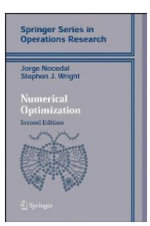
M05M11084 最优化理论、算法与应用

4 非精确一维搜索方法



Armijo, Wolfe, Goldstein Rules

参考:

1. 应用最优化方法及MATLAB实现—刘兴高 胡云卿，第四章
2. Numerical Optimization，Jorge Nocedal, Stephen J.Wright, Chapter 3

精确一维搜索方法的问题

✓ 多数优化方法的算法结构 ⇒ 迭代下降算法

$$x^{k+1} = x^k + \alpha d^k$$

- 寻找下降方向
- 确定步长

✓ 精确一维搜索方法获取最佳步长需要多次迭代

- 当初始步长与最佳步长相距较远时，精确一维搜索的计算效率很低
- 获取步长的计算量 ≫ 获取下降方向的计算量

非精确一维搜索方法的优势

① 选择搜索方向的重要性 ≫ 最佳步长的精确性

只要搜索方向选择恰当，步长即使不精确，对极小点的求取影响不大

②使用非精确一维搜索方法来获取步长不影响大多数优化方法的收敛性

相比精确一维搜索法，非精确一维搜索方法更为实用：

①进行步长搜索前，不需要确定包含最佳步长 的单谷区间

②计算量较小

国际上流行的优化软件中，几乎都使用非精确一维搜索方法来获取步长

“什么样” 的非精确步长合适？如何获得？

合适的非精确步长

例 用梯度下降法，迭代是否下降？迭代是否收敛至驻点？步长合适吗？

$$f(x)=\frac{1}{2}x^2,\qquad x_0=1,\qquad \alpha_k=2-\frac{1}{2^{k+2}|x_k|}\qquad\qquad x^*=0\quad f^*=0$$

$$\nabla f(x)=x,\qquad d_k=-\nabla f(x_k)=-x_k$$

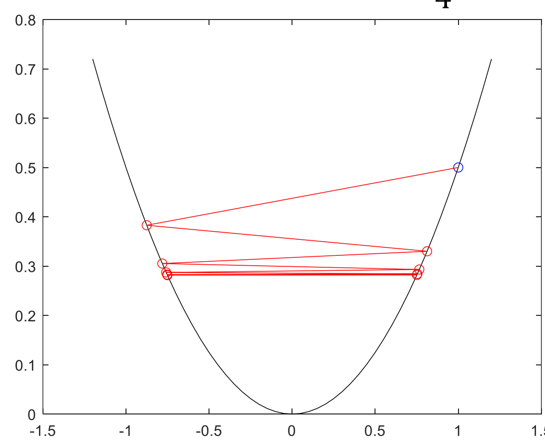
$$\begin{aligned}x_k&=x_{k-1}+\alpha_{k-1}d_{k-1}\\&=x_{k-1}-\left(2-\frac{1}{2^{k+1}|x_{k-1}|}\right)x_{k-1}\\&=(-1)^k\left(\frac{3}{4}+\frac{1}{2^{k+1}}\right)\to\pm\frac{3}{4}\qquad\neq x^*\end{aligned}$$

$$f(x_k)=\frac{1}{2}\left((-1)^k\left(\frac{3}{4}+\frac{1}{2^{k+1}}\right)\right)^2\to\frac{9}{32}\quad\neq f^*$$

$x_0 = 1 \qquad x_k \to \pm\frac{3}{4}$

f =	0.5000	0.3828	0.3301	0.3052
	0.2931	0.2871	0.2842	0.2827
	0.2820	0.2816	0.2814	0.2813
	0.2813	0.2813	0.2813	

α =	1.8750	1.9286	1.9615	1.9800
	1.9898	1.9948	1.9974	1.9987
	1.9993	1.9997	1.9998	1.9999
	2.0000	2.0000		



$f(x_k) \to \frac{9}{32}$

函数值$f(x_k)$

每一步，函数值都在下降，

但是，数列的极限 $f(x_k) \to \frac{9}{32}$

没有收敛到驻点

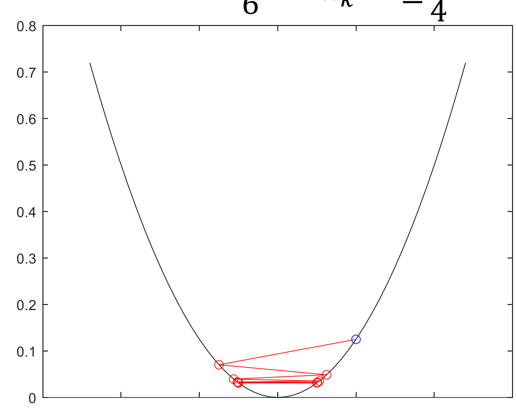
$x_0 = \frac{1}{6} \qquad x_k \to \pm\frac{1}{4}$

