

# 13. Unconstrained minimization

- terminology
- gradient and Hessian
- Newton's method

# Unconstrained minimization problem

$$\text{minimize } g(x_1, x_2, \dots, x_n)$$

$g : \mathbf{R}^n \rightarrow \mathbf{R}$  (a function that maps  $n$ -vectors to scalars)

- $x = (x_1, x_2, \dots, x_n)$  are the optimization *variables*
- $g$  is the *cost function* or *objective function*
- to solve a maximization problem (*i.e.*, maximize  $g(x)$ ), minimize  $-g(x)$

# Local and global optimum

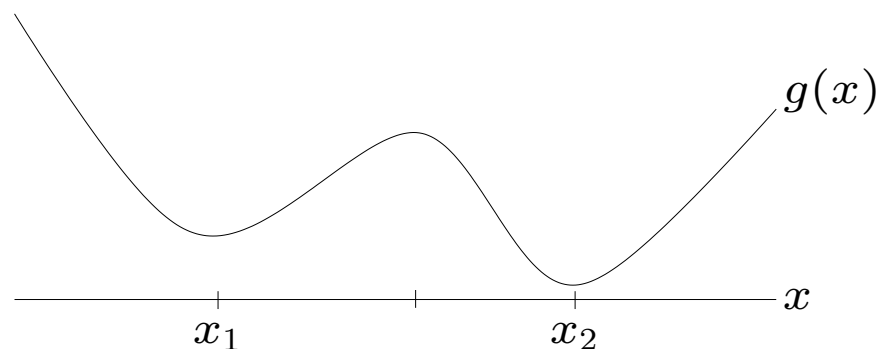
$x^*$  is an **optimal point** (or a **minimum**) if

$$g(x^*) \leq g(x) \quad \text{for all } x$$

also called *globally* optimal

$x^*$  is a **locally optimal point** (**local minimum**) if for some  $R > 0$

$$g(x^*) \leq g(x) \quad \text{for all } x \text{ with } \|x - x^*\| \leq R$$



$x_1$  is locally optimal;  $x_2$  is (globally) optimal

# Optimal value

the **optimal value** (or minimal value) of  $g$  is the largest  $\alpha$  such that

$$g(x) \geq \alpha \quad \text{for all } x$$

notation:  $\min g(x)$

- if  $x^*$  is optimal, then  $g(x^*) = \min g(x)$
- if  $g$  is unbounded below, we define  $\min g(x) = -\infty$

## examples

- $g(x) = (x - 1)^2$ :  $\min g(x) = 0$  (finite and attained at  $x^* = 1$ )
- $g(x) = 1/(x^2 + 1)$ :  $\min g(x) = 0$  (finite but not attained)
- $g(x) = x$ :  $\min g(x) = -\infty$  (unbounded below)

# Gradient

gradient of  $g : \mathbf{R}^n \rightarrow \mathbf{R}$  at  $\hat{x} \in \mathbf{R}^n$ :

$$\nabla g(\hat{x}) = \begin{bmatrix} \frac{\partial g(\hat{x})}{\partial x_1} \\ \frac{\partial g(\hat{x})}{\partial x_2} \\ \vdots \\ \frac{\partial g(\hat{x})}{\partial x_n} \end{bmatrix}$$

- $\nabla g(\hat{x})^T = Dg(\hat{x})$  is the derivative matrix of  $g$  at  $\hat{x}$
- special case ( $n = 1$ ):  $\nabla g(\hat{x}) = g'(\hat{x})$

first-order (affine) approximation of  $g$  around  $\hat{x}$ :

$$\begin{aligned} g(x) &\approx g(\hat{x}) + \frac{\partial g(\hat{x})}{\partial x_1}(x_1 - \hat{x}_1) + \cdots + \frac{\partial g(\hat{x})}{\partial x_n}(x_n - \hat{x}_n) \\ &= g(\hat{x}) + \nabla g(\hat{x})^T(x - \hat{x}) \end{aligned}$$

# Hessian

Hessian of  $g$  at  $\hat{x}$ :

$$\nabla^2 g(\hat{x}) = \begin{bmatrix} \frac{\partial^2 g(\hat{x})}{\partial x_1^2} & \frac{\partial^2 g(\hat{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 g(\hat{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 g(\hat{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 g(\hat{x})}{\partial x_2^2} & \cdots & \frac{\partial^2 g(\hat{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 g(\hat{x})}{\partial x_n \partial x_1} & \frac{\partial^2 g(\hat{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 g(\hat{x})}{\partial x_n^2} \end{bmatrix}$$

