmatrix} \to \underbrace{\begin{pmatrix}0\\0\\1\end{pmatrix} \to \begin{pmatrix}0\\0\\1\end{pmatrix}}^{\text{degeneracy}} \to \begin{pmatrix}2\\2\\0\end{pmatrix}$$

Lexicogr. method.

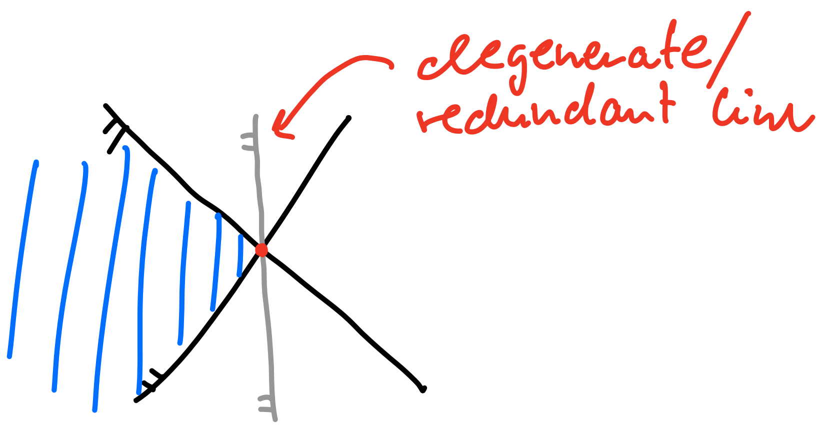
Perturbations with $\varepsilon_1 \gg \varepsilon_2 > 0$ "destroys" degeneracy.

Satisfying the planes:

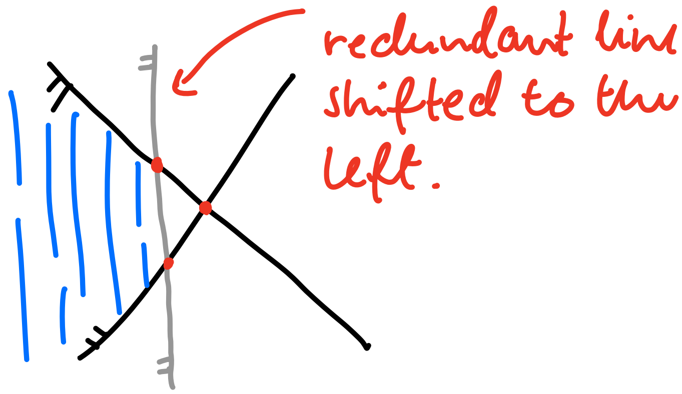
$$x_1 + 2x_3 = 2 + \varepsilon_1$$
$$x_2 + 2x_3 = 2 + \varepsilon_2$$



We are shifting the faces of our feasible set parallel in one direction.



Degeneracy means we have one redundant line.



Adding an ϵ means we shift the line (either to the left or to the right).

Notice that by introducing the ϵ we get a lot more vertices than before.

Efficiency of the SM

Each LP can be solved (if solution exist) by avoiding cycling.

How to measure efficiency?

Worst-case analysis

- measure effort (to be defined) to solve the "hardest" LP of a given size with SM.

- Easier: define upper bounds for effort from the solution of particular examples.

Avg.- case analysis

- Arranging the measured effort for all LPs of a given size.
- Easier : set of randomly created LP's must be analysed (only probability statements) $\Rightarrow$ empirical study.

Size of LP

\# of variables, # of constraints : m, n

- Data : A — (m,n) - matrix : $m\,n$

$b \in \mathbb{R}^m$: m

$c \in \mathbb{R}^n$: n

$mn + m + n$

many high-dim. problems have "many zeros". (sparse matrix)

Measuring the effort to solve an LP with SM.

- CPU time? (not so good)

Alg. and methods are iteration procedures.

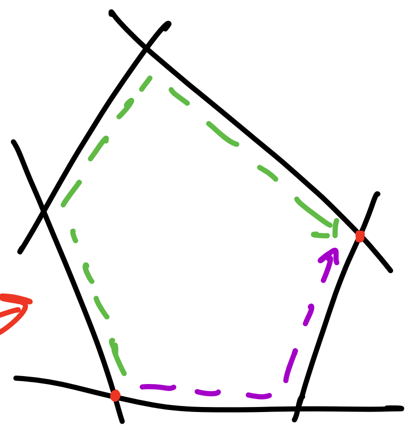
- $\boxed{\text{\# iterations, \# pivot steps}}$

This is our best method

Worst-case analysis for SM.

Upper bound (estimate) for iterations.

No cycling: each selection of BV appears at most once in the course of the SM.



For each case we don't visit all vertices.

