Interior-point methods

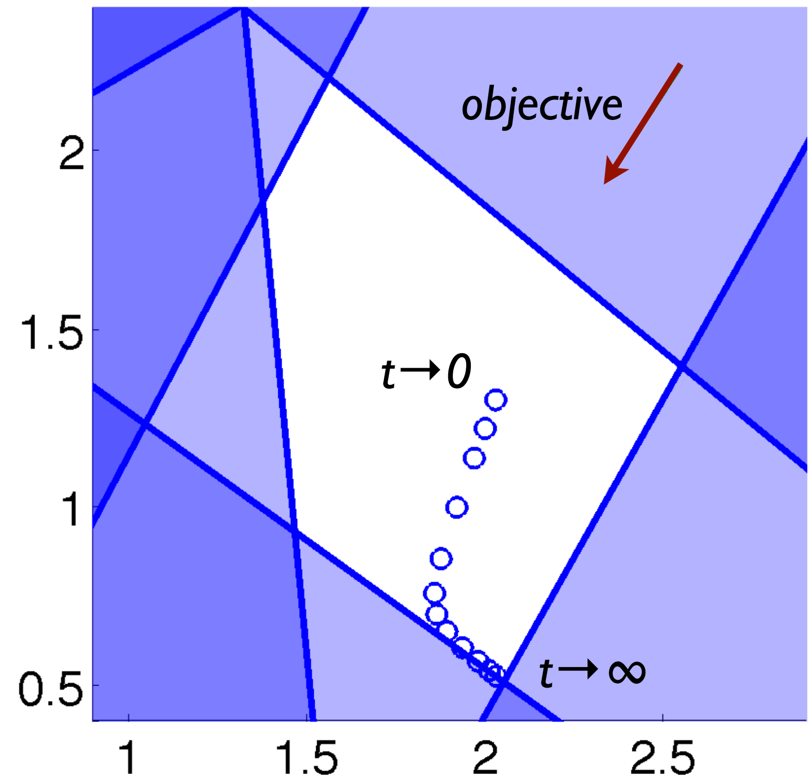
10-725 Optimization
Geoff Gordon
Ryan Tibshirani

Review

- Analytic center
  - force field interpretation
  - Newton's method: $y = 1./(Ax+b)$ $A^T Y^2 A \, \Delta x = A^T y$
- Dikin ellipsoid
  - unit ball of Hessian norm for log barrier
  - contained in feasible region
  - Dikin ellipsoid at analytic center: scale up by m, contains feasible region

Review

- Central path:
  - force field
  - Newton: $A^TY^2A\,\Delta x = A^Ty - tc$
  - trading centering v. optimality
- Affine invariance
- Constraint form v. penalty form of central path



- Primal-dual correspondence for central path
  - duality gap m/t

Primal-dual constraint form

- Primal-dual pair:
  - min $c^T x$ st $Ax + b \geq 0$
  - max $-b^T y$ st $A^T y = c \quad y \geq 0$
- KKT:
  - $Ax + b \geq 0$ (primal feasibility)
  - $y \geq 0 \quad A^T y = c$ (dual feasibility)
  - $c^T x + b^T y \leq 0$ (strong duality)
  - …or, $c^T x + b^T y \leq \lambda$ (relaxed strong duality)

