Practical Optimization Algorithms
实用优化算法

徐 翔

数学科学学院
浙江大学

April 15, 2021

Chapter V: Quasi-Newton Methods (拟牛顿法)

Outline

- The DFP (Davidon, Fletcher and Powell)
- The BFGS (Broyden, Fletcher, Goldfarb and Shanno)
- The SR1 (Symmetric-Rank-1)
- The Broyden Class (DFP+BFGS)
- Convergence Analysis

The DFP Method

- The approaching quadratic model of the objective function at the current iterate x_k is

$$f(x_k + p) \approx m_k(p) = f_k + \nabla_k^T p + \frac{1}{2} p^T B_k p \tag{5.1}$$

where B_k is an $n \times n$ symmetric positive definite (SPD) matrix that will be revised or updated at every iteration.

The DFP Method

- The approaching quadratic model of the objective function at the current iterate x_k is

$$f(x_k + p) \approx m_k(p) = f_k + \nabla_k^T p + \frac{1}{2} p^T B_k p \tag{5.1}$$

where B_k is an $n \times n$ symmetric positive definite (SPD) matrix that will be revised or updated at every iteration.

- The minimizer p_k of this convex quadratic model, which we can write explicitly as

$$p_k = -B_k^{-1} \nabla f_k \tag{5.2}$$

is the search direction.

The DFP Method

- The approaching quadratic model of the objective function at the current iterate x_k is

$$f(x_k + p) \approx m_k(p) = f_k + \nabla_k^T p + \frac{1}{2} p^T B_k p \tag{5.1}$$

  where B_k is an $n \times n$ symmetric positive definite (SPD) matrix that will be revised or updated at every iteration.

- The minimizer p_k of this convex quadratic model, which we can write explicitly as

$$p_k = -B_k^{-1} \nabla f_k \tag{5.2}$$

  is the search direction.
- The new iterate is

$$x_{k+1} = x_k + \alpha_k p_k \tag{5.3}$$

  where the step length α_k is chosen to satisfy the Wolfe conditions.

The DFP Method

- Instead of computing B_k afresh at every iteration, W.C. Davidon proposed to update it in a simple manner to account for the curvature measured during the most recent step (用简单的形式).

The DFP Method

- Instead of computing B_k afresh at every iteration, W.C. Davidon proposed to update it in a simple manner to account for the curvature measured during the most recent step (用简单的形式).
- Suppose that we have generated a new iterate x_{k+1} and wish to construct a new quadratic model, of the form

$$m_{k+1}(p) = f_{k+1} + \nabla f_{k+1}^T p + \frac{1}{2} p^T B_{k+1} p \tag{5.4}$$

The DFP Method

- Instead of computing B_k afresh at every iteration, W.C. Davidon proposed to update it in a simple manner to account for the curvature measured during the most recent step (用简单的形式).
- Suppose that we have generated a new iterate x_{k+1} and wish to construct a new quadratic model, of the form

$$m_{k+1}(p) = f_{k+1} + \nabla f_{k+1}^T p + \frac{1}{2} p^T B_{k+1} p \tag{5.4}$$

- What requirements should we impose on B_{k+1}? Based on the knowledge we have gained during the latest step.

Secant Equation

- The gradient of $m_{k+1}(p) = f_{k+1} + \nabla f_{k+1}^T p + \frac{1}{2} p^T B_{k+1} p$ should match the gradient of the objective function f at the latest two iterates x_{k+1} and x_k,

Secant Equation

- The gradient of $m_{k+1}(p) = f_{k+1} + \nabla f_{k+1}^T p + \frac{1}{2} p^T B_{k+1} p$ should match the gradient of the objective function f at the latest two iterates x_{k+1} and x_k, i.e., $\nabla m_{k+1}(0) = \nabla f_{k+1}$ (automatically) and

Secant Equation

- The gradient of $m_{k+1}(p) = f_{k+1} + \nabla f_{k+1}^T p + \frac{1}{2} p^T B_{k+1} p$ should match the gradient of the objective function f at the latest two iterates x_{k+1} and x_k, i.e., $\nabla m_{k+1}(0) = \nabla f_{k+1}$ (automatically) and

$$\nabla m_{k+1}(-\alpha_k p_k) = \nabla f_{k+1} - \alpha_k B_{k+1} p_k = \nabla f_k$$

Secant Equation

- The gradient of $m_{k+1}(p) = f_{k+1} + \nabla f_{k+1}^T p + \frac{1}{2} p^T B_{k+1} p$ should match the gradient of the objective function f at the latest two iterates x_{k+1} and x_k, i.e., $\nabla m_{k+1}(0) = \nabla f_{k+1}$ (automatically) and

$$\nabla m_{k+1}(-\alpha_k p_k) = \nabla f_{k+1} - \alpha_k B_{k+1} p_k = \nabla f_k$$

- Define

$$s_k = x_{k+1} - x_k = \alpha_k p_k, \quad y_k = \nabla f_{k+1} - \nabla f_k \tag{5.5}$$

Secant Equation

- The gradient of $m_{k+1}(p) = f_{k+1} + \nabla f_{k+1}^T p + \frac{1}{2} p^T B_{k+1} p$ should match the gradient of the objective function f at the latest two iterates x_{k+1} and x_k, i.e., $\nabla m_{k+1}(0) = \nabla f_{k+1}$ (automatically) and

$$\nabla m_{k+1}(-\alpha_k p_k) = \nabla f_{k+1} - \alpha_k B_{k+1} p_k = \nabla f_k$$

- Define

$$s_k = x_{k+1} - x_k = \alpha_k p_k, \quad y_k = \nabla f_{k+1} - \nabla f_k \tag{5.5}$$

- We get

$$B_{k+1} s_k = y_k \tag{5.6}$$

which is referred as the *secant equation (割线方程)*

Curvature Equation

Curvature Equation

- Given the displacement s_k and the change of gradients y_k , the secant equation requires that the symmetric positive definite (SPD) matrix B_{k+1} map s_k into y_k.

Curvature Equation

- Given the displacement s_k and the change of gradients y_k , the secant equation requires that the symmetric positive definite (SPD) matrix B_{k+1} map s_k into y_k.
- This will be possible only s_k,y_k satisfy the *curvature condition*

$$s_k^T y_k > 0 \tag{5.7}$$

Curvature Equation

- Given the displacement s_k and the change of gradients y_k , the secant equation requires that the symmetric positive definite (SPD) matrix B_{k+1} map s_k into y_k.
- This will be possible only s_k,y_k satisfy the *curvature condition*

$$s_k^T y_k > 0 \tag{5.7}$$

- By premultiplying (5.6) ($B_{k+1}s_k = y_k$) by s_k^T.

Curvature Equation

- Given the displacement s_k and the change of gradients y_k , the secant equation requires that the symmetric positive definite (SPD) matrix B_{k+1} map s_k into y_k.
- This will be possible only s_k,y_k satisfy the *curvature condition*

$$s_k^T y_k > 0 \tag{5.7}$$

- By premultiplying (5.6) ($B_{k+1}s_k = y_k$) by s_k^T.
- When f is strongly convex, the inequality (5.7) will be satisfied for any two points x_k and x_{k+1}.

Curvature Equation

- Given the displacement s_k and the change of gradients y_k , the secant equation requires that the symmetric positive definite (SPD) matrix B_{k+1} map s_k into y_k.
- This will be possible only s_k,y_k satisfy the *curvature condition*

$$s_k^T y_k > 0 \tag{5.7}$$

- By premultiplying (5.6) $(B_{k+1}s_k = y_k)$ by s_k^T.
- When f is strongly convex, the inequality (5.7) will be satisfied for any two points x_k and x_{k+1}. (Since $\nabla f_{k+1} - \nabla f_k = \nabla^2 f(x + \tau s_k)s_k,\ \tau \in (0,1)$)

Curvature Equation

- Given the displacement s_k and the change of gradients y_k , the secant equation requires that the symmetric positive definite (SPD) matrix B_{k+1} map s_k into y_k.
- This will be possible only s_k,y_k satisfy the *curvature condition*

$$s_k^T y_k > 0 \tag{5.7}$$

- By premultiplying (5.6) $(B_{k+1}s_k = y_k)$ by s_k^T.
- When f is strongly convex, the inequality (5.7) will be satisfied for any two points x_k and x_{k+1}. (Since $\nabla f_{k+1} - \nabla f_k = \nabla^2 f(x + \tau s_k)s_k, \ \tau \in (0,1)$)
- In fact, above condition is guaranteed to hold if we impose the Wolfe or strong Wolfe conditions on the line search.

Curvature Equation

- Given the displacement s_k and the change of gradients y_k , the secant equation requires that the symmetric positive definite (SPD) matrix B_{k+1} map s_k into y_k.
- This will be possible only s_k,y_k satisfy the *curvature condition*

$$s_k^T y_k > 0 \tag{5.7}$$

- By premultiplying (5.6) $(B_{k+1}s_k = y_k)$ by s_k^T.
- When f is strongly convex, the inequality (5.7) will be satisfied for any two points x_k and x_{k+1}. (Since $\nabla f_{k+1} - \nabla f_k = \nabla^2 f(x + \tau s_k)s_k,\ \tau \in (0,1)$)
- In fact, above condition is guaranteed to hold if we impose the Wolfe or strong Wolfe conditions on the line search.
Since $(\nabla f_{k+1})^T s_k \geq c_2(\nabla f_k)^T s_k$,

Curvature Equation

- Given the displacement s_k and the change of gradients y_k , the secant equation requires that the symmetric positive definite (SPD) matrix B_{k+1} map s_k into y_k.
- This will be possible only s_k,y_k satisfy the *curvature condition*

$$s_k^T y_k > 0 \tag{5.7}$$

- By premultiplying (5.6) ($B_{k+1}s_k = y_k$) by s_k^T.
- When f is strongly convex, the inequality (5.7) will be satisfied for any two points x_k and x_{k+1}. (Since $\nabla f_{k+1} - \nabla f_k = \nabla^2 f(x + \tau s_k)s_k,\ \tau \in (0,1)$)
- In fact, above condition is guaranteed to hold if we impose the Wolfe or strong Wolfe conditions on the line search.
  Since $(\nabla f_{k+1})^T s_k \geq c_2(\nabla f_k)^T s_k$,
  therefore $y_k^T s_k \geq (c_2 - 1)\alpha_k(\nabla f_k)^T p_k$.

