Convex Optimization in System and Control Theory

Stephen Boyd

Electrical Engineering Department
Stanford University

3rd SIAM Conf. Control & Applications

Basic idea

- Many problems arising in system and control theory can be cast as **convex optimization problems**
- Hence, are **fundamentally tractable**
- Recent interior-point methods can exploit problem structure to solve such problems **very efficiently**

Outline

- **Convex optimization**
- Some examples
- Interior-point methods

Convex optimization

minimize $f_0(x)$

subject to $f_1(x) \le 0, \ldots, f_L(x) \le 0$

$f_i : \mathbf{R}^n \to \mathbf{R}$ convex

- can have linear equality constraints
- differentiability not needed
- other formulations possible, e.g.:
  feasibility, multicriterion, monotone variational inequality

(Roughly speaking,)
convex optimization problems are **fundamentally tractable** in theory and practice

Algorithms:

- Classical optimization algorithms **do not** work
- Ellipsoid algorithm (Shor, Nemirovsky, Yudin 1970s)
  - very simple, universally applicable
  - efficient in terms of worst-case complexity theory
  - slow but robust in practice
- (General) interior-point methods (Nesterov, Nemirovsky 1980s)
  - efficient in theory and practice

Outline

- Convex optimization
- **Some examples**
- Interior-point methods

Well-known example: FIR filter design

transfer function: $H(z) \triangleq \sum_{i=0}^{n} h_i z^{-i}$

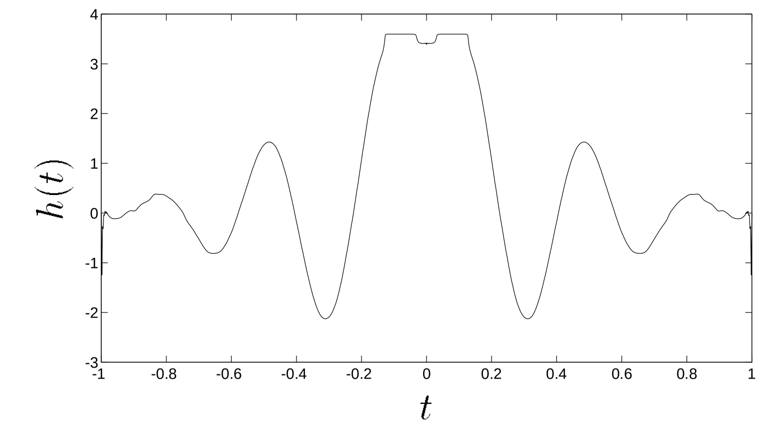
design variables: $x \triangleq [h_0 \; h_1 \; \ldots \; h_n]^T$

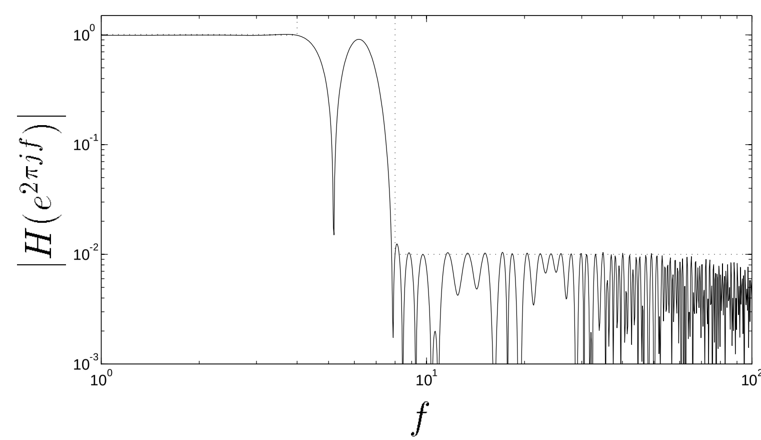
sample convex constraints:

- $H(e^{j0}) = 1$ (unity DC gain)
- $H(e^{j\omega_0}) = 0$ (notch at ω_0)
- $|H(e^{j\omega})| \leq 0.01$ for $\omega_s \leq \omega \leq \pi$
  (min. 40dB atten. in stop band)
- $|H(e^{j\omega})| \leq 1.12$ for $0 \leq \omega \leq \omega_b$
  (max. 1dB upper ripple in pass band)
- $h_i = h_{n-i}$ (linear phase constraint)
- $s(t) \triangleq \sum_{i=0}^{t} h_i \leq 1.1 H(e^{j0})$ (max. 10% step response overshoot)

FIR filter design example (M. Grant)

- sample rate $2/n$ sec^{-1}
- linear phase
- max ± 1dB ripple
  up to 0.4Hz
- min 40dB atten
  above 0.8Hz
- minimize $\max_i |h_i|$
- some solution times:
  $n = 255$: 5 sec
  $n = 2047$: 4 min

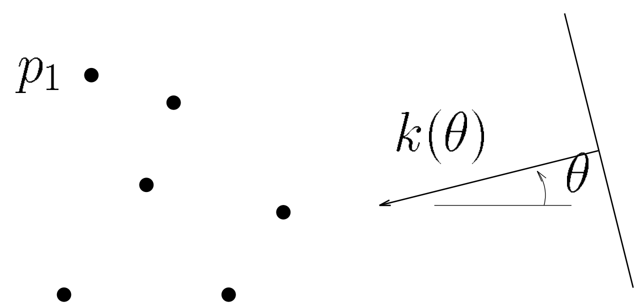




Beamforming

omnidirectional antenna elements at positions $p_1, \ldots, p_n \in \mathbf{R}^2$
plane wave incident from angle θ:

$$\exp j(k(\theta)^T p - \omega t), \qquad k(\theta) = -[\cos\theta \ \ \sin\theta]^T$$



demodulate to get $y_i = \exp(jk(\theta)^T p_i)$
form weighted sum $y(\theta) = \sum_{i=1}^{n} w_i y_i$

design variables: $x = [\mathbf{Re}\, w^T \ \mathbf{Im}\, w^T]^T$
(antenna array weights or shading coefficients)

$G(\theta) \triangleq |y(\theta)|$ antenna gain pattern

Sample convex constraints:

- $y(\theta_t) = 1$ (target direction normalization)
- $G(\theta_0) = 0$ (null in direction θ_0)
- w is real (amplitude only shading)
- $|w_i| \leq 1$ (attenuation only shading)

Sample convex objectives:

- $\max \{G(\theta) \mid |\theta - \theta_t| \geq 5^\circ\}$
  (sidelobe level with 10° beamwidth)
- $\sigma^2 \sum_i |w_i|^2$ (noise power in y)

Input design (trajectory planning)

Discrete-time linear system, input $u(t) \in \mathbf{R}^p$, output $y(t) \in \mathbf{R}^q$

Sample convex constraints:

- $|u_i(t)| \leq U$ (limit on input amplitude)
- $|u_i(t+1) - u_i(t)| \leq S$ (limit on input slew rate)
- $l_i(t) \leq y_i(t) \leq u_i(t)$ (envelope bounds for output)

