Sequential Convex Programming

- sequential convex programming
- alternating convex optimization
- convex-concave procedure

Methods for nonconvex optimization problems

- **convex optimization methods** are (roughly) always global, always fast
- for general nonconvex problems, we have to give up one
  - **local optimization methods** are fast, but need not find global solution (and even when they do, cannot certify it)
  - **global optimization methods** find global solution (and certify it), but are not always fast (indeed, are often slow)
- **this lecture**: local optimization methods that are based on solving a sequence of convex problems

Sequential convex programming (SCP)

- a local optimization method for nonconvex problems that leverages convex optimization
  - convex portions of a problem are handled 'exactly' and efficiently
- SCP is a **heuristic**
  - it can fail to find optimal (or even feasible) point
  - results can (and often do) depend on starting point
    (can run algorithm from many initial points and take best result)
- SCP often works well, *i.e.*, finds a feasible point with good, if not optimal, objective value

Problem

we consider nonconvex problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad j = 1, \ldots, p \end{array}$$

with variable $x \in \mathbf{R}^n$

- f_0 and f_i (possibly) nonconvex
- h_i (possibly) non-affine

Basic idea of SCP

- maintain estimate of solution $x^{(k)}$, and convex **trust region** $\mathcal{T}^{(k)} \subset \mathbf{R}^n$
- form convex approximation $\hat{f}_i$ of f_i over trust region $\mathcal{T}^{(k)}$
- form affine approximation $\hat{h}_i$ of h_i over trust region $\mathcal{T}^{(k)}$
- $x^{(k+1)}$ is optimal point for approximate convex problem

$$\begin{array}{ll} \text{minimize} & \hat{f}_0(x) \\ \text{subject to} & \hat{f}_i(x) \leq 0, \quad i = 1, \ldots, m \\ & \hat{h}_i(x) = 0, \quad i = 1, \ldots, p \\ & x \in \mathcal{T}^{(k)} \end{array}$$

Trust region

- typical trust region is box around current point:

$$\mathcal{T}^{(k)} = \{x \mid |x_i - x_i^{(k)}| \leq \rho_i, \ i = 1, \ldots, n\}$$

- if x_i appears only in convex inequalities and affine equalities, can take $\rho_i = \infty$

Affine and convex approximations via Taylor expansions

- (affine) first order Taylor expansion:

$$\hat{f}(x) = f(x^{(k)}) + \nabla f(x^{(k)})^T (x - x^{(k)})$$

- (convex part of) second order Taylor expansion:

$$\hat{f}(x) = f(x^{(k)}) + \nabla f(x^{(k)})^T (x - x^{(k)}) + (1/2)(x - x^{(k)})^T P (x - x^{(k)})$$

$P = \left(\nabla^2 f(x^{(k)})\right)_+$, PSD part of Hessian

- give local approximations, which don't depend on trust region radii ρ_i

Particle method

- particle method:
  - choose points $z_1, \ldots, z_K \in \mathcal{T}^{(k)}$
    ($e.g.$, all vertices, some vertices, grid, random, . . .)
  - evaluate $y_i = f(z_i)$
  - fit data (z_i, y_i) with convex (affine) function
    (using convex optimization)

- advantages:
  - handles nondifferentiable functions, or functions for which evaluating derivatives is difficult
  - gives **regional models**, which depend on current point and trust region radii ρ_i

Fitting affine or quadratic functions to data

fit convex quadratic function to data (z_i, y_i)

$$
\begin{array}{ll}
\mbox{minimize} & \sum_{i=1}^K \left((z_i - x^{(k)})^T P (z_i - x^{(k)}) + q^T (z_i - x^{(k)}) + r - y_i \right)^2 \\
\mbox{subject to} & P \succeq 0
\end{array}
$$

with variables $P \in \mathbf{S}^n$, $q \in \mathbf{R}^n$, $r \in \mathbf{R}$

- can use other objectives, add other convex constraints
- no need to solve exactly
- this problem is solved for each nonconvex constraint, each SCP step

