Linear programs

10-725 Optimization
Geoff Gordon
Ryan Tibshirani

Review: LPs

- LPs: m constraints, n vars
  - A: $R^{m \times n}$ b: R^m c: R^n x: R^n
  - ineq form
    - [min or max] $c^T x$ s.t. $Ax \le b$
    - $m \ge n$
  - std form
    - [min or max] $c^T x$ s.t. $Ax = b \quad x \ge 0$
    - $m \le n$

$$\begin{aligned} &\max\ 2x+3y \text{ s.t.} \\ &x + y \le 4 \\ &2x + 5y \le 12 \\ &x + 2y \le 5 \\ &x, y \ge 0 \end{aligned}$$

Review: LPs

- Polyhedral feasible set
  - infeasible (unhappy ball)
  - unbounded (where's my ball?)
- Optimum at a vertex (= a 0-face)
- Transforming LPs
  - changing ≥ to ≤ to =
  - getting rid of free vars or bounded vars

Review: LPs

- Tableau:

x	y	u	v	w	z	RHS
1	1	1	0	0	0	4
2	5	0	1	0	0	12
1	2	0	0	1	0	5
−2	−3	0	0	0	1	0

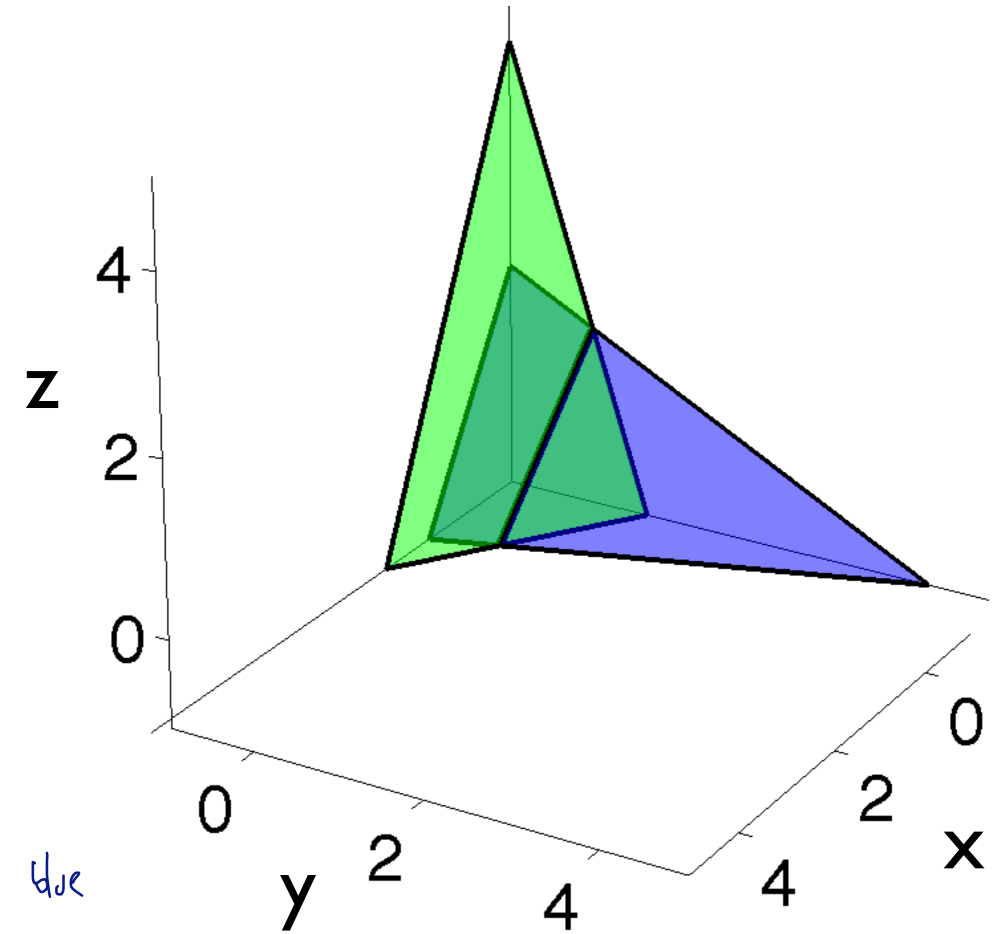
A

b

−c

- Row operations to get equivalent tableaux
- Basis (more or less corresponds to a corner)
  - use row ops to make m×m block of tableau = identity matrix
  - set nonbasic vars = 0: enough constraints to fully specify all other variables (so, a 0-face, if it's feasible)

Ineq form is projected std form



$$x, y, z \geq 0$$

$$A[x;y;z] = b$$

$$\begin{array}{ccc|cl} x & y & z & & \\ 3 & 1 & 2 & 5 & \text{blue} \\ 2 & 3 & 1 & 5 & \text{green} \end{array}$$

Three bases

$$\left[\begin{array}{ccc|c} 1 & 0 & 5/7 & 10/7 \\ 0 & 1 & -1/7 & 5/7 \end{array}\right]$$