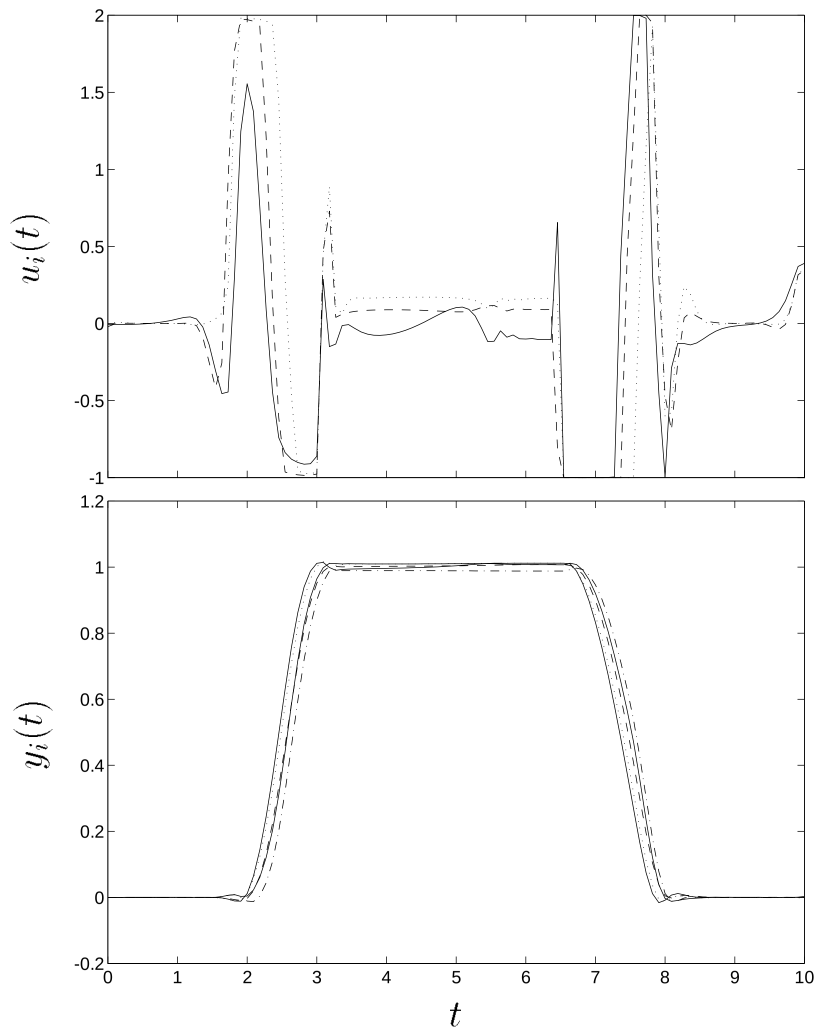
Sample convex objective:

- $\max_{t,i} |y_i(t) - y_i^{\text{des}}(t)|$ (peak tracking error)

Immediately extends to **multi-plant** (robust) case, predictive control, etc.

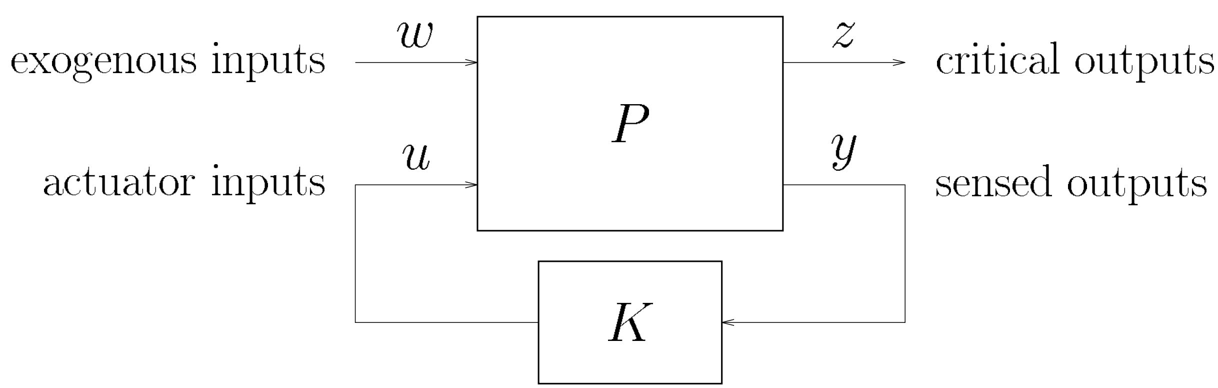
Input design example (M. Grant)

- rapid thermal processor
- 3 inputs, 8 outputs, 8 states
- amplitude limits on inputs
- slew limits on 3 outputs
- minimize peak tracking error on 5 outputs



Linear controller design

(static case for simplicity)



linear plant P given; design linear feedback controller K

closed-loop I/O relation: $z = Hw$,

$$H = P_{zw} + P_{zu}K(I - P_{yu}K)^{-1}P_{yw}$$

most specifications, objectives convex in H, **not** K

Transform to convex problem

Linear-fractional transformation:

$$Q \triangleq K(I - P_{yu}K)^{-1}$$

$H = P_{zw} + P_{zu}QP_{yw}$: constraints, objectives are convex in Q!