Quasi-linearization

- a cheap and simple method for affine approximation
- write $h(x)$ as $A(x)x + b(x)$ (many ways to do this)
- use $\hat{h}(x) = A(x^{(k)})x + b(x^{(k)})$
- example:

$$h(x) = (1/2)x^T Px + q^T x + r = ((1/2)Px + q)^T x + r$$

- $\hat{h}_{\mathrm{ql}}(x) = ((1/2)Px^{(k)} + q)^T x + r$
- $\hat{h}_{\mathrm{tay}}(x) = (Px^{(k)} + q)^T (x - x^{(k)}) + h(x^{(k)})$

Example

- nonconvex QP

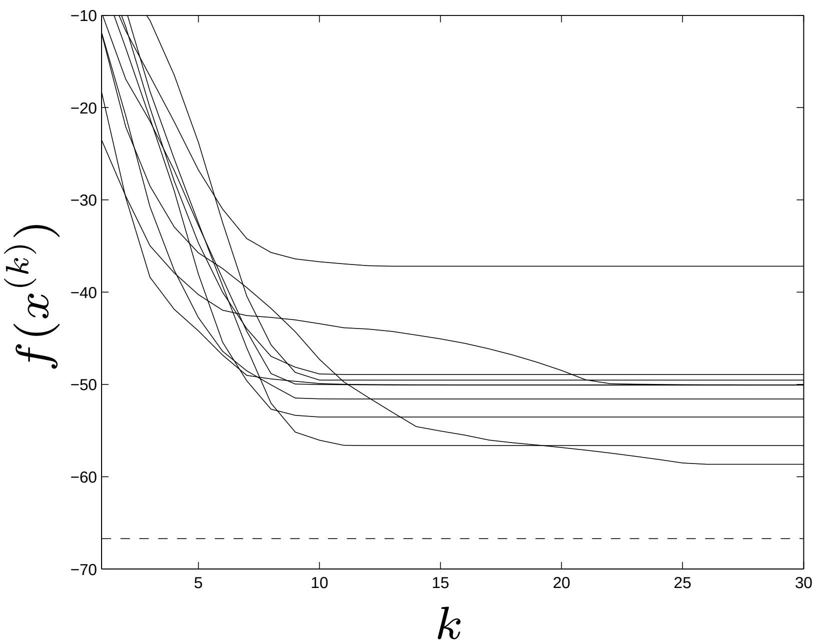
$$\begin{array}{ll} \text{minimize} & f(x) = (1/2)x^T P x + q^T x \\ \text{subject to} & \|x\|_\infty \leq 1 \end{array}$$

  with P symmetric but not PSD

- use approximation

$$f(x^{(k)}) + (Px^{(k)} + q)^T (x - x^{(k)}) + (1/2)(x - x^{(k)})^T P_+ (x - x^{(k)})$$

- example with $x \in \mathbf{R}^{20}$
- SCP with $\rho = 0.2$, started from 10 different points



- runs typically converge to points between -60 and -50
- dashed line shows lower bound on optimal value ≈ -66.5

Lower bound via Lagrange dual

- write constraints as $x_i^2 \leq 1$ and form Lagrangian

$$
\begin{aligned}
L(x,\lambda) &= (1/2)x^TPx + q^Tx + \sum_{i=1}^n \lambda_i(x_i^2-1) \\
&= (1/2)x^T\left(P + 2\,\mathbf{diag}(\lambda)\right)x + q^Tx - \mathbf{1}^T\lambda
\end{aligned}
$$

- $g(\lambda) = -(1/2)q^T\left(P+2\,\mathbf{diag}(\lambda)\right)^{-1}q - \mathbf{1}^T\lambda$; need $P + 2\,\mathbf{diag}(\lambda) \succ 0$

- solve dual problem to get best lower bound:

$$
\begin{array}{ll}
\text{maximize} & -(1/2)q^T\left(P+2\,\mathbf{diag}(\lambda)\right)^{-1}q - \mathbf{1}^T\lambda \\
\text{subject to} & \lambda \succeq 0, \quad P + 2\,\mathbf{diag}(\lambda) \succ 0
\end{array}
$$

Some (related) issues

- approximate convex problem can be infeasible

- how do we evaluate progress when $x^{(k)}$ isn't feasible?
  need to take into account
  - objective $f_0(x^{(k)})$
  - inequality constraint violations $f_i(x^{(k)})_+$
  - equality constraint violations $|h_i(x^{(k)})|$