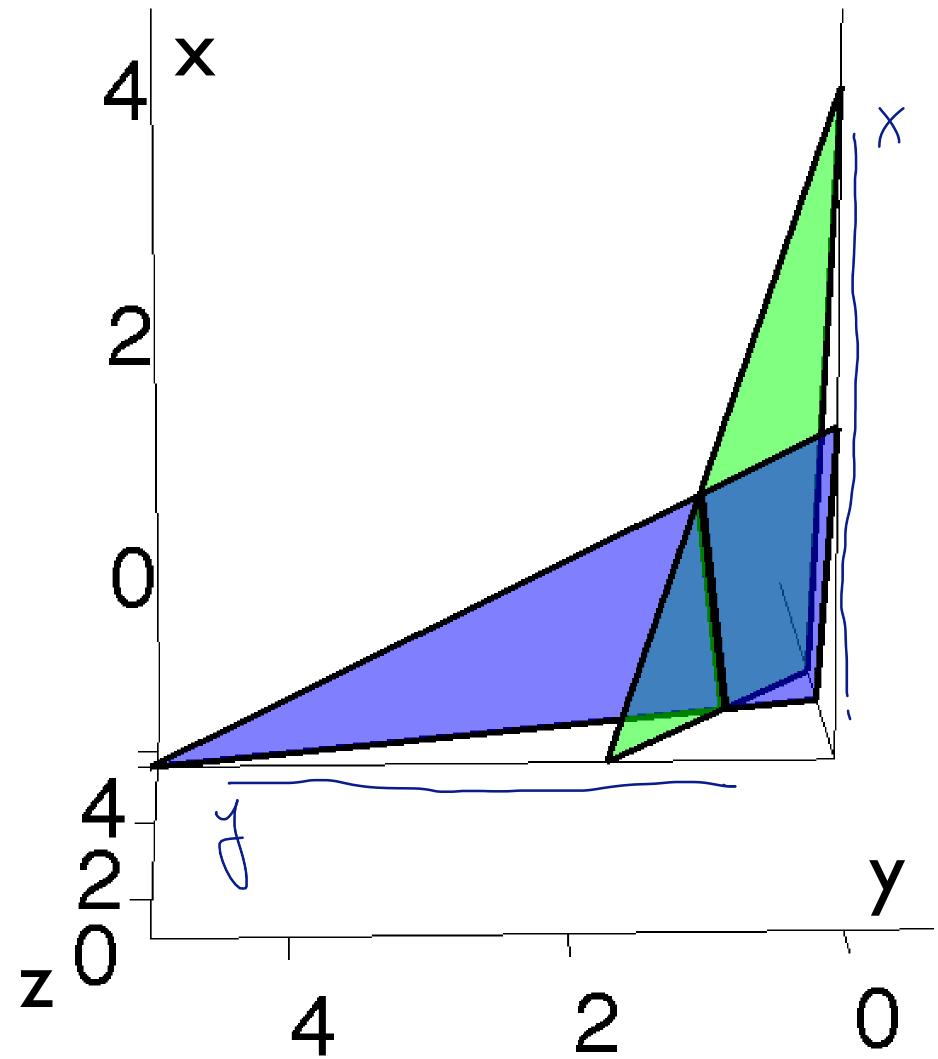
$\boxed{-5 \leq z \leq 2}$

$$\left[\begin{array}{ccc|c} 1 & 5 & 0 & 5 \\ 0 & -7 & 1 & -5 \end{array}\right]$$

$\boxed{5/7 \leq y \leq 1}$

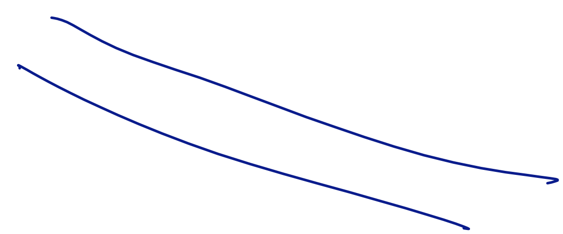
$$\left[\begin{array}{ccc|c} 1/5 & 1 & 0 & 1 \\ 7/5 & 0 & 1 & 2 \end{array}\right]$$

$\boxed{x \leq 5 \quad x \leq 10/7}$



What if we can't pick basis?

- E.g., suppose A doesn't have full row rank
  - can't pick m linearly independent cols
- Ex:
  - $3x + 2y + 1z = 3$
  - $6x + 4y + 2z = \cancel{6}\ 7$



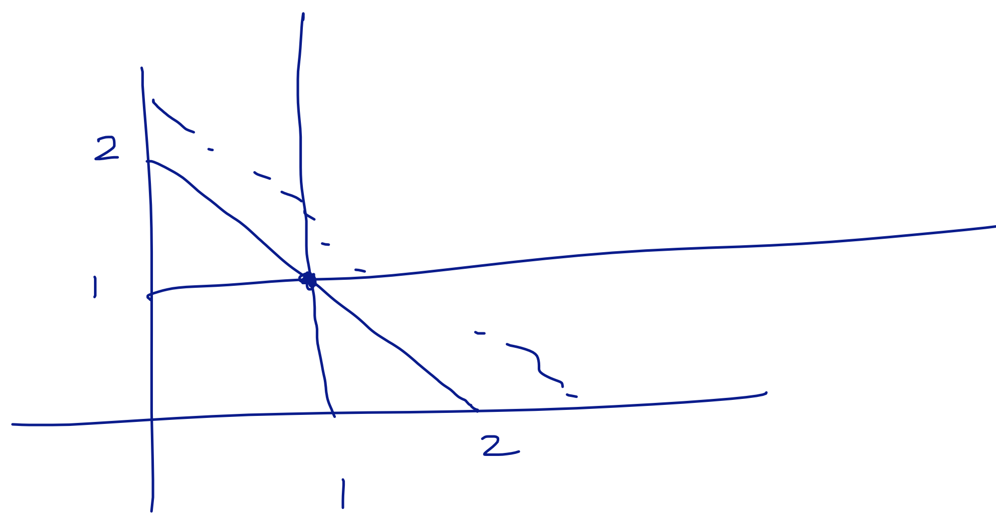
What if we can't pick basis?

- E.g., suppose fewer vars than constraints
  - A taller than it is wide, $m \geq n$
  - can't pick enough cols of A to make a square matrix
- Ex:

$\rightarrow x + y = 2 \quad \ldots = 3$

$x = 1$

$y = 1$



Nonsingular

- We can assume
  - $n \geq m$ (at least as many vars as constrs)
  - A has full row rank
- Else, drop rows (maintaining rank) until it's true
- Called ***nonsingular*** standard form LP

Naive (slooow) algorithm

- Put in nonsingular standard form
- Iterate through all subsets of ~~n~~ m vars out of n
  - if m constraints, how many subsets? $\binom{n}{m}$
- Check each for $O(m^3)$
  - full rank ("basis-ness")
  - feasibility (RHS ≥ 0)