- design Q via convex programming
- set $K = Q(I + P_{yu}Q)^{-1}$

Extends to dynamic case ...

- time and frequency domain limits on actuator effort, regulation, tracking error
- some robustness specifications

Quadratic Lyapunov function search

Is there a quadratic Lyapunov function $V(z) \triangleq z^T P z$ that proves stability of differential inclusion

$$\dot{x}(t) = A(t)x(t), \qquad A(t) \in \mathbf{Co}\{A_1, \ldots, A_L\} \ ?$$

Equivalent to: is there P s.t.

$$P > 0, \qquad A_i^T P + P A_i < 0, \quad i = 1, \ldots, L \ ?$$

- a convex feasibility problem in P; no analytic solution but **readily solved**
- looks simple, but **more powerful** than many well known methods (multivariable circle criteria, ...)
- extends to a huge variety of other problems

Other examples

- synthesis of Lyapunov functions, state feedback
- filter/controller realization
- system identification problems
- truss design
- VLSI transistor sizing
- design centering
- computational geometry

Outline

- Convex optimization
- Some examples
- **Interior-point methods**

Interior-point convex programming methods

History:

- Dikin; Fiacco & McCormick's SUMT (1960s)
- Karmarkar's LP algorithm (1984); many more since then
- Nesterov & Nemirovsky's general formulation (1989)

General:

- # iterations small, grows slowly with problem size (typical number: 5 – 50)
- each iteration is basically least-squares problem

Semidefinite programming (SDP)

Semidefinite program:

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & F_0 + \sum_{i=1}^{m} x_i F_i \geq 0 \end{array}$$

$F_i = F_i^T \in \mathbf{R}^{n \times n}$, c are given.

- special but wide class of nonlinear convex problems
- can pose most system and control theory convex problems as SDPs
- powerful interior-point methods for SDPs recently developed (Nesterov & Nemirovsky, Alizadeh, others)

SDP examples

- **linear program:**

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \leq b \end{array}$$

as SDP: take $F(x) = \mathbf{diag}(b - Ax)$

- **matrix norm minimization:**

$$\text{minimize} \;\; \|A(x)\|,$$

$$A(x) \triangleq A_0 + x_1 A_1 + \cdots + x_k A_k$$

as SDP:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \begin{bmatrix} tI & A(x) \\ A(x)^T & tI \end{bmatrix} \geq 0 \end{array}$$

Duality

primal SDP:

$$p^* = \min \quad c^T x \quad \text{subject to} \quad F_0 + \sum_{i=1}^{m} x_i F_i \geq 0$$

dual SDP:

$$\begin{aligned} d^* = \max \quad & -\mathbf{Tr} F_0 Z \\ \text{subject to} \quad & \mathbf{Tr} F_i Z = c_i, \quad i = 1, \ldots, m \\ & Z \geq 0 \end{aligned}$$

- Z dual feasible $\Longrightarrow p^* \geq -\mathbf{Tr} F_0 Z$ (easy)
- $p^* = d^*$ (usually)

duality gap $\stackrel{\Delta}{=} c^T x + \mathbf{Tr} F_0 Z$

- gap ≥ 0, gap $= 0$ at optimum

Primal-dual potential function

for x, Z strictly feasible

$$\varphi(x, Z) \triangleq q \log(\mathrm{gap}) + \log\det F(x)^{-1} + \log\det Z^{-1}$$

$q > n$ is a parameter

- first term rewards decrease in gap
- second term keeps $F(x) > 0$
- third term keeps $Z > 0$

main properties:

- gap $\leq \exp \frac{1}{q-n} \varphi(x, Z)$
- φ unbounded below
- hence, can solve SDP by minimizing smooth function φ

Primal-dual potential reduction algorithm

initialization: $x = x^{(0)}$, $Z = Z^{(0)}$ strictly feasible
repeat

1. find search directions δx, δZ by solving least-squares problem†
2. plane search: minimize $\varphi\,(x + \alpha\,\delta x, Z + \beta\,\delta Z)$ over $\alpha, \beta \in \mathbf{R}$
3. update: $x := x + \alpha\,\delta x$; $Z := Z + \beta\,\delta Z$

until duality gap $\leq \epsilon$.