- controlling the trust region size
  - ρ too large: approximations are poor, leading to bad choice of $x^{(k+1)}$
  - ρ too small: approximations are good, but progress is slow

Exact penalty formulation

- instead of original problem, we solve unconstrained problem

$$\text{minimize} \quad \phi(x) = f_0(x) + \lambda \left(\sum_{i=1}^m f_i(x)_+ + \sum_{i=1}^p |h_i(x)|\right)$$

  where $\lambda > 0$

- for λ large enough, minimizer of ϕ is solution of original problem
- for SCP, use convex approximation

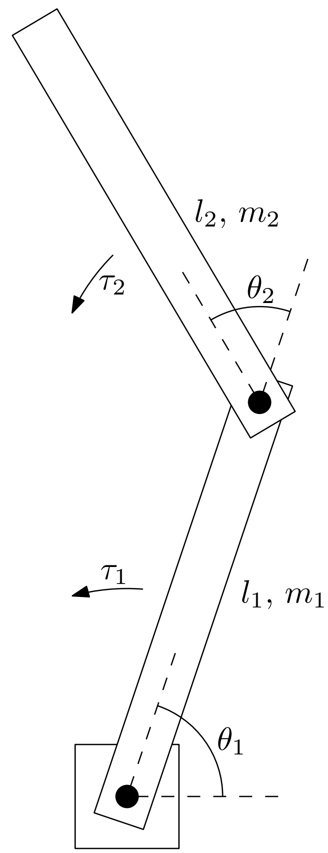
$$\hat{\phi}(x) = \hat{f}_0(x) + \lambda \left(\sum_{i=1}^m \hat{f}_i(x)_+ + \sum_{i=1}^p |\hat{h}_i(x)| \right)$$

- approximate problem always feasible

Trust region update

- judge algorithm progress by decrease in ϕ, using solution $\tilde{x}$ of approximate problem
- decrease with approximate objective: $\hat{\delta} = \phi(x^{(k)}) - \hat{\phi}(\tilde{x})$ (called *predicted decrease*)
- decrease with exact objective: $\delta = \phi(x^{(k)}) - \phi(\tilde{x})$
- if $\delta \geq \alpha\hat{\delta}$, $\rho^{(k+1)} = \beta^{\rm succ}\rho^{(k)}$, $x^{(k+1)} = \tilde{x}$ ($\alpha \in (0,1)$, $\beta^{\rm succ} \geq 1$; typical values $\alpha = 0.1$, $\beta^{\rm succ} = 1.1$)
- if $\delta < \alpha\hat{\delta}$, $\rho^{(k+1)} = \beta^{\rm fail}\rho^{(k)}$, $x^{(k+1)} = x^{(k)}$ ($\beta^{\rm fail} \in (0,1)$; typical value $\beta^{\rm fail} = 0.5$)
- interpretation: if actual decrease is more (less) than fraction α of predicted decrease then increase (decrease) trust region size

Nonlinear optimal control



- 2-link system, controlled by torques τ_1 and τ_2 (no gravity)

- dynamics given by $M(\theta)\ddot{\theta} + W(\theta, \dot{\theta})\dot{\theta} = \tau$, with

$$\begin{aligned}
M(\theta) &= \begin{bmatrix} (m_1+m_2)l_1^2 & m_2 l_1 l_2 (s_1 s_2 + c_1 c_2) \\ m_2 l_1 l_2 (s_1 s_2 + c_1 c_2) & m_2 l_2^2 \end{bmatrix} \\
W(\theta, \dot{\theta}) &= \begin{bmatrix} 0 & m_2 l_1 l_2 (s_1 c_2 - c_1 s_2)\dot{\theta}_2 \\ m_2 l_1 l_2 (s_1 c_2 - c_1 s_2)\dot{\theta}_1 & 0 \end{bmatrix}
\end{aligned}$$

$s_i = \sin\theta_i$, $c_i = \cos\theta_i$

- nonlinear optimal control problem:

$$\begin{array}{ll}
\text{minimize} & J = \int_0^T \|\tau(t)\|_2^2 \, dt \\
\text{subject to} & \theta(0) = \theta_{\rm init}, \quad \dot{\theta}(0) = 0, \quad \theta(T) = \theta_{\rm final}, \quad \dot{\theta}(T) = 0 \\
& \|\tau(t)\|_\infty \leq \tau_{\max}, \quad 0 \leq t \leq T
\end{array}$$

