9. Numerical linear algebra background

- matrix structure and algorithm complexity
- solving linear equations with factored matrices
- LU, Cholesky, LDL^T factorization
- block elimination and the matrix inversion lemma
- solving underdetermined equations

Matrix structure and algorithm complexity

cost (execution time) of solving $Ax = b$ with $A \in \mathbf{R}^{n\times n}$

- for general methods, grows as n^3
- less if A is structured (banded, sparse, Toeplitz, . . .)

flop counts

- flop (floating-point operation): one addition, subtraction, multiplication, or division of two floating-point numbers
- to estimate complexity of an algorithm: express number of flops as a (polynomial) function of the problem dimensions, and simplify by keeping only the leading terms
- not an accurate predictor of computation time on modern computers
- useful as a rough estimate of complexity

vector-vector operations (x, $y \in \mathbf{R}^n$)

- inner product $x^T y$: $2n - 1$ flops (or $2n$ if n is large)
- sum $x + y$, scalar multiplication αx: n flops

matrix-vector product $y = Ax$ with $A \in \mathbf{R}^{m \times n}$

- $m(2n - 1)$ flops (or $2mn$ if n large)
- $2N$ if A is sparse with N nonzero elements
- $2p(n + m)$ if A is given as $A = UV^T$, $U \in \mathbf{R}^{m \times p}$, $V \in \mathbf{R}^{n \times p}$

matrix-matrix product $C = AB$ with $A \in \mathbf{R}^{m \times n}$, $B \in \mathbf{R}^{n \times p}$

- $mp(2n - 1)$ flops (or $2mnp$ if n large)
- less if A and/or B are sparse
- $(1/2)m(m + 1)(2n - 1) \approx m^2 n$ if $m = p$ and C symmetric

Linear equations that are easy to solve

diagonal matrices ($a_{ij} = 0$ if $i \neq j$): n flops

$$x = A^{-1}b = (b_1/a_{11}, \ldots, b_n/a_{nn})$$

lower triangular ($a_{ij} = 0$ if $j > i$): n^2 flops

$$\begin{array}{rcl} x_1 & := & b_1/a_{11} \\ x_2 & := & (b_2 - a_{21}x_1)/a_{22} \\ x_3 & := & (b_3 - a_{31}x_1 - a_{32}x_2)/a_{33} \\ & \vdots & \\ x_n & := & (b_n - a_{n1}x_1 - a_{n2}x_2 - \cdots - a_{n,n-1}x_{n-1})/a_{nn} \end{array}$$

called forward substitution

upper triangular ($a_{ij} = 0$ if $j < i$): n^2 flops via backward substitution

orthogonal matrices: $A^{-1} = A^T$

- $2n^2$ flops to compute $x = A^T b$ for general A
- less with structure, *e.g.*, if $A = I - 2uu^T$ with $\|u\|_2 = 1$, we can compute $x = A^T b = b - 2(u^T b)u$ in $4n$ flops

permutation matrices:

$$a_{ij} = \begin{cases} 1 & j = \pi_i \\ 0 & \text{otherwise} \end{cases}$$

where $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ is a permutation of $(1, 2, \ldots, n)$

- interpretation: $Ax = (x_{\pi_1}, \ldots, x_{\pi_n})$
- satisfies $A^{-1} = A^T$, hence cost of solving $Ax = b$ is 0 flops

example:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \qquad A^{-1} = A^T = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

The factor-solve method for solving $Ax = b$

- factor A as a product of simple matrices (usually 2 or 3):

$$A = A_1 A_2 \cdots A_k$$

  (A_i diagonal, upper or lower triangular, etc)

- compute $x = A^{-1}b = A_k^{-1} \cdots A_2^{-1} A_1^{-1} b$ by solving k 'easy' equations

$$A_1 x_1 = b, \qquad A_2 x_2 = x_1, \qquad \ldots, \qquad A_k x = x_{k-1}$$

cost of factorization step usually dominates cost of solve step

equations with multiple righthand sides

$$A x_1 = b_1, \qquad A x_2 = b_2, \qquad \ldots, \qquad A x_m = b_m$$

cost: one factorization plus m solves

LU factorization

every nonsingular matrix A can be factored as

$$A = PLU$$

with P a permutation matrix, L lower triangular, U upper triangular

cost: $(2/3)n^3$ flops

Solving linear equations by LU factorization.

given a set of linear equations $Ax = b$, with A nonsingular.

1. *LU factorization.* Factor A as $A = PLU$ ($(2/3)n^3$ flops).
2. *Permutation.* Solve $Pz_1 = b$ (0 flops).
3. *Forward substitution.* Solve $Lz_2 = z_1$ (n^2 flops).
4. *Backward substitution.* Solve $Ux = z_2$ (n^2 flops).

---

cost: $(2/3)n^3 + 2n^2 \approx (2/3)n^3$ for large n

sparse LU factorization

$$A = P_1 L U P_2$$

- adding permutation matrix P_2 offers possibility of sparser L, U (hence, cheaper factor and solve steps)

- P_1 and P_2 chosen (heuristically) to yield sparse L, U

- choice of P_1 and P_2 depends on sparsity pattern and values of A

- cost is usually much less than $(2/3)n^3$; exact value depends in a complicated way on n, number of zeros in A, sparsity pattern

Cholesky factorization

every positive definite A can be factored as

$$A = LL^T$$

with L lower triangular

cost: $(1/3)n^3$ flops

Solving linear equations by Cholesky factorization.

given a set of linear equations $Ax = b$, with $A \in \mathbf{S}^n_{++}$.

1. *Cholesky factorization.* Factor A as $A = LL^T$ ($(1/3)n^3$ flops).
2. *Forward substitution.* Solve $Lz_1 = b$ (n^2 flops).
3. *Backward substitution.* Solve $L^T x = z_1$ (n^2 flops).

---

cost: $(1/3)n^3 + 2n^2 \approx (1/3)n^3$ for large n

sparse Cholesky factorization

$$A = PLL^T P^T$$

- adding permutation matrix P offers possibility of sparser L
- P chosen (heuristically) to yield sparse L
- choice of P only depends on sparsity pattern of A (unlike sparse LU)
- cost is usually much less than $(1/3)n^3$; exact value depends in a complicated way on n, number of zeros in A, sparsity pattern

LDL$^\mathsf{T}$ factorization

every nonsingular symmetric matrix A can be factored as

$$A = PLDL^TP^T$$

with P a permutation matrix, L lower triangular, D block diagonal with 1×1 or 2×2 diagonal blocks

cost: $(1/3)n^3$

- cost of solving symmetric sets of linear equations by LDL$^\mathsf{T}$ factorization: $(1/3)n^3 + 2n^2 \approx (1/3)n^3$ for large n
- for sparse A, can choose P to yield sparse L; cost $\ll (1/3)n^3$

Equations with structured sub-blocks

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{1}$$

- variables $x_1 \in \mathbf{R}^{n_1}$, $x_2 \in \mathbf{R}^{n_2}$; blocks $A_{ij} \in \mathbf{R}^{n_i \times n_j}$
- if A_{11} is nonsingular, can eliminate x_1: $x_1 = A_{11}^{-1}(b_1 - A_{12}x_2)$; to compute x_2, solve

$$(A_{22} - A_{21}A_{11}^{-1}A_{12})x_2 = b_2 - A_{21}A_{11}^{-1}b_1$$

Solving linear equations by block elimination.

given a nonsingular set of linear equations (1), with A_{11} nonsingular.

1. Form $A_{11}^{-1}A_{12}$ and $A_{11}^{-1}b_1$.
2. Form $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$ and $\tilde{b} = b_2 - A_{21}A_{11}^{-1}b_1$.
3. Determine x_2 by solving $Sx_2 = \tilde{b}$.
4. Determine x_1 by solving $A_{11}x_1 = b_1 - A_{12}x_2$.

---

dominant terms in flop count

- step 1: $f + n_2 s$ (f is cost of factoring A_{11}; s is cost of solve step)
- step 2: $2n_2^2 n_1$ (cost dominated by product of A_{21} and $A_{11}^{-1}A_{12}$)
- step 3: $(2/3)n_2^3$

total: $f + n_2 s + 2n_2^2 n_1 + (2/3)n_2^3$

examples

- general A_{11} ($f = (2/3)n_1^3$, $s = 2n_1^2$): no gain over standard method

$$\#\text{flops} = (2/3)n_1^3 + 2n_1^2 n_2 + 2n_2^2 n_1 + (2/3)n_2^3 = (2/3)(n_1 + n_2)^3$$

- block elimination is useful for structured A_{11} ($f \ll n_1^3$)

  for example, diagonal ($f = 0$, $s = n_1$): $\#\text{flops} \approx 2n_2^2 n_1 + (2/3)n_2^3$

Structured matrix plus low rank term

$$(A + BC)x = b$$

- $A \in \mathbf{R}^{n \times n}$, $B \in \mathbf{R}^{n \times p}$, $C \in \mathbf{R}^{p \times n}$
- assume A has structure ($Ax = b$ easy to solve)

first write as

$$\begin{bmatrix} A & B \\ C & -I \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

now apply block elimination: solve

$$(I + CA^{-1}B)y = CA^{-1}b,$$

then solve $Ax = b - By$

this proves the **matrix inversion lemma:** if A and $A + BC$ nonsingular,

$$(A + BC)^{-1} = A^{-1} - A^{-1}B(I + CA^{-1}B)^{-1}CA^{-1}$$

example: A diagonal, B, C dense

- method 1: form $D = A + BC$, then solve $Dx = b$

  cost: $(2/3)n^3 + 2pn^2$

- method 2 (via matrix inversion lemma): solve

$$(I + CA^{-1}B)y = A^{-1}b, \tag{2}$$

  then compute $x = A^{-1}b - A^{-1}By$

  total cost is dominated by (2): $2p^2n + (2/3)p^3$ ($i.e.$, linear in n)

Underdetermined linear equations

if $A \in \mathbf{R}^{p \times n}$ with $p < n$, $\mathbf{rank}\, A = p$,

$$\{x \mid Ax = b\} = \{Fz + \hat{x} \mid z \in \mathbf{R}^{n-p}\}$$

- $\hat{x}$ is (any) particular solution
- columns of $F \in \mathbf{R}^{n \times (n-p)}$ span nullspace of A
- there exist several numerical methods for computing F (QR factorization, rectangular LU factorization, ...)

