

# 15. Quasi-Newton methods

- variable metric methods
- quasi-Newton methods
- BFGS update
- limited-memory quasi-Newton methods

# Newton method for unconstrained minimization

minimize  $f(x)$

$f$  convex, twice continuously differentiable

## Newton method

$$x_{k+1} = x_k - t_k \nabla^2 f(x_k)^{-1} \nabla f(x_k)$$

- advantages: fast convergence, robustness, affine invariance
- disadvantages: requires second derivatives and solution of linear equation

can be too expensive for large scale applications

## Variable metric methods

$$x_{k+1} = x_k - t_k H_k^{-1} \nabla f(x_k)$$

the positive definite matrix  $H_k$  is an approximation of the Hessian at  $x_k$ , chosen to:

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

**“Variable metric” interpretation** (236B, lecture 10, page 11)

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

is the steepest descent direction at  $x$  for the quadratic norm

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

# Quasi-Newton methods

**given:** starting point  $x_0 \in \text{dom } f$ ,  $H_0 \succ 0$

**for**  $k = 0, 1, \dots$

1. compute quasi-Newton direction  $\Delta x_k = -H_k^{-1} \nabla f(x_k)$
  2. determine step size  $t_k$  (e.g., by backtracking line search)
  3. compute  $x_{k+1} = x_k + t_k \Delta x_k$
  4. compute  $H_{k+1}$
- different update rules exist for  $H_{k+1}$  in step 4
  - can also propagate  $H_k^{-1}$  or a factorization of  $H_k$  to simplify calculation of  $\Delta x_k$

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

## BFGS update

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

where

$$s = x_{k+1} - x_k, \quad y = \nabla f(x_{k+1}) - \nabla f(x_k)$$

## Inverse update

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

- note that  $y^T s > 0$  for strictly convex  $f$ ; see page 1.8
- cost of update or inverse update is  $O(n^2)$  operations

## Positive definiteness

- if  $y^T s > 0$ , BFGS update preserves positive definiteness of  $H_k$
- this ensures that  $\Delta x = -H_k^{-1} \nabla f(x_k)$  is a descent direction

*Proof:* from inverse update formula,

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

- if  $H_k > 0$ , both terms are nonnegative for all  $v$
- second term is zero only if  $s^T v = 0$ ; then first term is zero only if  $v = 0$

## Secant condition

the BFGS update satisfies the *secant condition*

$$H_{k+1}s = y$$

where  $s = x_{k+1} - x_k$  and  $y = \nabla f(x_{k+1}) - \nabla f(x_k)$

**Interpretation:** we define a quadratic approximation of  $f$  around  $x_{k+1}$

$$\tilde{f}(x) = f(x_{k+1}) + \nabla f(x_{k+1})^T (x - x_{k+1}) + \frac{1}{2}(x - x_{k+1})^T H_{k+1}(x - x_{k+1})$$

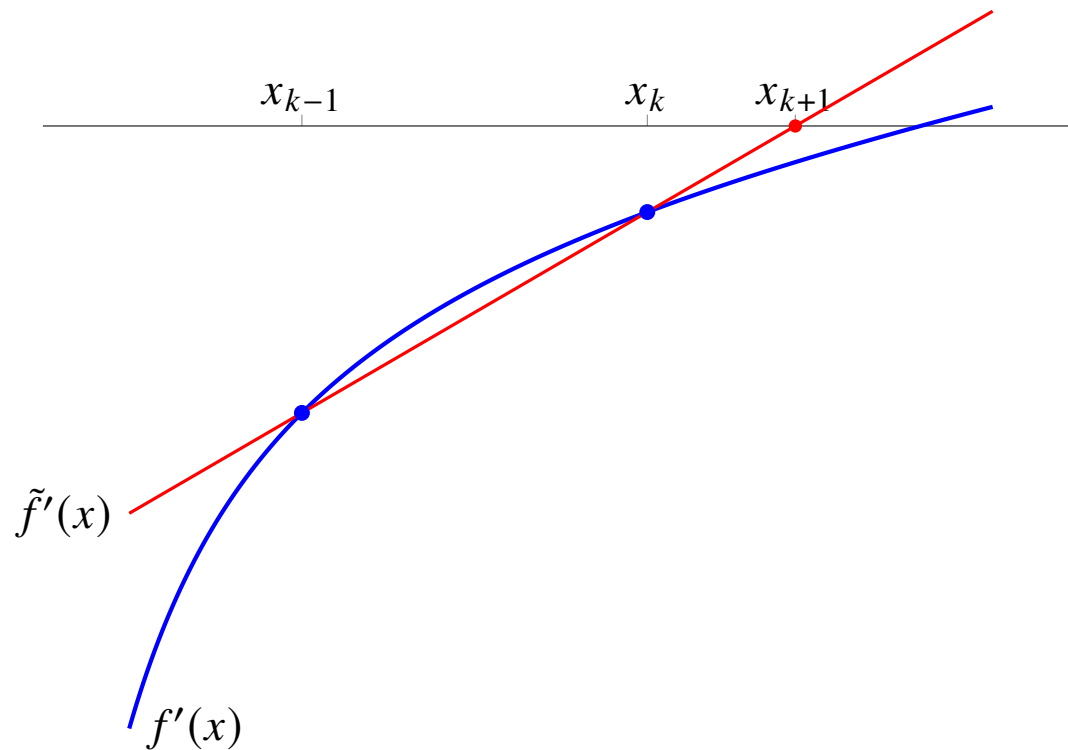
- by construction  $\nabla \tilde{f}(x_{k+1}) = \nabla f(x_{k+1})$
- secant condition implies that also  $\nabla \tilde{f}(x_k) = \nabla f(x_k)$ :