Discretization

- discretize with time interval $h = T/N$
- $J \approx h \sum_{i=1}^{N} \|\tau_i\|_2^2$, with $\tau_i = \tau(ih)$
- approximate derivatives as

$$\dot{\theta}(ih) \approx \frac{\theta_{i+1} - \theta_{i-1}}{2h}, \qquad \ddot{\theta}(ih) \approx \frac{\theta_{i+1} - 2\theta_i + \theta_{i-1}}{h^2}$$

- approximate dynamics as set of nonlinear equality constraints:

$$M(\theta_i)\frac{\theta_{i+1} - 2\theta_i + \theta_{i-1}}{h^2} + W\left(\theta_i, \frac{\theta_{i+1} - \theta_{i-1}}{2h}\right)\frac{\theta_{i+1} - \theta_{i-1}}{2h} = \tau_i$$

- $\theta_0 = \theta_1 = \theta_{\mathrm{init}}$; $\theta_N = \theta_{N+1} = \theta_{\mathrm{final}}$

- discretized nonlinear optimal control problem:

$$
\begin{array}{ll}
\text{minimize} & h \sum_{i=1}^{N} \|\tau_i\|_2^2 \\
\text{subject to} & \theta_0 = \theta_1 = \theta_{\rm init}, \quad \theta_N = \theta_{N+1} = \theta_{\rm final} \\
& \|\tau_i\|_\infty \leq \tau_{\max}, \quad i = 1, \ldots, N \\
& M(\theta_i) \frac{\theta_{i+1} - 2\theta_i + \theta_{i-1}}{h^2} + W\left(\theta_i, \frac{\theta_{i+1} - \theta_{i-1}}{2h}\right) \frac{\theta_{i+1} - \theta_{i-1}}{2h} = \tau_i
\end{array}
$$

- replace equality constraints with quasilinearized versions

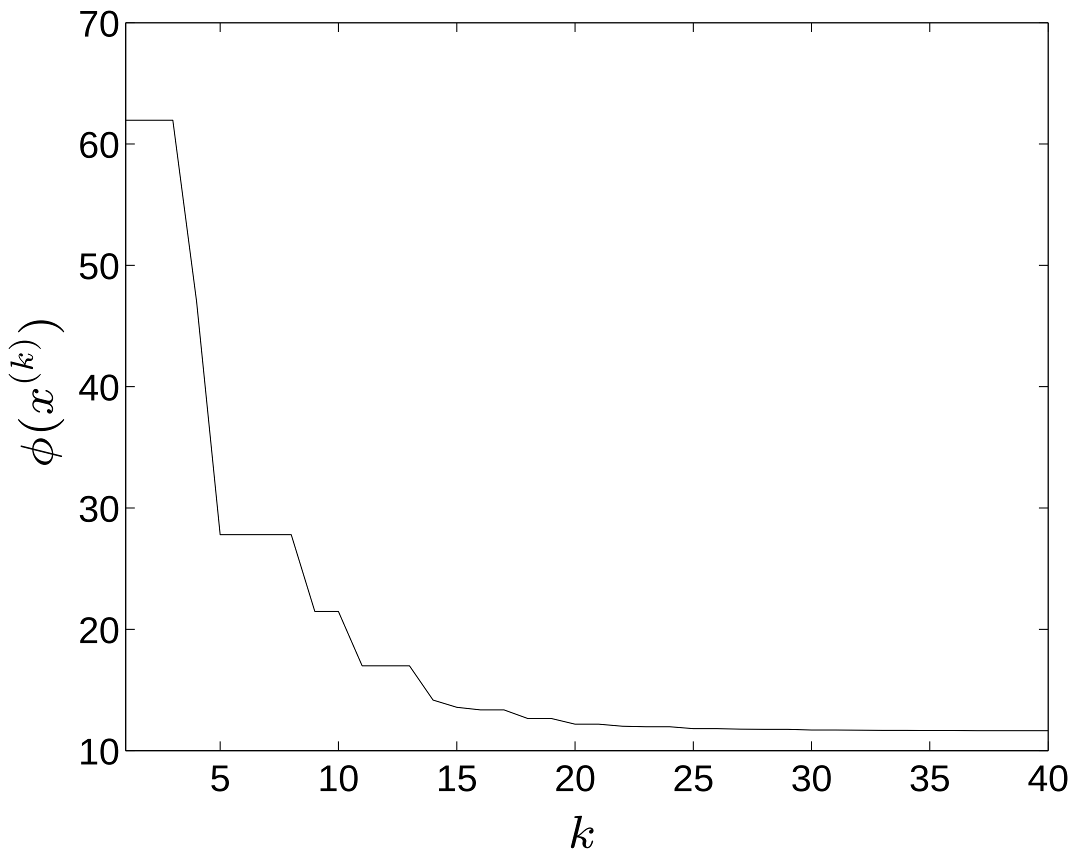
$$
M(\theta_i^{(k)}) \frac{\theta_{i+1} - 2\theta_i + \theta_{i-1}}{h^2} + W\left(\theta_i^{(k)}, \frac{\theta_{i+1}^{(k)} - \theta_{i-1}^{(k)}}{2h}\right) \frac{\theta_{i+1} - \theta_{i-1}}{2h} = \tau_i
$$