10. Unconstrained minimization

- terminology and assumptions
- gradient descent method
- steepest descent method
- Newton's method
- self-concordant functions
- implementation

Unconstrained minimization

$$\text{minimize} \quad f(x)$$

- f convex, twice continuously differentiable (hence $\textbf{dom}\, f$ open)
- we assume optimal value $p^\star = \inf_x f(x)$ is attained (and finite)

unconstrained minimization methods

- produce sequence of points $x^{(k)} \in \textbf{dom}\, f$, $k = 0, 1, \ldots$ with

$$f(x^{(k)}) \rightarrow p^\star$$

- can be interpreted as iterative methods for solving optimality condition

$$\nabla f(x^\star) = 0$$

Initial point and sublevel set

algorithms in this chapter require a starting point $x^{(0)}$ such that

- $x^{(0)} \in \mathbf{dom}\, f$
- sublevel set $S = \{x \mid f(x) \leq f(x^{(0)})\}$ is closed

2nd condition is hard to verify, except when *all* sublevel sets are closed:

- equivalent to condition that $\mathbf{epi}\, f$ is closed
- true if $\mathbf{dom}\, f = \mathbf{R}^n$
- true if $f(x) \to \infty$ as $x \to \mathbf{bd}\,\mathbf{dom}\, f$

examples of differentiable functions with closed sublevel sets:

$$f(x) = \log(\sum_{i=1}^{m} \exp(a_i^T x + b_i)), \qquad f(x) = -\sum_{i=1}^{m} \log(b_i - a_i^T x)$$

Strong convexity and implications

f is strongly convex on S if there exists an $m > 0$ such that

$$\nabla^2 f(x) \succeq mI \qquad \text{for all } x \in S$$

implications

- for $x, y \in S$,

  $$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{m}{2}\|x - y\|_2^2$$

  hence, S is bounded

- $p^\star > -\infty$, and for $x \in S$,

  $$f(x) - p^\star \leq \frac{1}{2m}\|\nabla f(x)\|_2^2$$

  useful as stopping criterion (if you know m)

Descent methods

$$x^{(k+1)} = x^{(k)} + t^{(k)} \Delta x^{(k)} \quad \text{with } f(x^{(k+1)}) < f(x^{(k)})$$

- other notations: $x^+ = x + t\Delta x$, $x := x + t\Delta x$
- Δx is the *step*, or *search direction*; t is the *step size*, or *step length*
- from convexity, $f(x^+) < f(x)$ implies $\nabla f(x)^T \Delta x < 0$ ($i.e.$, Δx is a *descent direction*)

General descent method.

given a starting point $x \in \mathbf{dom}\, f$.
repeat
1. Determine a descent direction Δx.
2. *Line search.* Choose a step size $t > 0$.
3. *Update.* $x := x + t\Delta x$.

until stopping criterion is satisfied.

Line search types

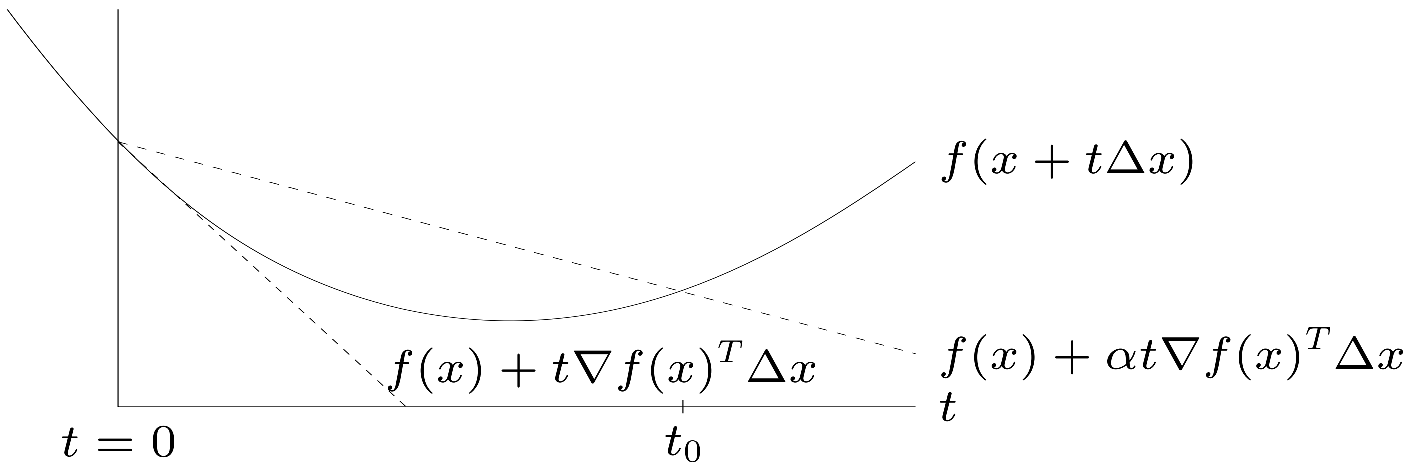
exact line search: $t = \mathrm{argmin}_{t>0} f(x + t\Delta x)$

backtracking line search (with parameters $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$)

- starting at $t = 1$, repeat $t := \beta t$ until

$$f(x + t\Delta x) < f(x) + \alpha t \nabla f(x)^T \Delta x$$

- graphical interpretation: backtrack until $t \leq t_0$



Gradient descent method

general descent method with $\Delta x = -\nabla f(x)$

---

given a starting point $x \in \mathbf{dom}\, f$.
repeat
1. $\Delta x := -\nabla f(x)$.
2. *Line search.* Choose step size t via exact or backtracking line search.
3. *Update.* $x := x + t\Delta x$.

until stopping criterion is satisfied.

- stopping criterion usually of the form $\|\nabla f(x)\|_2 \leq \epsilon$
- convergence result: for strongly convex f,

$$f(x^{(k)}) - p^\star \leq c^k (f(x^{(0)}) - p^\star)$$

  $c \in (0, 1)$ depends on m, $x^{(0)}$, line search type
- very simple, but often very slow; rarely used in practice

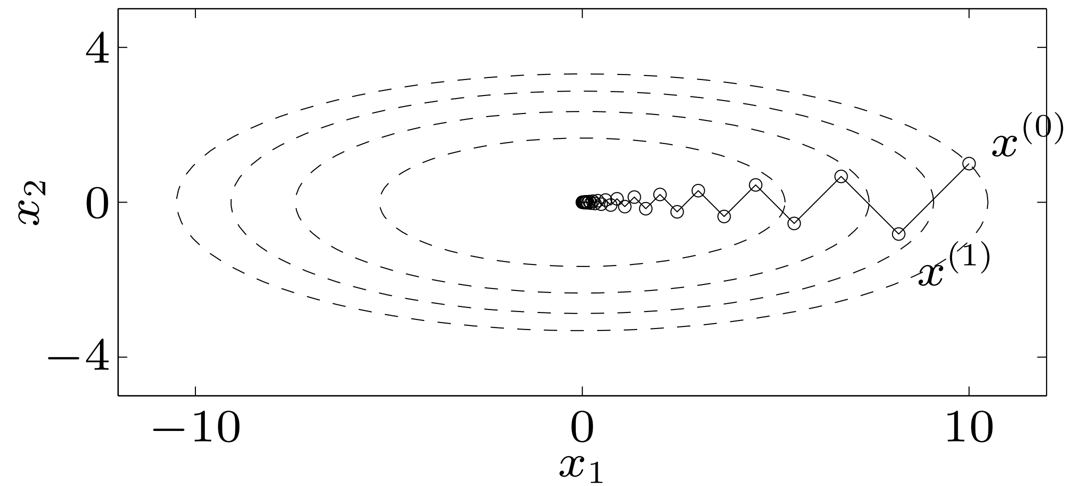
quadratic problem in $\mathbf{R}^2$

$$f(x) = (1/2)(x_1^2 + \gamma x_2^2) \qquad\qquad (\gamma > 0)$$

with exact line search, starting at $x^{(0)} = (\gamma, 1)$:

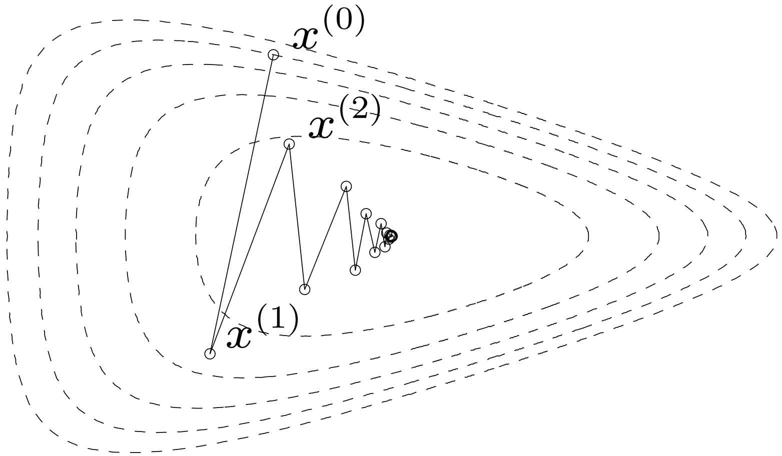
$$x_1^{(k)} = \gamma \left(\frac{\gamma - 1}{\gamma + 1}\right)^k, \qquad x_2^{(k)} = \left(-\frac{\gamma - 1}{\gamma + 1}\right)^k$$

- very slow if $\gamma \gg 1$ or $\gamma \ll 1$
- example for $\gamma = 10$:

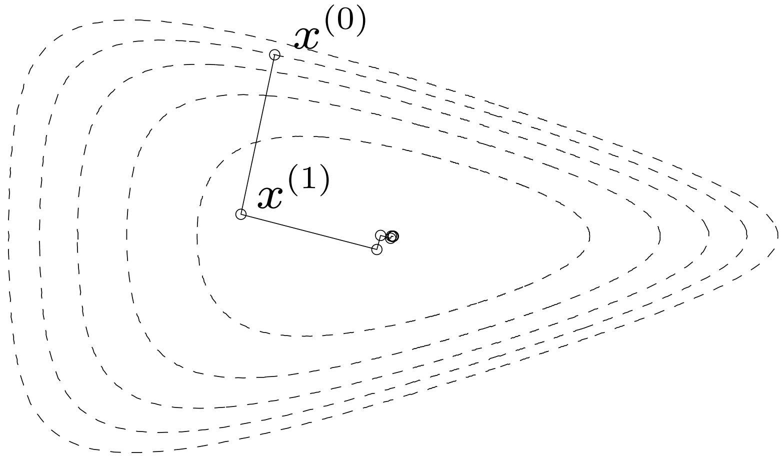


nonquadratic example

$$f(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}$$



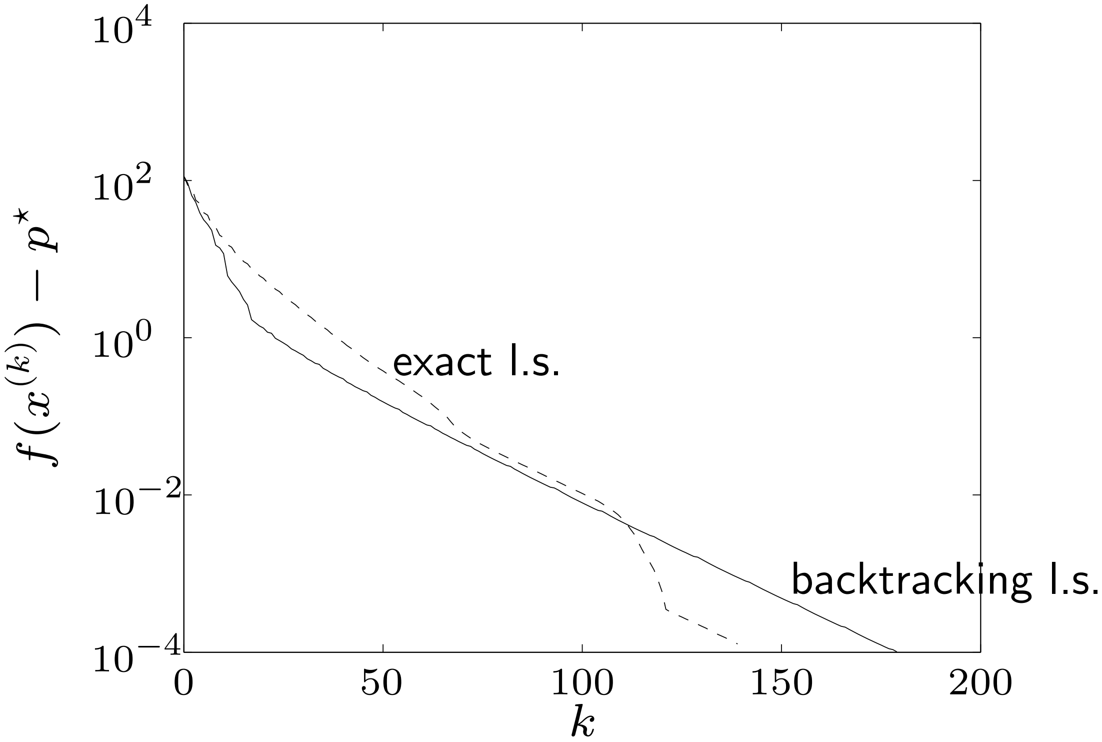
backtracking line search



exact line search

a problem in $\mathbf{R}^{100}$

$$f(x) = c^T x - \sum_{i=1}^{500} \log(b_i - a_i^T x)$$



'linear' convergence, *i.e.*, a straight line on a semilog plot

Steepest descent method

normalized steepest descent direction (at x, for norm $\|\cdot\|$):

$$\Delta x_{\mathrm{nsd}} = \mathrm{argmin}\{\nabla f(x)^T v \mid \|v\| = 1\}$$

interpretation: for small v, $f(x+v) \approx f(x) + \nabla f(x)^T v$;
direction Δx_{nsd} is unit-norm step with most negative directional derivative

(unnormalized) steepest descent direction

$$\Delta x_{\mathrm{sd}} = \|\nabla f(x)\|_* \Delta x_{\mathrm{nsd}}$$

satisfies $\nabla f(x)^T \Delta_{\mathrm{sd}} = -\|\nabla f(x)\|_*^2$

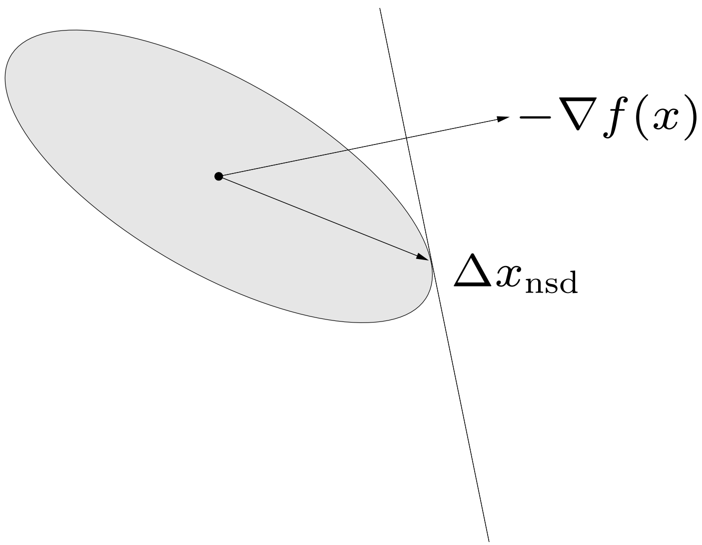
steepest descent method

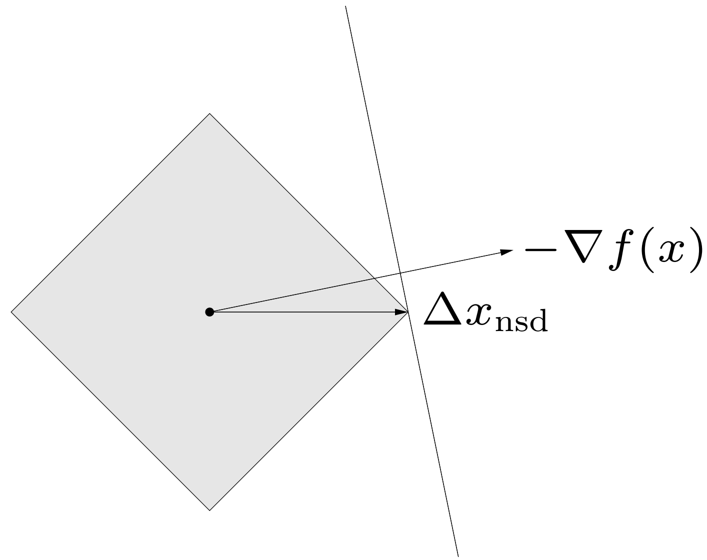
- general descent method with $\Delta x = \Delta x_{\mathrm{sd}}$
- convergence properties similar to gradient descent