$$\begin{aligned}\nabla \tilde{f}(x_k) &= \nabla f(x_{k+1}) + H_{k+1}(x_k - x_{k+1}) \\ &= \nabla f(x_k)\end{aligned}$$

# 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  $x_0$ ,  $H_0 > 0$

## Local convergence

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

$$\|x_{k+1} - x^\star\|_2 \leq c_k \|x_k - x^\star\|_2$$

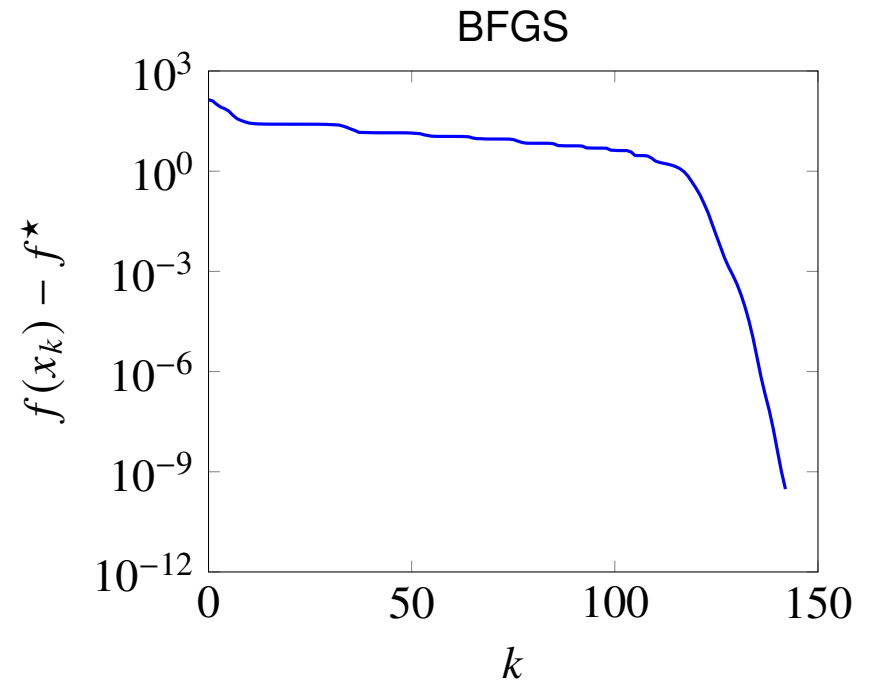
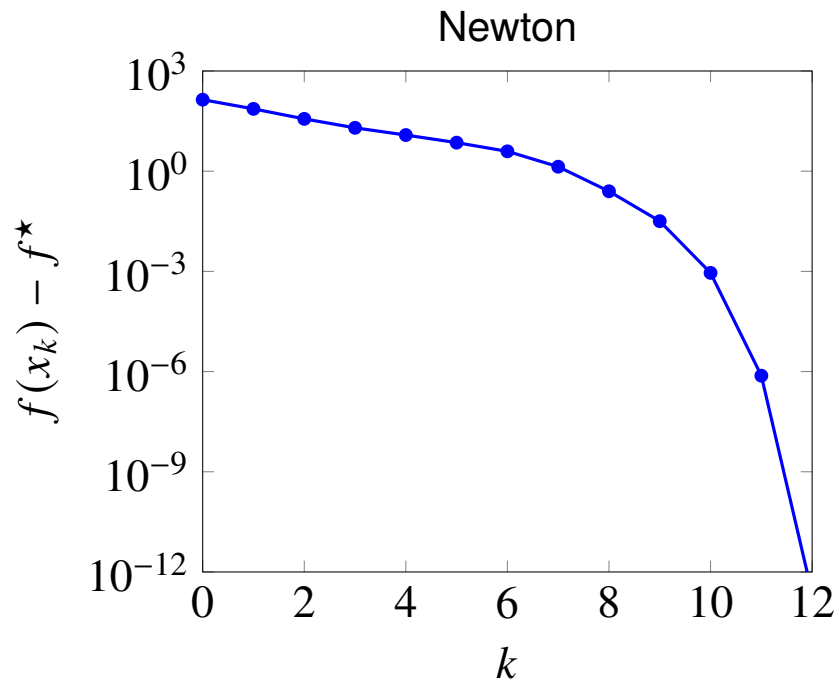
where  $c_k \rightarrow 0$