函数值不断地下降，但是，**下降量的总和没有达到足够大，步长不合适**

所以，需要一些条件，来保证函数值“真的” 能够充分下降

因而，出现了Armijo, Armijo-Goldstein, Wolfe-Powell （强、弱） 条件 ，保障步长合适

称为： 可接受步长



$f(x_k) \to \frac{1}{32}$

$\nrightarrow 0$

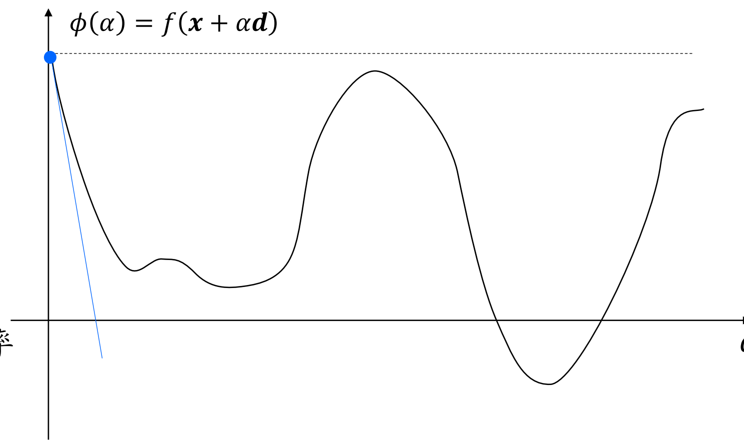
$\phi(\alpha)=f(\boldsymbol{x}+\alpha\boldsymbol{d})$随$\alpha$变化的情况

由于$\boldsymbol{d}$是点$\boldsymbol{x}$处的下降方向，随着步长α从0开始增加，$\phi(\alpha)=f(\boldsymbol{x}+\alpha\boldsymbol{d})$的曲线首先出现下降趋势

此后，根据f的定义不同，$f(\boldsymbol{x}+\alpha\boldsymbol{d})$的曲线可能出现多个波峰和波谷

$\phi'(0)=\nabla f(\boldsymbol{x})^T\boldsymbol{d}<0$

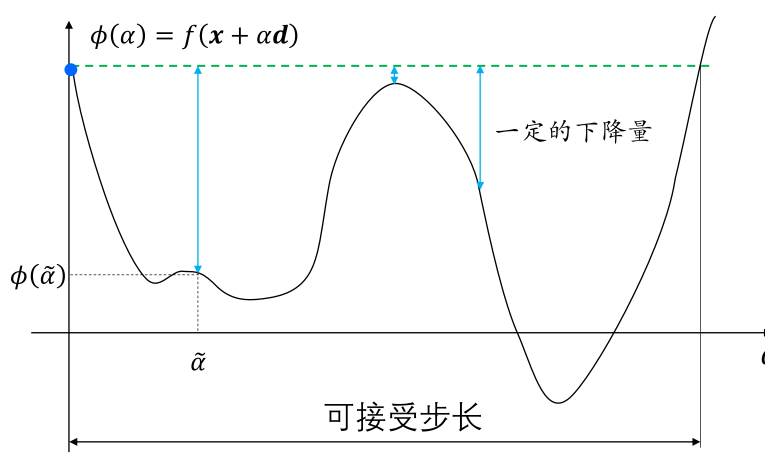
曲线 $\phi(\alpha)$在点$\boldsymbol{x}$处的切线的斜率



一定的下降量与可接受步长

目标函数f从点$\boldsymbol{x}$出发、沿着下降方向$\boldsymbol{d}$的一个步长$\tilde{\alpha}$，使得$f(\boldsymbol{x}+\tilde{\alpha}\boldsymbol{d})$比$f(\boldsymbol{x})$有一定量的减小，即，有一定的下降量 随$\tilde{\alpha}$不同而不同

步长$\tilde{\alpha}$被称为**可接受步长**



定义一条直线$L_B(\alpha)$作为衡量下降量的“标尺”

f从点$\boldsymbol{x}$出发、沿着下降方向$\boldsymbol{d}$的一个步长$\tilde{\alpha}$，使$f(\boldsymbol{x}+\tilde{\alpha}\boldsymbol{d})$比$L_B(\alpha)$有一定量的减小，即满足 $\phi(\tilde{\alpha}) \le L_B(\alpha)$