Curvature Equation

- Given the displacement s_k and the change of gradients y_k , the secant equation requires that the symmetric positive definite (SPD) matrix B_{k+1} map s_k into y_k.
- This will be possible only s_k,y_k satisfy the *curvature condition*

$$s_k^T y_k > 0 \tag{5.7}$$

- By premultiplying (5.6) ($B_{k+1} s_k = y_k$) by s_k^T.
- When f is strongly convex, the inequality (5.7) will be satisfied for any two points x_k and x_{k+1}. (Since $\nabla f_{k+1} - \nabla f_k = \nabla^2 f(x + \tau s_k) s_k, \ \tau \in (0, 1)$)
- In fact, above condition is guaranteed to hold if we impose the Wolfe or strong Wolfe conditions on the line search.
  Since $(\nabla f_{k+1})^T s_k \geq c_2 (\nabla f_k)^T s_k$,
  therefore $y_k^T s_k \geq (c_2 - 1) \alpha_k (\nabla f_k)^T p_k$.
  Sice $c_2 < 1$ and p_k is a decent direction, the right term is positive.

The DFP Method

- When the curvature condition is satisfied, the secant equation always has a solution B_{k+1}.

The DFP Method

- When the curvature condition is satisfied, the secant equation always has a solution B_{k+1}.
- In fact, it admits an infinite number of solutions, since there are $\frac{n(n+1)}{2}$ degrees of freedom in a symmetric matrix, and the secant equation represents only n conditions.

The DFP Method

- When the curvature condition is satisfied, the secant equation always has a solution B_{k+1}.
- In fact, it admits an infinite number of solutions, since there are $\frac{n(n+1)}{2}$ degrees of freedom in a symmetric matrix, and the secant equation represents only n conditions.
- To determine B_{k+1} uniquely, then, we impose the additional condition that among all symmetric matrices satisfying the secant equation, B_{k+1} is, in some sense, closest to the current matrix B_k.

The DFP Method

- When the curvature condition is satisfied, the secant equation always has a solution B_{k+1}.
- In fact, it admits an infinite number of solutions, since there are $\frac{n(n+1)}{2}$ degrees of freedom in a symmetric matrix, and the secant equation represents only n conditions.
- To determine B_{k+1} uniquely, then, we impose the additional condition that among all symmetric matrices satisfying the secant equation, B_{k+1} is, in some sense, closest to the current matrix B_k.
- In other words, we solve the problem

$$\min_{B} \quad \|B - B_k\| \tag{5.8a}$$

$$s.t. \quad B = B^T, \quad Bs_k = y_k, \tag{5.8b}$$

where s_k and y_k satisfy the curvature condition (5.7) and B_k is symmetric and positive definite.

The DFP Method

- When the curvature condition is satisfied, the secant equation always has a solution B_{k+1}.
- In fact, it admits an infinite number of solutions, since there are $\frac{n(n+1)}{2}$ degrees of freedom in a symmetric matrix, and the secant equation represents only n conditions.
- To determine B_{k+1} uniquely, then, we impose the additional condition that among all symmetric matrices satisfying the secant equation, B_{k+1} is, in some sense, closest to the current matrix B_k.
- In other words, we solve the problem

$$\min_{B} \quad \|B - B_k\| \tag{5.8a}$$

$$s.t. \quad B = B^T, \quad Bs_k = y_k, \tag{5.8b}$$

  where s_k and y_k satisfy the curvature condition (5.7) and B_k is symmetric and positive definite.
- Many matrix norms can be used in (5.8a), and each norm gives rise to a different quasi-Newton method.

The DFP Method

Theorem

Assume $B \in R^{n\times n}$ is symmetric, $c,\ s,\ y \in R^n$, satisfying $c^T s > 0$. Suppose $M \in R^{n\times n}$ is a nonsingular symmetric matrix, satisfying $c = M^{-2}s$, then

$$\bar{B} = B + \frac{(y - Bs)c^T + c(y - Bs)^T}{c^T s} - \frac{(y - Bs)^T s}{(c^T s)^2} cc^T$$

is the unique solution of the following minization problem

$$\min \left\{ \|\hat{B} - B\|_{M,F}, \quad s.t.\ \ \hat{B}s = y, \quad \hat{B}^T = \hat{B} \right\}$$

where $\|B\|_{M,F} = \|MBM\|_F$ and $\|\cdot\|_F$ is the Frobenius norm defined by $\|C\|_F = \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}^2$

The DFP Method

$$\bar{B} - B = \frac{(y - Bs)c^T + c(y - Bs)^T}{c^T s} - \frac{(y - Bs)^T s}{(c^T s)^2} c c^T$$

Proof

The DFP Method

$$\bar{B} - B = \frac{(y - Bs)c^T + c(y - Bs)^T}{c^T s} - \frac{(y - Bs)^T s}{(c^T s)^2} cc^T$$

Proof

- Let $Mc = M^{-1}s = z$, $E = M(\hat{B} - B)M$, $\bar{E} = M(\bar{B} - B)M$.

The DFP Method

$$\bar{B} - B = \frac{(y - Bs)c^T + c(y - Bs)^T}{c^T s} - \frac{(y - Bs)^T s}{(c^T s)^2} c c^T$$

Proof

- Let $Mc = M^{-1}s = z$, $E = M(\hat{B} - B)M$, $\bar{E} = M(\bar{B} - B)M$.
- Since $y = \hat{B}s$, premultiplying M on both sides of $(\bar{B} - B)$ derives

$$\bar{E} = \frac{E z z^T + z z^T E}{z^T z} - \frac{z^T E z}{(z^T z)^2} z z^T.$$

The DFP Method

$$\bar{B} - B = \frac{(y - Bs)c^T + c(y - Bs)^T}{c^T s} - \frac{(y - Bs)^T s}{(c^T s)^2} cc^T$$

Proof

- Let $Mc = M^{-1}s = z$, $E = M(\hat{B} - B)M$, $\bar{E} = M(\bar{B} - B)M$.
- Since $y = \hat{B}s$, premultiplying M on both sides of $(\bar{B} - B)$ derives

$$\bar{E} = \frac{Ezz^T + zz^T E}{z^T z} - \frac{z^T E z}{(z^T z)^2} zz^T.$$

- Obiviously, $\bar{E}z = Ez$, i.e., $\|\bar{E}z\|_2 = \|Ez\|_2$.

The DFP Method

$$\bar{B} - B = \frac{(y - Bs)c^T + c(y - Bs)^T}{c^T s} - \frac{(y - Bs)^T s}{(c^T s)^2} cc^T$$

Proof

- Let $Mc = M^{-1}s = z$, $E = M(\hat{B} - B)M$, $\bar{E} = M(\bar{B} - B)M$.
- Since $y = \hat{B}s$, premultiplying M on both sides of $(\bar{B} - B)$ derives

$$\bar{E} = \frac{Ezz^T + zz^T E}{z^T z} - \frac{z^T Ez}{(z^T z)^2} zz^T.$$

- Obiviously, $\bar{E}z = Ez$, i.e., $\|\bar{E}z\|_2 = \|Ez\|_2$.
- Moreover, if $v^T z = 0$, then $\|\bar{E}v\|_2 \leq \|Ev\|_2$.

The DFP Method

$$\bar{B} - B = \frac{(y - Bs)c^T + c(y - Bs)^T}{c^T s} - \frac{(y - Bs)^T s}{(c^T s)^2} cc^T$$

Proof

- Let $Mc = M^{-1}s = z$, $E = M(\hat{B} - B)M$, $\bar{E} = M(\bar{B} - B)M$.
- Since $y = \hat{B}s$, premultiplying M on both sides of $(\bar{B} - B)$ derives

$$\bar{E} = \frac{Ezz^T + zz^T E}{z^T z} - \frac{z^T E z}{(z^T z)^2} zz^T.$$

- Obiviously, $\bar{E}z = Ez$, i.e., $\|\bar{E}z\|_2 = \|Ez\|_2$.
- Moreover, if $v^T z = 0$, then $\|\bar{E}v\|_2 \leq \|Ev\|_2$.
- Hence $\|\bar{E}\|_F \leq \|E\|_F$.

The DFP Method

$$\bar{B} - B = \frac{(y - Bs)c^T + c(y - Bs)^T}{c^T s} - \frac{(y - Bs)^T s}{(c^T s)^2} cc^T$$

Proof

- Let $Mc = M^{-1}s = z$, $E = M(\hat{B} - B)M$, $\bar{E} = M(\bar{B} - B)M$.
- Since $y = \hat{B}s$, premultiplying M on both sides of $(\bar{B} - B)$ derives

$$\bar{E} = \frac{Ezz^T + zz^T E}{z^T z} - \frac{z^T E z}{(z^T z)^2} zz^T.$$

- Obiviously, $\bar{E}z = Ez$, i.e., $\|\bar{E}z\|_2 = \|Ez\|_2$.
- Moreover, if $v^T z = 0$, then $\|\bar{E}v\|_2 \leq \|Ev\|_2$.
- Hence $\|\bar{E}\|_F \leq \|E\|_F$.
- Moreover, $f(\hat{B}) = \|\hat{B} - B\|_{M,F}$ is strongly convex on the convex set $\{\hat{B} | \hat{B}s = y, \hat{B}^T = \hat{B}\}$, hence the solution $\hat{B}$ is the unique solution.