$O(m^3 \binom{n}{m})$

- If pass both tests, compute objective
- Maintain running winner, return at end

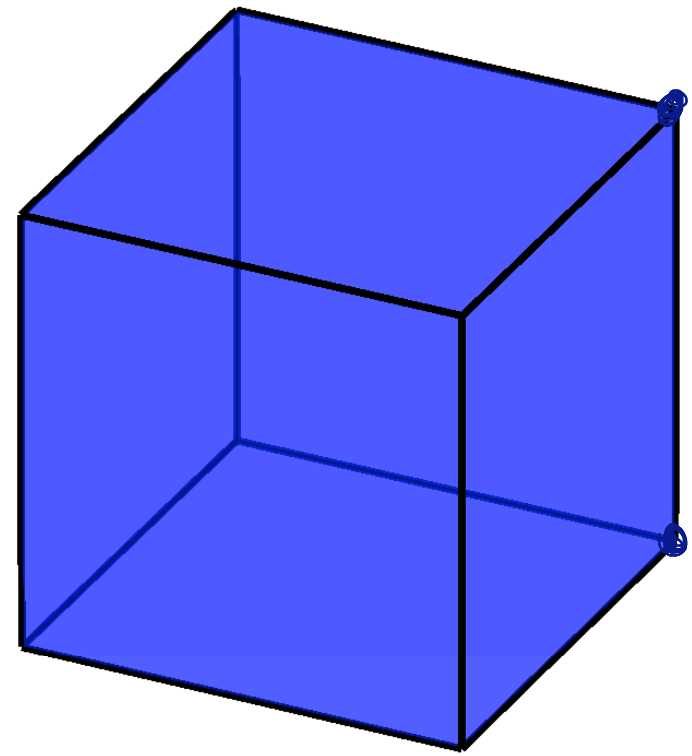
Improving our search

- Naive: enumerate all possible bases
- Smarter: maybe neighbors of good bases are also good?
- Simplex algorithm: repeatedly move to a neighboring basis to improve objective
  - continue to assume nonsingular standard form LP

Neighboring bases

- Two bases are ***neighbors*** if they share (m–1) variables
- Neighboring feasible bases correspond to vertices connected by an edge

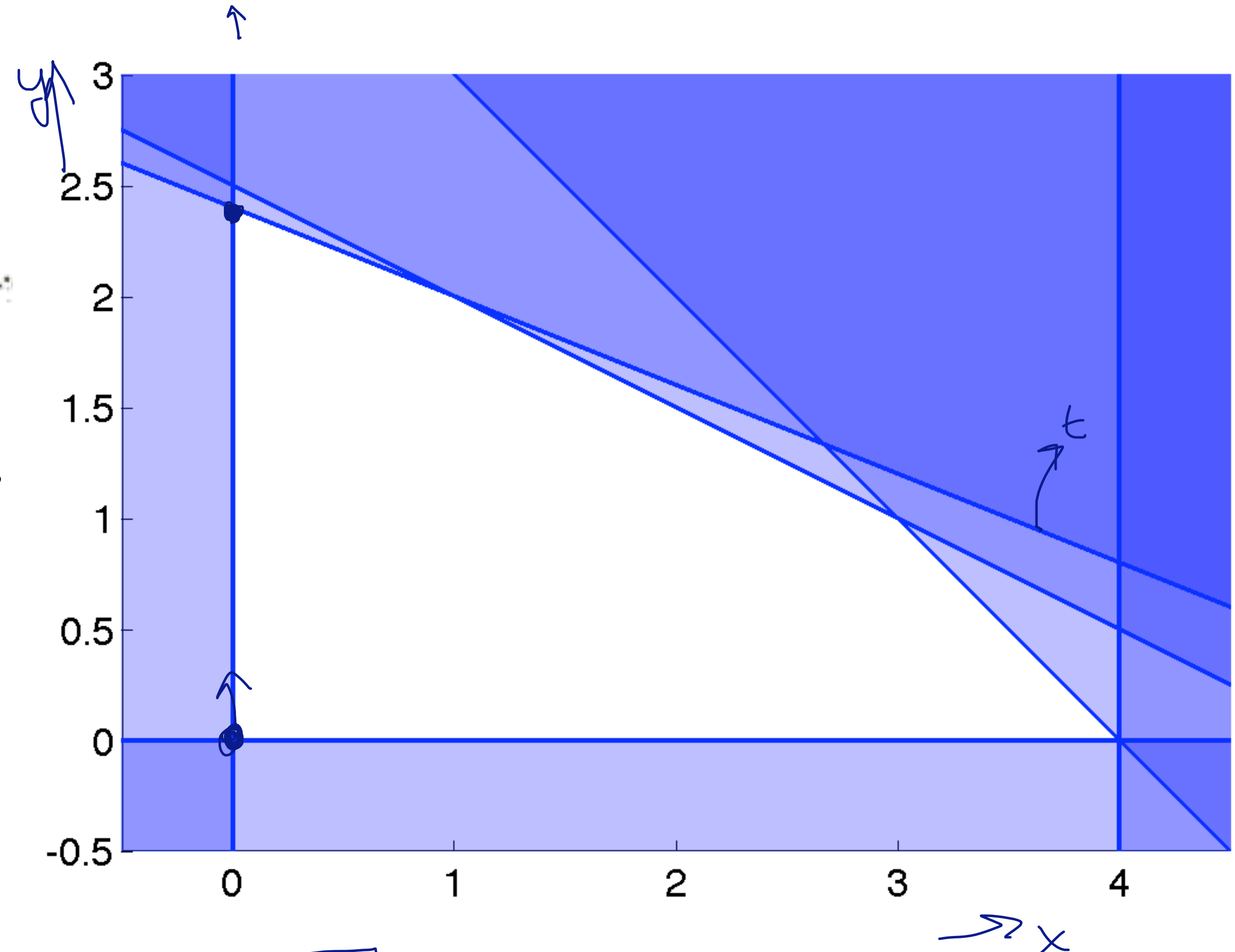
x	y	z	u	v	w	RHS
1	0	0	1	0	0	1
0	1	0	0	1	0	1
0	0	1	0	0	1	1



