10. Unconstrained minimization

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

Unconstrained minimization

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

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

We will assume that x^* = argmin_x f(x) exists and is unique
Recall p^* = f(x^*)

unconstrained minimization methods

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

$$f(x^{(k)}) \to p^\star$$ **as k -> infinity**

x^(0), x^(1), ... is a minimizing sequence to the problem
Algorithm stops when f(x^(k)) - p^* <= epsilon, for some tolerance epsilon > 0

- can be interpreted as iterative methods for solving optimality condition

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

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

Assume f is twice differentiable
By Taylor's theorem, there exists a z in the line segment from x to y such that
f(y) = f(x) + df(x)'(y-x) + ½ (y-x)' d^2 f(z) (y-x)
>= f(x) + df(x)'(y-x) + ½ (y-x)' (m I) (y-x) … since f is strongly convex
= f(x) + df(x)'(y-x) + ½ m |y-x|_2^2

(Taylor's theorem is a generalization of the mean value theorem,
and is very related to, but is not exactly the same as Taylor series)

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*)

From convexity (slide 3-7):
f(x^+) >= f(x) + df(x)'(x^+ - x)
= f(x) + t df(x)'Δx
Thus: f(x^+) - f(x) >= t df(x)'Δx

If f(x^+) < f(x) then:
0 > f(x^+) - f(x) >= t df(x)'Δx
Thus: df(x)'Δx < 0

General descent method.

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

1. Determine a descent direction Δx. **(Each algorithm has its own way for choosing 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 $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$)

(one of the many inexact methods)

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

since β < 1, t := β t reduces t

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

(Armijo–Goldstein condition)

Since Δx is a descent direction (see previous slide) then df(x)'Δx < 0
For small t, we have:

f(x + t Δx) ≈ f(x) + t df(x)'Δx < f(x) + α t df(x)'Δx

Thus, the procedure will eventually terminate.

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,

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

(linear convergence)