The DFP Method

$$\bar{B} - B = \frac{(y - Bs)c^T + c(y - Bs)^T}{c^T s} - \frac{(y - Bs)^T s}{(c^T s)^2} cc^T$$

Theorem Application

The DFP Method

$$\bar{B} - B = \frac{(y - Bs)c^T + c(y - Bs)^T}{c^T s} - \frac{(y - Bs)^T s}{(c^T s)^2} c c^T$$

Theorem Application

- In particular, let $c = y_k, \; B = B_k, \; s = s_k, \; y = y_k$ in above, we have the DFP formula

$$\begin{aligned} B_{k+1} &= B_k + \frac{(y_k - B_k s_k){y_k}^T + y_k(y_k - B_k s_k)^T}{{y_k}^T s_k} - \frac{(y_k - B_k s_k)^T s_k}{({y_k}^T s_k)^2} y_k {y_k}^T \\ &= B_k + \rho_k\big((y_k - B_k s_k){y_k}^T + y_k(y_k - B_k s_k)^T\big) + \rho_k^2 (y_k - B_k s_k)^T s_k y_k y \\ &= (I - \rho_k y_k s_k^T) B_k (I - \rho_k s_k y_k^T) + \rho_k y_k y_k^T \end{aligned}$$

The DFP Method

$$\bar{B} - B = \frac{(y - Bs)c^T + c(y - Bs)^T}{c^T s} - \frac{(y - Bs)^T s}{(c^T s)^2} c c^T$$

Theorem Application

- In particular, let $c = y_k$, $B = B_k$, $s = s_k$, $y = y_k$ in above, we have the DFP formula

$$\begin{aligned} B_{k+1} &= B_k + \frac{(y_k - B_k s_k){y_k}^T + y_k (y_k - B_k s_k)^T}{{y_k}^T s_k} - \frac{(y_k - B_k s_k)^T s_k}{({y_k}^T s_k)^2} y_k {y_k}^T \\ &= B_k + \rho_k \big((y_k - B_k s_k){y_k}^T + y_k (y_k - B_k s_k)^T \big) + \rho_k^2 (y_k - B_k s_k)^T s_k y_k y \\ &= (I - \rho_k y_k s_k^T) B_k (I - \rho_k s_k y_k^T) + \rho_k y_k y_k^T \end{aligned}$$

- Denote by $H_k = B_k^{-1}$. Utilizing Sherman-Morrison-Woodbury formula to derive

$$H_{k+1} = H_k - \frac{H_k y_k y_k H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k}$$

The DFP Method

Remark

- Rank two update H_{k+1}.

The DFP Method

Remark

- Rank two update H_{k+1}.
- This is the fundamental idea of quasi-Newton updating: 每步迭代中并不重新计算近似Hessian(或其逆)，而是充分利用当前得到的信息s_k, y_k以及已有的近似Hessian B_k(或其逆H_k).

The DFP Method

Remark

- Rank two update H_{k+1}.
- This is the fundamental idea of quasi-Newton updating: 每步迭代中并不重新计算近似Hessian(或其逆)，而是充分利用当前得到的信息s_k, y_k以及已有的近似Hessian B_k(或其逆H_k).
- $M = ?$

The DFP Method

Remark

- Rank two update H_{k+1}.
- This is the fundamental idea of quasi-Newton updating: 每步迭代中并不重新计算近似Hessian(或其逆)，而是充分利用当前得到的信息s_k, y_k以及已有的近似Hessian B_k(或其逆H_k).
- $M =?$
- since $M^2 c = s$, $c = y_k$ and $s = s_k$ in DFP, hence $M^2 y_k = s_k$

The DFP Method

- One of Choices: $M^2 = \bar{G}_k^{-1}$ where $\bar{G}_k$ is the average Hessian defined by

$$\bar{G}_k = \int_0^1 \nabla^2 f(x_k + \tau\alpha_k p_k)d\tau.$$

The DFP Method

- One of Choices: $M^2 = \bar{G}_k^{-1}$ where $\bar{G}_k$ is the average Hessian defined by

$$\bar{G}_k = \int_0^1 \nabla^2 f(x_k + \tau\alpha_k p_k) d\tau.$$

- From Taylor's Theorem, we have

$$y_k = \bar{G}_k \alpha_k p_k = \bar{G}_k s_k$$

The DFP Method

- One of Choices: $M^2 = \bar{G}_k^{-1}$ where $\bar{G}_k$ is the average Hessian defined by

$$\bar{G}_k = \int_0^1 \nabla^2 f(x_k + \tau \alpha_k p_k) d\tau.$$

- From Taylor's Theorem, we have

$$y_k = \bar{G}_k \alpha_k p_k = \bar{G}_k s_k$$

- With this weighting matrix and this norm, the unique solution of (5.8a) is

$$B_{k+1} = (I - \gamma_k y_k s_k^T) B_k (I - \gamma_k y_k s_k^T) + \gamma_k y_k y_k^T, \tag{5.9}$$

where $\gamma_k = \frac{1}{y_k^T s_k}$.

The DFP Method

- One of Choices: $M^2 = \bar{G}_k^{-1}$ where $\bar{G}_k$ is the average Hessian defined by

$$\bar{G}_k = \int_0^1 \nabla^2 f(x_k + \tau \alpha_k p_k) d\tau.$$

- From Taylor's Theorem, we have

$$y_k = \bar{G}_k \alpha_k p_k = \bar{G}_k s_k$$

- With this weighting matrix and this norm, the unique solution of (5.8a) is

$$B_{k+1} = (I - \gamma_k y_k s_k^T) B_k (I - \gamma_k y_k s_k^T) + \gamma_k y_k y_k^T, \tag{5.9}$$

  where $\gamma_k = \frac{1}{y_k^T s_k}$.

- This formula is called the DFP updating formula, since it is the one originally proposed by Davidon in 1959, and subsequently studied, implemented, and popularized by Fletcher and Powell (1962).

The DFP Method

Properties

The DFP Method

Properties

- 对于二次目标函数 $f = \frac{1}{2}x^T Ax - b^T x$ (采用精确线搜索方法)

The DFP Method

Properties

- 对于二次目标函数 $f = \frac{1}{2}x^T Ax - b^T x$ (采用精确线搜索方法)

The DFP Method

Properties

- 对于二次目标函数 $f = \frac{1}{2}x^T Ax - b^T x$ (采用精确线搜索方法)
  - 具有二次终止性质， 即 $H_n = A^{-1}$

The DFP Method

Properties

- 对于二次目标函数 $f = \frac{1}{2}x^T Ax - b^T x$ (采用精确线搜索方法)
  - 具有二次终止性质， 即$H_n = A^{-1}$
  - 具有遗传性质，即$H_i y_j = s_j$，$j < i$

The DFP Method

Properties

- 对于二次目标函数 $f = \frac{1}{2}x^T Ax - b^T x$ (采用精确线搜索方法)
  - 具有二次终止性质， 即$H_n = A^{-1}$
  - 具有遗传性质，即$H_i y_j = s_j$，$j < i$
  - 当$H_0 = I$时，产生共轭方向和共轭梯度

The DFP Method

Properties

- 对于二次目标函数 $f = \frac{1}{2}x^T Ax - b^T x$ (采用精确线搜索方法)
  - 具有二次终止性质， 即$H_n = A^{-1}$
  - 具有遗传性质，即$H_i y_j = s_j$, $j < i$
  - 当$H_0 = I$时，产生共轭方向和共轭梯度
- 对于一般的目标函数时

The DFP Method

Properties

- 对于二次目标函数 $f = \frac{1}{2}x^T Ax - b^T x$ (采用精确线搜索方法)
  - 具有二次终止性质， 即$H_n = A^{-1}$
  - 具有遗传性质，即$H_i y_j = s_j$，$j < i$
  - 当$H_0 = I$时，产生共轭方向和共轭梯度
- 对于一般的目标函数时
  - H_k保持正定性, 因而下降性质成立

The DFP Method

Properties

- 对于二次目标函数 $f = \frac{1}{2}x^TAx - b^Tx$ (采用精确线搜索方法)
  - 具有二次终止性质， 即$H_n = A^{-1}$
  - 具有遗传性质，即$H_iy_j = s_j,\ j < i$
  - 当$H_0 = I$时，产生共轭方向和共轭梯度
- 对于一般的目标函数时
  - H_k保持正定性, 因而下降性质成立
  - 每次迭代需要 $3n^2 + \mathcal{O}(n)$次乘法运算

The DFP Method

Properties

- 对于二次目标函数 $f = \frac{1}{2}x^T Ax - b^T x$ (采用精确线搜索方法)
  - 具有二次终止性质， 即$H_n = A^{-1}$
  - 具有遗传性质，即$H_i y_j = s_j$，$j < i$
  - 当$H_0 = I$时，产生共轭方向和共轭梯度
- 对于一般的目标函数时
  - H_k保持正定性, 因而下降性质成立
  - 每次迭代需要 $3n^2 + \mathcal{O}(n)$次乘法运算
  - 方法具有超线性收敛速度

The DFP Method

Properties

- 对于二次目标函数 $f = \frac{1}{2}x^TAx - b^Tx$ (采用精确线搜索方法)
  - 具有二次终止性质， 即$H_n = A^{-1}$
  - 具有遗传性质，即$H_iy_j = s_j$，$j < i$
  - 当$H_0 = I$时，产生共轭方向和共轭梯度
- 对于一般的目标函数时
  - H_k保持正定性, 因而下降性质成立
  - 每次迭代需要 $3n^2 + \mathcal{O}(n)$次乘法运算
  - 方法具有超线性收敛速度
  - 当采用精确线性搜索时，对于凸函数，方法具有全局收敛性。

THE DFP METHOD

Theorem (二次终止性定理)

如果 f是二次目标函数，A是其正定的Hessian矩阵，那么当采用精确线性搜索时， DFP方法具有遗传性质和方向共轭性质，即对于$i = 0, 1, \cdots, m$, 有

$$
\begin{aligned}
H_{i+1} y_j &= s_j, \quad j = 0, 1, \cdots, i \text{ 遗传性质} \\
s_i^T A s_j &= 0, \quad j = 0, 1, \cdots, i-1 \text{ 方向共轭性}
\end{aligned}
$$

方法在$m + 1 \le n$步迭代后终止。如果$m = n - 1$, 则$H_n = A^{-1}$。

The DFP Method

Proof

The DFP Method

Proof

- 使用归纳法证明遗传性质和方向共轭性质

The DFP Method

Proof

- 使用归纳法证明遗传性质和方向共轭性质
- $i=0$时，显然成立。假定当i时成立，需要证明$i+1$时也成立。

The DFP Method

Proof

- 使用归纳法证明遗传性质和方向共轭性质
- $i=0$时，显然成立。假定当i时成立，需要证明$i+1$时也成立。
- 由于$r_{i+1}\neq 0$，由精确一维搜索和归纳假设可以得到，对于$j\leq i$，有

$$
\begin{aligned}
r_{i+1}^T s_j &= r_{j+1}^T s_j + \sum_{k=j+1}^{i} (r_{k+1}-r_k)^T s_j \\
&= r_{j+1}^T s_j + \sum_{k=j+1}^{i} y_k^T s_j = 0 + \sum_{k=j+1}^{i} (s_k^T A) s_j = 0
\end{aligned}
$$

The DFP Method

Proof

- 使用归纳法证明遗传性质和方向共轭性质
- $i=0$时，显然成立。假定当i时成立，需要证明$i+1$时也成立。
- 由于$r_{i+1}\neq 0$，由精确一维搜索和归纳假设可以得到，对于$j\leq i$，有

$$
\begin{aligned}
r_{i+1}^T s_j &= r_{j+1}^T s_j + \sum_{k=j+1}^{i} (r_{k+1}-r_k)^T s_j \\
&= r_{j+1}^T s_j + \sum_{k=j+1}^{i} y_k^T s_j = 0 + \sum_{k=j+1}^{i} (s_k^T A) s_j = 0
\end{aligned}
$$

- 利用归纳假设 $H_{i+1}y_j = s_j$,, $As_j = A(x_{j+1}-x_j) = y_j$和上式,得到

$$
s_{i+1}^T A s_j = \alpha_{i+1} p_{i+1}^T A s_j = \alpha_{i+1}(-H_{i+1}r_{i+1})^T y_j = -\alpha_{i+1} g_{i+1}^T s_j = 0
$$

这就证明方向共轭性对于$i+1$也是成立的。

The DFP Method

Proof (Continue...)