- a symmetric  $n \times n$  matrix
- the derivative matrix  $Df(x)$  of  $f(x) = \nabla g(x)$
- special case ( $n = 1$ ):  $\nabla^2 g(\hat{x}) = g''(\hat{x})$

second-order (quadratic) approximation of  $g$  around  $\hat{x}$ :

$$g(x) \approx g(\hat{x}) + \nabla g(\hat{x})^T (x - \hat{x}) + \frac{1}{2} (x - \hat{x})^T \nabla^2 g(\hat{x}) (x - \hat{x})$$

# Examples

- affine function  $g(x) = a^T x + b$

$$\nabla g(x) = a, \quad \nabla^2 g(x) = 0$$

- quadratic function  $g(x) = x^T P x + q^T x + r$

$$\nabla g(x) = 2P x + q, \quad \nabla^2 g(x) = 2P$$

- $g(x) = \|Ax - b\|^2 = x^T A^T A x - 2b^T A x + b^T b$

$$\nabla g(x) = 2A^T A x - 2A^T b, \quad \nabla^2 g(x) = 2A^T A$$

## Useful properties

- $g(x) = g_1(x) + g_2(x)$

$$\nabla g(x) = \nabla g_1(x) + \nabla g_2(x), \quad \nabla^2 g(x) = \nabla^2 g_1(x) + \nabla^2 g_2(x)$$

- $g(x) = \alpha h(x)$

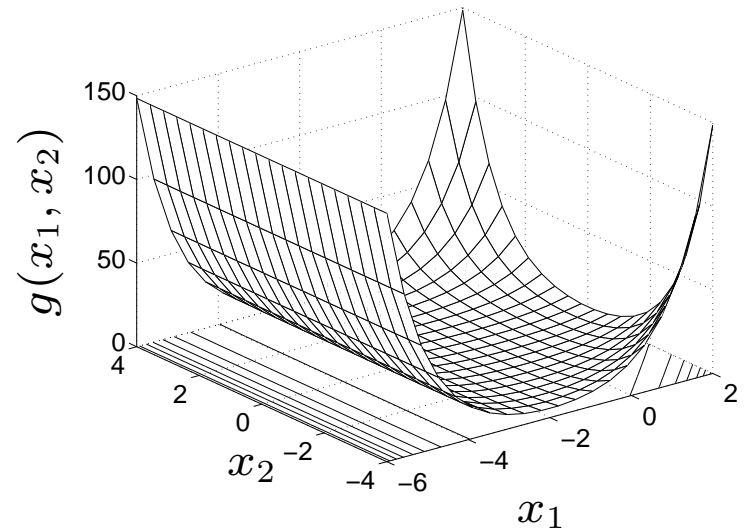
$$\nabla g(x) = \alpha \nabla h(x), \quad \nabla^2 g(x) = \alpha \nabla^2 h(x)$$

- $g(x) = h(Cx + d)$

$$\nabla g(x) = C^T \nabla h(Cx + d), \quad \nabla^2 g(x) = C^T \nabla^2 h(Cx + d) C$$

# Example

$$g(x_1, x_2) = e^{x_1+x_2-1} + e^{x_1-x_2-1} + e^{-x_1-1}$$



$$\nabla g(x) = \begin{bmatrix} e^{x_1+x_2-1} + e^{x_1-x_2-1} - e^{-x_1-1} \\ e^{x_1+x_2-1} - e^{x_1-x_2-1} \end{bmatrix}$$

$$\nabla^2 g(x) = \begin{bmatrix} e^{x_1+x_2-1} + e^{x_1-x_2-1} + e^{-x_1-1} & e^{x_1+x_2-1} - e^{x_1-x_2-1} \\ e^{x_1+x_2-1} - e^{x_1-x_2-1} & e^{x_1+x_2-1} + e^{x_1-x_2-1} \end{bmatrix}$$

**alternative derivation** (from last property on page 13-8)

$g = h(Cx + d)$  where  $h(y_1, y_2, y_3) = \exp(y_1) + \exp(y_2) + \exp(y_3)$

$$C = \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 0 \end{bmatrix}, \quad d = \begin{bmatrix} -1 \\ -1 \\ -1 \end{bmatrix}$$

therefore,

$$\begin{aligned} \nabla g(x) &= C^T \nabla h(Cx + d) \\ &= \begin{bmatrix} 1 & 1 & -1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} e^{x_1+x_2-1} \\ e^{x_1-x_2-1} \\ e^{-x_1-1} \end{bmatrix} \end{aligned}$$

$$\begin{aligned} \nabla^2 g(x) &= C^T \nabla^2 h(Cx + d) C \\ &= \begin{bmatrix} 1 & 1 & -1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} e^{x_1+x_2-1} & 0 & 0 \\ 0 & e^{x_1-x_2-1} & 0 \\ 0 & 0 & e^{-x_1-1} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 0 \end{bmatrix} \end{aligned}$$