examples

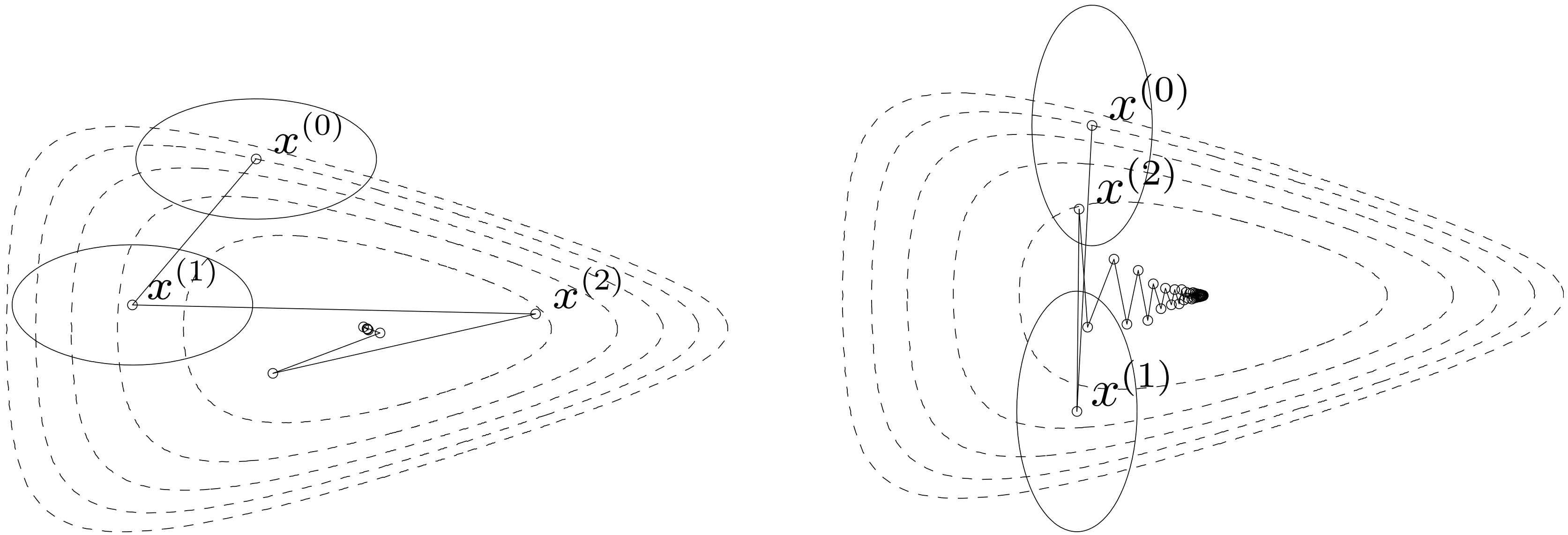
- Euclidean norm: $\Delta x_{\mathrm{sd}} = -\nabla f(x)$
- quadratic norm $\|x\|_P = (x^T P x)^{1/2}$ ($P \in \mathbf{S}^n_{++}$): $\Delta x_{\mathrm{sd}} = -P^{-1}\nabla f(x)$
- ℓ_1-norm: $\Delta x_{\mathrm{sd}} = -(\partial f(x)/\partial x_i)e_i$, where $|\partial f(x)/\partial x_i| = \|\nabla f(x)\|_\infty$

unit balls and normalized steepest descent directions for a quadratic norm and the ℓ_1-norm:





choice of norm for steepest descent



- steepest descent with backtracking line search for two quadratic norms
- ellipses show $\{x \mid \|x - x^{(k)}\|_P = 1\}$
- equivalent interpretation of steepest descent with quadratic norm $\|\cdot\|_P$: gradient descent after change of variables $\bar{x} = P^{1/2}x$

shows choice of P has strong effect on speed of convergence

Newton step

$$\Delta x_{\rm nt} = -\nabla^2 f(x)^{-1}\nabla f(x)$$

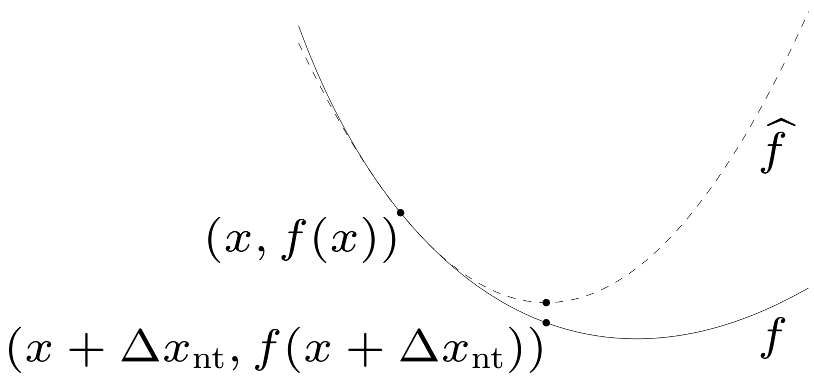
interpretations

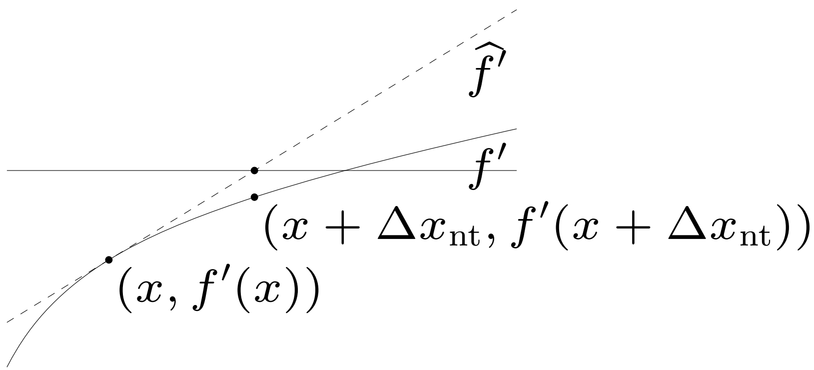
- $x + \Delta x_{\rm nt}$ minimizes second order approximation

$$\widehat{f}(x+v) = f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v$$

- $x + \Delta x_{\rm nt}$ solves linearized optimality condition

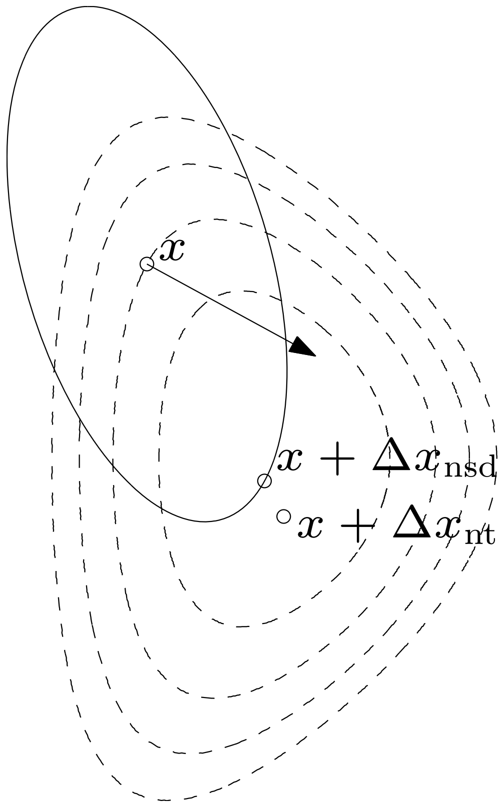
$$\nabla f(x+v) \approx \nabla \widehat{f}(x+v) = \nabla f(x) + \nabla^2 f(x) v = 0$$





- Δx_{nt} is steepest descent direction at x in local Hessian norm

$$\|u\|_{\nabla^2 f(x)} = \left(u^T \nabla^2 f(x) u\right)^{1/2}$$



dashed lines are contour lines of f; ellipse is $\{x + v \mid v^T \nabla^2 f(x) v = 1\}$

arrow shows $-\nabla f(x)$

Newton decrement

$$\lambda(x) = \left(\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x)\right)^{1/2}$$

a measure of the proximity of x to $x^\star$

properties

- gives an estimate of $f(x) - p^\star$, using quadratic approximation $\widehat{f}$:

$$f(x) - \inf_y \widehat{f}(y) = \frac{1}{2}\lambda(x)^2$$

- equal to the norm of the Newton step in the quadratic Hessian norm

$$\lambda(x) = \left(\Delta x_{\mathrm{nt}} \nabla^2 f(x) \Delta x_{\mathrm{nt}}\right)^{1/2}$$

- directional derivative in the Newton direction: $\nabla f(x)^T \Delta x_{\mathrm{nt}} = -\lambda(x)^2$
- affine invariant (unlike $\|\nabla f(x)\|_2$)

Newton's method

---

given a starting point $x \in \mathbf{dom}\, f$, tolerance $\epsilon > 0$.
repeat

1. *Compute the Newton step and decrement.*
   $\Delta x_{\rm nt} := -\nabla^2 f(x)^{-1} \nabla f(x); \quad \lambda^2 := \nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x).$
2. *Stopping criterion.* **quit** if $\lambda^2/2 \leq \epsilon$.
3. *Line search.* Choose step size t by backtracking line search.
4. *Update.* $x := x + t\Delta x_{\rm nt}$.