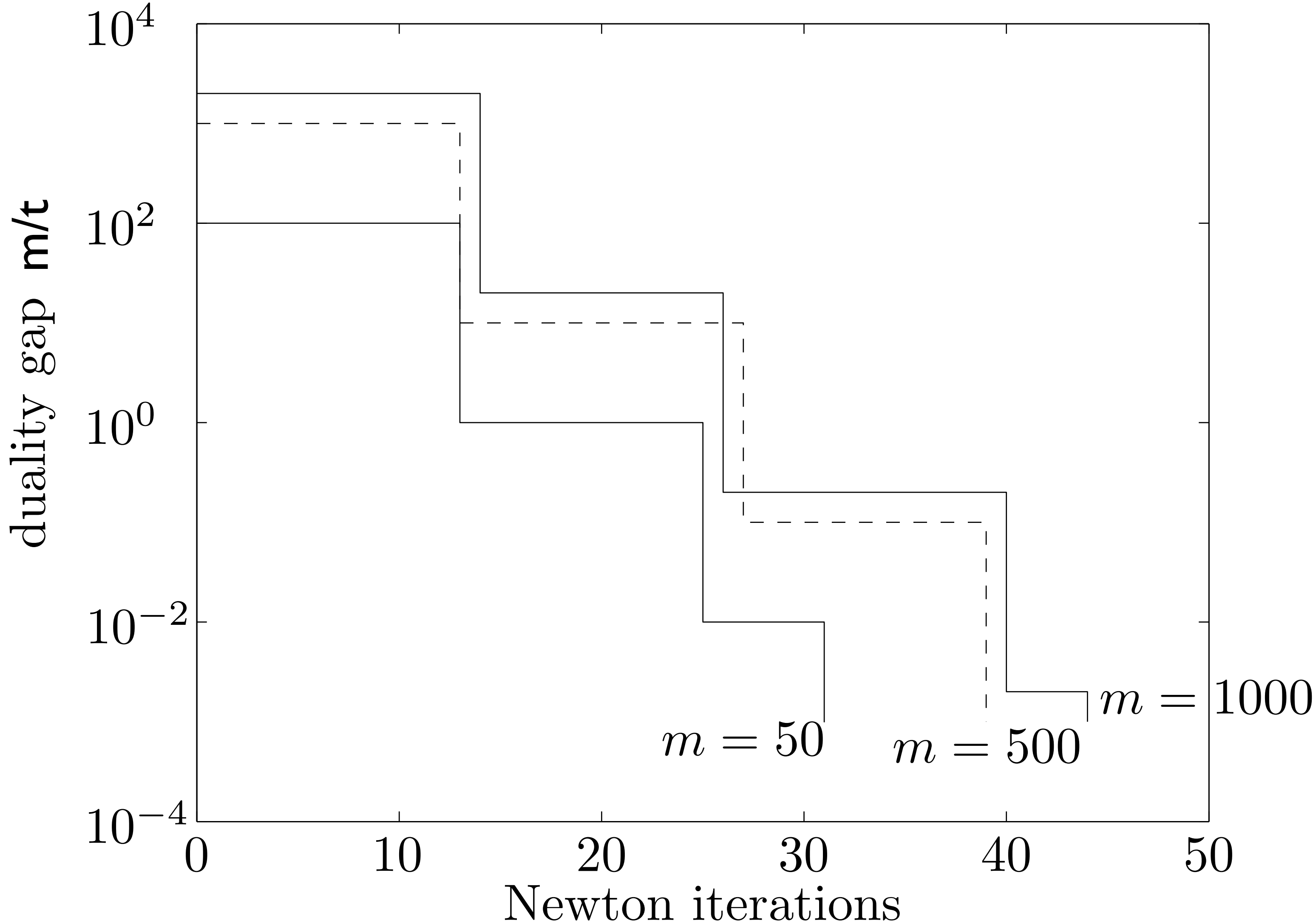
Analytic center of relaxed KKT

- Relaxed KKT conditions:
  - $Ax + b = s \geq 0$
  - $y \geq 0$
  - $A^T y = c$
  - $c^T x + b^T y \leq \lambda$
- Central path = {analytic centers of relaxed KKT}

A simple algorithm

- $t := 1$, $y := 1^m$, $x := 0^n$
- Repeat:
  - Use infeasible-start Newton to find point y on dual central path (and corresponding multipliers x)
  - $t := \alpha t \quad (\alpha > 1)$
- After any outer iteration:
  - Multipliers x are primal feasible; gap $c^T x + b^T y = m/t$
  - or, recover w/ duality: $s = 1./ty \quad x = A\backslash(s-b)$

Example



duality gap m/t
10^4
10^2
10^0
10^{-2}
10^{-4}
0
10
20
30
40
50
Newton iterations
$m = 50$
$m = 500$
$m = 1000$

