

# Variable metric methods

$$x^+ = x - tH^{-1}\nabla f(x)$$

$H \succ 0$  is approximation of the Hessian at  $x$ , chosen to:

- avoid calculation of second derivatives
- simplify computation of search direction

**‘Variable metric’ interpretation** (EE236B, lecture 10, page 11)

$$\Delta x = -H^{-1}\nabla f(x)$$

is steepest descent direction at  $x$  for quadratic norm

$$\|z\|_H = (z^T H z)^{1/2}$$

# Quasi-Newton methods

**given** starting point  $x^{(0)} \in \text{dom } f$ ,  $H_0 \succ 0$

1. compute quasi-Newton direction  $\Delta x = -H_{k-1}^{-1} \nabla f(x^{(k-1)})$
  2. determine step size  $t$  (e.g., by backtracking line search)
  3. compute  $x^{(k)} = x^{(k-1)} + t\Delta x$
  4. compute  $H_k$
- different methods use different rules for updating  $H$  in step 4
  - can also propagate  $H_k^{-1}$  to simplify calculation of  $\Delta x$

# Broyden-Fletcher-Goldfarb-Shanno (BFGS) update

## BFGS update

$$H_k = H_{k-1} + \frac{yy^T}{y^T s} - \frac{H_{k-1} s s^T H_{k-1}}{s^T H_{k-1} s}$$

where

$$s = x^{(k)} - x^{(k-1)}, \quad y = \nabla f(x^{(k)}) - \nabla f(x^{(k-1)})$$

## Inverse update

$$H_k^{-1} = \left( I - \frac{sy^T}{y^T s} \right) H_{k-1}^{-1} \left( I - \frac{ys^T}{y^T s} \right) + \frac{ss^T}{y^T s}$$

- note that  $y^T s > 0$  for strictly convex  $f$ ; see page 1-9
- cost of update or inverse update is  $O(n^2)$  operations

## Positive definiteness

if  $y^T s > 0$ , BFGS update preserves positive definiteness of  $H_k$

**Proof:** from inverse update formula,

$$v^T H_k^{-1} v = \left( v - \frac{s^T v}{s^T y} y \right)^T H_{k-1}^{-1} \left( v - \frac{s^T v}{s^T y} y \right) + \frac{(s^T v)^2}{y^T s}$$

- if  $H_{k-1} \succ 0$ , both terms are nonnegative for all  $v$
- second term is zero only if  $s^T v = 0$ ; then first term is zero only if  $v = 0$

this ensures that  $\Delta x = -H_k^{-1} \nabla f(x^k)$  is a descent direction

## Secant condition

the BFGS update satisfies the *secant condition*  $H_k s = y$ , i.e.,

$$H_k(x^{(k)} - x^{(k-1)}) = \nabla f(x^{(k)}) - \nabla f(x^{(k-1)})$$

**Interpretation:** define second-order approximation at  $x^{(k)}$

$$f_{\text{quad}}(z) = f(x^{(k)}) + \nabla f(x^{(k)})^T (z - x^{(k)}) + \frac{1}{2}(z - x^{(k)})^T H_k (z - x^{(k)})$$