# Optimality conditions

(for twice differentiable  $g$ )

- **necessary condition:** if  $x^*$  is locally optimal, then

$$\nabla g(x^*) = 0 \quad \text{and} \quad \nabla^2 g(x^*) \text{ is positive semidefinite}$$

- **sufficient condition:** if  $x^*$  satisfies

$$\nabla g(x^*) = 0 \quad \text{and} \quad \nabla^2 g(x^*) \text{ is positive definite}$$

then  $x^*$  is locally optimal

- **necessary and sufficient condition for convex functions:** if  $\nabla^2 g(x)$  is positive semidefinite everywhere (' $g$  is convex'), then  $x^*$  is optimal iff

$$\nabla g(x^*) = 0$$

## Examples ( $n = 1$ )

- $g(x) = \log(e^x + e^{-x})$

$$g'(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad g''(x) = \frac{4}{(e^x + e^{-x})^2}$$

$g''(x) \geq 0$  everywhere;  $x^* = 0$  is the unique optimal point

- $g(x) = x^4$

$$g'(x) = 4x^3, \quad g''(x) = 12x^2$$

$g''(x) \geq 0$  everywhere;  $x^* = 0$  is the unique optimal point

- $g(x) = x^3$

$$g'(x) = 3x^2, \quad g''(x) = 6x$$

$g'(0) = 0, g''(0) = 0$  but  $x = 0$  is not locally optimal

## Examples

- $g(x) = x^T P x + q^T x + r$  ( $P$  is positive definite)

$$\nabla g(x) = 2Px + q, \quad \nabla^2 g(x) = 2P$$

$\nabla^2 g(x)$  is positive definite everywhere, hence the unique optimal point is

$$x^* = -(1/2)P^{-1}q$$

- $g(x) = \|Ax - b\|^2$  ( $A$  is a matrix with zero nullspace)

$$\nabla g(x) = 2A^T Ax - 2A^T b, \quad \nabla^2 g(x) = 2A^T A$$

$\nabla^2 g(x)$  is positive definite everywhere, hence the unique optimal point is

$$x^* = (A^T A)^{-1} A^T b$$

- example of page 13-9: we can express  $\nabla^2 g(x)$  as

$$\nabla^2 g(x) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} e^{x_1+x_2-1} & 0 & 0 \\ 0 & e^{x_1-x_2-1} & 0 \\ 0 & 0 & e^{-x_1-1} \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ 1 & 0 \end{bmatrix}$$

this shows that  $\nabla^2 g(x)$  is positive definite for all  $x$

therefore  $x^*$  is optimal if and only if

$$\nabla g(x^*) = \begin{bmatrix} e^{x_1^*+x_2^*-1} + e^{x_1^*-x_2^*-1} - e^{-x_1^*-1} \\ e^{x_1^*+x_2^*-1} - e^{x_1^*-x_2^*-1} \end{bmatrix} = 0$$

two nonlinear equations in two variables

# Newton's method for minimizing a convex function

if  $\nabla^2 g(x)$  is positive definite everywhere, we can minimize  $g(x)$  by solving

$$\nabla g(x) = 0$$

using Newton's method

---

**given** initial  $x$ , tolerance  $\epsilon > 0$

**repeat**

1. evaluate  $\nabla g(x)$  and  $\nabla^2 g(x)$ .
2. **if**  $\|\nabla g(x)\| \leq \epsilon$ , **return**  $x$ .
3. Solve  $\nabla^2 g(x)v = -\nabla g(x)$ .
4.  $x := x + v$ .

**until** maximum number of iterations is exceeded

---

- $v = -\nabla^2 g(x)^{-1} \nabla g(x)$  is called the *Newton step* at  $x$
- converges if started sufficiently close to the solution

# Interpretations of Newton step

- construct affine approximation of  $f(y) = \nabla g(y)$  around  $x$ :