An algorithm and proof

- ***Feasible*** for KKT conditions:
  - $Ax + b = s \geq 0$
  - $y \geq 0$
  - $A^T y = c$
- ***Optimal*** for KKT conditions:
  - $c^T x + b^T y \leq 0$ or $s^T y \leq 0$
- A potential combining feasibility & optimality:
  - $p(s,y) = (m+k) \ln y^T s - \sum \ln y_i - \sum \ln s_i$

[Kojima, Mizuno, Yoshise, Math. Prog., 1991]

Potential reduction

- Potential:
  - $p(s,y) = (m+k) \ln y^T s - \sum \ln y_i - \sum \ln s_i$
    $= k \ln y^T s + [m \ln y^T s - \sum \ln y_i - \sum \ln s_i]$
- Algorithm strategy:
  - start w/ strictly feasible (x, y, s)
  - update by (Δx, Δy, Δs):
  - reduce p(s,y) by at least δ per iter:

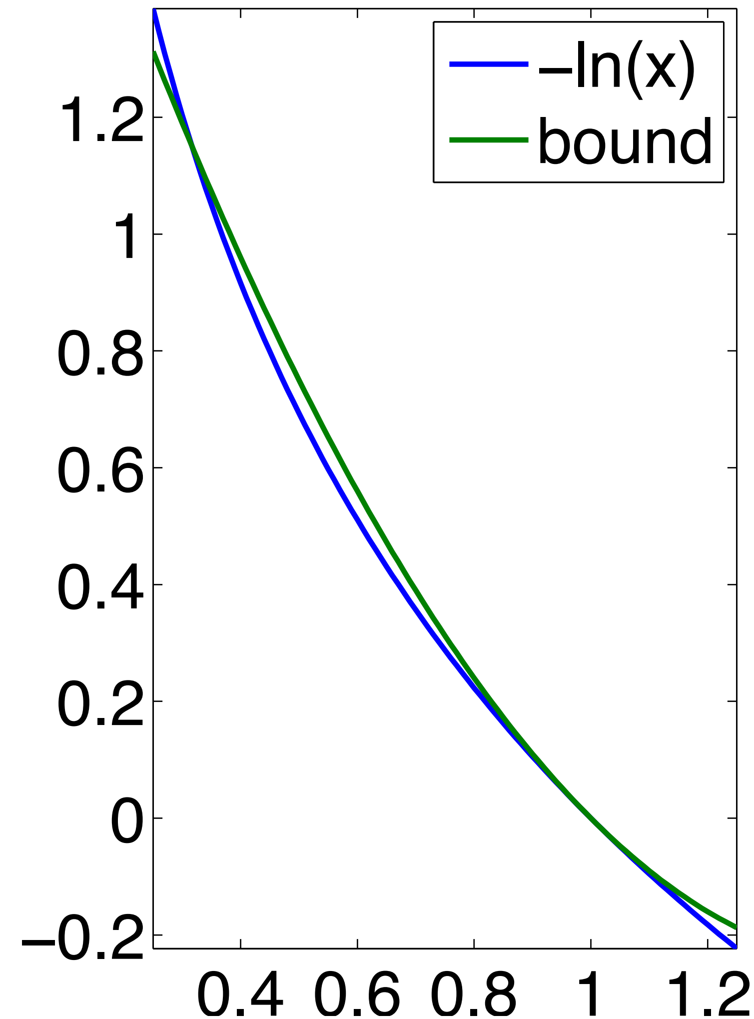
Potential reduction strategy

- Upper bound p(s,y) locally with a quadratic
  - will look like Hessian from Newton's method
  - analyze upper bound: reduce by at least δ / iter
  - $p(s,y) = (m+k) \ln y^T s - \sum \ln y_i - \sum \ln s_i$