$c \in (0,1)$ depends on m, $x^{(0)}$, line search type

- very simple, but often very slow; rarely used in practice

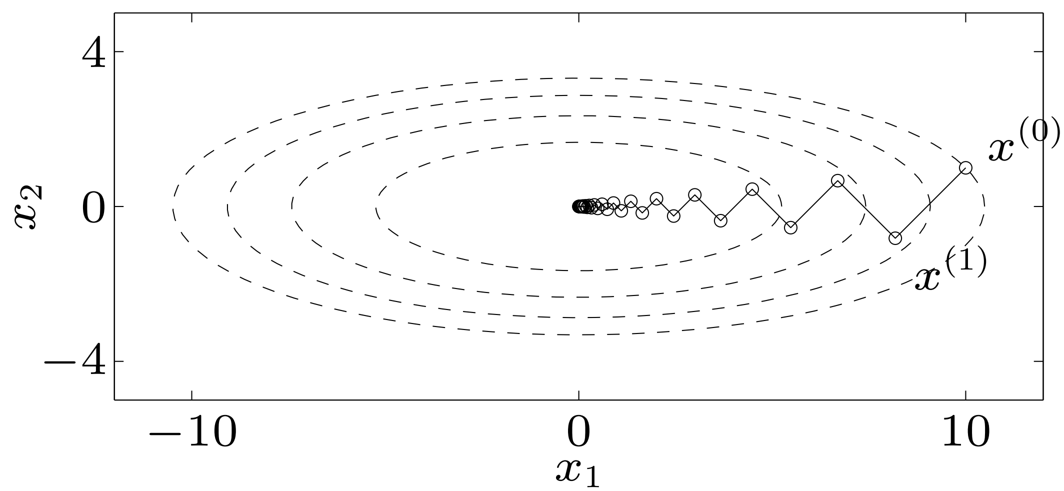
quadratic problem in $\mathbf{R}^2$

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

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

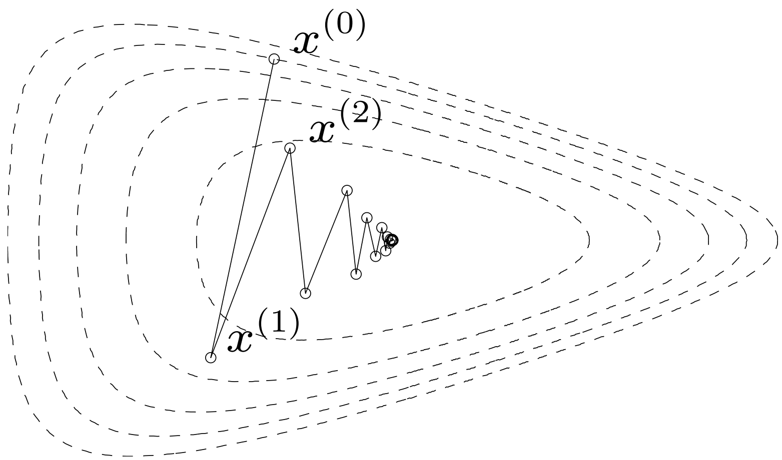
$$x_1^{(k)} = \gamma \left(\frac{\gamma - 1}{\gamma + 1}\right)^k, \qquad x_2^{(k)} = \left(-\frac{\gamma - 1}{\gamma + 1}\right)^k$$