def'n: pivot, enter, exit

Example

max z = 2x + 3y s.t.
x + y ≤ 4
2x + 5y ≤ 12
x + 2y ≤ 5
x ≤ 4



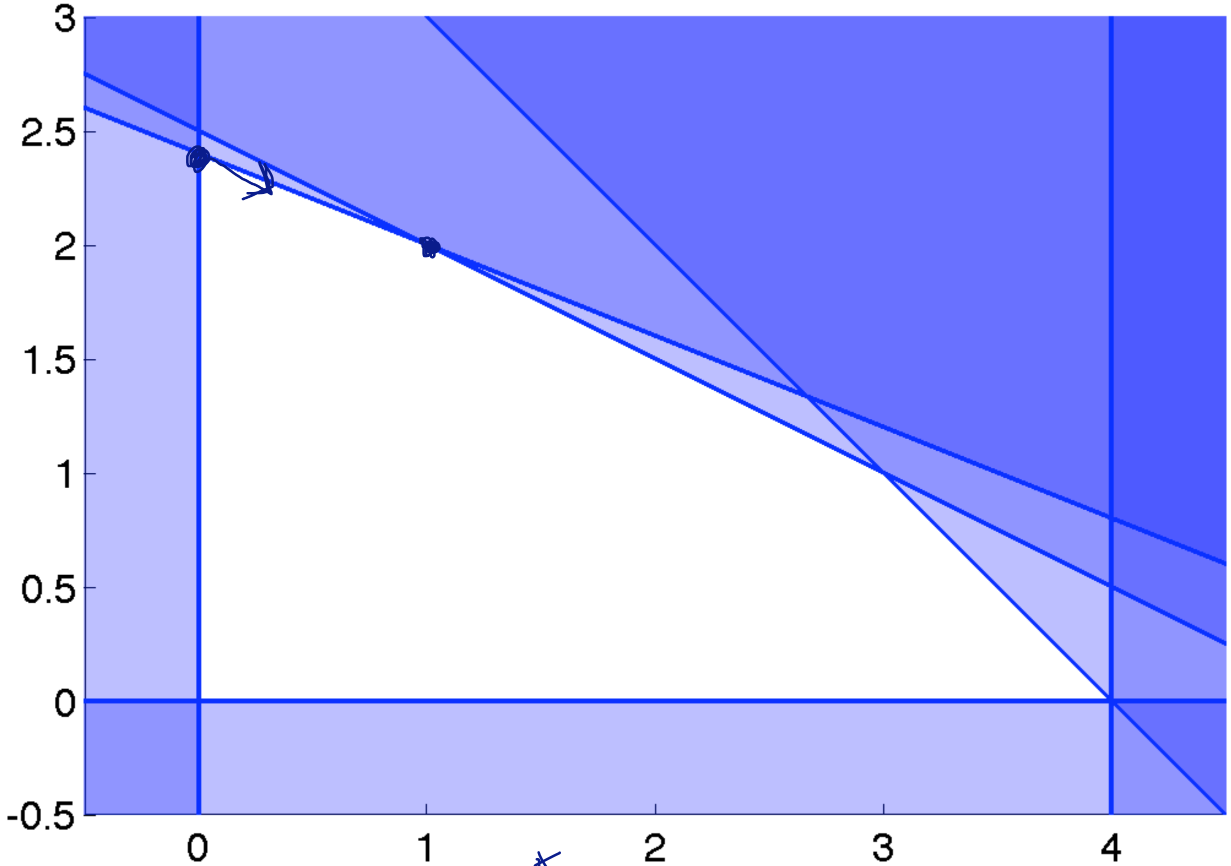
enter

leave

x	y	s	t	u	v	z	RHS
1	1	1	0	0	0	0	4
2	5	0	1	0	0	0	12
1	2	0	0	1	0	0	5
1	0	0	0	0	1	0	4
−2	−3	0	0	0	0	1	0

y += Δ
s −= Δ
t −= 5Δ
u −= 2Δ
v : no change
z += 3Δ

y ≤ 4
y = 12/5
y ≤ 5/2



reduced costs

in

out

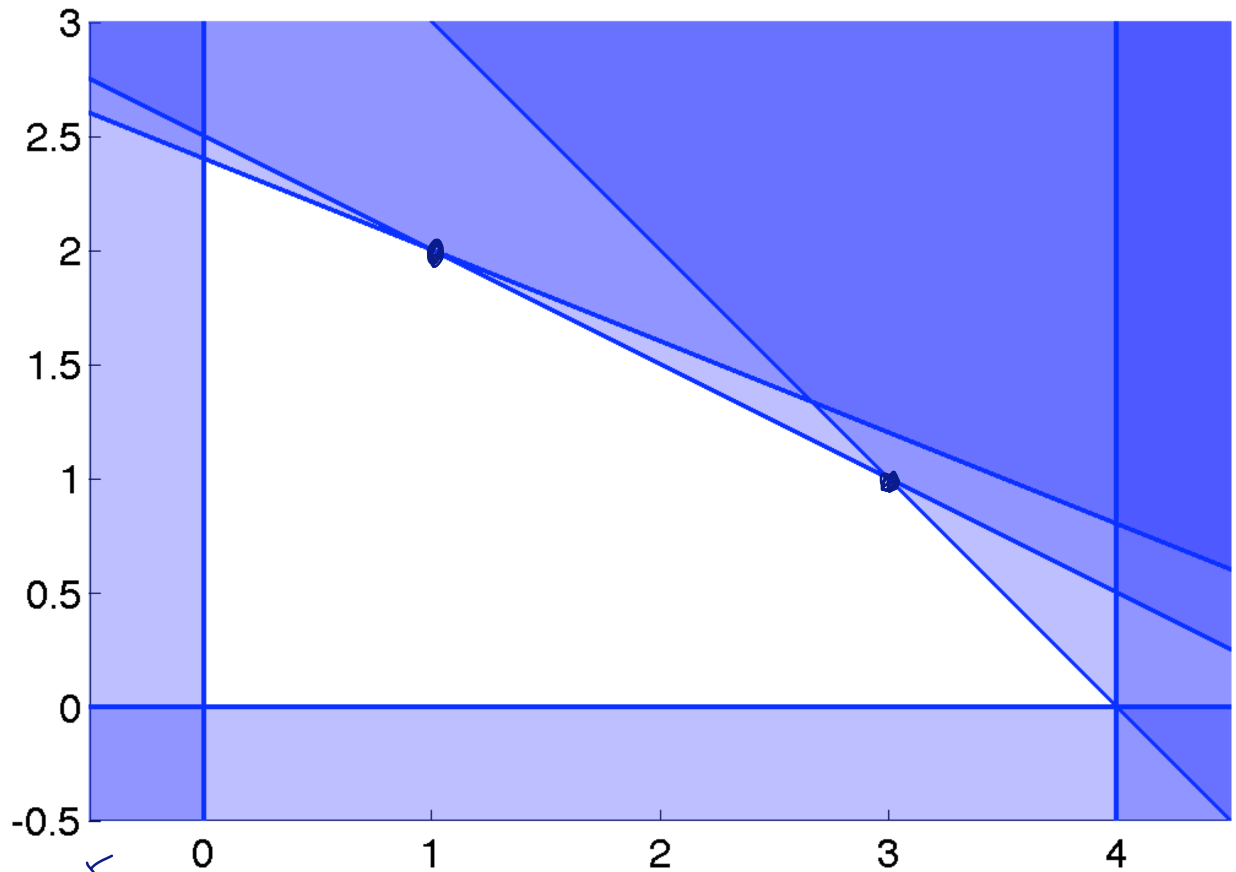
x	y	s	t	u	v	z	RHS
0.4	1	0	0.2	0	0	0	2.4
0.6	0	1	−0.2	0	0	0	1.6
0.2	0	0	−0.4	1	0	0	0.2
1	0	0	0	0	1	0	4
−0.8	0	0	0.6	0	0	1	7.2