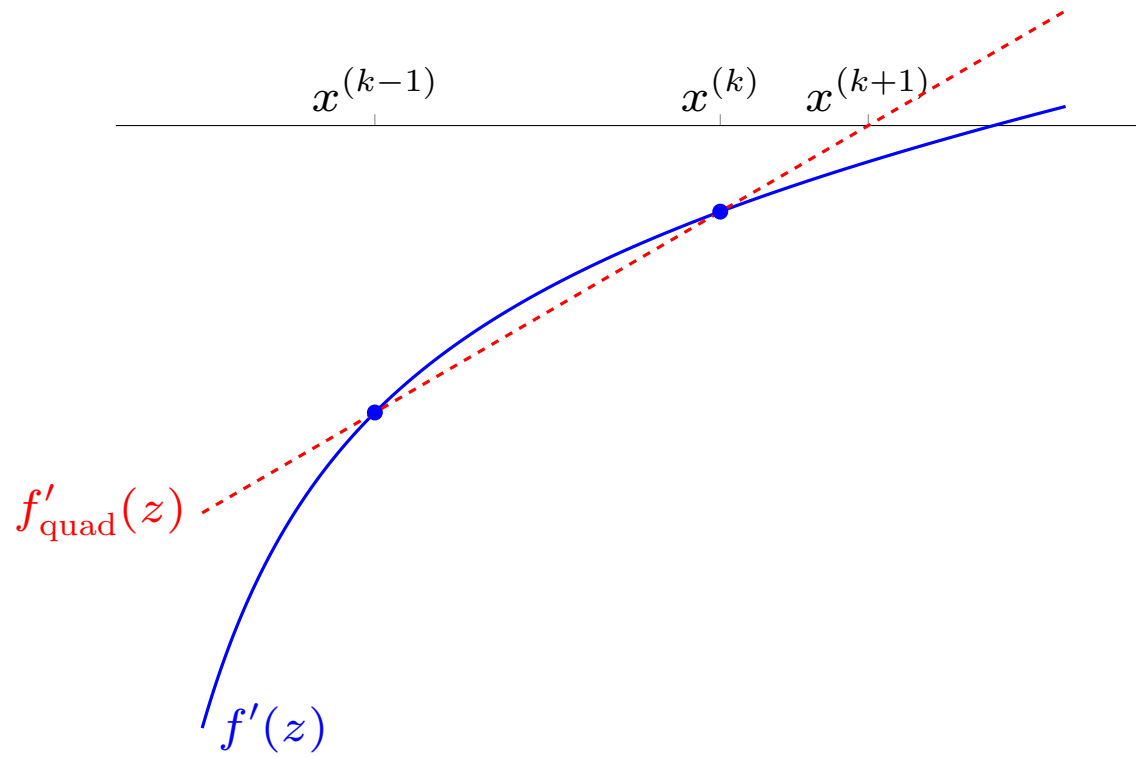
secant condition implies that gradient of  $f_{\text{quad}}$  agrees with  $f$  at  $x^{(k-1)}$ :

$$\begin{aligned} \nabla f_{\text{quad}}(x^{(k-1)}) &= \nabla f(x^{(k)}) + H_k(x^{(k-1)} - x^{(k)}) \\ &= \nabla f(x^{(k-1)}) \end{aligned}$$

# Secant method

for  $f : \mathbf{R} \rightarrow \mathbf{R}$ , BFGS with unit step size gives the secant method

$$x^{(k+1)} = x^{(k)} - \frac{f'(x^{(k)})}{H_k}, \quad H_k = \frac{f'(x^{(k)}) - f'(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}$$



# Convergence

## Global result

if  $f$  is strongly convex, BFGS with backtracking line search (EE236B, lecture 10-6) converges from any  $x^{(0)}$ ,  $H_0 \succ 0$

## Local convergence

if  $f$  is strongly convex and  $\nabla^2 f(x)$  is Lipschitz continuous, local convergence is *superlinear*: for sufficiently large  $k$ ,