---

affine invariant, $i.e.$, independent of linear changes of coordinates:

Newton iterates for $\tilde{f}(y) = f(Ty)$ with starting point $y^{(0)} = T^{-1}x^{(0)}$ are

$$y^{(k)} = T^{-1}x^{(k)}$$

Classical convergence analysis

assumptions

- f strongly convex on S with constant m
- $\nabla^2 f$ is Lipschitz continuous on S, with constant $L > 0$:

$$\|\nabla^2 f(x) - \nabla^2 f(y)\|_2 \leq L\|x - y\|_2$$

(L measures how well f can be approximated by a quadratic function)

outline: there exist constants $\eta \in (0, m^2/L)$, $\gamma > 0$ such that

- if $\|\nabla f(x)\|_2 \geq \eta$, then $f(x^{(k+1)}) - f(x^{(k)}) \leq -\gamma$
- if $\|\nabla f(x)\|_2 < \eta$, then

$$\frac{L}{2m^2}\|\nabla f(x^{(k+1)})\|_2 \leq \left(\frac{L}{2m^2}\|\nabla f(x^{(k)})\|_2\right)^2$$

damped Newton phase ($\|\nabla f(x)\|_2 \geq \eta$)

- most iterations require backtracking steps
- function value decreases by at least γ
- if $p^\star > -\infty$, this phase ends after at most $(f(x^{(0)}) - p^\star)/\gamma$ iterations

quadratically convergent phase ($\|\nabla f(x)\|_2 < \eta$)

- all iterations use step size $t = 1$
- $\|\nabla f(x)\|_2$ converges to zero quadratically: if $\|\nabla f(x^{(k)})\|_2 < \eta$, then

$$\frac{L}{2m^2}\|\nabla f(x^l)\|_2 \leq \left(\frac{L}{2m^2}\|\nabla f(x^k)\|_2\right)^{2^{l-k}} \leq \left(\frac{1}{2}\right)^{2^{l-k}}, \qquad l \geq k$$

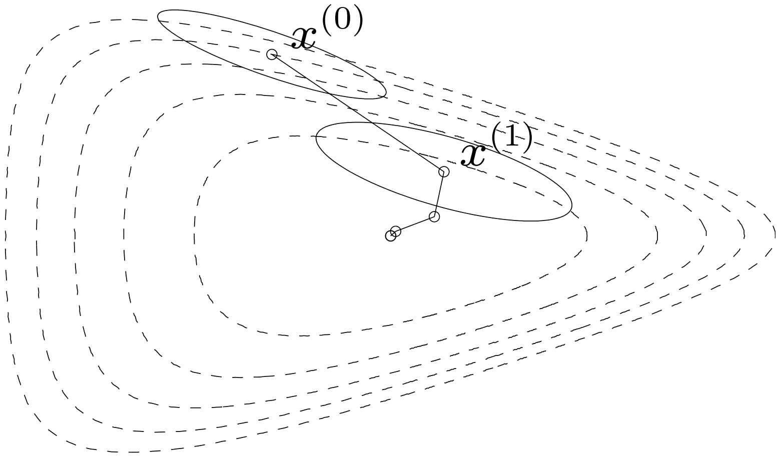
conclusion: number of iterations until $f(x) - p^\star \le \epsilon$ is bounded above by

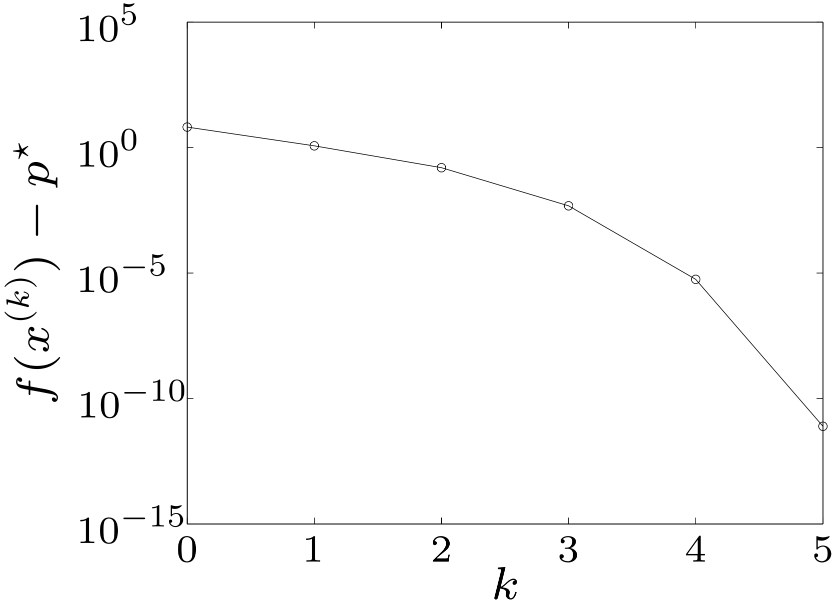
$$\frac{f(x^{(0)}) - p^\star}{\gamma} + \log_2 \log_2(\epsilon_0/\epsilon)$$

- γ, ϵ_0 are constants that depend on m, L, $x^{(0)}$
- second term is small (of the order of 6) and almost constant for practical purposes
- in practice, constants m, L (hence γ, ϵ_0) are usually unknown
- provides qualitative insight in convergence properties ($i.e.$, explains two algorithm phases)

Examples

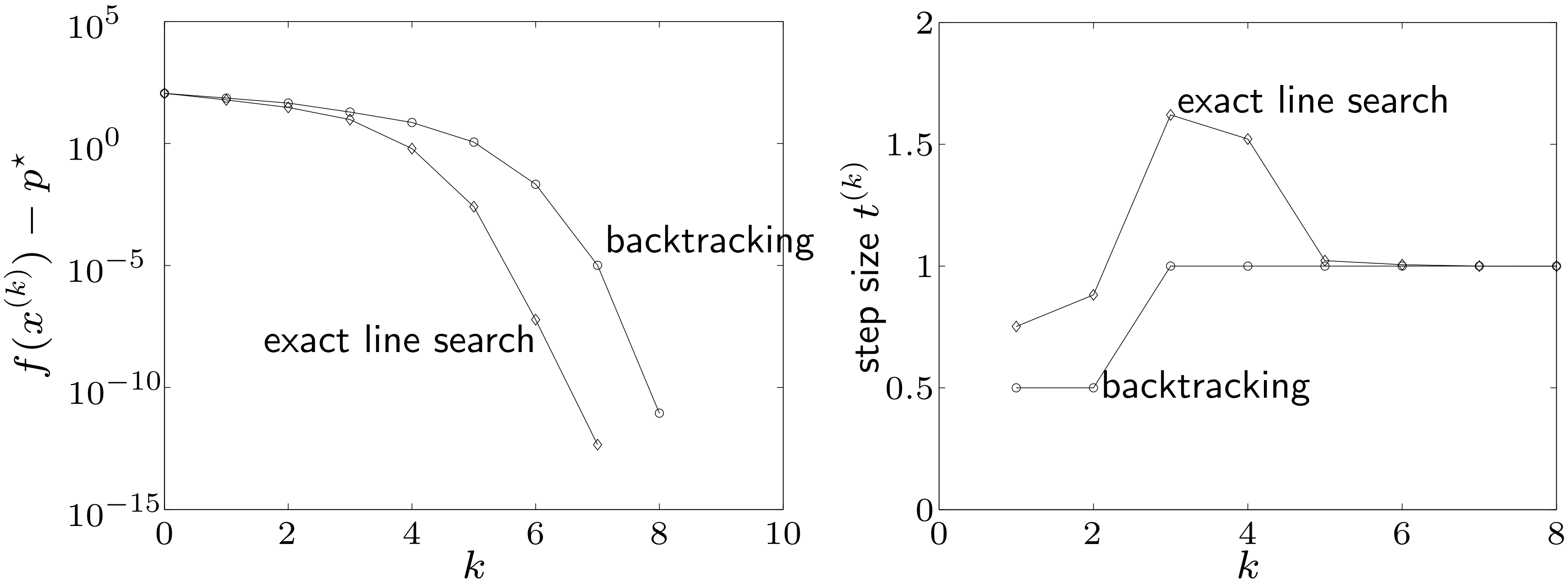
example in $\mathbf{R}^2$ (page 10–9)





- backtracking parameters $\alpha = 0.1$, $\beta = 0.7$
- converges in only 5 steps
- quadratic local convergence

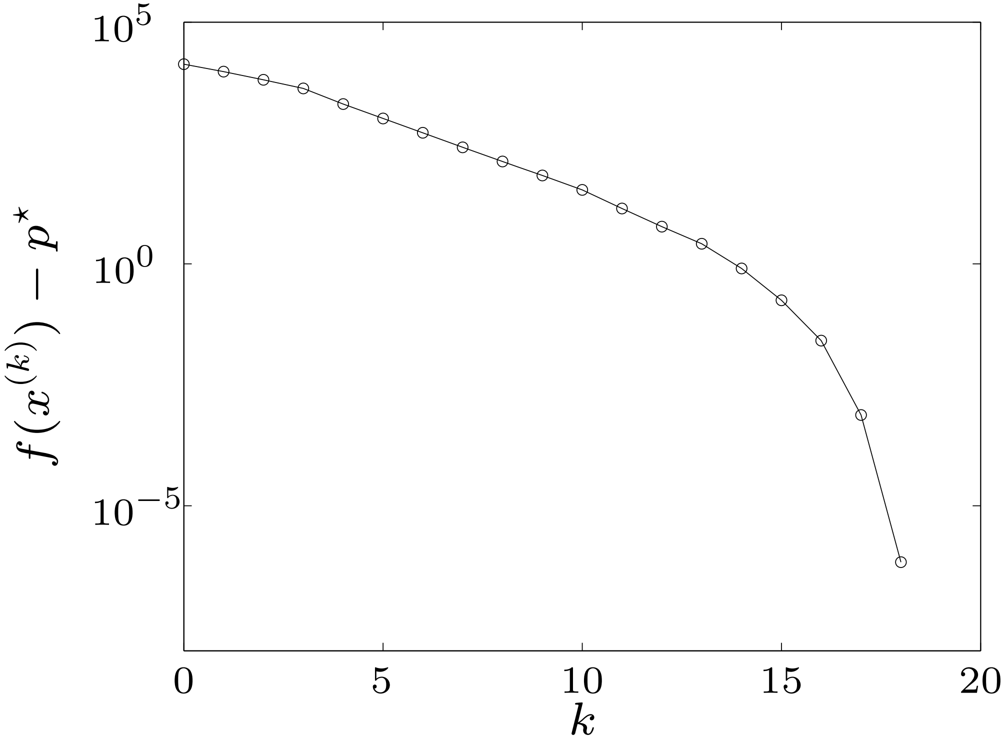
example in R^{100} (page 10–10)