$2.4/0.4 = 6$

$1.6/0.6 = 2\,^{2}/_{3}$

1

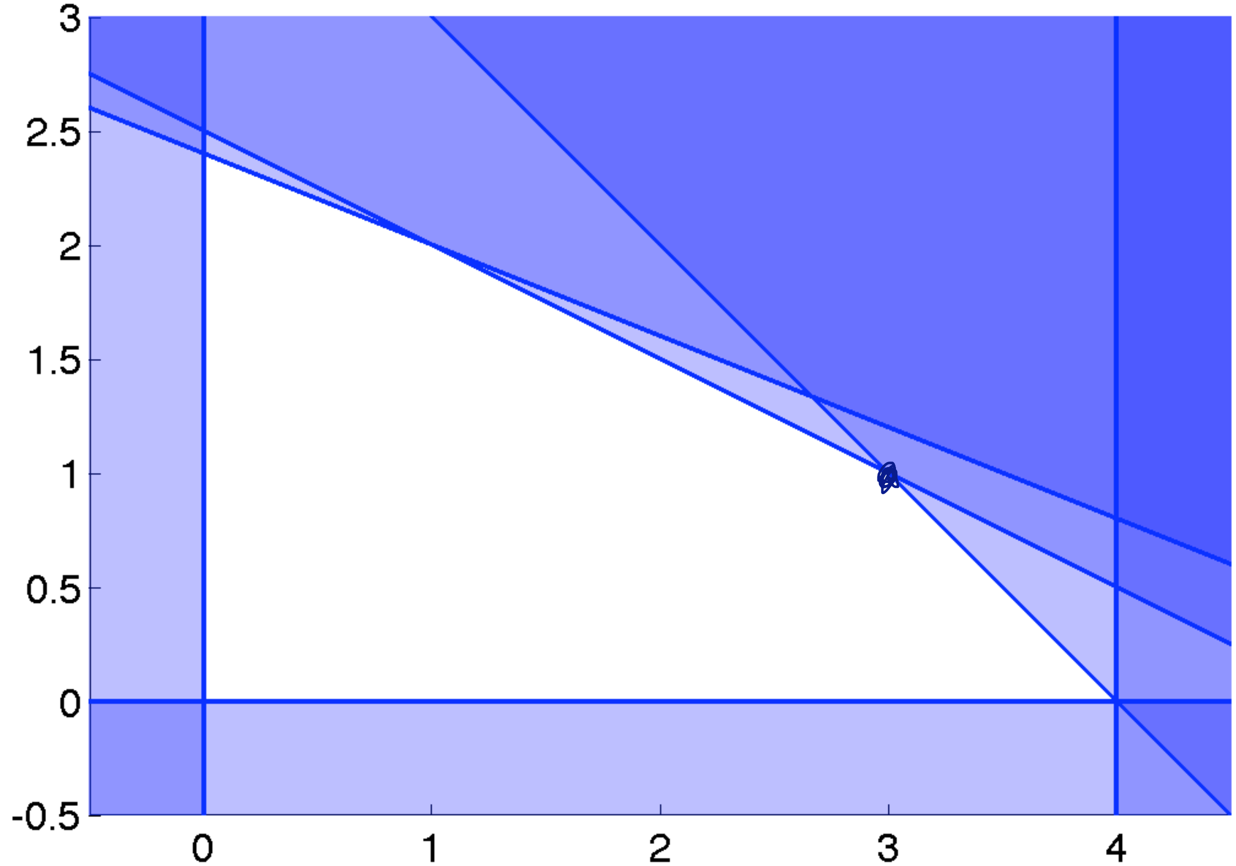
$4/1 = 4$



out

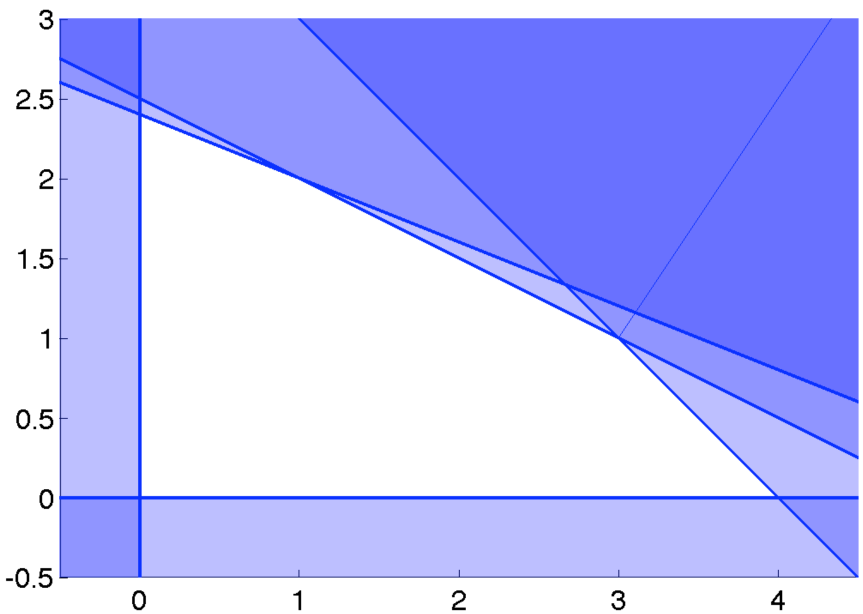
in

x	y	s	t	u	v	z	RHS
1	0	0	−2	5	0	0	1
0	1	0	1	−2	0	0	2
0	0	1	1	−3	0	0	1
0	0	0	2	−5	1	0	3
0	0	0	−1	4	0	1	8



x	y	s	t	u	v	z	RHS
1	0	2	0	−1	0	0	3
0	1	−1	0	1	0	0	1
0	0	1	1	−3	0	0	1
0	0	−2	0	1	1	0	1
0	0	1	0	1	0	1	9