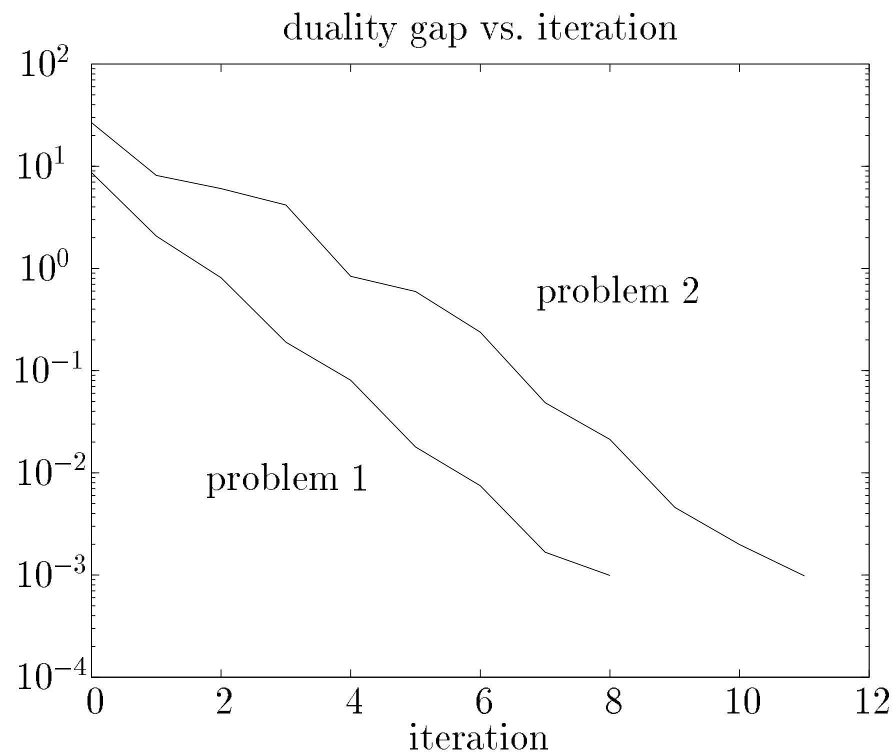
- **theorem:** $\varphi(x, Z) - \varphi(x^+, Z^+) \geq \delta > 0$
- **corollary:** (polynomial) convergence
- in practice, $\varphi(x, Z) - \varphi(x^+, Z^+) \gg \delta$

†least-squares problem, $\delta > 0$ depend on particular method

Typical example: matrix norm minimization

$$\text{minimize} \ \ \|A_0 + x_1 A_1 + \cdots + x_k A_k\|$$

two specific problems: 5 matrices, 5×5; 50 matrices, 50×50



Cost per iteration:
computing Newton direction, a **least squares problem** with same structure as original problem (Toeplitz, etc.)

Hence:
cost of solving **convex problem**
$\approx 5 - 50\times$ cost of solving similar **least-squares** problem

Hence:
can solve **least-squares problem** efficiently
$\Longrightarrow$ can solve **convex problem** efficiently

Exploiting problem structure via CG

Conjugate Gradients: solve $\min_x \|Ax - b\|$, $x \in \mathbf{R}^m$ via m evaluations of $x \to Ax$ and $y \to A^T y$