How many choices of BV are possible?

$$\binom{n+m}{m} \geq \text{\# of iterations}$$

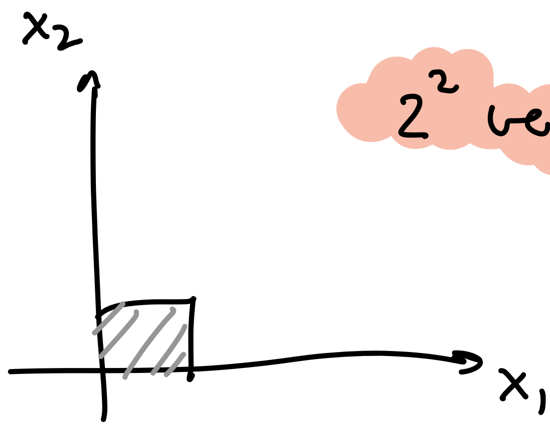
<u>for $m = n$</u>

$$\frac{1}{2} 2^{2n} \leq \binom{2n}{n} \leq 2^{2n}$$

Exponential bounds

Example: n-dimensional cube

2-dim.

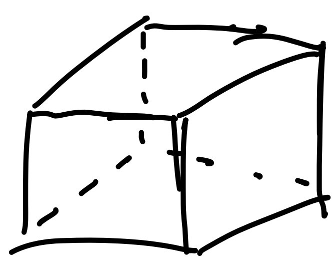


2^2 vertices

$x_1 \leq 1, \; x_2 \leq 1$

$x_1, x_2 \geq 0$

3-dim.



$n = 3$

2^3 vertices

$\vdots$

2^n vertices

Klee-Minty (1972)

Constr. an example which req $2^n - 1$ Pivot steps.

With the largest coeff. rule for the choice of the pivot column.

$$\max.\ 100x_1 + 10x_2 + x_3$$

$$\text{s.t.} \quad \left.\begin{aligned} x_1 &\le 1 \\ 20x_1 + x_2 &\le 100 \\ 200x_1 + 20x_2 + x_3 &\le 10000 \end{aligned}\right\} *$$

$$x_1, x_2, x_3 \ge 0$$

Klee-Minty cube

*

constrained by :

$$\left\{ x \in \mathbb{R}^n \;\middle|\; 0 \le x_1 \le 1,\ 0 \le x_2 \le 100,\ 0 \le x_3 \le 10000 \right\}$$

$$0 \le x_n \le 100^{n-1}$$

Let $1 = b_1 \ll b_2 \ll b_3$ (symbolic parameters)

Klee-Minty problem: (going over <u>all</u> vetices)

$$\max.\quad 100x_1 - \frac{100}{2}b_1 + 10x_2 - \frac{10}{2}b_2 + x_3 - \frac{1}{2}b_3$$

s.t.

$$
\begin{aligned}
x_1 &\le b_1 \\
20x_1 + x_2 &\le 10b_1 + b_2 \\
200x_1 + 20x_2 + x_3 &\le 100b_1 + 10b_2 + b_3
\end{aligned}
$$

We choose the largest coeff. from x_i (not from b_i).

Hence, we choose x_1.

Exch. x_1 and w_1.

since $x_1 = b_1 - w_1$

$w_1 = b_1 - x_1$

(se lecture/book for example)

Exch. x_2 and w_2

$$x_2 = -10b_1 + b_2 + 20w_1 - w_2$$

- The next dict has the same absolute values of the coefficients

- $\boxed{7}$ pivot steps unnecessary with the following BV-sets:

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} \longrightarrow \begin{pmatrix} x_1 \\ w_2 \\ w_3 \end{pmatrix} \longrightarrow \begin{pmatrix} x_1 \\ x_2 \\ w_3 \end{pmatrix} \longrightarrow \begin{pmatrix} w_1 \\ x_2 \\ w_3 \end{pmatrix}$$

$$\rightarrow \begin{pmatrix} w_1 \\ x_2 \\ x_3 \end{pmatrix} \rightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \rightarrow \begin{pmatrix} x_1 \\ w_2 \\ x_3 \end{pmatrix} \rightarrow \begin{pmatrix} w_1 \\ w_2 \\ x_3 \end{pmatrix}^{*}$$

All interchanges are x_i/w_i.
All dictionaries have coeff. with the same absolute value.

$\boxed{7} = 2^3 - 1$ pivot steps. (In general: $2^n - 1$).

Note! The choice of x_3 as pivot col. in the first step would deliver the optimal solution in one step.

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} \rightarrow \begin{pmatrix} w_1 \\ w_2 \\ x_3 \end{pmatrix}^{*}$$

Computational complexity

P - set of problem classes which have polynomial complexity. That is, each problem of this class can be solved by $P(n)$ steps where P is a polynomial and n is the problem size.

Until 1979 it was not clear wether LP was polynomial.

In 1979 a new alg. was found for LP which was polynomial. (interior point method).

will learn more later.

In practice it is not better than SM.