Initial basis



x	y	u	v	w	RHS
1	1	1	0	0	4
2	5	0	1	0	12
1	2	0	0	1	5

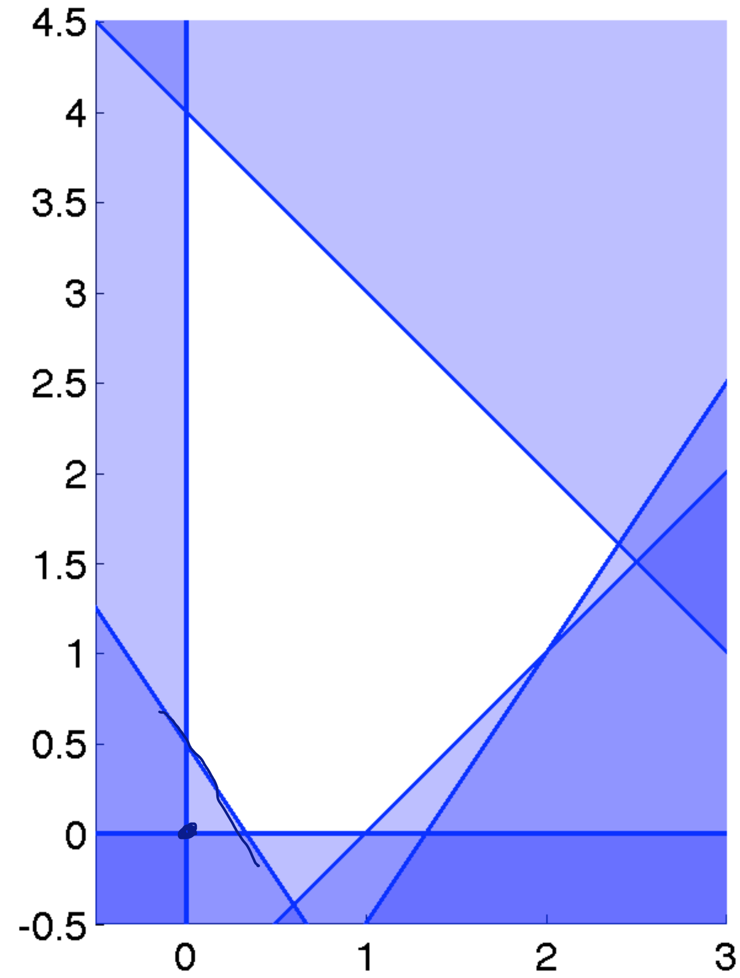
- So far, assumed we started w/ feasible basic solution—in fact, it was trivial to find one
- Not always so easy in general

Big M

0 ≤ x, y, s1..s6 $q \geq 0$

max x − 2y $-\underline{M}q$

x	y	slacks				z	q	RHS
1	1	1	0	0	0	0		4
3	−2	0	1	0	0	0		4
1	−1	0	0	1	0	0		1
+3	+2	0	0	0	−1	0	+1	+1
−1	2	0	0	0	0	1		0



- Can make it easy: variant of slack trick
  - For each violated constraint, add var w/ coeff −1
  - Penalize in objective; negate constraint

Simplex in one slide

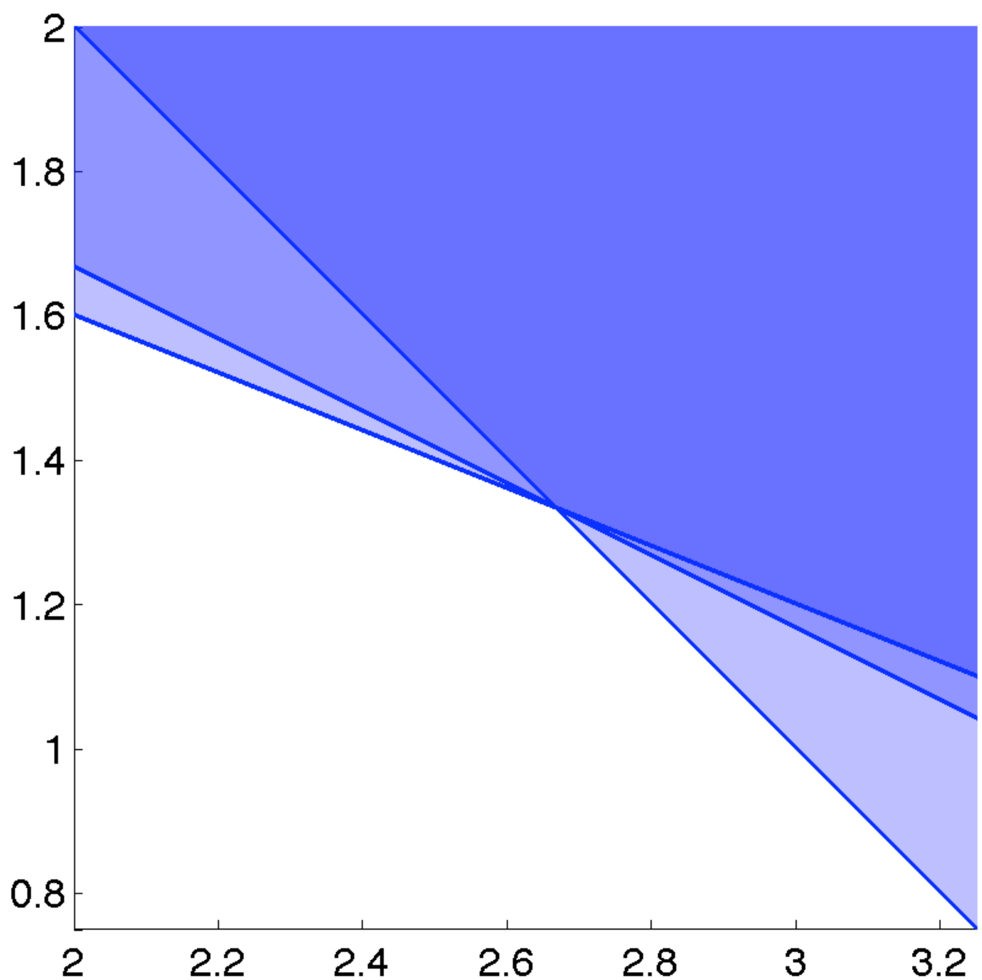
(skipping degeneracy handling)