The DFP Method

Proof (Continue...)

- 对于遗传性质，我们要证明$H_{i+2}y_j = s_j,\ j = 0, \cdots, i+1.$

The DFP Method

Proof (Continue...)

- 对于遗传性质，我们要证明$H_{i+2}y_j = s_j$，$j = 0, \cdots, i+1$.
- 对于DFG算法，我们有$H_{i+2}y_{i+1} = s_{i+1}$.

The DFP Method

Proof (Continue...)

- 对于遗传性质，我们要证明$H_{i+2}y_j = s_j$，$j = 0, \cdots, i+1$.
- 对于DFG算法，我们有$H_{i+2}y_{i+1} = s_{i+1}$.
- 对于$j \leq i$，由于方向共轭性和归纳假设，我们有

$$
\begin{aligned}
s_{i+1}^T y_j &= s_{i+1}^T A s_j = 0, \\
y_{i+1}^T H_{i+1} y_j &= y_{i+1}^T s_j = s_{i+1}^T A s_j = 0
\end{aligned}
$$

The DFP Method

Proof (Continue...)

- 对于遗传性质，我们要证明$H_{i+2}y_j = s_j$，$j = 0, \cdots, i+1$.
- 对于DFG算法，我们有$H_{i+2}y_{i+1} = s_{i+1}$.
- 对于$j \le i$，由于方向共轭性和归纳假设，我们有

$$
\begin{aligned}
s_{i+1}^T y_j &= s_{i+1}^T A s_j = 0, \\
y_{i+1}^T H_{i+1} y_j &= y_{i+1}^T s_j = s_{i+1}^T A s_j = 0
\end{aligned}
$$

- 因此

$$
\begin{aligned}
H_{i+2}y_j &= H_{i+1}y_j + \frac{s_{i+1}s_{i+1}^T y_j}{s_{i+1}^T y_{i+1}} - \frac{H_{i+1}y_{i+1}y_{i+1}^T H_{i+1}y_j}{y_{i+1}^T H_{i+1} y_{i+1}} \\
&= H_{i+1}y_j = s_j
\end{aligned}
$$

这就证明了遗传性质对$i+1$也是成立的。

The DFP Method

Proof (Continue...)

The DFP Method

Proof (Continue...)

- 由于s_i共轭，$i = 0, \cdots, m$，因此该方法是共轭梯度法。根据线性共轭梯度方法是二次终止方法，该方法至多n步终止。

The DFP Method

Proof (Continue...)

- 由于s_i共轭，$i=0,\cdots,m$，因此该方法是共轭梯度法。根据线性共轭梯度方法是二次终止方法，该方法至多n步终止。
- 当$m=n-1$时，由于s_i线性无关，$i=0,\cdots,n-1$，故 $H_n y_j = s_j$, $j=0,\cdots,n-1$, 此即 $H_n A s_j = s_j$, $j=0,\cdots,n-1$.

从而有$H_n = A^{-1}$。

The DFP Method

Theorem (DFP方法的正定性)

当且仅当$s_k^T y_k > 0$时，DFP的校正公式$H_{k+1} = H_k + \frac{s_k s_k^T}{s_k^T y_k} - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k}$保持正定性。

BFGS Mehtod (Broyden, Fletcher, Goldfarb and Shanno)

- The DFP updating formula is quite effective, but it was soon superseded by the BFGS formula, which is presently considered to be the most effective of all quasi-Newton updating formulae.

BFGS Mehtod (Broyden, Fletcher, Goldfarb and Shanno)

- The DFP updating formula is quite effective, but it was soon superseded by the BFGS formula, which is presently considered to be the most effective of all quasi-Newton updating formulae.
- Instead of imposing conditions on the Hessian approximations B_k, we impose similar conditions on the inverses H_k.

BFGS Mehtod (Broyden, Fletcher, Goldfarb and Shanno)

- The DFP updating formula is quite effective, but it was soon superseded by the BFGS formula, which is presently considered to be the most effective of all quasi-Newton updating formulae.
- Instead of imposing conditions on the Hessian approximations B_k, we impose similar conditions on the inverses H_k.
- The updated approximation H_{k+1} must be symmetric and positive definite (SPD), and must satisfy the secant equation (割线方程) (5.6), can be written as

$$H_{k+1} y_k = s_k \tag{5.10}$$

BFGS Mehtod (Broyden, Fletcher, Goldfarb and Shanno)

- The DFP updating formula is quite effective, but it was soon superseded by the BFGS formula, which is presently considered to be the most effective of all quasi-Newton updating formulae.
- Instead of imposing conditions on the Hessian approximations B_k, we impose similar conditions on the inverses H_k.
- The updated approximation H_{k+1} must be symmetric and positive definite (SPD), and must satisfy the secant equation (割线方程) (5.6), can be written as

$$H_{k+1} y_k = s_k \tag{5.10}$$

- Infinite solutions of H_{k+1}.

BFGS Method

- The condition of closeness to H_k is now specified by

$$\min_{H} \quad \|H - H_k\|_{M,F} \tag{5.11}$$

$$s.t. \quad H = H^T, \quad H y_k = s_k. \tag{5.12}$$

- The norm is again the weighted Frobenius norm described above, where the weight matrix M^2 is now any matrix satisfying

$$M^2 s_k = y_k$$

- Assume again that M^2 is given by the average Hessian $\bar{G}_k$. The unique solution H_{k+1} to (5.11) is given by

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k s_k y_k^T) + \rho_k s_k s_k^T, \tag{5.13}$$

with $\rho_k = \frac{1}{y_k^T s_k}$.

$$(DFP): B_{k+1} = (I - \rho_k y_k s_k^T) B_k (I - \rho_k s_k y_k^T) + \rho_k y_k y_k^T$$

BFGS Method

- We can derive a version of the BFGS algorithm that works with the Hessian approximation B_k rather than H_k.

BFGS Method

- We can derive a version of the BFGS algorithm that works with the Hessian approximation B_k rather than H_k.
- The update formula for B_k is obtained by simply applying the Sherman-Morrison-Woodbury formula to (5.13) to obtain

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \tag{5.14}$$

BFGS Method

- We can derive a version of the BFGS algorithm that works with the Hessian approximation B_k rather than H_k.
- The update formula for B_k is obtained by simply applying the Sherman-Morrison-Woodbury formula to (5.13) to obtain

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} \tag{5.14}$$

- It is interesting to note that the DFP and BFGS updating formulae are duals of each other (互为对偶), in the sense that one can be obtained from the other by the interchanges $s \leftrightarrow y$, $B \leftrightarrow H$.

BFGS Method

Algorithm 1: (BFGS Method)

Given initial guess x_0, convergence tolerance $\epsilon > 0$ and inverse Hessian approximation H_0

BFGS Method

Algorithm 1: (BFGS Method)

Given initial guess x_0, convergence tolerance $\epsilon > 0$ and inverse Hessian approximation H_0

Set $k \leftarrow 0$;

while $\|\nabla f_k\| > \epsilon$, **do**

BFGS Method

Algorithm 1: (BFGS Method)

Given initial guess x_0, convergence tolerance $\epsilon > 0$ and inverse Hessian approximation H_0

Set $k \leftarrow 0$;

while $\|\nabla f_k\| > \epsilon$, **do**

Compute search direction $p_k = -H_k \nabla f_k$;

BFGS Method

Algorithm 1: (BFGS Method)

Given initial guess x_0, convergence tolerance $\epsilon > 0$ and inverse Hessian approximation H_0

Set $k \leftarrow 0$;

while $\|\nabla f_k\| > \epsilon$, **do**

Compute search direction $p_k = -H_k \nabla f_k$;

Set $x_{k+1} = x_k + \alpha_k p_k$ where α_k is computed from a line search procedure to satisfy the Wolfe conditions

BFGS Method

Algorithm 1: (BFGS Method)

Given initial guess x_0, convergence tolerance $\epsilon > 0$ and inverse Hessian approximation H_0

Set $k \leftarrow 0$;

while $\|\nabla f_k\| > \epsilon$, **do**

Compute search direction $p_k = -H_k \nabla f_k$;

Set $x_{k+1} = x_k + \alpha_k p_k$ where α_k is computed from a line search procedure to satisfy the Wolfe conditions

Define $s_k = x_{k+1} - x_k$ and $y_k = \nabla f_{k+1} - \nabla f_k$;

BFGS Method

Algorithm 1: (BFGS Method)

Given initial guess x_0, convergence tolerance $\epsilon > 0$ and inverse Hessian approximation H_0

Set $k \leftarrow 0$;

while $\|\nabla f_k\| > \epsilon$, **do**

Compute search direction $p_k = -H_k \nabla f_k$;

Set $x_{k+1} = x_k + \alpha_k p_k$ where α_k is computed from a line search procedure to satisfy the Wolfe conditions

Define $s_k = x_{k+1} - x_k$ and $y_k = \nabla f_{k+1} - \nabla f_k$;

Compute H_{k+1} by means of BFGS

BFGS Method

Algorithm 1: (BFGS Method)

Given initial guess x_0, convergence tolerance $\epsilon > 0$ and inverse Hessian approximation H_0

Set $k \leftarrow 0$;

while $\|\nabla f_k\| > \epsilon$, **do**

Compute search direction $p_k = -H_k \nabla f_k$;

Set $x_{k+1} = x_k + \alpha_k p_k$ where α_k is computed from a line search procedure to satisfy the Wolfe conditions

Define $s_k = x_{k+1} - x_k$ and $y_k = \nabla f_{k+1} - \nabla f_k$;

Compute H_{k+1} by means of BFGS

$k \leftarrow k + 1$;

End(while)

BFGS Method

Computational Complexity and Advantages

BFGS Method

Computational Complexity and Advantages

- Each iteration can be performed at a cost of $\mathcal{O}(n^2)$ arithmetic operations (plus the cost of function and gradient evaluations);

BFGS Method

Computational Complexity and Advantages

- Each iteration can be performed at a cost of $\mathcal{O}(n^2)$ arithmetic operations (plus the cost of function and gradient evaluations);
- there are no $\mathcal{O}(n^3)$ operations such as linear system solves or matrix-matrix operations.