(*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:  $O(n^2)$

## Square root BFGS update

to improve numerical stability, propagate  $H_k$  in factored form  $H_k = L_k L_k^T$

- if  $H_k = L_k L_k^T$  then  $H_{k+1} = L_{k+1} L_{k+1}^T$  with

$$L_{k+1} = L_k \left( I + \frac{(\alpha \tilde{y} - \tilde{s}) \tilde{s}^T}{\tilde{s}^T \tilde{s}} \right),$$

where

$$\tilde{y} = L_k^{-1} y, \quad \tilde{s} = L_k^T s, \quad \alpha = \left( \frac{\tilde{s}^T \tilde{s}}{y^T s} \right)^{1/2}$$

- if  $L_k$  is triangular, cost of reducing  $L_{k+1}$  to triangular form is  $O(n^2)$

## Optimality of BFGS update

$X = H_{k+1}$  solves the convex optimization problem

$$\begin{aligned} & \text{minimize} && \text{tr}(H_k^{-1}X) - \log \det(H_k^{-1}X) - n \\ & \text{subject to} && Xs = y \end{aligned}$$

- cost function is nonnegative, equal to zero only if  $X = H_k$
- also known as relative entropy between densities  $N(0, X)$ ,  $N(0, H_k)$
- BFGS update is a *least-change secant update*

optimality result follows from KKT conditions:  $X = H_{k+1}$  satisfies

$$X^{-1} = H_k^{-1} - \frac{1}{2}(sv^T + vs^T), \quad Xs = y, \quad X \succ 0$$

with

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

# Davidon–Fletcher–Powell (DFP) update

switch  $H_k$  and  $X$  in objective on previous page

$$\begin{array}{ll} \text{minimize} & \text{tr}(H_k X^{-1}) - \log \det(H_k X^{-1}) - n \\ \text{subject to} & Xs = y \end{array}$$

- minimize relative entropy between  $N(0, H_k)$  and  $N(0, X)$
- problem is convex in  $X^{-1}$  (with constraint written as  $s = X^{-1}y$ )
- solution is ‘dual’ of BFGS formula

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

(known as DFP update)

predates BFGS update, but is less often used

# Limited memory quasi-Newton methods

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

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

- instead we store up to  $m$  (e.g.,  $m = 30$ ) values of

$$s_j = x_{j+1} - x_j, \quad y_j = \nabla f(x_{j+1}) - \nabla f(x_j)$$

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

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

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

- an alternative is to restart after  $m$  iterations
- cost per iteration is  $O(nm)$ , storage is  $O(nm)$

# Interpretation of CG as restarted BFGS method

first two iterations of BFGS (page 15.5) if  $H_0 = I$ :

$$x_1 = x_0 - t_0 \nabla f(x_0), \quad x_2 = x_1 - t_1 H_1^{-1} \nabla f(x_1)$$

where  $H_1$  is computed from  $s = x_1 - x_0$  and  $y = \nabla f(x_1) - \nabla f(x_0)$  via

$$H_1^{-1} = I + \left(1 + \frac{y^T y}{s^T y}\right) \frac{ss^T}{y^T s} - \frac{ys^T + sy^T}{y^T s}$$

- if  $t_0$  is determined by exact line search, then  $\nabla f(x_1)^T s = 0$
- quasi-Newton step in second iteration simplifies to

$$-H_1^{-1} \nabla f(x_1) = -\nabla f(x_1) + \frac{y^T \nabla f(x_1)}{y^T s} s$$

this is the Hestenes–Stiefel conjugate gradient update

nonlinear CG can be interpreted as L-BFGS with  $m = 1$

# References

- J. E. Dennis, Jr., and R. B. Schabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (1996), chapter 9.
- C. T. Kelley, *Iterative Methods for Optimization* (1999), chapter 4.
- J. Nocedal and S. J. Wright, *Numerical Optimization* (2006), chapter 6 and section 7.2.