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

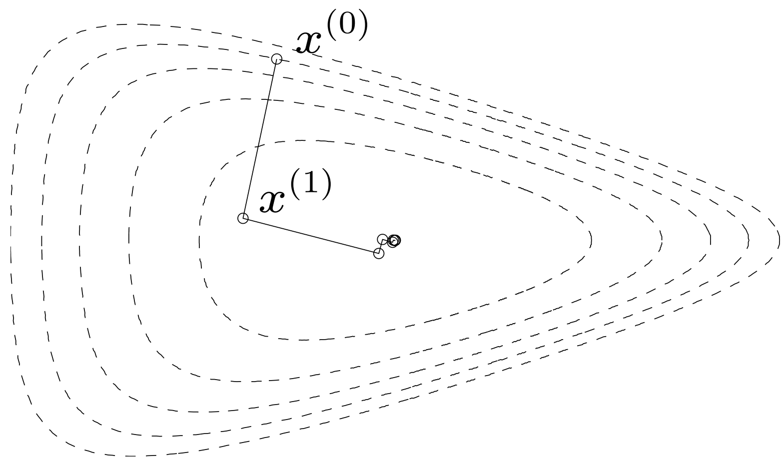


nonquadratic example

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



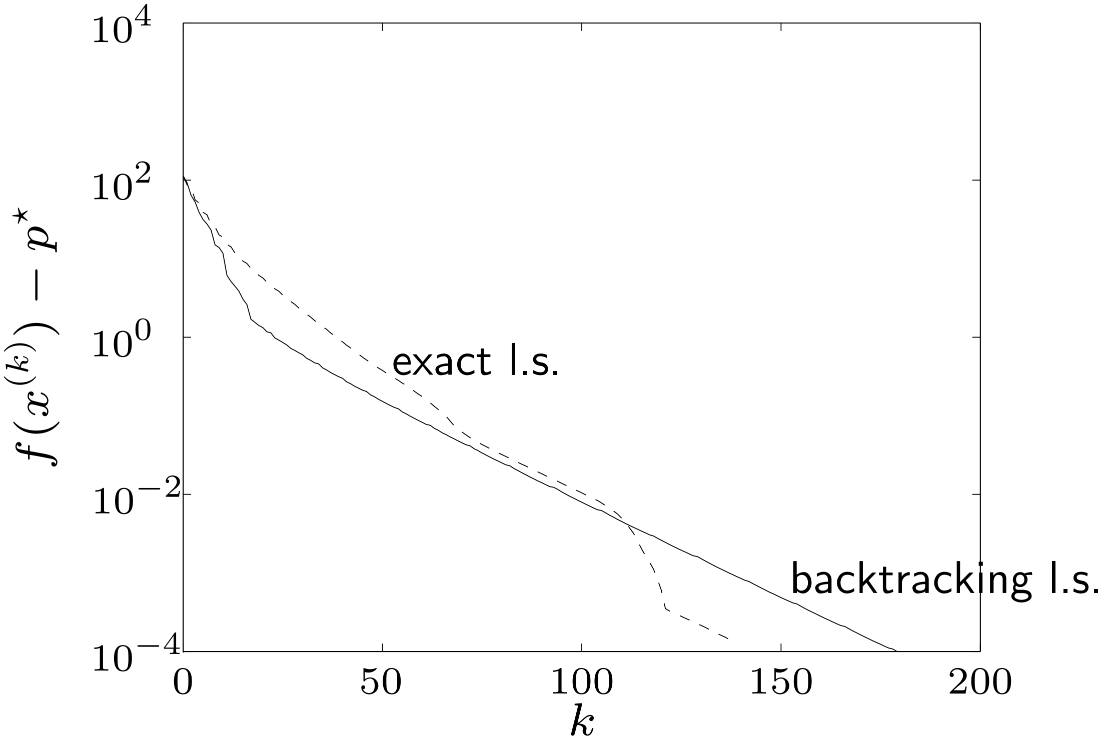
backtracking line search



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 $\|\cdot\|$):

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

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

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

steepest descent method

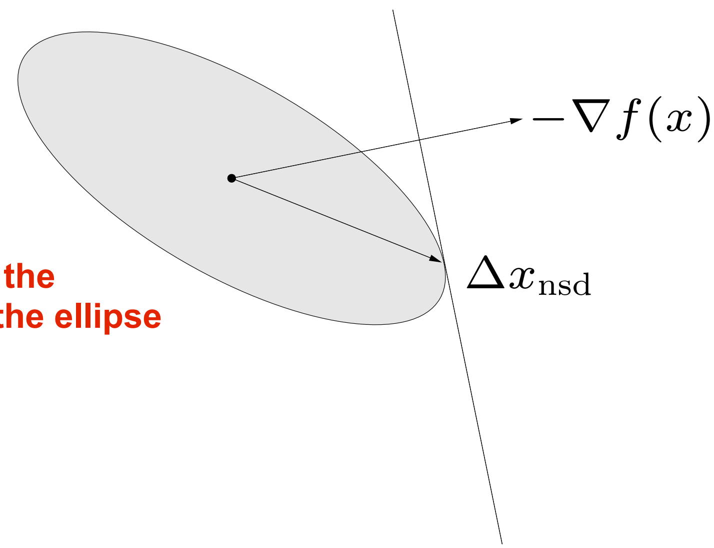
- general descent method with $\Delta x = \Delta x_{\text{sd}}$
- convergence properties similar to gradient descent

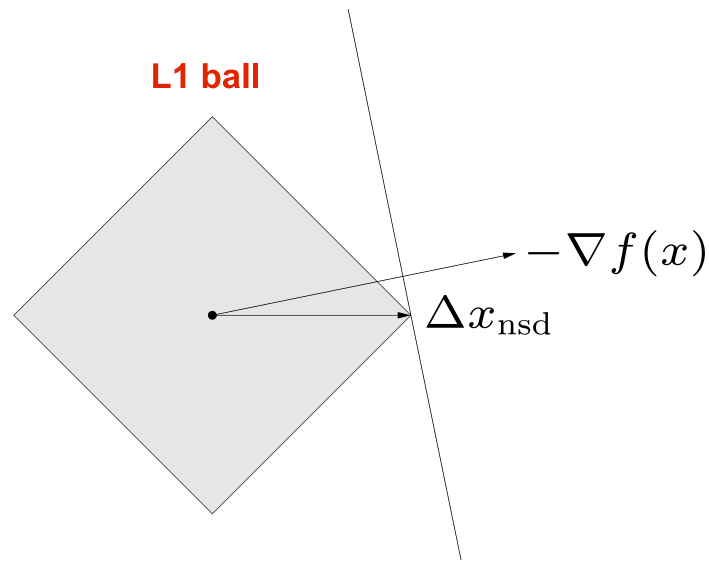
examples

- 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 $|\partial f(x)/\partial x_i| = \|\nabla f(x)\|_\infty$