BFGS Method

Computational Complexity and Advantages

- Each iteration can be performed at a cost of $\mathcal{O}(n^2)$ arithmetic operations (plus the cost of function and gradient evaluations);
- there are no $\mathcal{O}(n^3)$ operations such as linear system solves or matrix-matrix operations.
- The algorithm is robust, and its rate of convergence is superlinear, which is fast enough for most practical purposes.

BFGS Method

Computational Complexity and Advantages

- Each iteration can be performed at a cost of $\mathcal{O}(n^2)$ arithmetic operations (plus the cost of function and gradient evaluations);
- there are no $\mathcal{O}(n^3)$ operations such as linear system solves or matrix-matrix operations.
- The algorithm is robust, and its rate of convergence is superlinear, which is fast enough for most practical purposes.
- Though Newton's method converges more rapidly (that is quadratically), its cost per iteration is higher because it requires the solution of a linear system.

BFGS Method

Computational Complexity and Advantages

- Each iteration can be performed at a cost of $\mathcal{O}(n^2)$ arithmetic operations (plus the cost of function and gradient evaluations);
- there are no $\mathcal{O}(n^3)$ operations such as linear system solves or matrix-matrix operations.
- The algorithm is robust, and its rate of convergence is superlinear, which is fast enough for most practical purposes.
- Though Newton's method converges more rapidly (that is quadratically), its cost per iteration is higher because it requires the solution of a linear system.
- A more important advantage for BFGS is, of course, that it does not require calculation of second derivatives.

The BFGS Method

Properties of BFGS

The BFGS Method

Properties of BFGS

- H_{k+1} will be SPD whenever H_k is SPD.

The BFGS Method

Properties of BFGS

- H_{k+1} will be SPD whenever H_k is SPD. Since for any nonzero vector z, we have

$$
\begin{aligned}
z^T H_{k+1} z &= z^T (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) z + z^T \rho_k s_k s_k^T z \\
&= w^T H_k w + \rho_k (z^T s_k)^2 > 0
\end{aligned}
$$

The BFGS Method

Properties of BFGS

- H_{k+1} will be SPD whenever H_k is SPD. Since for any nonzero vector z, we have
$$\begin{aligned} z^T H_{k+1} z &= z^T (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) z + z^T \rho_k s_k s_k^T z \\ &= w^T H_k w + \rho_k (z^T s_k)^2 > 0 \end{aligned}$$
- The updating formula is invariant to changes in the variables. (算法对线性变换保持不变.)

The BFGS Method

Properties of BFGS

- H_{k+1} will be SPD whenever H_k is SPD. Since for any nonzero vector z, we have
$$\begin{aligned} z^T H_{k+1} z &= z^T(I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) z + z^T \rho_k s_k s_k^T z \\ &= w^T H_k w + \rho_k (z^T s_k)^2 > 0 \end{aligned}$$
- The updating formula is invariant to changes in the variables. (算法对线性变换保持不变.)
- The updating formula has very effective self-correcting properties:

The BFGS Method

Properties of BFGS

- H_{k+1} will be SPD whenever H_k is SPD. Since for any nonzero vector z, we have

$$
\begin{aligned}
z^T H_{k+1} z &= z^T (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) z + z^T \rho_k s_k s_k^T z \\
&= w^T H_k w + \rho_k (z^T s_k)^2 > 0
\end{aligned}
$$

- The updating formula is invariant to changes in the variables. (算法对线性变换保持不变.)
- The updating formula has very effective self-correcting properties:
    - If H_k incorrectly estimates the curvature in the objective function, and if this bad estimate slows down the iteration

The BFGS Method

Properties of BFGS

- H_{k+1} will be SPD whenever H_k is SPD. Since for any nonzero vector z, we have

$$\begin{aligned} z^T H_{k+1} z &= z^T (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) z + z^T \rho_k s_k s_k^T z \\ &= w^T H_k w + \rho_k (z^T s_k)^2 > 0 \end{aligned}$$

- The updating formula is invariant to changes in the variables. (算法对线性变换保持不变.)
- The updating formula has very effective self-correcting properties:
  - If H_k incorrectly estimates the curvature in the objective function, and if this bad estimate slows down the iteration
  - then the Hessian approximation will tend to correct itself within a few steps

The BFGS Method

Properties of BFGS

- H_{k+1} will be SPD whenever H_k is SPD. Since for any nonzero vector z, we have

$$\begin{aligned} z^T H_{k+1} z &= z^T(I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) z + z^T \rho_k s_k s_k^T z \\ &= w^T H_k w + \rho_k (z^T s_k)^2 > 0 \end{aligned}$$

- The updating formula is invariant to changes in the variables. (算法对线性变换保持不变.)
- The updating formula has very effective self-correcting properties:
  - If H_k incorrectly estimates the curvature in the objective function, and if this bad estimate slows down the iteration
  - then the Hessian approximation will tend to correct itself within a few steps
  - The self correcting properties of BFGS hold only when an adequate line search is performed.

The BFGS Method

Properties of BFGS

- H_{k+1} will be SPD whenever H_k is SPD. Since for any nonzero vector z, we have
$$
\begin{aligned}
z^T H_{k+1} z &= z^T (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) z + z^T \rho_k s_k s_k^T z \\
&= w^T H_k w + \rho_k (z^T s_k)^2 > 0
\end{aligned}
$$
- The updating formula is invariant to changes in the variables. (算法对线性变换保持不变.)
- The updating formula has very effective self-correcting properties:
    - If H_k incorrectly estimates the curvature in the objective function, and if this bad estimate slows down the iteration
    - then the Hessian approximation will tend to correct itself within a few steps
    - The self correcting properties of BFGS hold only when an adequate line search is performed.
    - In particular, the Wolfe line search conditions ensure that the quadratic model to capture appropriate curvature information.

The BFGS Method - Implementation Details

Initial Approximation H_0

The BFGS Method - Implementation Details

Initial Approximation H_0

- We can use specific information about the problem, for instance by setting it to the inverse of an approximate Hessian calculated by finite differences at x_0

The BFGS Method - Implementation Details

Initial Approximation H_0

- We can use specific information about the problem, for instance by setting it to the inverse of an approximate Hessian calculated by finite differences at x_0
- The initial matrix H_0 often is set to some multiple βI of the identity, but there is no good general strategy for choosing β.

The BFGS Method - Implementation Details

Initial Approximation H_0

- We can use specific information about the problem, for instance by setting it to the inverse of an approximate Hessian calculated by finite differences at x_0
- The initial matrix H_0 often is set to some multiple βI of the identity, but there is no good general strategy for choosing β.
  - If β is "too large" so that the first step $p_0 = -\beta g_0$ is too long, many function evaluations may be required to find a suitable value for the step length α_0.

The BFGS Method - Implementation Details

Initial Approximation H_0

- We can use specific information about the problem, for instance by setting it to the inverse of an approximate Hessian calculated by finite differences at x_0
- The initial matrix H_0 often is set to some multiple βI of the identity, but there is no good general strategy for choosing β.
    - If β is "too large" so that the first step $p_0 = -\beta g_0$ is too long, many function evaluations may be required to find a suitable value for the step length α_0.
    - Some software asks the user to prescribe a value δ for the norm of the first step, and then set $H_0 = \delta \|g_0\|^{-1} I$ to achieve this norm.

The BFGS Method - Implementation

Initial Approximation H_0

The BFGS Method - Implementation

Initial Approximation H_0

- A heuristic (启发式) that is often quite effective is to scale the starting matrix after the first step has been computed but before the first BFGS update is performed.

The BFGS Method - Implementation

Initial Approximation H_0

- A heuristic (启发式) that is often quite effective is to scale the starting matrix after the first step has been computed but before the first BFGS update is performed.
- We change the provisional value $H_0 = I$ (临时的H_0) by setting

$$H_0 \leftarrow \frac{y_k^T s_k}{y_k^T y_k} I$$

before applying the update to obtain H_1.

THE BFGS METHOD - IMPLEMENTATION

Initial Approximation H_0

- A heuristic (启发式) that is often quite effective is to scale the starting matrix after the first step has been computed but before the first BFGS update is performed.
- We change the provisional value $H_0 = I$ (临时的H_0) by setting

$$H_0 \leftarrow \frac{y_k^T s_k}{y_k^T y_k} I$$

  before applying the update to obtain H_1.
- This formula attempts to make the size of H_0 similar to that of $[\nabla^2 f(x_0)]^{-1}$ by approximates an eigenvalue of $[\nabla^2 f(x_0)]^{-1}$. Why?