- backtracking parameters $\alpha = 0.01$, $\beta = 0.5$
- backtracking line search almost as fast as exact l.s. (and much simpler)
- clearly shows two phases in algorithm

example in R^{10000} (with sparse a_i)

$$f(x) = -\sum_{i=1}^{10000} \log(1 - x_i^2) - \sum_{i=1}^{100000} \log(b_i - a_i^T x)$$



- backtracking parameters $\alpha = 0.01$, $\beta = 0.5$.
- performance similar as for small examples

Self-concordance

shortcomings of classical convergence analysis

- depends on unknown constants (m, L, . . .)
- bound is not affinely invariant, although Newton's method is

convergence analysis via self-concordance (Nesterov and Nemirovski)

- does not depend on any unknown constants
- gives affine-invariant bound
- applies to special class of convex functions ('self-concordant' functions)
- developed to analyze polynomial-time interior-point methods for convex optimization

Self-concordant functions

definition

- $f : \mathbf{R} \to \mathbf{R}$ is self-concordant if $|f'''(x)| \leq 2f''(x)^{3/2}$ for all $x \in \mathbf{dom}\, f$
- $f : \mathbf{R}^n \to \mathbf{R}$ is self-concordant if $g(t) = f(x + tv)$ is self-concordant for all $x \in \mathbf{dom}\, f$, $v \in \mathbf{R}^n$

examples on R

- linear and quadratic functions
- negative logarithm $f(x) = -\log x$
- negative entropy plus negative logarithm: $f(x) = x \log x - \log x$

affine invariance: if $f : \mathbf{R} \to \mathbf{R}$ is s.c., then $\tilde{f}(y) = f(ay + b)$ is s.c.:

$$\tilde{f}'''(y) = a^3 f'''(ay + b), \qquad \tilde{f}''(y) = a^2 f''(ay + b)$$

Self-concordant calculus

properties

- preserved under positive scaling $\alpha \geq 1$, and sum
- preserved under composition with affine function
- if g is convex with $\mathbf{dom}\, g = \mathbf{R}_{++}$ and $|g'''(x)| \leq 3g''(x)/x$ then

$$f(x) = \log(-g(x)) - \log x$$

  is self-concordant

examples: properties can be used to show that the following are s.c.

- $f(x) = -\sum_{i=1}^{m} \log(b_i - a_i^T x)$ on $\{x \mid a_i^T x < b_i,\ i = 1, \ldots, m\}$
- $f(X) = -\log\det X$ on $\mathbf{S}_{++}^n$
- $f(x) = -\log(y^2 - x^T x)$ on $\{(x, y) \mid \|x\|_2 < y\}$

Convergence analysis for self-concordant functions

summary: there exist constants $\eta \in (0, 1/4]$, $\gamma > 0$ such that

- if $\lambda(x) > \eta$, then
$$f(x^{(k+1)}) - f(x^{(k)}) \leq -\gamma$$
- if $\lambda(x) \leq \eta$, then
$$2\lambda(x^{(k+1)}) \leq \left(2\lambda(x^{(k)})\right)^2$$

(η and γ only depend on backtracking parameters α, β)

complexity bound: number of Newton iterations bounded by

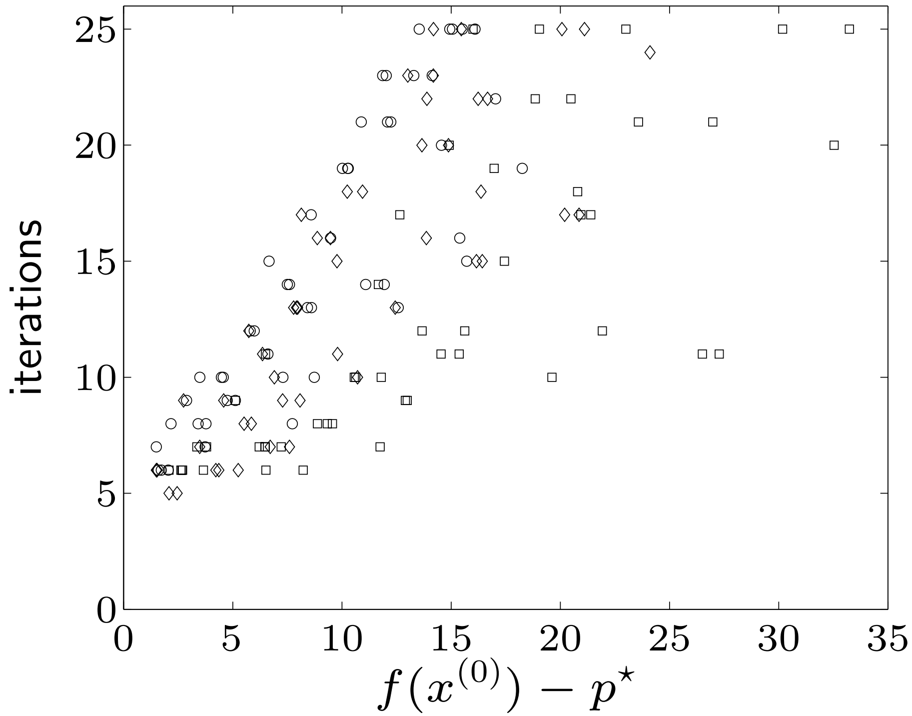
$$\frac{f(x^{(0)}) - p^\star}{\gamma} + \log_2 \log_2(1/\epsilon)$$

for $\alpha = 0.1$, $\beta = 0.8$, $\epsilon = 10^{-10}$, bound evaluates to $375(f(x^{(0)}) - p^\star) + 6$

numerical example: 150 randomly generated instances of

$$\text{minimize} \quad f(x) = -\sum_{i=1}^{m} \log(b_i - a_i^T x)$$

○: $m = 100$, $n = 50$
□: $m = 1000$, $n = 500$
◇: $m = 1000$, $n = 50$



- number of iterations much smaller than $375(f(x^{(0)}) - p^\star) + 6$
- bound of the form $c(f(x^{(0)}) - p^\star) + 6$ with smaller c (empirically) valid

Implementation

main effort in each iteration: evaluate derivatives and solve Newton system

$$H\Delta x = g$$

where $H = \nabla^2 f(x)$, $g = -\nabla f(x)$

via Cholesky factorization

$$H = LL^T, \qquad \Delta x_{\rm nt} = L^{-T}L^{-1}g, \qquad \lambda(x) = \|L^{-1}g\|_2$$

- cost $(1/3)n^3$ flops for unstructured system
- cost $\ll (1/3)n^3$ if H sparse, banded

example of dense Newton system with structure

$$f(x) = \sum_{i=1}^{n} \psi_i(x_i) + \psi_0(Ax+b), \qquad H = D + A^T H_0 A$$

- assume $A \in \mathbf{R}^{p \times n}$, dense, with $p \ll n$
- D diagonal with diagonal elements $\psi_i''(x_i)$; $H_0 = \nabla^2 \psi_0(Ax+b)$

method 1: form H, solve via dense Cholesky factorization: (cost $(1/3)n^3$)

method 2 (page 9–15): factor $H_0 = L_0 L_0^T$; write Newton system as

$$D\Delta x + A^T L_0 w = -g, \qquad L_0^T A \Delta x - w = 0$$

eliminate Δx from first equation; compute w and Δx from

$$(I + L_0^T A D^{-1} A^T L_0) w = -L_0^T A D^{-1} g, \qquad D\Delta x = -g - A^T L_0 w$$

cost: $2p^2 n$ (dominated by computation of $L_0^T A D^{-1} A^T L_0$)

11. Equality constrained minimization

- equality constrained minimization
- eliminating equality constraints
- Newton's method with equality constraints
- infeasible start Newton method
- implementation

Equality constrained minimization

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \end{array}$$

- f convex, twice continuously differentiable
- $A \in \mathbf{R}^{p \times n}$ with $\mathbf{rank}\, A = p$
- we assume $p^\star$ is finite and attained

optimality conditions: $x^\star$ is optimal iff there exists a $\nu^\star$ such that

$$\nabla f(x^\star) + A^T \nu^\star = 0, \qquad Ax^\star = b$$

equality constrained quadratic minimization (with $P \in \mathbf{S}^n_+$)

$$\begin{array}{ll} \text{minimize} & (1/2)x^T Px + q^T x + r \\ \text{subject to} & Ax = b \end{array}$$

optimality condition:

$$\begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^\star \\ \nu^\star \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix}$$