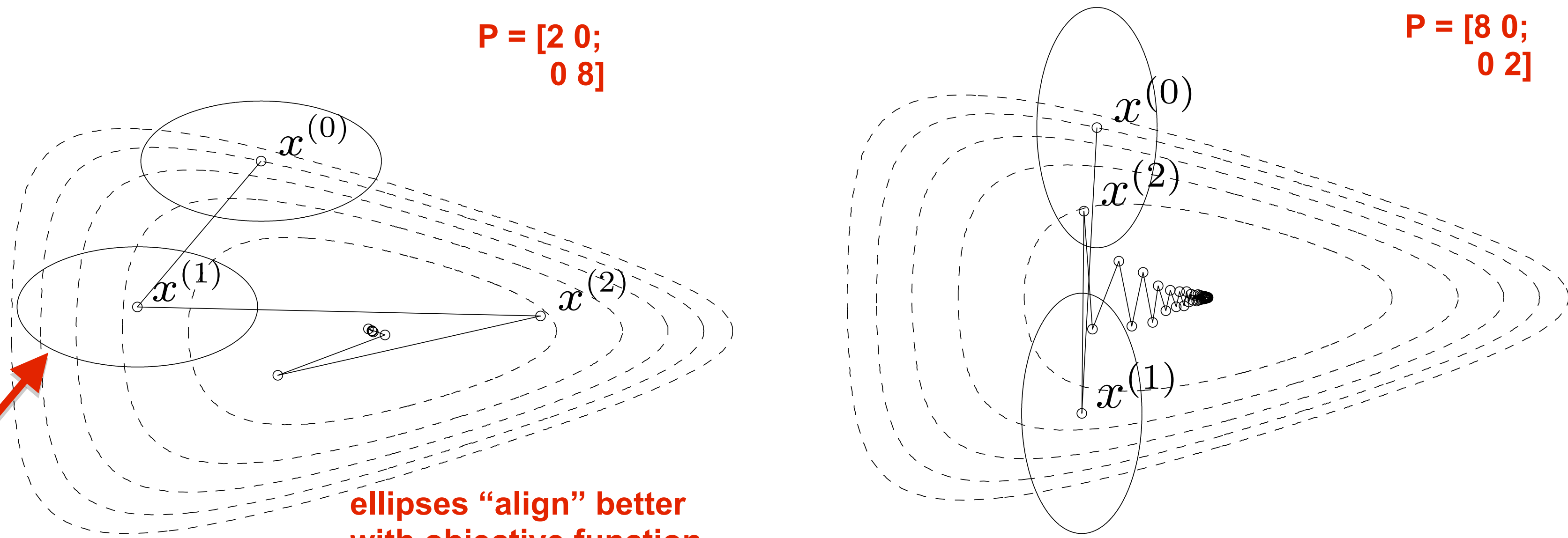
unit balls and normalized steepest descent directions for a quadratic norm and the ℓ_1-norm:

P defines the shape of the ellipse





choice of norm for steepest descent



ellipses "align" better
with objective function
thus convergence is faster

- steepest descent with backtracking line search for two quadratic norms

(two different P's)

- ellipses show $\{x \mid \|x - x^{(k)}\|_P = 1\}$

See Figure 9.13

- equivalent interpretation of steepest descent with quadratic norm $\|\cdot\|_P$: gradient descent after change of variables $\bar{x} = P^{1/2}x$

See Figures 9.14, 9.15

shows choice of P has strong effect on speed of convergence

Newton step

(Uses the Hessian as a good ellipse, see previous slide)

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

interpretations

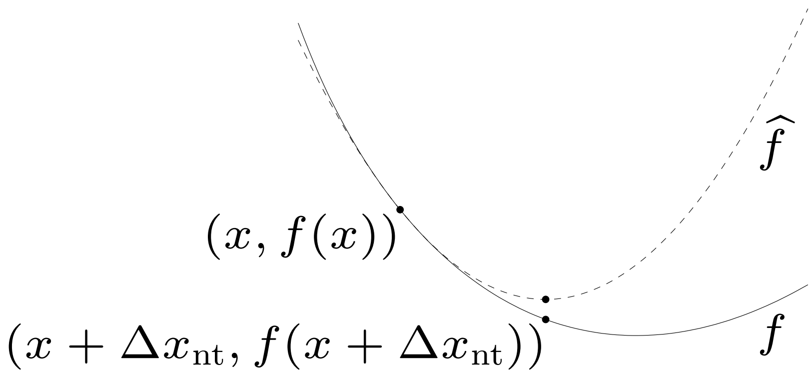
- $x + \Delta x_{\mathrm{nt}}$ minimizes second order approximation