- Given a nonsingular standard-form max LP
- Start from a feasible basis and its tableau
  - big-M if needed
- Pick non-basic variable w/ coeff in objective ≤ 0
- Pivot it into basis, getting neighboring basis
  - select exiting variable to keep feasibility
- Repeat until all non-basic variables have objective ≥ 0

Degeneracy

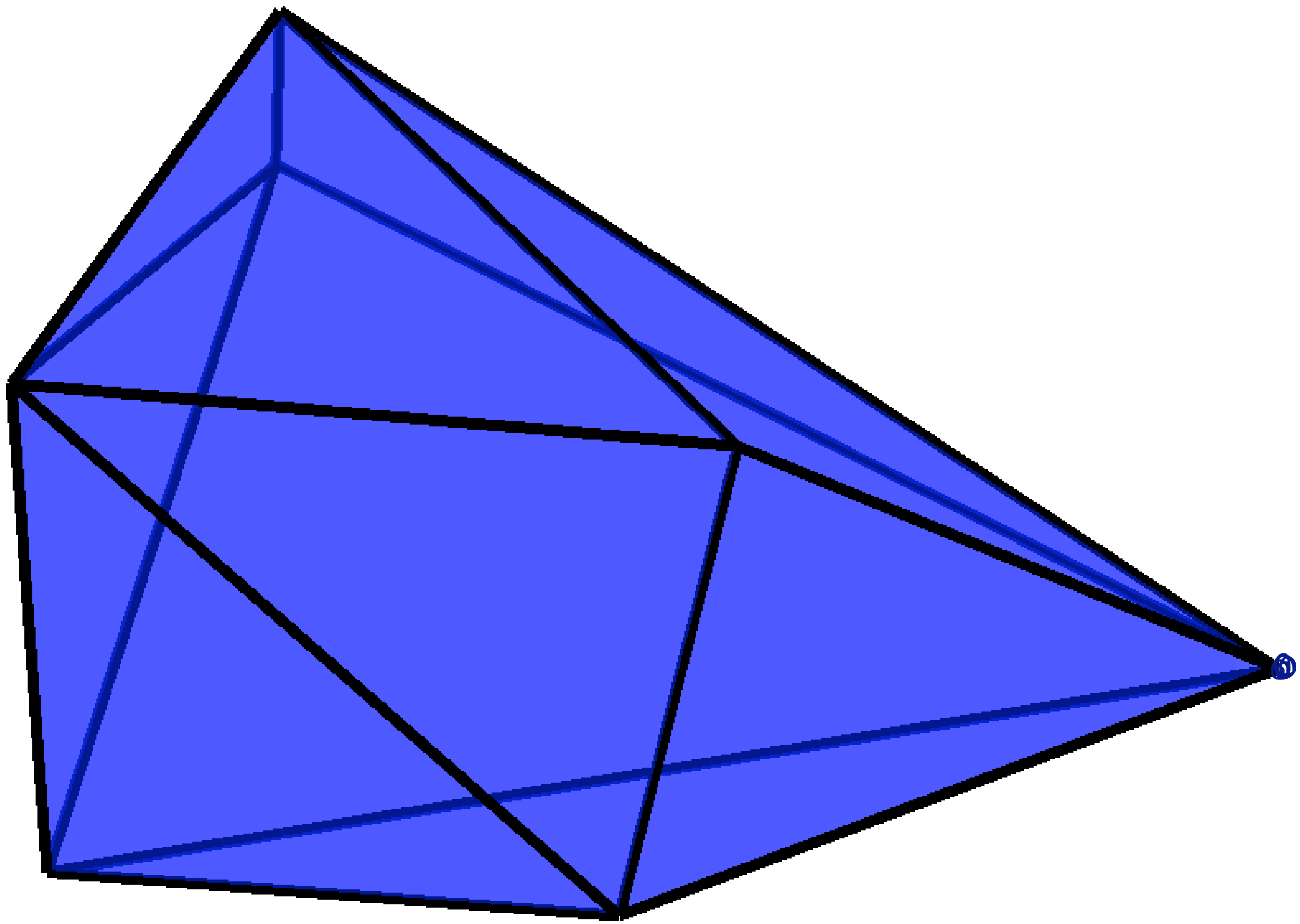
- Not every set of m variables yields a corner
  - some have rank < m (not a basis)
  - some are infeasible
- Can the reverse be true? Can two bases yield the same corner?