- $p_1(s,y) \leq$

Upper bound, cont'd

- $p_2(s,y) = -\sum \ln y_i - \sum \ln s_i$



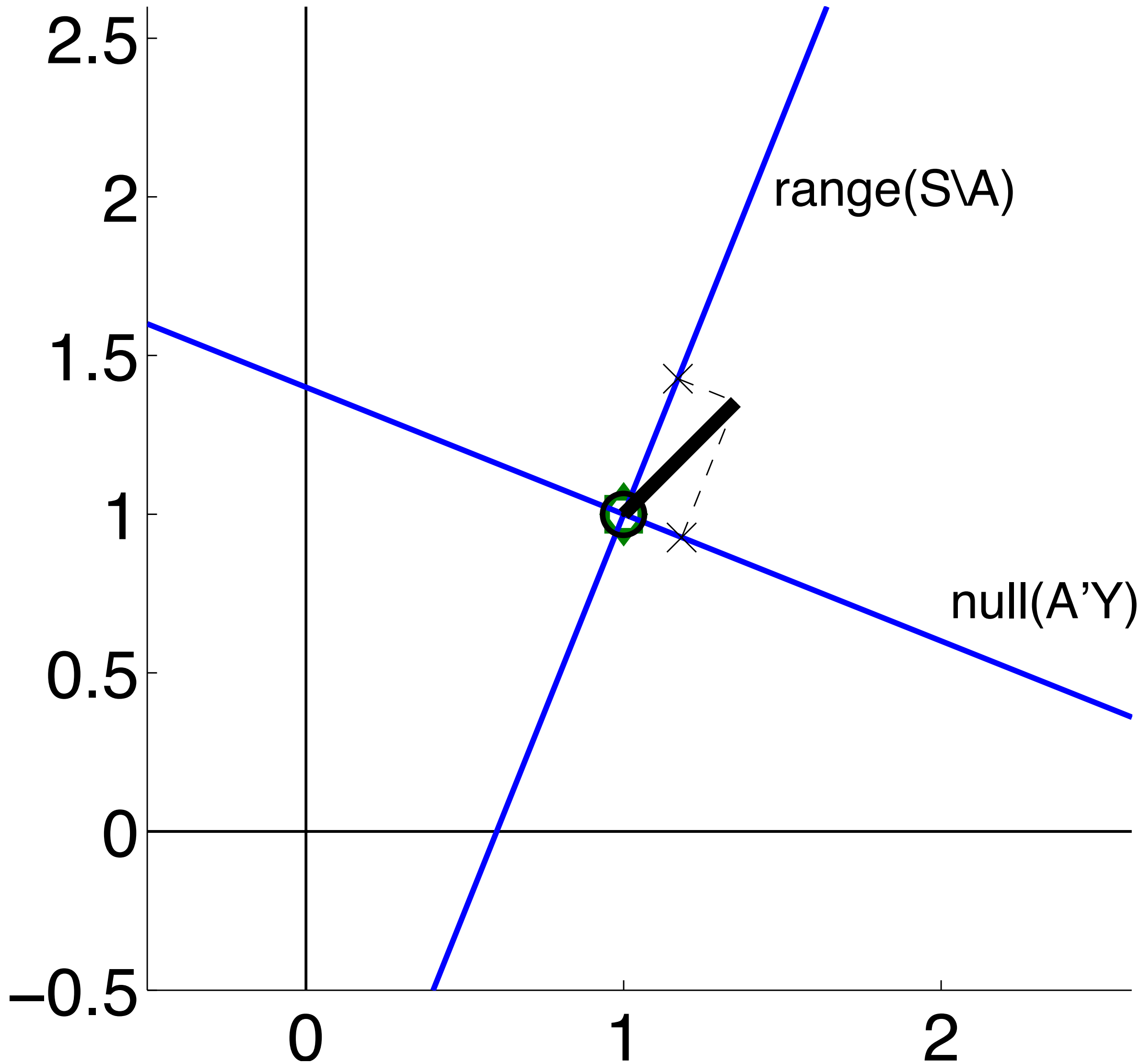
Algorithm: repeat…

- Choose $(\Delta x, \Delta y, \Delta s)$ to minimize $\bar{p}_1 + \bar{p}_2$ st
  - $A^T\Delta y=0 \quad \Delta s=A\Delta x$
  - $\Delta y^T Y^{-2} \Delta y + \Delta s^T S^{-2} \Delta s \leq (2/3)^2$
    - stronger than box constraint
- Step along $(\Delta x, \Delta y, \Delta s)$ while keeping $y>0$, $s>0$
- Claim: can always decrease potential by $\delta=1/4$ per iteration