- trust region: only on θ_i
- initialize with $\theta_i = ((i-1)/(N-1))(\theta_{\rm final} - \theta_{\rm init})$, $i = 1, \ldots, N$

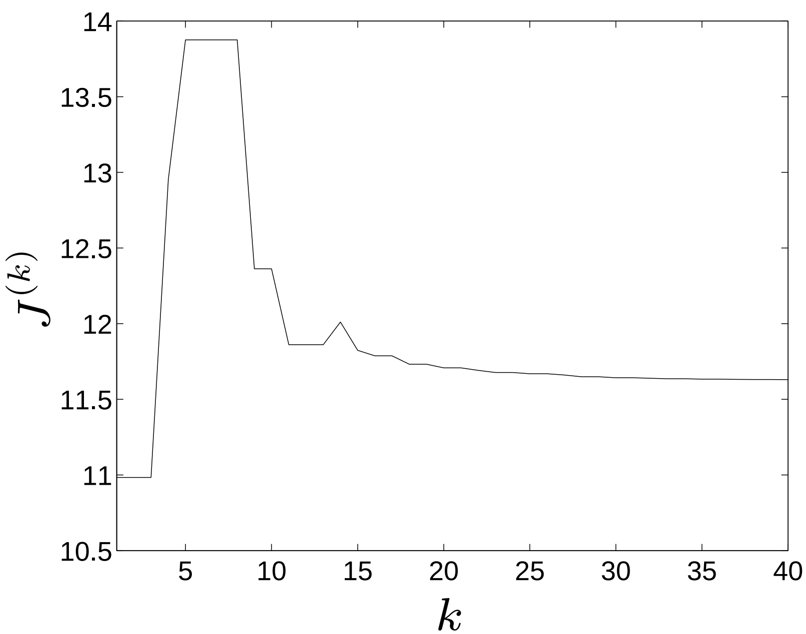
Numerical example

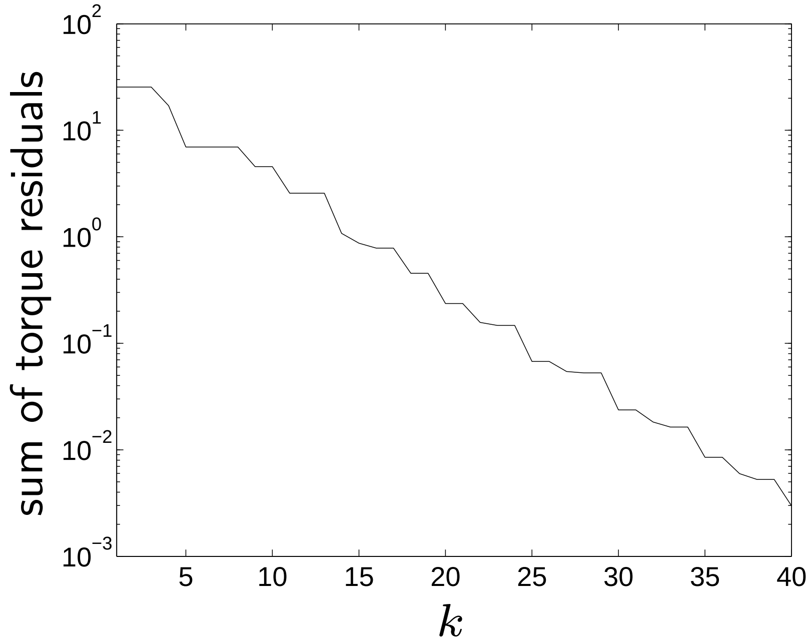
- $m_1 = 1$, $m_2 = 5$, $l_1 = 1$, $l_2 = 1$
- $N = 40$, $T = 10$
- $\theta_{\mathrm{init}} = (0, -2.9)$, $\theta_{\mathrm{final}} = (3, 2.9)$
- $\tau_{\mathrm{max}} = 1.1$
- $\alpha = 0.1$, $\beta^{\mathrm{succ}} = 1.1$, $\beta^{\mathrm{fail}} = 0.5$, $\rho^{(1)} = 90^\circ$
- $\lambda = 2$