$$\|x^{(k+1)} - x^*\|_2 \leq c_k \|x^{(k)} - x^*\|_2 \rightarrow 0$$

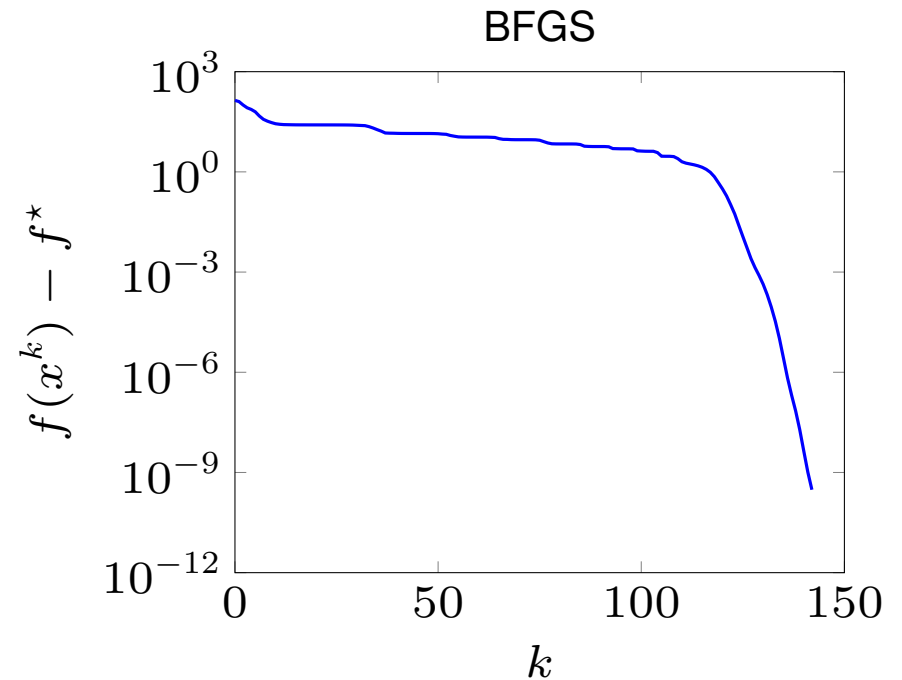
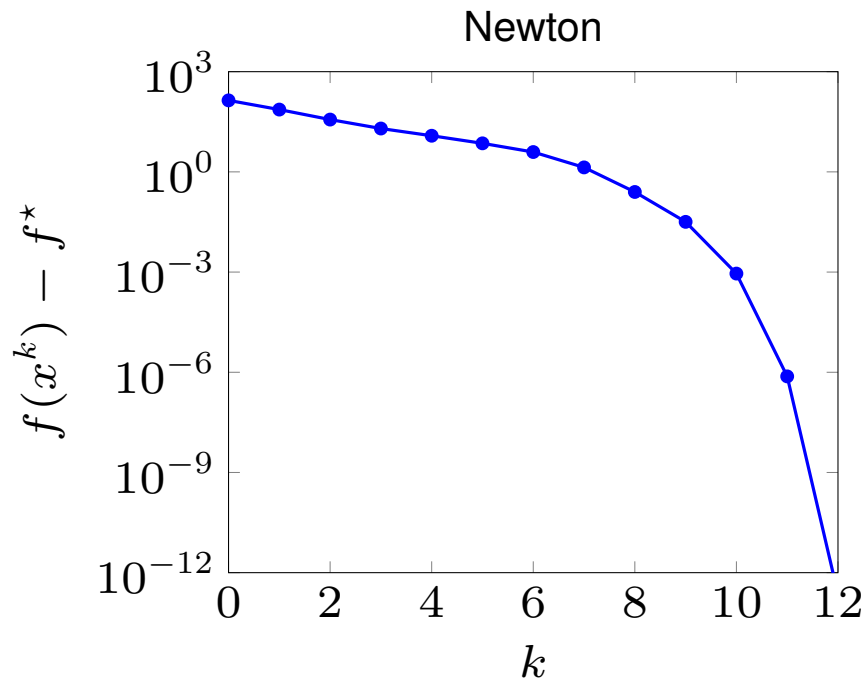
where  $c_k \rightarrow 0$

(*cf.*, quadratic local convergence of Newton method)

# Example

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

$$n = 100, m = 500$$



- cost per Newton iteration:  $O(n^3)$  plus computing  $\nabla^2 f(x)$
- cost per BFGS iteration:  $O(n^2)$



# Limited memory quasi-Newton methods

main disadvantage of quasi-Newton method is need to store  $H_k$  or  $H_k^{-1}$

**Limited-memory BFGS (L-BFGS):** do not store  $H_k^{-1}$  explicitly

- instead we store the  $m$  (e.g.,  $m = 30$ ) most recent values of

$$s_j = x^{(j)} - x^{(j-1)}, \quad y_j = \nabla f(x^{(j)}) - \nabla f(x^{(j-1)})$$

- we evaluate  $\Delta x = H_k^{-1} \nabla f(x^{(k)})$  recursively, using

$$H_j^{-1} = \left( I - \frac{s_j y_j^T}{y_j^T s_j} \right) H_{j-1}^{-1} \left( I - \frac{y_j s_j^T}{y_j^T s_j} \right) + \frac{s_j s_j^T}{y_j^T s_j}$$

for  $j = k, k - 1, \dots, k - m + 1$ , assuming, for example,  $H_{k-m}^{-1} = I$

- cost per iteration is  $O(nm)$ ; storage is  $O(nm)$