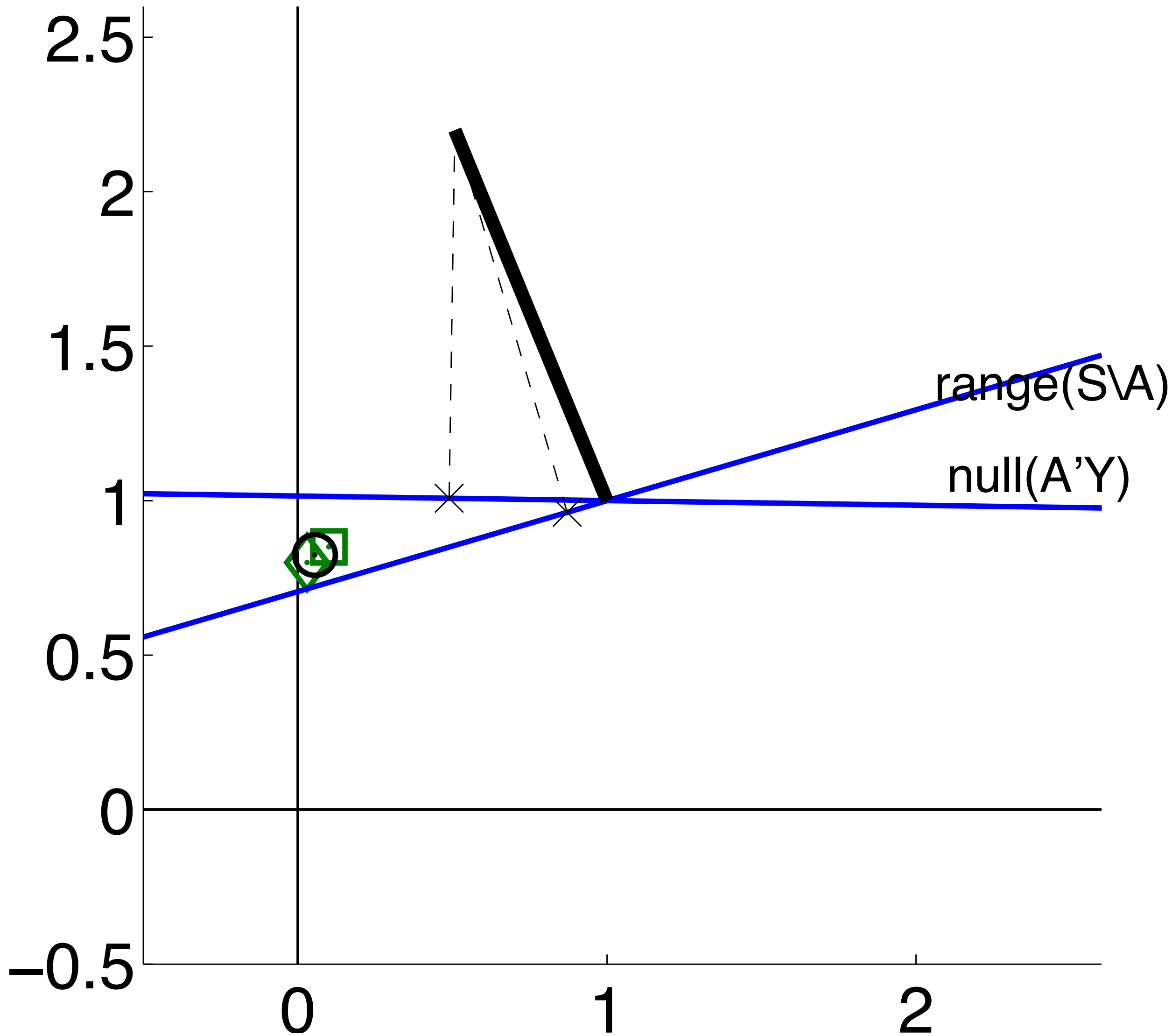
Intuition

- Suppose $s = y = 1^m$
  - $\bar{p}_1 = p_1 + [(m+k)/m]\ [1^T\Delta y + 1^T\Delta s]$
  - $\bar{p}_2 = p_2 - 1^T\Delta y - 1^T\Delta s + \Delta y^T\Delta y + \Delta s^T\Delta s$
  - $\Delta p \leq$
- How much decrease is possible?