$$f_{\text{aff}}(y) = \nabla g(x) + \nabla^2 g(x)(y - x)$$

$x + v$  is solution of

$$f_{\text{aff}}(y) = 0$$

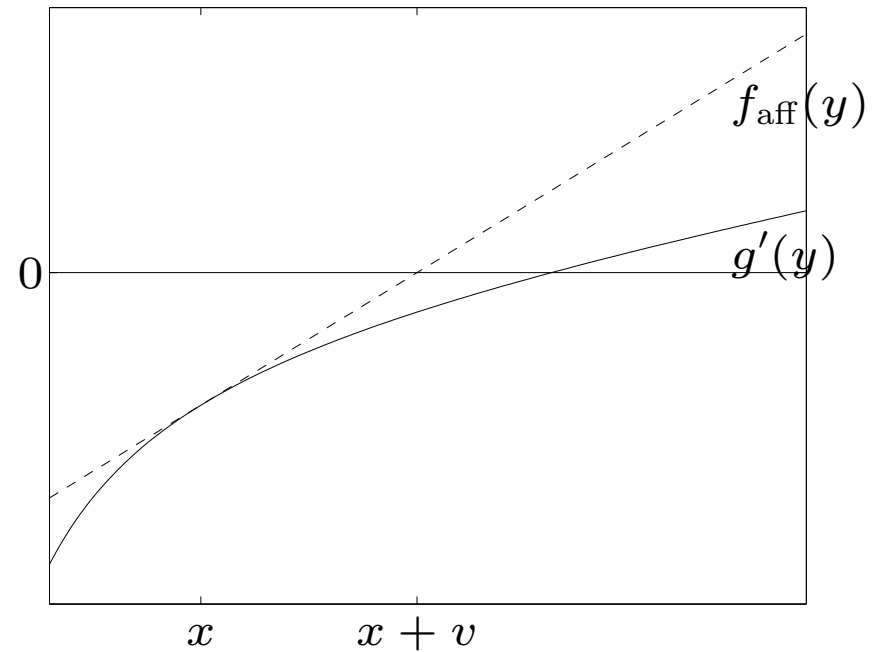
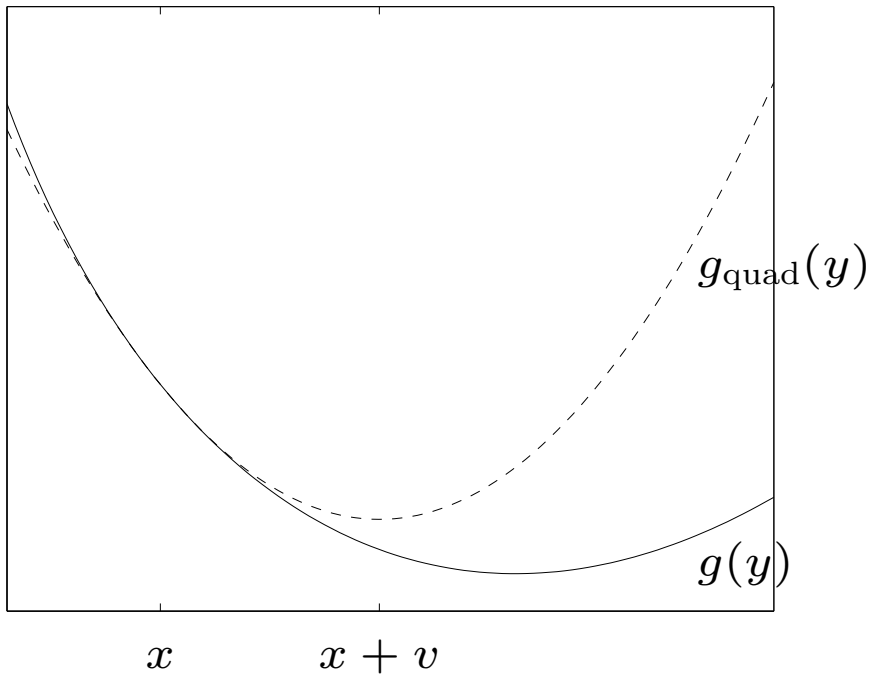
- construct second-order approximation of  $g(y)$  around  $x$ :

$$g_{\text{quad}}(y) = g(x) + \nabla g(x)^T (y - x) + \frac{1}{2}(y - x)^T \nabla^2 g(x)(y - x)$$