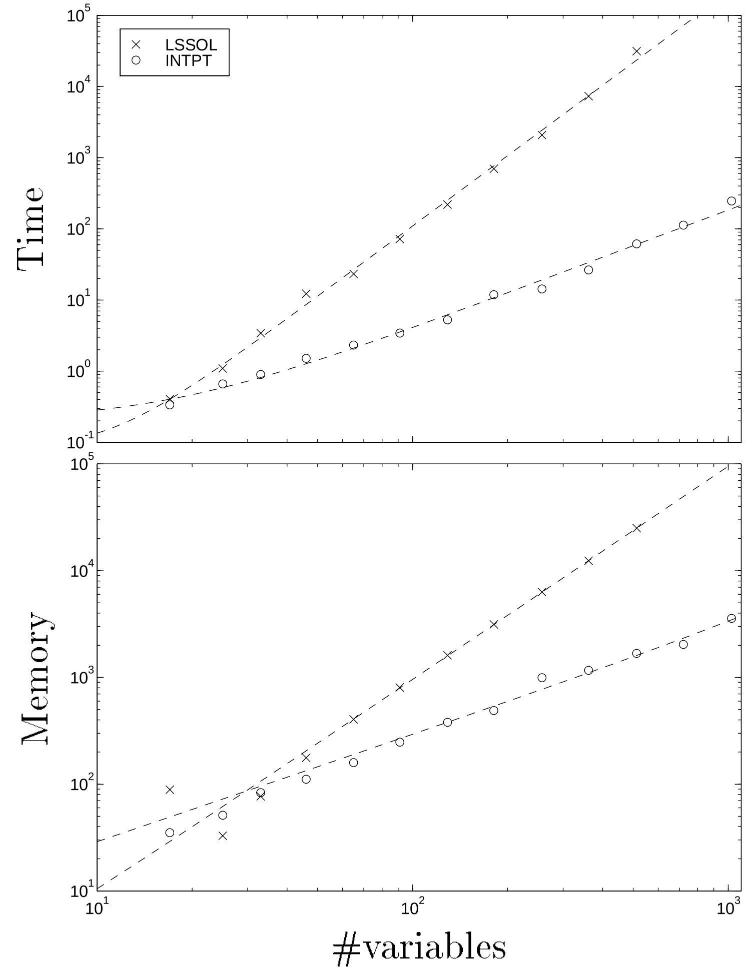
- roughly: can evaluate response and adjoint fast
  $\Longrightarrow$ can solve least-squares problem fast
  ($\Longrightarrow$ can solve convex problem fast)
- don't need exact solution for interior-point methods (allows early termination)
- preconditioning (problem specific)

Examples:

- FIR filter: fast ($N \log N$) convolution
- Input design: system state, co-state simulation
- Lyapunov function search: matrix (Kronecker) structure

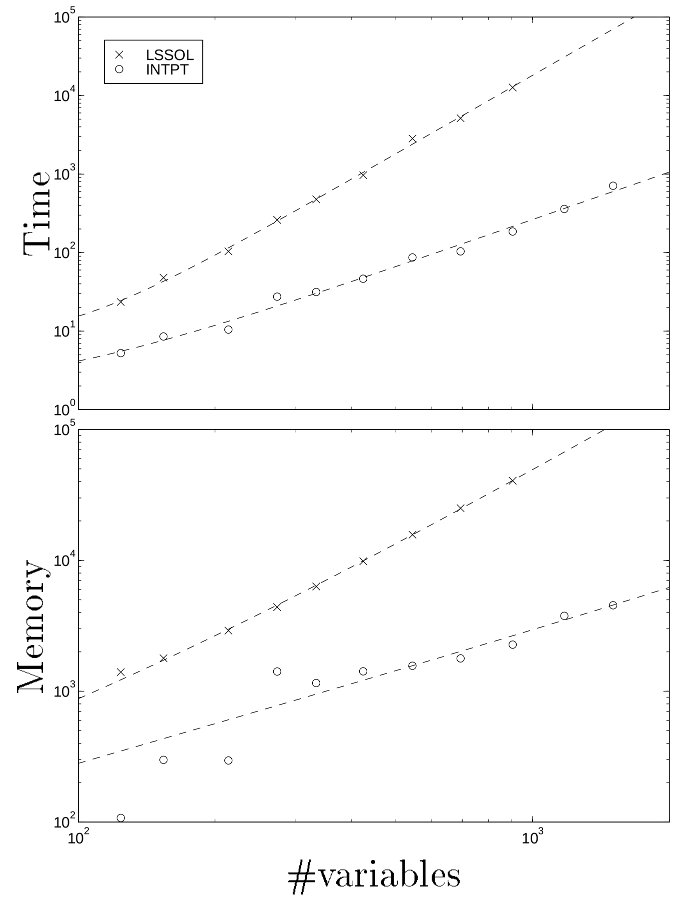
FIR filter design example (M. Grant)