- coefficient matrix is called KKT matrix
- KKT matrix is nonsingular if and only if

$$Ax = 0, \quad x \neq 0 \qquad \Longrightarrow \qquad x^T Px > 0$$

- equivalent condition for nonsingularity: $P + A^T A \succ 0$

Eliminating equality constraints

represent solution of $\{x \mid Ax = b\}$ as

$$\{x \mid Ax = b\} = \{Fz + \hat{x} \mid z \in \mathbf{R}^{n-p}\}$$

- $\hat{x}$ is (any) particular solution
- range of $F \in \mathbf{R}^{n \times (n-p)}$ is nullspace of A ($\mathbf{rank}\, F = n - p$ and $AF = 0$)

reduced or eliminated problem

$$\text{minimize} \quad f(Fz + \hat{x})$$

- an unconstrained problem with variable $z \in \mathbf{R}^{n-p}$
- from solution $z^\star$, obtain $x^\star$ and $\nu^\star$ as

$$x^\star = Fz^\star + \hat{x}, \qquad \nu^\star = -(AA^T)^{-1} A \nabla f(x^\star)$$

example: optimal allocation with resource constraint

$$\begin{array}{ll} \text{minimize} & f_1(x_1) + f_2(x_2) + \cdots + f_n(x_n) \\ \text{subject to} & x_1 + x_2 + \cdots + x_n = b \end{array}$$

eliminate $x_n = b - x_1 - \cdots - x_{n-1}$, *i.e.*, choose

$$\hat{x} = be_n, \qquad F = \begin{bmatrix} I \\ -\mathbf{1}^T \end{bmatrix} \in \mathbf{R}^{n\times(n-1)}$$

reduced problem:

$$\text{minimize} \quad f_1(x_1) + \cdots + f_{n-1}(x_{n-1}) + f_n(b - x_1 - \cdots - x_{n-1})$$

(variables x_1, . . . , x_{n-1})

Newton step

Newton step of f at feasible x is given by (1st block) of solution of

$$\begin{bmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{\mathrm{nt}} \\ w \end{bmatrix} = \begin{bmatrix} -\nabla f(x) \\ 0 \end{bmatrix}$$

interpretations

- Δx_{nt} solves second order approximation (with variable v)

$$\begin{array}{ll} \text{minimize} & \widehat{f}(x+v) = f(x) + \nabla f(x)^T v + (1/2) v^T \nabla^2 f(x) v \\ \text{subject to} & A(x+v) = b \end{array}$$

- equations follow from linearizing optimality conditions

$$\nabla f(x + \Delta x_{\mathrm{nt}}) + A^T w = 0, \qquad A(x + \Delta x_{\mathrm{nt}}) = b$$

Newton decrement

$$\lambda(x) = \left(\Delta x_{\mathrm{nt}}^T \nabla^2 f(x) \Delta x_{\mathrm{nt}}\right)^{1/2} = \left(-\nabla f(x)^T \Delta x_{\mathrm{nt}}\right)^{1/2}$$

properties

- gives an estimate of $f(x) - p^\star$ using quadratic approximation $\widehat{f}$:

$$f(x) - \inf_{Ay=b} \widehat{f}(y) = \frac{1}{2}\lambda(x)^2$$

- directional derivative in Newton direction:

$$\frac{d}{dt} f(x + t\Delta x_{\mathrm{nt}})\bigg|_{t=0} = -\lambda(x)^2$$

- in general, $\lambda(x) \neq \left(\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x)\right)^{1/2}$

Newton's method with equality constraints

---

given starting point $x \in \mathbf{dom}\, f$ with $Ax = b$, tolerance $\epsilon > 0$.
repeat
1. Compute the Newton step and decrement Δx_{nt}, $\lambda(x)$.
2. *Stopping criterion.* **quit** if $\lambda^2/2 \leq \epsilon$.
3. *Line search.* Choose step size t by backtracking line search.
4. *Update.* $x := x + t\Delta x_{\mathrm{nt}}$.

- a feasible descent method: $x^{(k)}$ feasible and $f(x^{(k+1)}) < f(x^{(k)})$
- affine invariant

Newton's method and elimination

Newton's method for reduced problem

$$\text{minimize} \quad \tilde{f}(z) = f(Fz + \hat{x})$$

- variables $z \in \mathbf{R}^{n-p}$
- $\hat{x}$ satisfies $A\hat{x} = b$; $\mathbf{rank}\, F = n - p$ and $AF = 0$
- Newton's method for $\tilde{f}$, started at $z^{(0)}$, generates iterates $z^{(k)}$

Newton's method with equality constraints

when started at $x^{(0)} = Fz^{(0)} + \hat{x}$, iterates are

$$x^{(k+1)} = Fz^{(k)} + \hat{x}$$

hence, don't need separate convergence analysis

Newton step at infeasible points

2nd interpretation of page 11–6 extends to infeasible x ($i.e.$, $Ax \neq b$)

linearizing optimality conditions at infeasible x (with $x \in \mathbf{dom}\, f$) gives

$$\left[\begin{array}{cc} \nabla^2 f(x) & A^T \\ A & 0 \end{array}\right] \left[\begin{array}{c} \Delta x_{\rm nt} \\ w \end{array}\right] = - \left[\begin{array}{c} \nabla f(x) \\ Ax - b \end{array}\right] \qquad (1)$$

primal-dual interpretation

- write optimality condition as $r(y) = 0$, where

$$y = (x, \nu), \qquad r(y) = (\nabla f(x) + A^T \nu, Ax - b)$$

- linearizing $r(y) = 0$ gives $r(y + \Delta y) \approx r(y) + Dr(y)\Delta y = 0$:

$$\left[\begin{array}{cc} \nabla^2 f(x) & A^T \\ A & 0 \end{array}\right] \left[\begin{array}{c} \Delta x_{\rm nt} \\ \Delta \nu_{\rm nt} \end{array}\right] = - \left[\begin{array}{c} \nabla f(x) + A^T \nu \\ Ax - b \end{array}\right]$$

  same as (1) with $w = \nu + \Delta \nu_{\rm nt}$

Infeasible start Newton method

---

given starting point $x \in \mathbf{dom}\, f$, ν, tolerance $\epsilon > 0$, $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$.
repeat
1. Compute primal and dual Newton steps $\Delta x_{\rm nt}$, $\Delta \nu_{\rm nt}$.
2. *Backtracking line search on* $\|r\|_2$.
   $t := 1$.
   while $\|r(x + t\Delta x_{\rm nt}, \nu + t\Delta \nu_{\rm nt})\|_2 > (1 - \alpha t)\|r(x, \nu)\|_2, \quad t := \beta t$.
3. *Update.* $x := x + t\Delta x_{\rm nt}$, $\nu := \nu + t\Delta \nu_{\rm nt}$.

until $Ax = b$ and $\|r(x, \nu)\|_2 \leq \epsilon$.

- not a descent method: $f(x^{(k+1)}) > f(x^{(k)})$ is possible
- directional derivative of $\|r(y)\|_2^2$ in direction $\Delta y = (\Delta x_{\rm nt}, \Delta \nu_{\rm nt})$ is

$$\frac{d}{dt}\left.\|r(y + \Delta y)\|_2\right|_{t=0} = -\|r(y)\|_2$$

Solving KKT systems

$$\begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = - \begin{bmatrix} g \\ h \end{bmatrix}$$

solution methods

- LDL$^{\mathrm{T}}$ factorization
- elimination (if H nonsingular)

$$AH^{-1}A^T w = h - AH^{-1}g, \qquad Hv = -(g + A^T w)$$

- elimination with singular H: write as

$$\begin{bmatrix} H + A^T Q A & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = - \begin{bmatrix} g + A^T Q h \\ h \end{bmatrix}$$