$x + v$  is solution of

$$\nabla g_{\text{quad}}(y) = 0$$

**example** ( $n = 1$ )

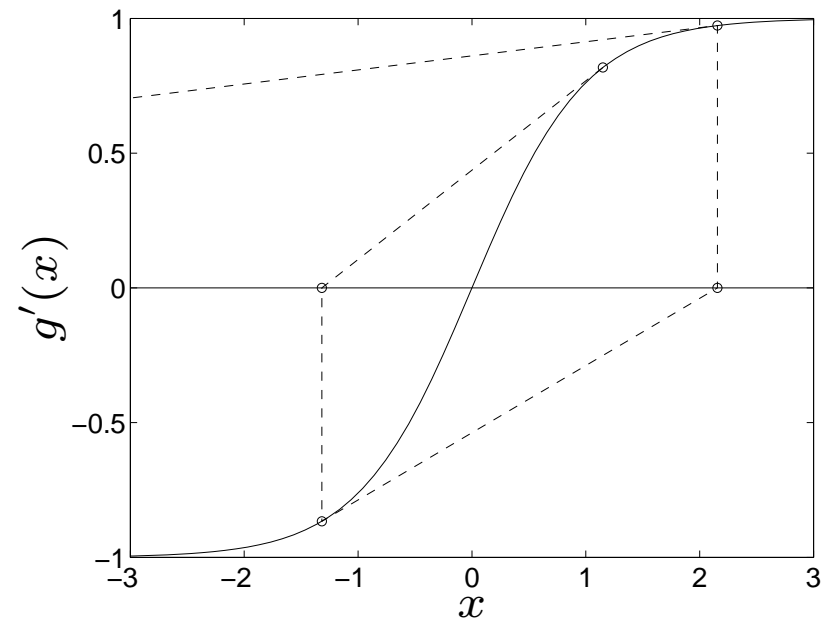
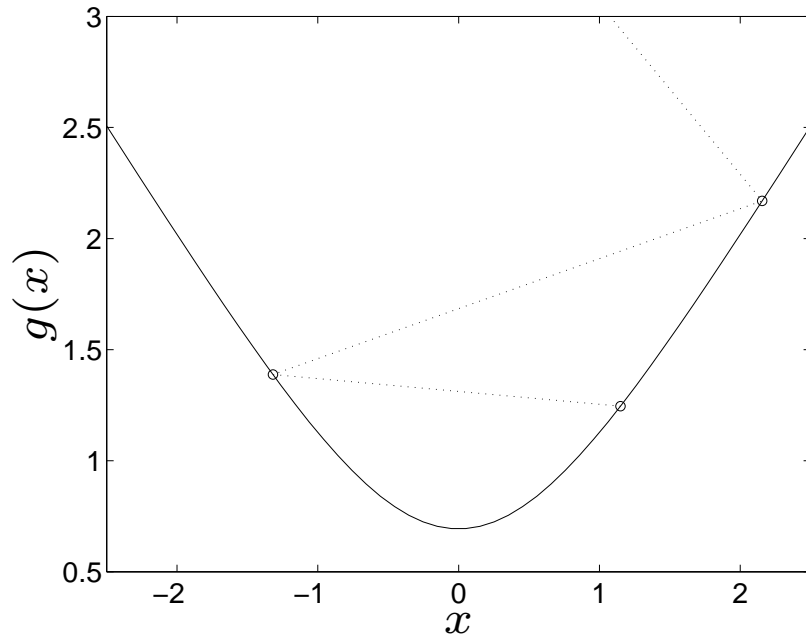


$$g_{\text{quad}}(y) = g(x) + g'(x)(y - x) + \frac{1}{2}g''(x)(y - x)^2$$

$$f_{\text{aff}}(y) = g'(x) + g''(x)(y - x)$$

# Example

$$g(x) = \log(e^x + e^{-x}), \quad g'(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad g''(x) = \frac{4}{(e^x + e^{-x})^2}$$



does not converge when started at  $x = 1.15$

# Newton's method with backtracking line search

use update  $x^+ = x + tv$ ; choose  $t$  so that  $g(x^+) < g(x)$

---

**given** initial  $x$ , tolerance  $\epsilon > 0$ , parameter  $\alpha \in (0, 1/2)$ .

**repeat**

1. evaluate  $\nabla g(x)$  and  $\nabla^2 g(x)$ .

2. **if**  $\|\nabla g(x)\| \leq \epsilon$ , **return**  $x$ .

3. Solve  $\nabla^2 g(x)v = -\nabla g(x)$ .

4.  $t := 1$ .

**while**  $g(x + tv) > g(x) + \alpha t \nabla g(x)^T v$ ,  $t := t/2$ .

5.  $x := x + tv$ .

**until** maximum number of iterations is exceeded

---

- typical values of  $\alpha$  are small (*e.g.*,  $\alpha = 0.01$ )
- $t$  is called the *step size*
- inner loop is called *backtracking line search*