The BFGS Method - Implementation

Initial Approximation H_0

- A heuristic (启发式) that is often quite effective is to scale the starting matrix after the first step has been computed but before the first BFGS update is performed.
- We change the provisional value $H_0 = I$ (临时的H_0) by setting

$$H_0 \leftarrow \frac{y_k^T s_k}{y_k^T y_k} I$$

  before applying the update to obtain H_1.
- This formula attempts to make the size of H_0 similar to that of $[\nabla^2 f(x_0)]^{-1}$ by approximates an eigenvalue of $[\nabla^2 f(x_0)]^{-1}$. Why?
- In BFGS, $M^2 = \bar{G}_k$ and $M^2 y_k = s_k$. Let $z_k = M^{-1} s_k$, 则$y_k = M^{-1} z_k$,

$$\frac{y_k^T s_k}{y_k^T y_k} = \frac{(M^{-1} z_k)^T M z_k}{z_k M^{-2} z_k} = \frac{z_k^T z_k}{z_k \bar{G}_k z_k}$$

The BFGS Method - Implementation

Line Search

The BFGS Method - Implementation

Line Search

- The line search, which should satisfy either the Wolfe conditions or the strong Wolfe conditions, should always try the step length $\alpha_k = 1$ frst, because this step length will eventually always be accepted (under certain conditions), thereby producing superlinear convergence of the overall algorithm.

The BFGS Method - Implementation

Line Search

- The line search, which should satisfy either the Wolfe conditions or the strong Wolfe conditions, should always try the step length $\alpha_k = 1$ frst, because this step length will eventually always be accepted (under certain conditions), thereby producing superlinear convergence of the overall algorithm.
- Computational observations strongly suggest that it is more economical, in terms of function evaluations, to perform a fairly inaccurate line search. The values $c_1 = 10^{-4}$ and $c_2 = 0.9$ are commonly used.

THE BFGS METHOD - IMPLEMENTATION

Line Search

- The line search, which should satisfy either the Wolfe conditions or the strong Wolfe conditions, should always try the step length $\alpha_k = 1$ frst, because this step length will eventually always be accepted (under certain conditions), thereby producing superlinear convergence of the overall algorithm.
- Computational observations strongly suggest that it is more economical, in terms of function evaluations, to perform a fairly inaccurate line search. The values $c_1 = 10^{-4}$ and $c_2 = 0.9$ are commonly used.
- The performance of the BFGS method can degrade if the line search is not based on the Wolfe conditions.

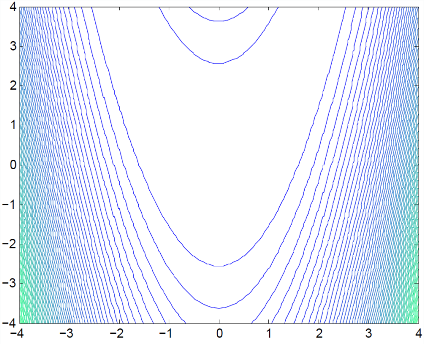
Numerical Experiment Results

Rosenbrock's function
$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$

Numerical Experiment Results

Rosenbrock's function

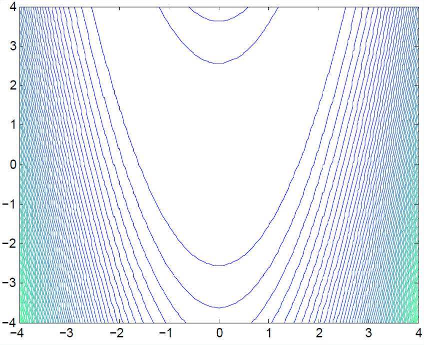
$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$

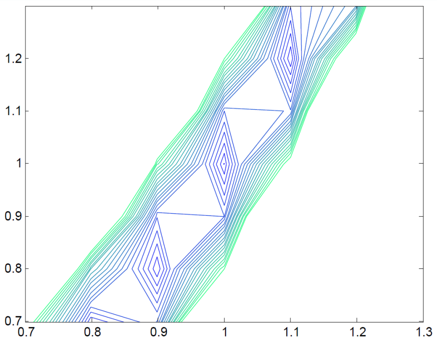


Numerical Experiment Results

Rosenbrock's function
$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$

The optimal solution is $x^* = (1,1)^T$, $f(x^*) = 0$.





Numerical Experiment Results

- Use the steepest descent, BFGS, and an inexact Newton method

Numerical Experiment Results

- Use the steepest descent, BFGS, and an inexact Newton method
- The Wolfe conditions were imposed on the step length in all three methods.

Numerical Experiment Results

- Use the steepest descent, BFGS, and an inexact Newton method
- The Wolfe conditions were imposed on the step length in all three methods.
- The initial point $x_0 = (-1.2, 1)$.

Numerical Experiment Results

- Use the steepest descent, BFGS, and an inexact Newton method
- The Wolfe conditions were imposed on the step length in all three methods.
- The initial point $x_0 = (-1.2, 1)$.
- The steepest descent method required 5264 iterations, whereas BFGS and Newton took only 34 and 21 iterations, respectively to reduce the gradient norm to 10^{-5}.

steepest descent	BFGS	Newton
1.827e-04	1.70e-03	3.48e-02
1.826e-04	1.17e-03	1.44e-02
1.824e-04	1.34e-04	1.82e-04
1.823e-04	1.01e-06	1.17e-08

The value of $\|x_k - x^*\|$ in last few iterations of the steepest descent, BFGS, and an inexact Newton method on Rosenbrock's function

Symmetric-Rank-1 (SR1)

Symmetric-Rank-1 (SR1)

- In the BFGS and DFP updating formulae, the updated matrix B_{k+1} (or H_{k+1}) differs from its predecessor B_k (or H_k) by a rank-2 matrix.

Symmetric-Rank-1 (SR1)

- In the BFGS and DFP updating formulae, the updated matrix B_{k+1} (or H_{k+1}) differs from its predecessor B_k (or H_k) by a rank-2 matrix.
- In fact, there is a simpler rank-1 update that maintains symmetry of the matrix and allows it to satisfy the secant equation.

Symmetric-Rank-1 (SR1)

- In the BFGS and DFP updating formulae, the updated matrix B_{k+1} (or H_{k+1}) differs from its predecessor B_k (or H_k) by a rank-2 matrix.
- In fact, there is a simpler rank-1 update that maintains symmetry of the matrix and allows it to satisfy the secant equation.
- Unlike the rank-two update formulae, this *symmetric-rank-1* , or SR1, update does not guarantee that the updated matrix maintains positive definiteness.

Symmetric-Rank-1 (SR1)

- In the BFGS and DFP updating formulae, the updated matrix B_{k+1} (or H_{k+1}) differs from its predecessor B_k (or H_k) by a rank-2 matrix.
- In fact, there is a simpler rank-1 update that maintains symmetry of the matrix and allows it to satisfy the secant equation.
- Unlike the rank-two update formulae, this *symmetric-rank-1* , or SR1, update does not guarantee that the updated matrix maintains positive definiteness. Good numerical results have been obtained with algorithms based on SR1.

SR1 Derivation

SR1 Derivation

- Assume the symmetric rank-1 update has the general form $B_{k+1} = B_k + \sigma v v^T$, where $\sigma = \pm 1$.

SR1 Derivation

- Assume the symmetric rank-1 update has the general form $B_{k+1} = B_k + \sigma v v^T$, where $\sigma = \pm 1$.
- B_{k+1} satisfies the secant equation $y_k = B_{k+1} s_k$, i.e.,

$$y_k = B_k s_k + [\sigma v^T s_k] v.$$

SR1 Derivation

- Assume the symmetric rank-1 update has the general form $B_{k+1} = B_k + \sigma v v^T$, where $\sigma = \pm 1$.
- B_{k+1} satisfies the secant equation $y_k = B_{k+1} s_k$, i.e.,

$$y_k = B_k s_k + [\sigma v^T s_k] v.$$

- Thus, $v = \delta(y_k - B_k s_k)$ for some scalar δ.
- Substituting this form of v into the secant equation, we obtain

$$(y_k - B_k s_k) = \sigma \delta^2 [s_k^T (y_k - B_k s_k)] (y_k - B_k s_k)$$

SR1 Derivation

- Assume the symmetric rank-1 update has the general form $B_{k+1} = B_k + \sigma v v^T$, where $\sigma = \pm 1$.
- B_{k+1} satisfies the secant equation $y_k = B_{k+1} s_k$, i.e.,

$$y_k = B_k s_k + [\sigma v^T s_k] v.$$

- Thus, $v = \delta(y_k - B_k s_k)$ for some scalar δ.
- Substituting this form of v into the secant equation, we obtain

$$(y_k - B_k s_k) = \sigma \delta^2 [s_k^T (y_k - B_k s_k)](y_k - B_k s_k)$$

- Therefore, we choose the parameters δ and σ to be

$$\sigma = \mathrm{sign}[s_k^T (y_k - B_k s_k)], \quad \delta = \pm [|s_k^T (y_k - B_k s_k)|]^{-\frac{1}{2}}.$$

SR1 Derivation

SR1 Derivation

- The only symmetric rank-1 updating formula that satisfies the secant equation is given by

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k} \tag{5.15}$$

SR1 Derivation

- The only symmetric rank-1 updating formula that satisfies the secant equation is given by

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k} \tag{5.15}$$

- By applying the Sherman-Morrison formula, we obtain the corresponding update formula for the inverse Hessian approximation H_k

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k} \tag{5.16}$$

SR1 Derivation

- The only symmetric rank-1 updating formula that satisfies the secant equation is given by

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k} \tag{5.15}$$

- By applying the Sherman-Morrison formula, we obtain the corresponding update formula for the inverse Hessian approximation H_k

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k} \tag{5.16}$$

- SR1 method is self-dual, i.e. the inverse formula H_k can be obtained simply by replacing B, s and y by H, y and s, respectively.

SR1 Properties

Properties

SR1 Properties

Properties

- If B_k is positive definite, B_{k+1} may not have this property; the same is, of course, true of H_k.

SR1 Properties

Properties

- If B_k is positive definite, B_{k+1} may not have this property; the same is, of course, true of H_k.
- This observation was considered a major drawback in the early days of nonlinear optimization when only line search iterations were used.

SR1 PROPERTIES

Properties

- If B_k is positive definite, B_{k+1} may not have this property; the same is, of course, true of H_k.
- This observation was considered a major drawback in the early days of nonlinear optimization when only line search iterations were used.
- However, with the advent of trust-region methods, the SR1 updating formula has proved to be quite useful.