SCP progress

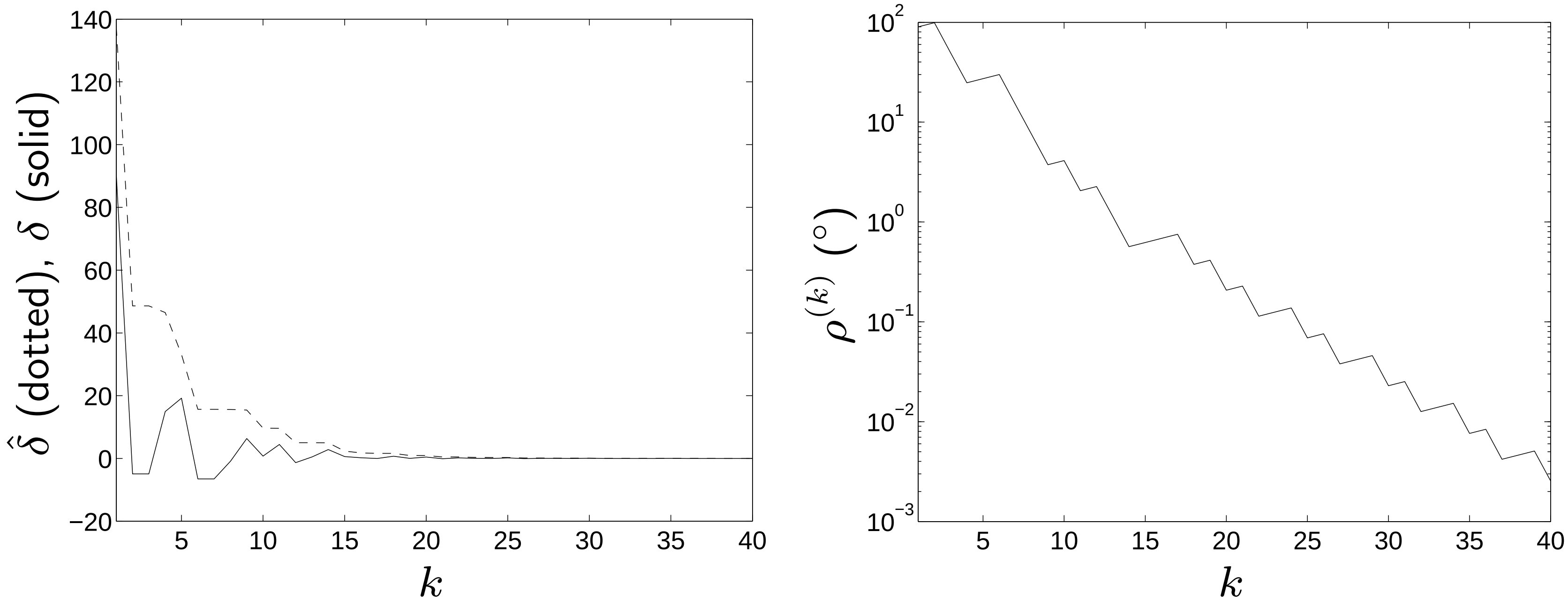


Convergence of J and torque residuals

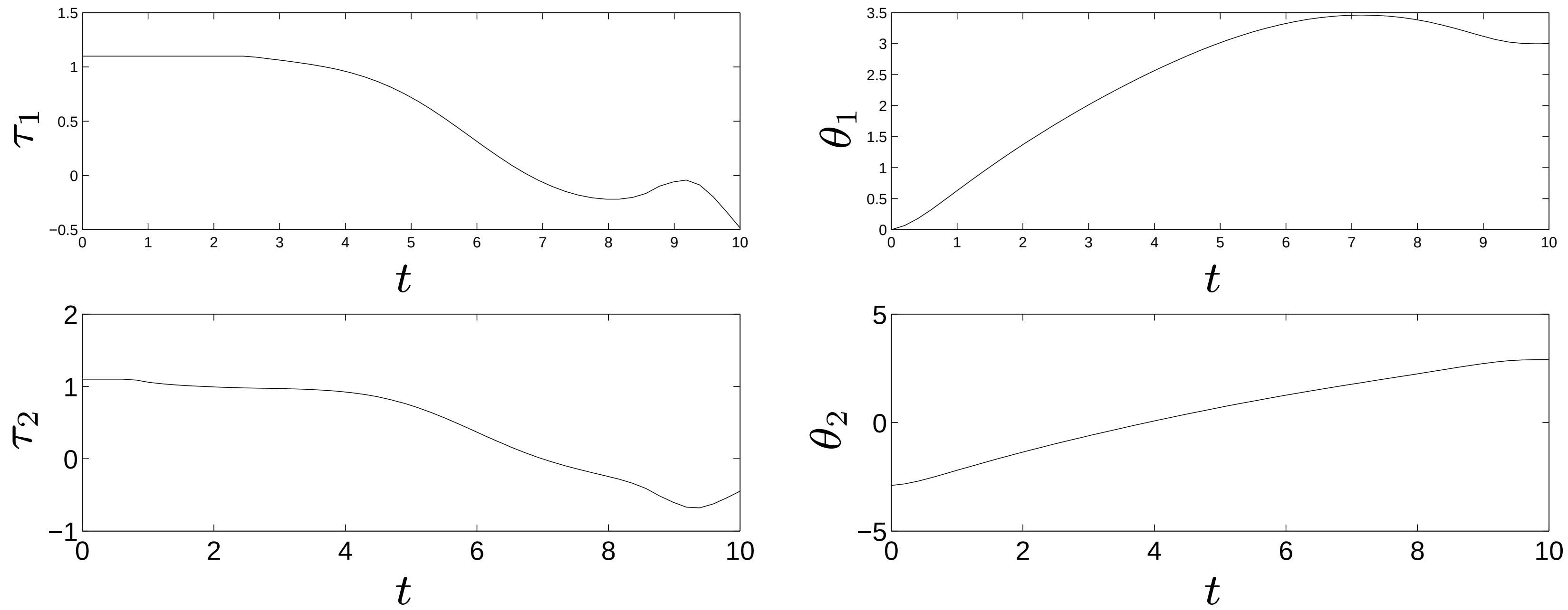




Predicted and actual decreases in ϕ



Trajectory plan



‘Difference of convex’ programming

- express problem as

$$\begin{array}{ll} \text{minimize} & f_0(x) - g_0(x) \\ \text{subject to} & f_i(x) - g_i(x) \le 0, \quad i = 1, \ldots, m \end{array}$$

  where f_i and g_i are convex

- $f_i - g_i$ are called ‘difference of convex’ functions
- problem is sometimes called ‘difference of convex programming’

Convex-concave procedure

- obvious convexification at $x^{(k)}$: replace $f(x) - g(x)$ with

$$\hat{f}(x) = f(x) - g(x^{(k)}) - \nabla g(x^{(k)})^T (x - x^{(k)})$$

- since $\hat{f}(x) \geq f(x)$ for all x, no trust region is needed
  - true objective at $\tilde{x}$ is better than convexified objective
  - true feasible set contains feasible set for convexified problem

- SCP sometimes called 'convex-concave procedure'

Example (BV §7.1)

- given samples $y_1, \ldots, y_N \in \mathbf{R}^n$ from $\mathcal{N}(0, \Sigma^{\rm true})$
- negative log-likelihood function is

$$f(\Sigma) = \log\det\Sigma + \mathbf{Tr}(\Sigma^{-1}Y), \qquad Y = (1/N)\sum_{i=1}^N y_i y_i^T$$