## Interpretation of line search

to determine a suitable step size, consider the function  $h : \mathbf{R} \rightarrow \mathbf{R}$

$$h(t) = g(x + tv)$$

$x = x^{(k)}$  is the current iterate;  $v$  is the Newton step at  $x$

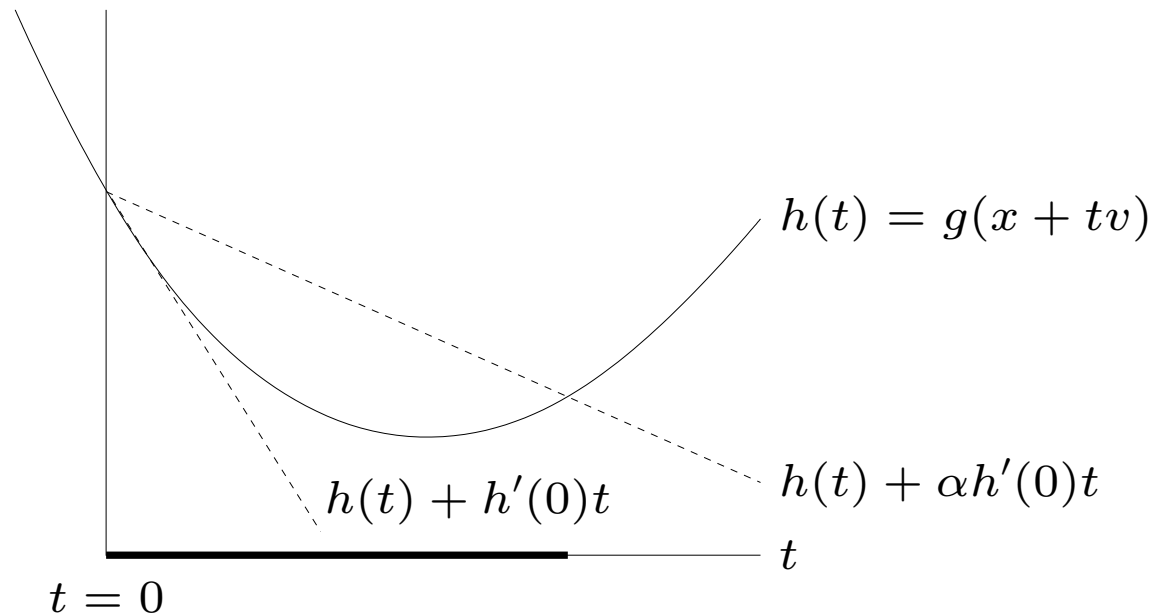
- derivative  $h'(t)$ :

$$\begin{aligned} h'(t) &= \frac{\partial g(x + tv)}{\partial x_1} v_1 + \frac{\partial g(x + tv)}{\partial x_2} v_2 + \cdots + \frac{\partial g(x + tv)}{\partial x_n} v_n \\ &= \nabla g(x + tv)^T v \end{aligned}$$

- at  $t = 0$ :  $h(0) = g(x)$ ,  $h'(0) = \nabla g(x)^T v$
- linear approximation of  $h$  at  $t = 0$ :

$$h(0) + h'(0)t = g(x) + t\nabla g(x)^T v$$

**backtracking:** start with  $t = 1$ ; divide  $t$  by 2 until  $h(t) \leq h(0) + \alpha h'(0)t$

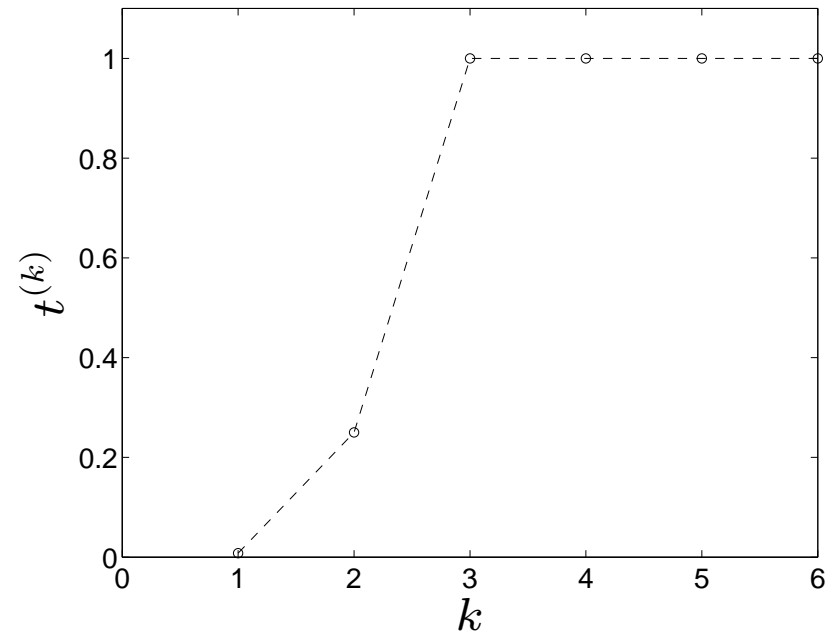
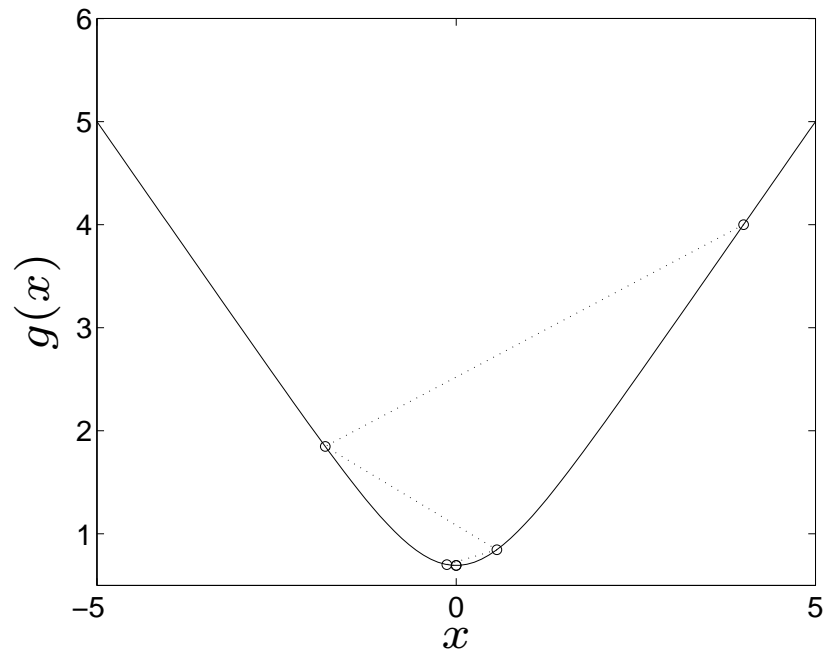