The simple case



Farther from equilibrium



In general

Bounding g

- $g = (m+k) y \circ s / y^T s - 1$
- $\min \;\; g^T \Delta u + g^T \Delta v + \Delta u^T \Delta u + \Delta v^T \Delta v$
  - s.t. $A^T Y \Delta u = 0 \quad \Delta v = S^{-1} A \Delta x$
- $||\pi(g,g)|| \geq g^T \Delta u + g^T \Delta v \quad \forall$ feasible $||(\Delta u, \Delta v)|| \leq 1$

Step size

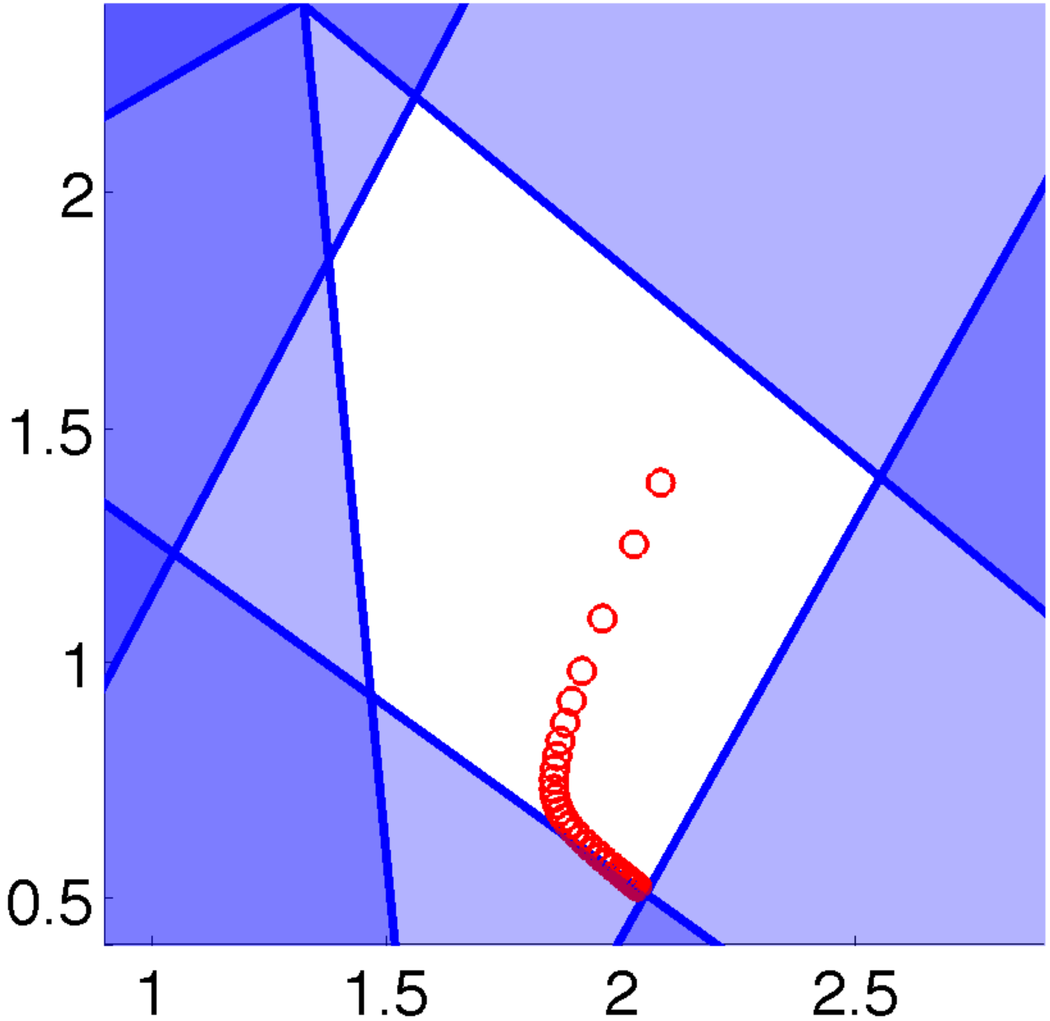
- $||(g,g)|| \geq k/\sqrt{m}$
  - step size:
  - decrease:

Algorithm summary

- Pick parameters $k>0$, $\tau>1$ and feasible (x, y, s)
- Repeat until $y^T s$ is small enough:
  - choose $(\Delta x, \Delta y, \Delta s)$ to minimize
    - $((m+k)s/y^T s - 1/y)^T \Delta y + ((m+k)y/y^T s - 1/s)^T \Delta s + \tau \Delta y^T Y^{-2} \Delta y/2 + \tau \Delta s^T S^{-2} \Delta s/2$
    - $\Delta s = A\Delta x \quad A^T \Delta y = 0 \quad \Delta y^T Y^{-2} \Delta y + \Delta s^T S^{-2} \Delta s \le f(\tau)$
    - quadratic w/ linear constrs—looks like Newton
  - line search for best step length with $s>0$, $y>0$
  - update (x, y, s) with our direction and step length

Example

- Infeasible initializer
- $k=\sqrt{m}$
- $\tau=2$
- $A \in R^{7\times 2}$

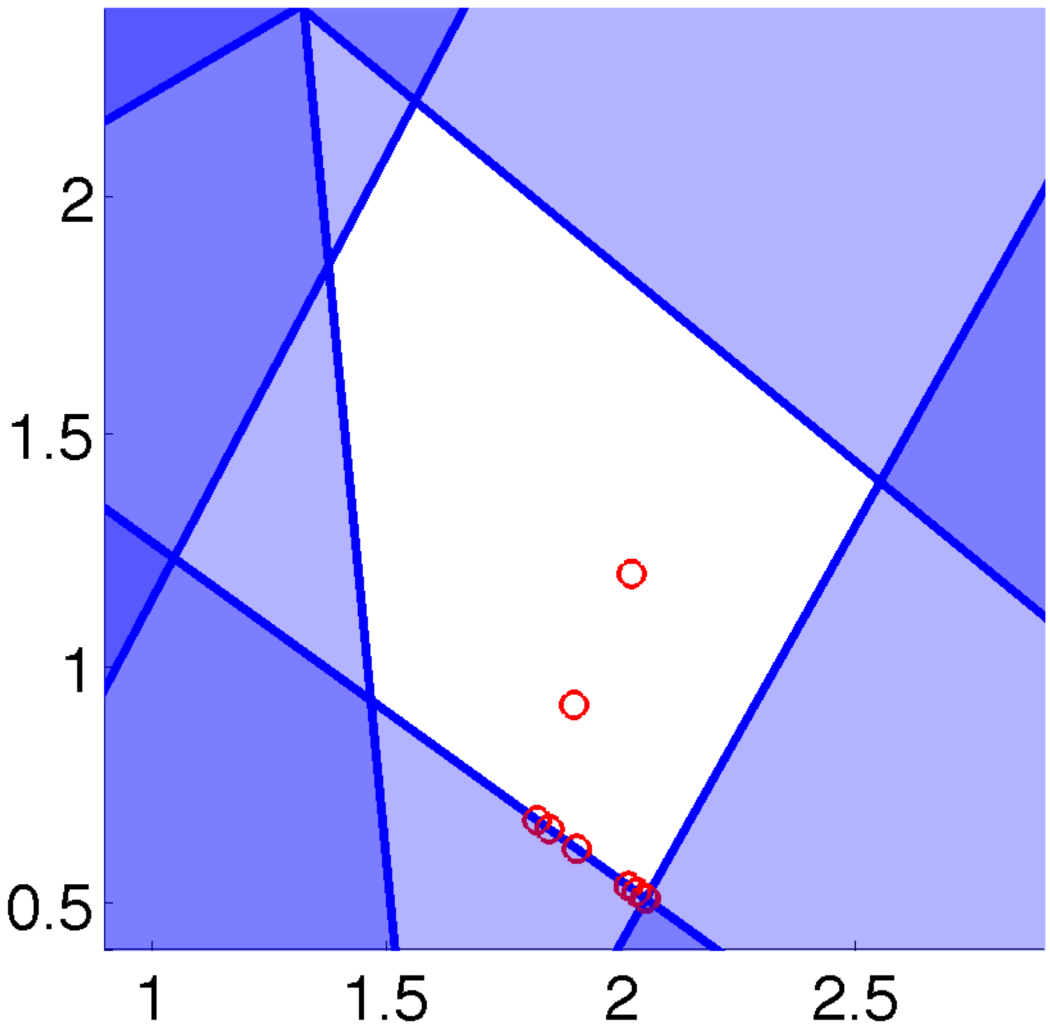


Diagnostics

```
 1: step 1.0000, mean gap 10^ -0.4057, pot  17.2774
 2: step 1.0000, mean gap 10^ -0.5184, pot  15.9290
 3: step 1.0000, mean gap 10^ -0.6024, pot  15.1694
 4: step 1.0000, mean gap 10^ -0.6908, pot  14.5695
…
11: step 1.0000, mean gap 10^ -1.3256, pot  10.6940
12: step 1.0000, mean gap 10^ -1.4165, pot  10.1402
13: step 1.0000, mean gap 10^ -1.5074, pot   9.5864
14: step 1.0000, mean gap 10^ -1.5983, pot   9.0327
…
21: step 1.0000, mean gap 10^ -2.2340, pot   5.1602
22: step 1.0000, mean gap 10^ -2.3248, pot   4.6069