  with $Q \succeq 0$ for which $H + A^T Q A \succ 0$, and apply elimination

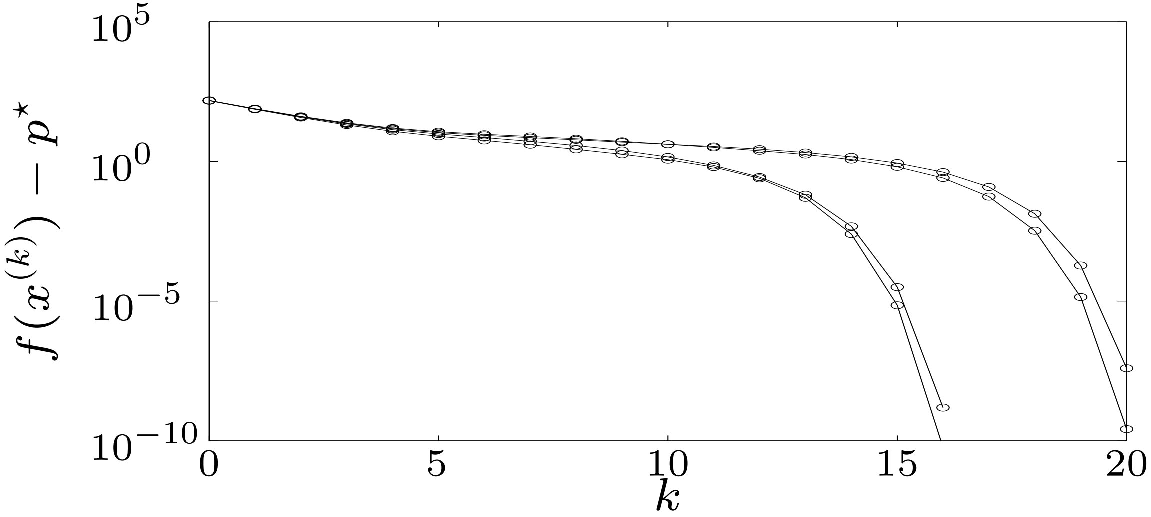
Equality constrained analytic centering

primal problem: minimize $-\sum_{i=1}^{n} \log x_i$ subject to $Ax = b$

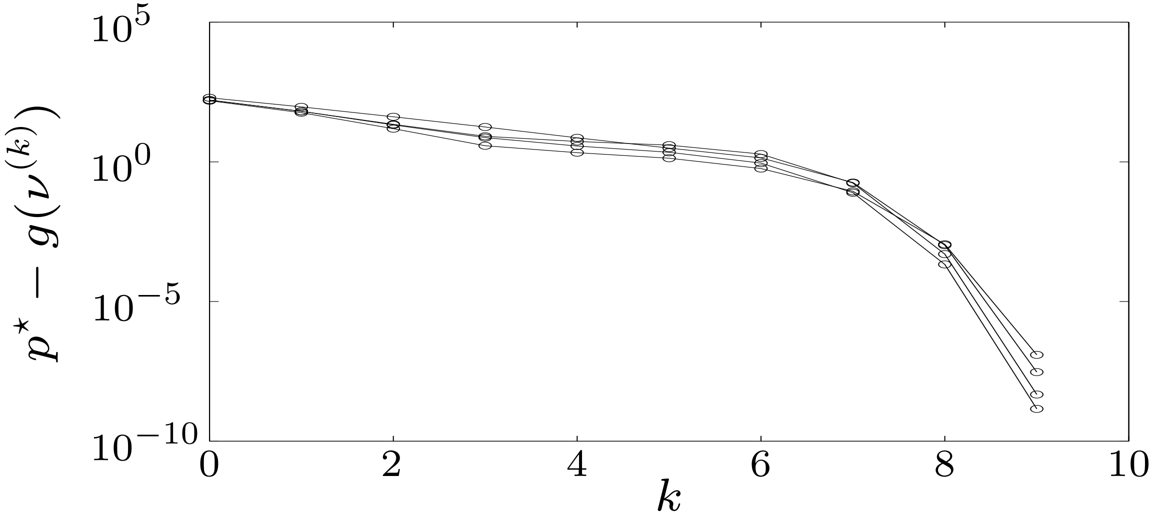
dual problem: maximize $-b^T\nu + \sum_{i=1}^{n} \log(A^T\nu)_i + n$

three methods for an example with $A \in \mathbf{R}^{100\times 500}$, different starting points

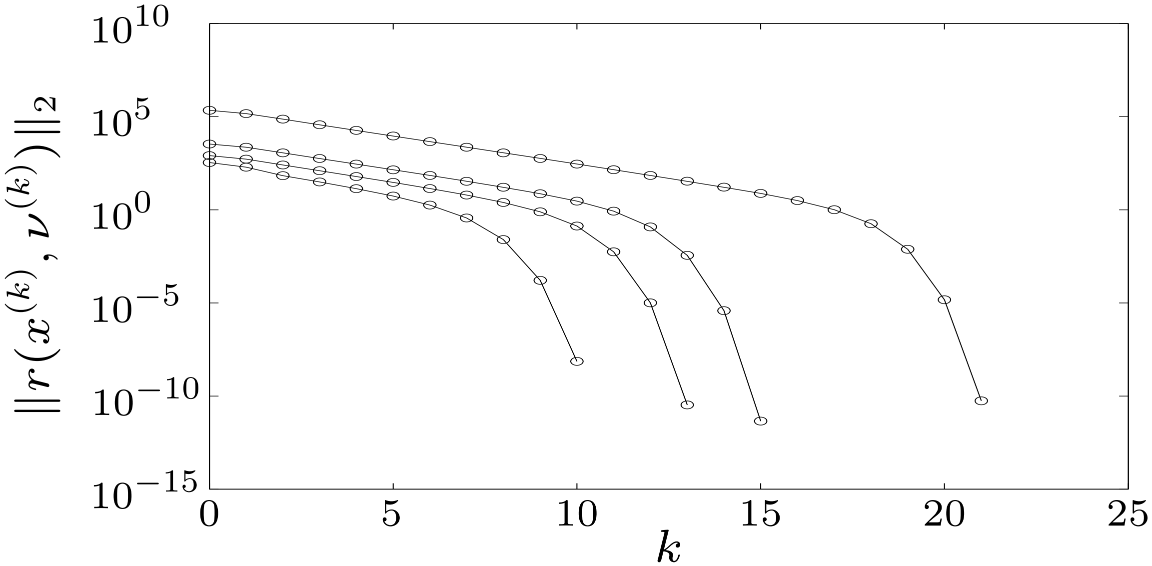
1. Newton method with equality constraints (requires $x^{(0)} \succ 0$, $Ax^{(0)} = b$)



2. Newton method applied to dual problem (requires $A^T\nu^{(0)} \succ 0$)



3. infeasible start Newton method (requires $x^{(0)} \succ 0$)



complexity per iteration of three methods is identical

1. use block elimination to solve KKT system

$$\begin{bmatrix} \mathbf{diag}(x)^{-2} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ w \end{bmatrix} = \begin{bmatrix} \mathbf{diag}(x)^{-1}\mathbf{1} \\ 0 \end{bmatrix}$$

   reduces to solving $A\,\mathbf{diag}(x)^2 A^T w = b$

2. solve Newton system $A\,\mathbf{diag}(A^T\nu)^{-2}A^T\Delta\nu = -b + A\,\mathbf{diag}(A^T\nu)^{-1}\mathbf{1}$

3. use block elimination to solve KKT system

$$\begin{bmatrix} \mathbf{diag}(x)^{-2} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\nu \end{bmatrix} = \begin{bmatrix} \mathbf{diag}(x)^{-1}\mathbf{1} \\ Ax - b \end{bmatrix}$$

   reduces to solving $A\,\mathbf{diag}(x)^2 A^T w = 2Ax - b$

conclusion: in each case, solve $ADA^T w = h$ with D positive diagonal

Network flow optimization

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n \phi_i(x_i) \\ \text{subject to} & Ax = b \end{array}$$

- directed graph with n arcs, $p+1$ nodes
- x_i: flow through arc i; ϕ_i: cost flow function for arc i (with $\phi_i''(x) > 0$)
- node-incidence matrix $\tilde{A} \in \mathbf{R}^{(p+1)\times n}$ defined as

$$\tilde{A}_{ij} = \begin{cases} 1 & \text{arc } j \text{ leaves node } i \\ -1 & \text{arc } j \text{ enters node } i \\ 0 & \text{otherwise} \end{cases}$$

- reduced node-incidence matrix $A \in \mathbf{R}^{p\times n}$ is $\tilde{A}$ with last row removed
- $b \in \mathbf{R}^p$ is (reduced) source vector
- $\mathbf{rank}\, A = p$ if graph is connected

KKT system

$$\begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = - \begin{bmatrix} g \\ h \end{bmatrix}$$

- $H = \mathbf{diag}(\phi_1''(x_1), \ldots, \phi_n''(x_n))$, positive diagonal
- solve via elimination:

$$AH^{-1}A^T w = h - AH^{-1}g, \qquad Hv = -(g + A^T w)$$

sparsity pattern of coefficient matrix is given by graph connectivity

$$\begin{aligned} (AH^{-1}A^T)_{ij} \neq 0 \quad &\Longleftrightarrow \quad (AA^T)_{ij} \neq 0 \\ &\Longleftrightarrow \quad \text{nodes } i \text{ and } j \text{ are connected by an arc} \end{aligned}$$

Analytic center of linear matrix inequality

$$\begin{array}{ll} \text{minimize} & -\log\det X \\ \text{subject to} & \mathbf{tr}(A_i X) = b_i, \quad i = 1, \ldots, p \end{array}$$

variable $X \in \mathbf{S}^n$

optimality conditions

$$X^\star \succ 0, \qquad -(X^\star)^{-1} + \sum_{j=1}^p \nu_j^\star A_i = 0, \qquad \mathbf{tr}(A_i X^\star) = b_i, \quad i = 1, \ldots, p$$

Newton equation at feasible X:

$$X^{-1}\Delta X X^{-1} + \sum_{j=1}^p w_j A_i = X^{-1}, \qquad \mathbf{tr}(A_i \Delta X) = 0, \quad i = 1, \ldots, p$$

- follows from linear approximation $(X + \Delta X)^{-1} \approx X^{-1} - X^{-1}\Delta X X^{-1}$
- $n(n+1)/2 + p$ variables ΔX, w

solution by block elimination

- eliminate ΔX from first equation: $\Delta X = X - \sum_{j=1}^{p} w_j X A_j X$
- substitute ΔX in second equation

$$\sum_{j=1}^{p} \mathbf{tr}(A_i X A_j X) w_j = b_i, \quad i = 1, \ldots, p \tag{2}$$

a dense positive definite set of linear equations with variable $w \in \mathbf{R}^p$

flop count (dominant terms) using Cholesky factorization $X = LL^T$:

- form p products $L^T A_j L$: $(3/2)pn^3$
- form $p(p+1)/2$ inner products $\mathbf{tr}((L^T A_i L)(L^T A_j L))$: $(1/2)p^2 n^2$
- solve (2) via Cholesky factorization: $(1/3)p^3$