SR1 Properties

Properties

- If B_k is positive definite, B_{k+1} may not have this property; the same is, of course, true of H_k.
- This observation was considered a major drawback in the early days of nonlinear optimization when only line search iterations were used.
- However, with the advent of trust-region methods, the SR1 updating formula has proved to be quite useful.
- its ability to generate indefinite Hessian approximations can actually be regarded as one of its chief advantages.

SR1 Properties

By reasoning in terms of B_k (similar arguments can be applied to H_k), we see that there are three cases:

SR1 Properties

By reasoning in terms of B_k (similar arguments can be applied to H_k), we see that there are three cases:

- If $(y_k - B_k s_k)^T s_k \neq 0$, then the arguments above show that there is a unique rank-1 updating formula satisfying the secant equation;

SR1 Properties

By reasoning in terms of B_k (similar arguments can be applied to H_k), we see that there are three cases:

- If $(y_k - B_k s_k)^T s_k \neq 0$, then the arguments above show that there is a unique rank-1 updating formula satisfying the secant equation;
- If $y_k = B_k s_k$, then the only updating formula satisfying the secant equation is simply $B_{k+1} = B_k$;

SR1 Properties

By reasoning in terms of B_k (similar arguments can be applied to H_k), we see that there are three cases:

- If $(y_k - B_k s_k)^T s_k \neq 0$, then the arguments above show that there is a unique rank-1 updating formula satisfying the secant equation;
- If $y_k = B_k s_k$, then the only updating formula satisfying the secant equation is simply $B_{k+1} = B_k$;
- If $y_k \neq B_k s_k$ and $(y_k - B_k s_k)^T s_k = 0$, then there is no symmetric rank-one updating formula satisfying the secant equation.

SR1 Properties

SR1 Properties

- The last case clouds an otherwise simple and elegant derivation, and suggests that numerical instabilities and even breakdown of the method can occur.

SR1 Properties

- The last case clouds an otherwise simple and elegant derivation, and suggests that numerical instabilities and even breakdown of the method can occur.
- It suggests that rank-one updating does not provide enough freedom to develop a matrix with all the desired characteristics, and that a rank-two correction is required.

SR1 Properties

- The last case clouds an otherwise simple and elegant derivation, and suggests that numerical instabilities and even breakdown of the method can occur.
- It suggests that rank-one updating does not provide enough freedom to develop a matrix with all the desired characteristics, and that a rank-two correction is required.
- This reasoning leads us back to the BFGS method, in which positive definiteness (and thus nonsingularity) of all Hessian approximations is guaranteed.

SR1 Properties

A strategy to prevent the SR1 method from breaking down

SR1 PROPERTIES

A strategy to prevent the SR1 method from breaking down

- It has been observed in practice that SR1 performs well simply by skipping the update if the denominator is small.

SR1 Properties

A strategy to prevent the SR1 method from breaking down

- It has been observed in practice that SR1 performs well simply by skipping the update if the denominator is small.
- More specifically, the SR1 update is applied only if

$$|s_k^T(y_k - B_k s_k)| \geq r\|s_k\|\|y_k - B_k s_k\| \tag{5.17}$$

where $r \in (0,1)$ is a small number, for example, $r = 10^{-8}$.

SR1 Properties

A strategy to prevent the SR1 method from breaking down

- It has been observed in practice that SR1 performs well simply by skipping the update if the denominator is small.
- More specifically, the SR1 update is applied only if

$$|s_k^T(y_k - B_k s_k)| \geq r\|s_k\|\|y_k - B_k s_k\| \tag{5.17}$$

  where $r \in (0, 1)$ is a small number, for example, $r = 10^{-8}$.
- If (5.17) does not hold, we set $B_{k+1} = B_k$.

SR1 Properties

A strategy to prevent the SR1 method from breaking down

- It has been observed in practice that SR1 performs well simply by skipping the update if the denominator is small.
- More specifically, the SR1 update is applied only if

$$|s_k^T(y_k - B_k s_k)| \geq r\|s_k\|\|y_k - B_k s_k\| \tag{5.17}$$

  where $r \in (0, 1)$ is a small number, for example, $r = 10^{-8}$.
- If (5.17) does not hold, we set $B_{k+1} = B_k$.
- Most implementations of the SR1 method use a skipping rule of this kind.

SR1 Properties

Why do we advocate skipping of updates for the SR1 method, when in the previous section we discouraged this strategy in the case of BFGS?

SR1 Properties

Why do we advocate skipping of updates for the SR1 method, when in the previous section we discouraged this strategy in the case of BFGS?

- $s_k^T(y_k - B_k s_k) \approx 0$ occurs infrequently, since it requires certain vectors to be aligned in a specific way.

SR1 Properties

Why do we advocate skipping of updates for the SR1 method, when in the previous section we discouraged this strategy in the case of BFGS?

- $s_k^T(y_k - B_k s_k) \approx 0$ occurs infrequently, since it requires certain vectors to be aligned in a specific way.
- When it does occur, skipping the update appears to have no negative effects on the iteration, since the skipping condition implies that $s_k^T \bar{G} s_k \approx s_k^T B_k s_k$, where $\bar{G}$ is the average Hessian over the last step-meaning that the curvature of B_k along s_k is already correct.

SR1 Properties

Why do we advocate skipping of updates for the SR1 method, when in the previous section we discouraged this strategy in the case of BFGS?

- $s_k^T(y_k - B_k s_k) \approx 0$ occurs infrequently, since it requires certain vectors to be aligned in a specific way.
- When it does occur, skipping the update appears to have no negative effects on the iteration, since the skipping condition implies that $s_k^T \bar{G} s_k \approx s_k^T B_k s_k$, where $\bar{G}$ is the average Hessian over the last step-meaning that the curvature of B_k along s_k is already correct.

- $s_k^T y_k \geq 0$ required for BFGS updating may easily fail if the line search does not impose the Wolfe conditions (e.g., if the step is not long enough).

SR1 Properties

Why do we advocate skipping of updates for the SR1 method, when in the previous section we discouraged this strategy in the case of BFGS?

- $s_k^T(y_k - B_k s_k) \approx 0$ occurs infrequently, since it requires certain vectors to be aligned in a specific way.
- When it does occur, skipping the update appears to have no negative effects on the iteration, since the skipping condition implies that $s_k^T \bar{G} s_k \approx s_k^T B_k s_k$, where $\bar{G}$ is the average Hessian over the last step-meaning that the curvature of B_k along s_k is already correct.

- $s_k^T y_k \geq 0$ required for BFGS updating may easily fail if the line search does not impose the Wolfe conditions (e.g., if the step is not long enough).
- Therefore skipping the BFGS update can occur often and can degrade the quality of the Hessian approximation.

SR1 Properties

定理：二次终止性

Suppose that $f : R^n \to R$ is a strongly convex quadratic function $f(x) = b^T x + \frac{1}{2}x^T Ax$, where A is symmetric positive definite.
Then for any starting point x_0 and any symmetric starting matrix H_0, the iterates $\{x_k\}$ generated by the SR1 method converge to the minimizer in at most n steps, provided that $(s_k - H_k y_k)^T y_k \neq 0$ for all k. Moreover, if n steps are performed, and if the search directions p_k are linearly independent, then $H_n = A^{-1}$.

THE BROYDEN CLASS

The Broyden Class

- A family of updates specified by the following general formula

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} + \phi_k (s_k^T B_k s_k) v_k v_k^T \tag{5.18}$$

where ϕ_k is a scalar parameter and

$$v_k = \left(\frac{y_k}{y_k^T s_k} - \frac{B_k s_k}{s_k^T B_k s_k} \right).$$

The Broyden Class

The Broyden Class

- The BFGS and DFP methods are members of the Broyden class-we recover BFGS by setting $\phi_k = 0$, DFP by setting $\phi_k = 1$, and SR1 by setting $\phi_k = \frac{s_k^T y_k}{(s_k - H_k y_k)^T y_k}$ in (5.18).

The Broyden Class

- The BFGS and DFP methods are members of the Broyden class-we recover BFGS by setting $\phi_k = 0$, DFP by setting $\phi_k = 1$, and SR1 by setting $\phi_k = \frac{s_k^T y_k}{(s_k - H_k y_k)^T y_k}$ in (5.18).

The Broyden Class

- The BFGS and DFP methods are members of the Broyden class-we recover BFGS by setting $\phi_k = 0$, DFP by setting $\phi_k = 1$, and SR1 by setting $\phi_k = \frac{s_k^T y_k}{(s_k - H_k y_k)^T y_k}$ in (5.18).
-
- We can therefore rewrite (5.18) as a "linear combination" of these two methods, that is,

$$B_{k+1} = (1 - \phi_k) B_{k+1}^{BFGS} + \phi_k B_{k+1}^{DFP} \tag{5.19}$$

The Broyden Class

- The BFGS and DFP methods are members of the Broyden class-we recover BFGS by setting $\phi_k = 0$, DFP by setting $\phi_k = 1$, and SR1 by setting $\phi_k = \frac{s_k^T y_k}{(s_k - H_k y_k)^T y_k}$ in (5.18).
-
- We can therefore rewrite (5.18) as a "linear combination" of these two methods, that is,

$$B_{k+1} = (1-\phi_k) B_{k+1}^{BFGS} + \phi_k B_{k+1}^{DFP} \tag{5.19}$$

- This relationship indicates that all members of the Broyden class satisfy the secant equation

The Broyden Class

- The BFGS and DFP methods are members of the Broyden class-we recover BFGS by setting $\phi_k = 0$, DFP by setting $\phi_k = 1$, and SR1 by setting $\phi_k = \frac{s_k^T y_k}{(s_k - H_k y_k)^T y_k}$ in (5.18).
-
- We can therefore rewrite (5.18) as a "linear combination" of these two methods, that is,

$$B_{k+1} = (1 - \phi_k) B_{k+1}^{BFGS} + \phi_k B_{k+1}^{DFP} \tag{5.19}$$

- This relationship indicates that all members of the Broyden class satisfy the secant equation
- members with $0 \leq \phi_k \leq 1$(restricted Broyden class) preserve positive definiteness of the Hessian approximations when $s_k^T y_k > 0$.
(由于DFP方法保持正定性，当$\phi \geq 0$由联锁特征值定理可知，Broyden校正后的特征值不小于H_{k+1}^{DFP}的最小特征值, 可以得到保持正定性.)