(dropping a constant and positive scale factor)

- ML estimate of Σ, with prior knowledge $\Sigma_{ij} \geq 0$:

$$\begin{array}{ll} \mbox{minimize} & f(\Sigma) = \log\det\Sigma + \mathbf{Tr}(\Sigma^{-1}Y) \\ \mbox{subject to} & \Sigma_{ij} \geq 0, \quad i,j = 1,\ldots,n \end{array}$$

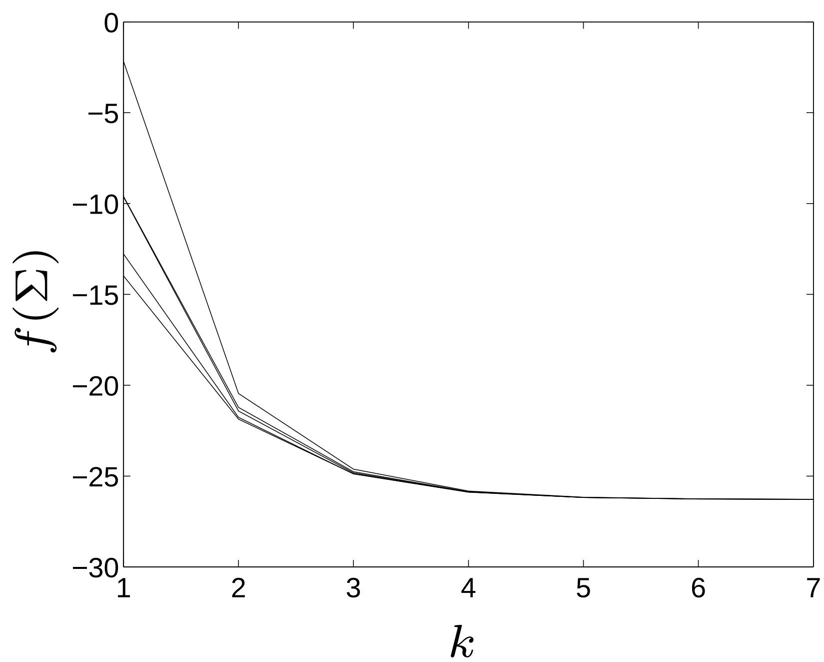
with variable Σ (constraint $\Sigma \succ 0$ is implicit)

- first term in f is concave; second term is convex
- linearize first term in objective to get

$$\hat{f}(\Sigma) = \log\det\Sigma^{(k)} + \mathbf{Tr}\left((\Sigma^{(k)})^{-1}(\Sigma - \Sigma^{(k)})\right) + \mathbf{Tr}(\Sigma^{-1}Y)$$

Numerical example

convergence of problem instance with $n = 10$, $N = 15$



Alternating convex optimization

- given nonconvex problem with variable $(x_1, \ldots, x_n) \in \mathbf{R}^n$
- $\mathcal{I}_1, \ldots, \mathcal{I}_k \subset \{1, \ldots, n\}$ are index subsets with $\bigcup_j \mathcal{I}_j = \{1, \ldots, n\}$
- suppose problem is convex in subset of variables x_i, $i \in \mathcal{I}_j$, when x_i, $i \notin \mathcal{I}_j$ are fixed
- alternating convex optimization method: cycle through j, in each step optimizing over variables x_i, $i \in \mathcal{I}_j$
- special case: bi-convex problem
  - $x = (u, v)$; problem is convex in u (v) with v (u) fixed
  - alternate optimizing over u and v

Nonnegative matrix factorization

- NMF problem:

$$\begin{array}{ll} \text{minimize} & \|A - XY\|_F \\ \text{subject to} & X_{ij},\ Y_{ij} \geq 0 \end{array}$$

  variables $X \in \mathbf{R}^{m \times k}$, $Y \in \mathbf{R}^{k \times n}$, data $A \in \mathbf{R}^{m \times n}$

- difficult problem, except for a few special cases ($e.g.$, $k = 1$)

- alternating convex optimation: solve QPs to optimize over X, then Y, then X . . .

Example

- convergence for example with $m = n = 50$, $k = 5$ (five starting points)