- forward, adjoint operator: FFT
- #taps $\approx 2 \cdot$ #variables
- #constraints $\approx 10 \cdot$ #variables
- $>$ 1000 variables, $>$ 10000 constraints solved in 4 min, 4Mb



Input design example (M. Grant)

- forward, adjoint operator: state, co-state simulation
- #vbles $= 3 \cdot$ #time steps
- #constr $\approx 7 \cdot$ #vbles
- $>$ 1500 variables, $>$ 10000 constraints solved in 12 min, 5Mb

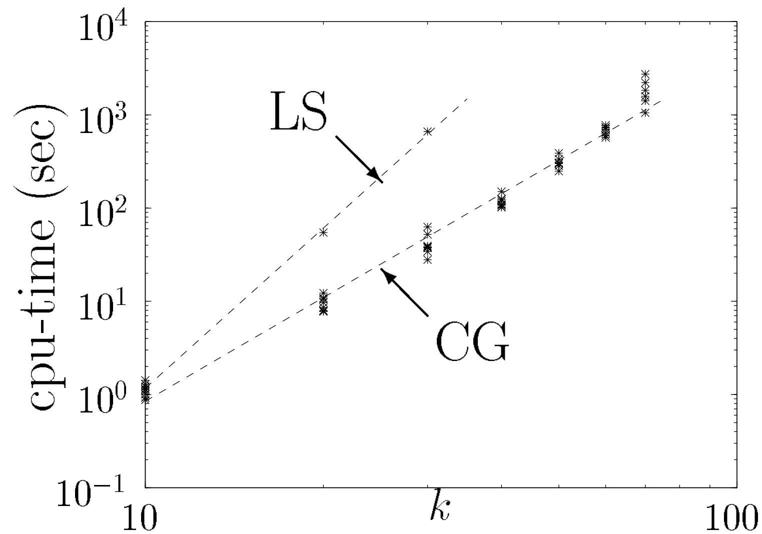


Lyapunov function search (Vandenberghe)

feasibility problem in $P = P^T \in \mathbf{R}^{k \times k}$

$$P > 0, \quad A_i^T P + P A_i < 0, \quad i = 1, \ldots, L$$

- two matrices, size $k \times k$
- $k = 10, \ldots, 70$
- # vbles $\approx 50, \ldots, 2500$
- LS: $O(k^{5.7})$; CG: $O(k^{3.9})$
- $k = 30$ (# vbles ≈ 450)
  LS: 12 min; CG: 30 sec



Exploiting structure in convex problems

can evaluate response, adjoint fast *(exploiting structure)*

$\Downarrow$

can solve least-squares problem fast *(using conj grad)*

$\Downarrow$

can solve convex problem fast *(using int-pt methods)*

Main point

- Many problems arising in engineering analysis and design can be cast as **convex optimization problems**
- Hence, can be efficiently solved by **interior-point methods** that **exploit problem structure**

(A few) references

- Nesterov and Nemirovsky, *Interior-point polynomial algorithms in convex programming*, SIAM, 1994.
- Boyd, El Ghaoui, Feron, Balakrishnan, *Linear matrix inequalities in system and control theory*, SIAM, 1994.
- Vandenberghe and Boyd, *Semidefinite programming*, ftp `isl.stanford.edu`

Software

- **LMI-lab** (Gahinet, Nemirovsky, Laub, Chilali)
  Matlab toolbox for control analysis/design
- **LMI-tool** (El Ghaoui, Delebecque, Nikoukhah)
  Matlab (ftp `ftp.ensta.fr`)
- **SP** (Vandenberghe, Boyd)
  C with Matlab interface (ftp `isl.stanford.edu`)
- **SDPSOL** (Boyd, Wu)
  parser/solver for SDPs (ftp `isl.stanford.edu`)

. . . the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity.

— R. Rockafellar, SIAM Review 1993