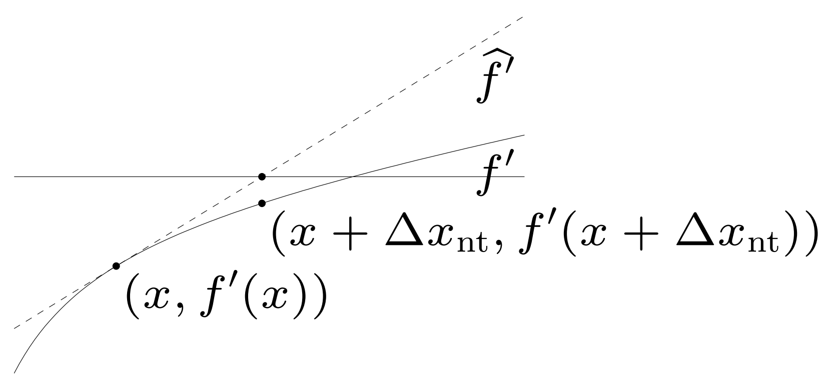
$$\widehat{f}(x+v) = f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v$$

Second order Taylor series approximation (we are discarding the remainder term)

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

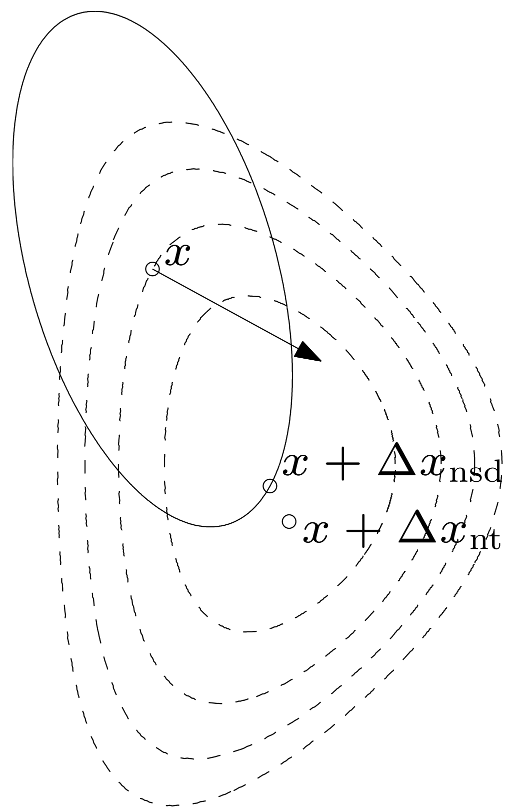
$$\nabla f(x+v) \approx \nabla \widehat{f}(x+v) = \nabla f(x) + \nabla^2 f(x) v = 0$$





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

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



Let H = d^2 f(x)
d = df(x)
From slide 10-11 we have:

min d'u
s.t. u' H u = 1

Let u = H^-½ s

min (H^-½ d)'s
s.t. s's = 1

L(s,v) = (H^-½ d)'s + v (s's - 1)
dL/ds = H^-½ d + 2 v s = 0
s^* = -1/(2v) H^-½ d

Then:
u^* = H^-½ s^*
= -1/(2v) H^-1 d

which is the direction of Δx_nt !

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 $x^\star$

Remember p^* = inf_y f(y)

properties

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

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

Let H = d^2 f(x)
d = df(x)
λ = λ(x)
Δx = Δx_nt = -H^-1 d

inf_y f^(y) = f^ (x + Δx)
= f(x) + d'Δx + ½ Δx' H Δx
= f(x) - ½ d' H^-1 d

f(x) - inf_y f^(y) = ½ d' H^-1 d = ½ λ^2

Thus λ = sqrt(d' H^-1 d)

Newton's method

---

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

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

---

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

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

x = T y
Let Hf~(y) = d^2 f~(y)
df~(y) = T' df(T y) = T' df(x)
Hf~(y) = T' Hf(T y) T = T' Hf(x) T

y = T^-1 x
Δy = - Hf~(y)^-1 df(y) = - (T' Hf(x) T)^-1 T' df(x)
= - T^-1 Hf(x)^-1 df(x) = T^-1 Δx
y^(k) = y + Δy = T^-1 (x + Δx) = T^-1 x^(k)

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

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

- 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 $L_0^T A D^{-1} A^T L_0$)