- guarantees that  $g(x^{(k+1)}) < g(x^{(k)})$
- works if  $h'(0) < 0$  ( $v$  is a **descent** direction)
- if  $\nabla^2 g(x)$  is positive definite, the Newton step is a descent direction

$$h'(0) = \nabla g(x)^T v = v^T \nabla^2 g(x) v < 0$$

# Examples

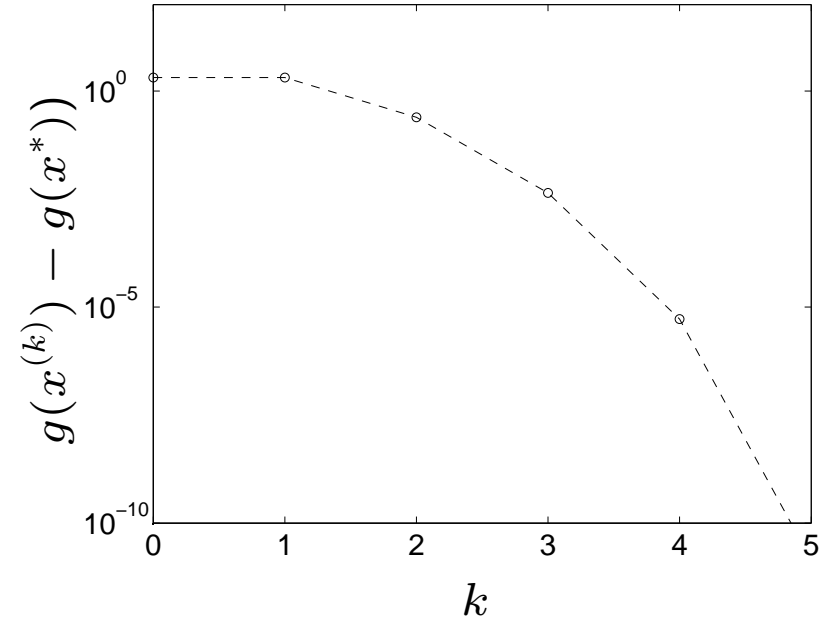
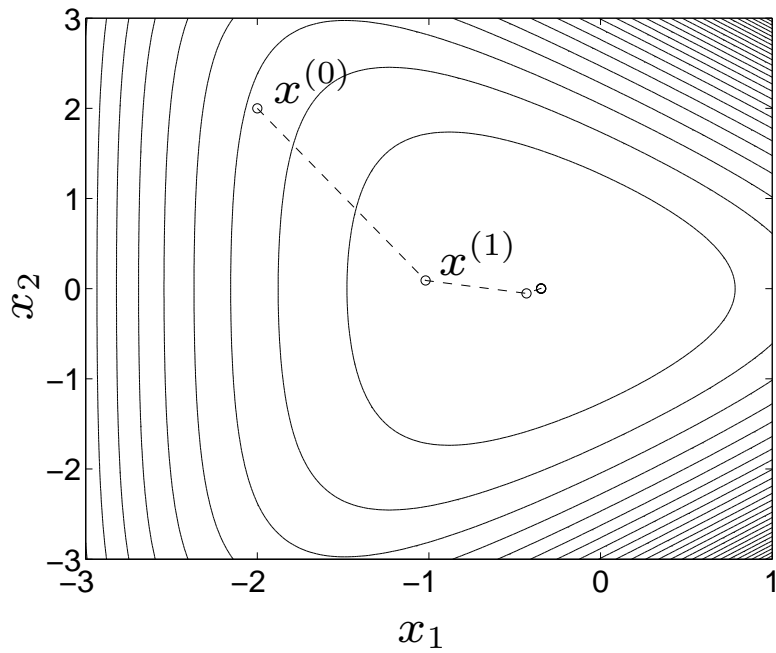
$$g(x) = \log(e^x + e^{-x}), \quad x^{(0)} = 10$$