Properties of the Broyden Class

Properties of the Broyden Class

- The last term in (5.18) is a rank-one correction.

Properties of the Broyden Class

- The last term in (5.18) is a rank-one correction.
- As we decrease ϕ_k , this matrix eventually becomes singular and then indefinite.

Properties of the Broyden Class

- The last term in (5.18) is a rank-one correction.
- As we decrease ϕ_k , this matrix eventually becomes singular and then indefinite.
- A little computation shows that B_{k+1} is singular when ϕ_k has the value

$$\phi_k^c = \frac{1}{1-\mu_k}, \text{ with } \mu_k = \frac{(y_k^T B_k^{-1} y_k)(s_k^T B_k s_k)}{(y_k^T s_k)^2}. \tag{5.20}$$

Properties of the Broyden Class

- The last term in (5.18) is a rank-one correction.
- As we decrease ϕ_k , this matrix eventually becomes singular and then indefinite.
- A little computation shows that B_{k+1} is singular when ϕ_k has the value

$$\phi_k^c = \frac{1}{1-\mu_k}, \text{ with } \mu_k = \frac{(y_k^T B_k^{-1} y_k)(s_k^T B_k s_k)}{(y_k^T s_k)^2}. \tag{5.20}$$

- By applying the Cauchy-Schwarz inequality to (5.20) we see that $\mu_k \geq 1$ and therefore $\phi_k^c \leq 0$.

Properties of the Broyden Class

- The last term in (5.18) is a rank-one correction.
- As we decrease ϕ_k , this matrix eventually becomes singular and then indefinite.
- A little computation shows that B_{k+1} is singular when ϕ_k has the value

$$\phi_k^c = \frac{1}{1-\mu_k}, \text{ with } \mu_k = \frac{(y_k^T B_k^{-1} y_k)(s_k^T B_k s_k)}{(y_k^T s_k)^2}. \tag{5.20}$$

- By applying the Cauchy-Schwarz inequality to (5.20) we see that $\mu_k \geq 1$ and therefore $\phi_k^c \leq 0$.
- Hence, if the initial Hessian approximation B_0 is symmetric and positive definite, and if $s_k^T y_k > 0$ and $\phi_k > \phi_k^c$ for each k, then all the matrices B_k generated by Broyden's formula (5.18) remain symmetric and positive definite.

Properties of the Broyden Class

- When the line search is exact, all methods in the Broyden class with $\phi_k \geq \phi_k^c$ generate the same sequence of iterates (由于p_k与ϕ无关).

Properties of the Broyden Class

- When the line search is exact, all methods in the Broyden class with $\phi_k \geq \phi_k^c$ generate the same sequence of iterates (由于p_k与ϕ无关).
- This result applies to general nonlinear functions and is based on the observation that when all the line searches are exact, the directions generated by Broyden-class methods differ only in their lengths.

Properties of the Broyden Class

- When the line search is exact, all methods in the Broyden class with $\phi_k \geq \phi_k^c$ generate the same sequence of iterates (由于p_k与ϕ无关).
- This result applies to general nonlinear functions and is based on the observation that when all the line searches are exact, the directions generated by Broyden-class methods differ only in their lengths.
- The line searches identify the same minima along the chosen search direction, though the values of the line search parameter may differ because of the different scaling.

Properties of the Broyden Class

The Broyden class has several remarkable properties when applied with exact line searches to quadratic functions.

Theorem: 二次终止性、遗传性和共轭性

- Suppose that a method in the Broyden class is applied to a strongly convex quadratic function $f : \mathcal{R}^n \to \mathcal{R}$, where x_0 is the starting point and B_0 is any symmetric and positive definite matrix.
- Assume that α_k is the exact step length and the chosen value of ϕ_k did not produce a singular update matrix.
- Then the following statements are true:

Properties of the Broyden Class

Theorem(continue..)

Properties of the Broyden Class

Theorem(continue..)

1. The iterates converge to the solution in at most n iterations.

Properties of the Broyden Class

Theorem(continue..)

1. The iterates converge to the solution in at most n iterations.
2. The secant equation is satisfied for all previous search directions, that is,

$$B_k s_j = y_j, j = k-1, \cdots, 1.$$

Properties of the Broyden Class

Theorem(continue..)

1. The iterates converge to the solution in at most n iterations.
2. The secant equation is satisfied for all previous search directions, that is,

$$B_k s_j = y_j, j = k-1, \cdots, 1.$$

3. If the starting matrix is $B_0 = I$, then the iterates are identical to those generated by the conjugate gradient method. In particular, the search directions are conjugate, that is,

$$s_i^T A s_j = 0 \text{ for } i \neq j$$

where A is the Hessian of the quadratic function.

Properties of the Broyden Class

Theorem(continue..)

1. The iterates converge to the solution in at most n iterations.
2. The secant equation is satisfied for all previous search directions, that is,

$$B_k s_j = y_j, j = k-1, \cdots, 1.$$

3. If the starting matrix is $B_0 = I$, then the iterates are identical to those generated by the conjugate gradient method. In particular, the search directions are conjugate, that is,

$$s_i^T A s_j = 0 \text{ for } i \neq j$$

where A is the Hessian of the quadratic function.

4. If the starting matrix B_0 is not the identity matrix, then the Broyden-class method is identical to the preconditioned conjugate gradient method that uses B_0 as preconditioner.

Properties of the Broyden Class

Theorem(continue..)

1. The iterates converge to the solution in at most n iterations.
2. The secant equation is satisfied for all previous search directions, that is,

$$B_k s_j = y_j, j = k-1, \cdots, 1.$$

3. If the starting matrix is $B_0 = I$, then the iterates are identical to those generated by the conjugate gradient method. In particular, the search directions are conjugate, that is,

$$s_i^T A s_j = 0 \text{ for } i \neq j$$

where A is the Hessian of the quadratic function.

4. If the starting matrix B_0 is not the identity matrix, then the Broyden-class method is identical to the preconditioned conjugate gradient method that uses B_0 as preconditioner.
5. If n iterations are performed, we have $B_{n+1} = A$.

Properties of the Broyden Class

Remark

- The results in the above theorem would appear to be mainly of theoretical interest,

Properties of the Broyden Class

Remark

- The results in the above theorem would appear to be mainly of theoretical interest,
- since the inexact line searches used in practical implementations of Broyden-class methods (and all other quasi-Newton methods) cause their performance to differ markedly.

Properties of the Broyden Class

Remark

- The results in the above theorem would appear to be mainly of theoretical interest,
- since the inexact line searches used in practical implementations of Broyden-class methods (and all other quasi-Newton methods) cause their performance to differ markedly.
- Nevertheless, this type of analysis guided most of the development of quasi-Newton methods.

CONVERGENCE ANALYSIS

- Although the BFGS and SR1 methods are known to be remarkably robust in practice, we will not be able to establish truly global convergence results for general nonlinear objective functions.
- That is, we cannot prove that the iterates of these quasi-Newton methods approach a stationary point of the problem from any starting point and any (suitable) initial Hessian approximation.
- In fact, it is not yet known if the algorithms enjoy such properties.
- In our analysis we will either assume that the objective function is convex or that the iterates satisfy certain properties.
- On the other hand, there are well known local, superlinear convergence results that are true under reasonable assumptions.

Global Convergence of BFGS

Theorem

- Let B_0 be any symmetric positive definite initial matrix, and let x_0 be a starting point for which

Global Convergence of BFGS

Theorem

- Let B_0 be any symmetric positive definite initial matrix, and let x_0 be a starting point for which
  1. the objective function f is twice continuously differentiable.
  2. level set $\mathcal{L} = \{x \in \mathcal{R}^n | f(x) \leq f(x_0)\}$ is convex
  3. there exist positive constants m and M such that

$$m\|z\|^2 \leq z^T \nabla^2 f(x) z \leq M\|z\|^2$$

- Then the sequence $\{x_k\}$ generated by Algorithm 1 converges to the minimizer x^* of f.

Global Convergence of BFGS

Remark

Global Convergence of BFGS

Remark

- The above theorem has been generalized to the entire restricted Broyden class, except for the DFP method;

Global Convergence of BFGS

Remark

- The above theorem has been generalized to the entire restricted Broyden class, except for the DFP method;
- An extension of the analysis shows that the rate of convergence of the iterates is linear.

Global Convergence of BFGS

Remark

- The above theorem has been generalized to the entire restricted Broyden class, except for the DFP method;
- An extension of the analysis shows that the rate of convergence of the iterates is linear.
- In particular, we can show that the sequence $\|x_k - x^*\|$ converges to zero rapidly enough that

$$\sum_{k=1}^{\infty} \|x_k - x^*\| < \infty. \tag{5.21}$$

Superlinear Convergence of BFGS

Theorem

- Suppose that f is twice continuously differentiable
- Hessian matrix $\nabla^2 f$ is Lipschitz continuous at x^* that is,

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

  for all x near x^*, where L is a positive constant.
- Suppose (5.21) holds.

Superlinear Convergence of BFGS

Theorem

- Suppose that f is twice continuously differentiable
- Hessian matrix $\nabla^2 f$ is Lipschitz continuous at x^* that is,

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

  for all x near x^*, where L is a positive constant.
- Suppose (5.21) holds.
- Then x_k converges to x^* at a superlinear rate.

Convergence of SR1 Method

Theorem

- Suppose that the iterates x_k are generated by Algorithm 2 . Suppose also that the following conditions hold:
  1. The sequence of iterates does not terminate, but remains in a closed, bounded, convex set $\mathcal{D}$, on which the function f is twice continuously differentiable, and in which f has a unique stationary point x^*;
  2. the Hessian $\nabla^2 f(x^*)$ is positive definite, and $\nabla^2 f(x)$ is Lipschitz continuous in a neighborhood of x^*;
  3. the sequence of matrices $\{B_k\}$ is bounded in norm;
  4. condition (5.17) holds at every iteration, where r is some constant in $(0, 1)$.
- Then

$$\lim_{k\to\infty} x_k = x^*, \text{ and } \lim_{k\to\infty} \frac{\|x_{k+n+1} - x^*\|}{\|x_k - x^*\|} = 0$$

Thanks For Your Attention