23: step 1.0000, mean gap 10^ -2.4157, pot   4.0534
24: step 1.0000, mean gap 10^ -2.5066, pot   3.4997
…
30: step 1.0000, mean gap 10^ -3.0522, pot   0.1756
```

Example

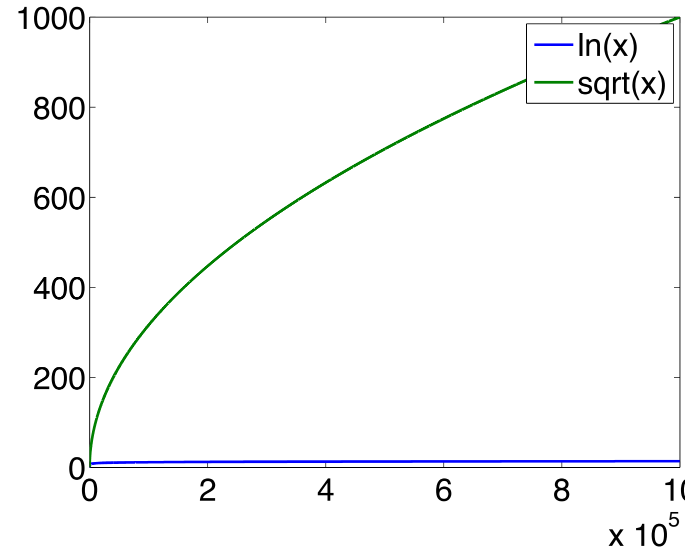
- Same initializer
- k=.999m
- τ=1.95
- $A \in R^{7\times 2}$



Diagnostics

```
  1: step 1.0000, mean gap 10^ -0.6266, pot  18.1732
  2: step 0.9109, mean gap 10^ -0.9666, pot  13.4386
  3: step 0.9997, mean gap 10^ -1.4694, pot  10.6936
  4: step 0.7258, mean gap 10^ -1.9010, pot   2.4038
  5: step 0.6761, mean gap 10^ -2.2711, pot  -4.7473
  6: step 0.9258, mean gap 10^ -2.8463, pot -14.3558
  7: step 0.6785, mean gap 10^ -3.3540, pot -24.9006
…
 17: step 0.9767, mean gap 10^ -8.1569, pot -98.7712
…
 30: step 1.0000, mean gap 10^-13.7609, pot -193.9617
```

When is IP useful?



- Newton: naively cubic in min(n,m)
  - unless we can take advantage of structure, limited to 1000s of variables
  - but structure often present!
- Convergence rate is on a different level from first-order methods: $\ln(1/\epsilon)$ vs. (at best) $1/\sqrt{\epsilon}$
  - and the latter requires more smoothness
  - so, great if accuracy requirements high / bad condition
- Intuition from IP/duality can help algorithm design