close to the solution: very fast convergence, no backtracking steps

**example** (page 13-9):

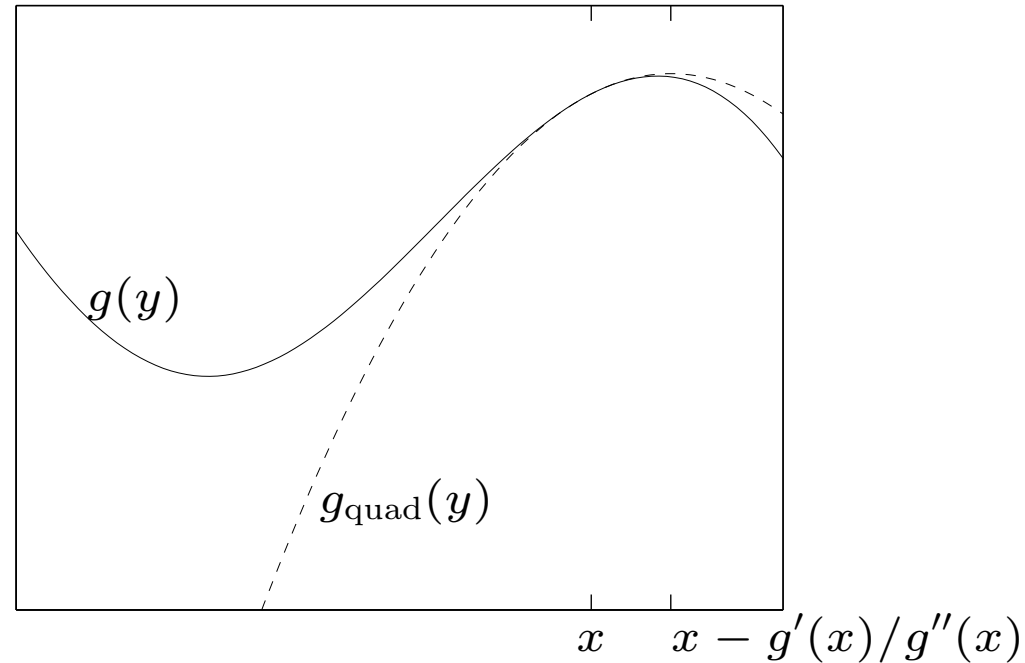
$$g(x_1, x_2) = e^{x_1+x_2-1} + e^{x_1-x_2-1} + e^{-x_1-1}$$



# Newton's method for nonconvex functions

if  $\nabla^2 g(x)$  not positive definite, Newton step may not be a descent direction

**example**



we move uphill because  $g''(x) < 0$

**solution:** always use a descent direction  $v$ , for example,  $v = -\nabla g(x)$

---

**given** initial  $x$ , tolerance  $\epsilon > 0$ , parameter  $\alpha \in (0, 1/2)$ .

**repeat**

1. evaluate  $\nabla g(x)$  and  $\nabla^2 g(x)$ .
2. **if**  $\|\nabla g(x)\| \leq \epsilon$ , **return**  $x$ .
3. **if**  $\nabla^2 g(x)$  is positive definite, solve  $\nabla^2 g(x)v = -\nabla g(x)$  for  $v$   
**else**,  $v := -\nabla g(x)$ .
4.  $t := 1$ .  
**while**  $g(x + tv) > g(x) + \alpha t \nabla g(x)^T v$ ,  $t := t/2$ .
5.  $x := x + tv$ .

**until** maximum number of iterations is exceeded

---

practical methods use more sophisticated choices of  $v$  if  $\nabla^2 g(x)$  is not positive definite