Degeneracy



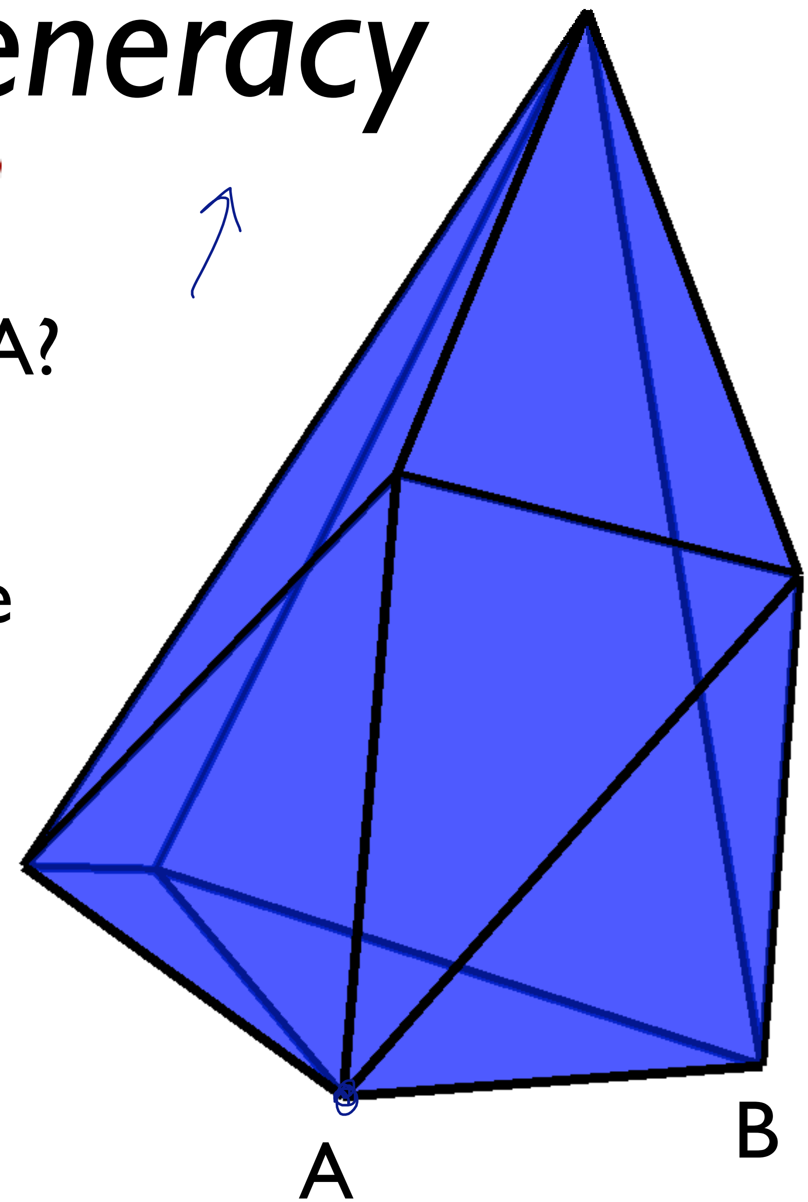
x	y	u	v	w	RHS
1	1	1	0	0	4
2	5	0	1	0	12
1	2	0	0	1	16/3
1	0	0	-2	5	8/3
0	1	0	1	-2	4/3
0	0	1	1	-3	0
1	0	2	0	-1	8/3
0	1	-1	0	1	4/3
0	0	1	1	-3	0

Degeneracy in 3D



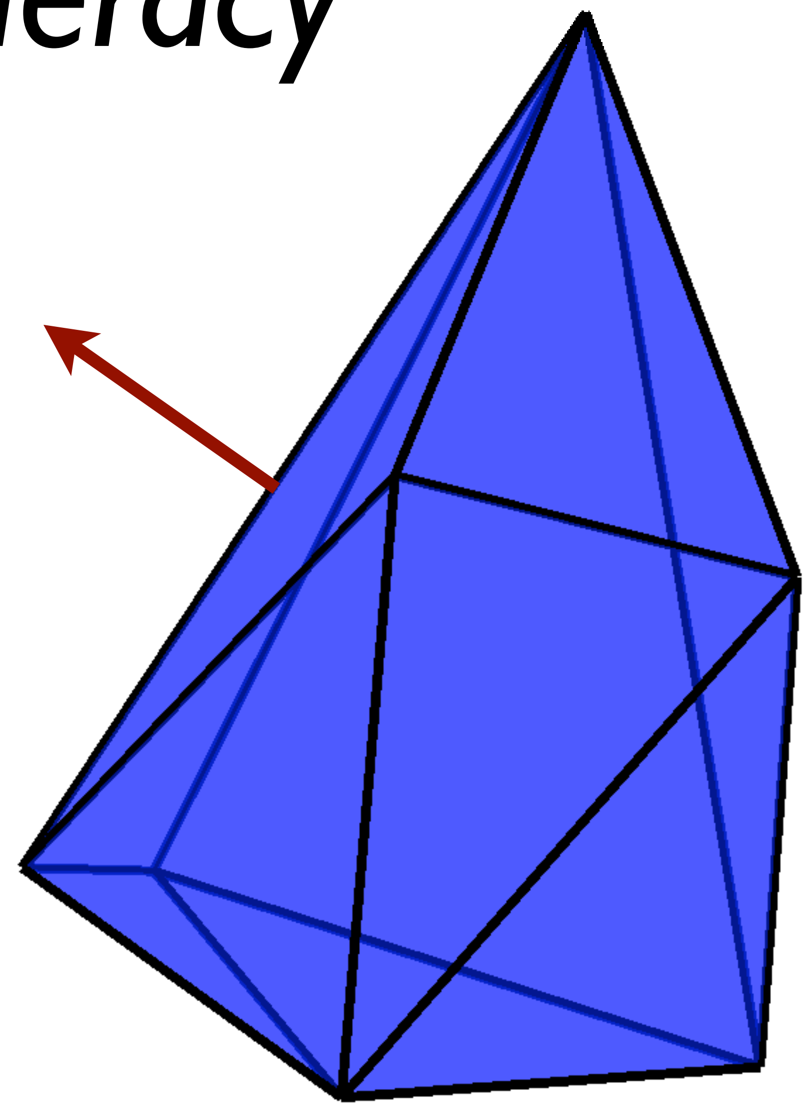
Bases & degeneracy

- How many bases for vertex A?
  - $\binom{5}{3}$
- Are they all neighbors of one another?
  - no
- Are they all neighbors of B?
  - no



Dual degeneracy

- More than m entries in objective row = 0
  - so, a nonbasic variable has reduced cost = 0
  - objective orthogonal to a d-face for d ≥ 1



Handling degeneracy

- Sometimes have to make pivots that don't improve objective
  - stay at same corner (exiting variable was already 0)
  - move to another corner w/ same objective (coeff of entering variable in objective was 0)

ordinary primal

dual

- Problem of cycling
  - need an anti-cycling rule (there are many…)
  - e.g.: add tiny random numbers to obj, RHS