步长$\tilde{\alpha}$被称为**可接受步长** $\alpha_{\text{acceptable}}$

$$\alpha_k = \alpha_{\text{acceptable}} \quad \Leftrightarrow \quad \phi(\alpha) \le L_B(\alpha)$$

- 在直线$L_B(\alpha)$上方
  所以，**不可接受**

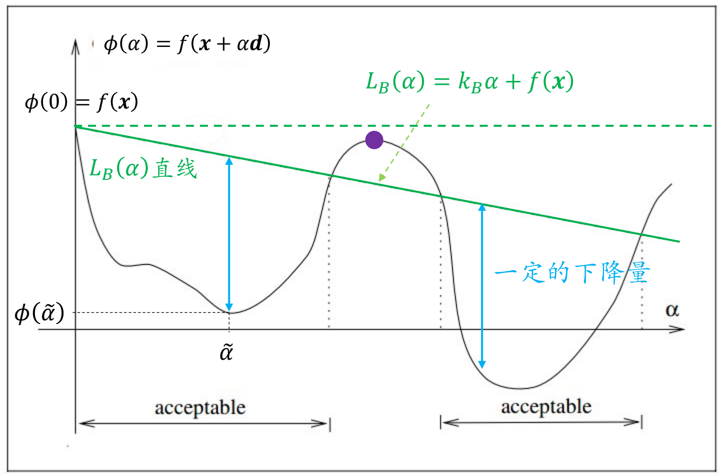


Figure 3.3 Sufficient decrease condition.

定义直线$L_B(\alpha)$ $= f(\boldsymbol{x}) + \rho[\nabla f(\boldsymbol{x})^T\boldsymbol{d}]\alpha$

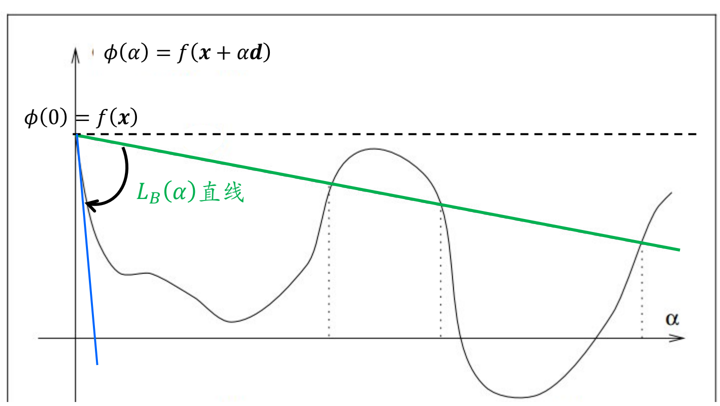
$\alpha=0$处切线 $L_A(\alpha) = f(\boldsymbol{x}) + [\nabla f(\boldsymbol{x})^T\boldsymbol{d}]\alpha \qquad \Leftarrow \ \phi'(0)=\nabla f(\boldsymbol{x})^T\boldsymbol{d},\ \phi(0)=f(\boldsymbol{x})$

取$L_B(\alpha)$的斜率 $k_B = \rho[\nabla f(\boldsymbol{x})^T\boldsymbol{d}]$

$\rho: 0 \to 1, \qquad \rho \in (0,1)$

$L_B(\alpha)$从水平位置转向切线$L_A(\alpha)$

一般地， $\rho = 10^{-2}$~10^{-4}



$$f(\boldsymbol{x}+\alpha\boldsymbol{d}) \approx \boxed{f(\boldsymbol{x}) + [\nabla f(\boldsymbol{x})^T\boldsymbol{d}]\alpha \le f(\boldsymbol{x}) + \rho[\nabla f(\boldsymbol{x})^T\boldsymbol{d}]\alpha}$$

α充分小时，一定成立 $\Leftarrow$ f是连续的 + $\boldsymbol{d}$是下降方向

可接受步长的判定条件 $f(\boldsymbol{x}+\alpha\boldsymbol{d}) \le f(\boldsymbol{x}) + [\rho\nabla f(\boldsymbol{x})^T\boldsymbol{d}]\alpha$， $\rho \in (0,1)$

在$L_B(\alpha)$之下的$\phi(\alpha)$对应的α是可接受步长

①对应的$\phi(\tilde{\alpha})$　在$L_B(\alpha)$之下，
$\Longrightarrow$可接受步长

②对应的$\phi(\hat{\alpha})$　在$L_B(\alpha)$之上，
$\Longrightarrow$非可接受步长

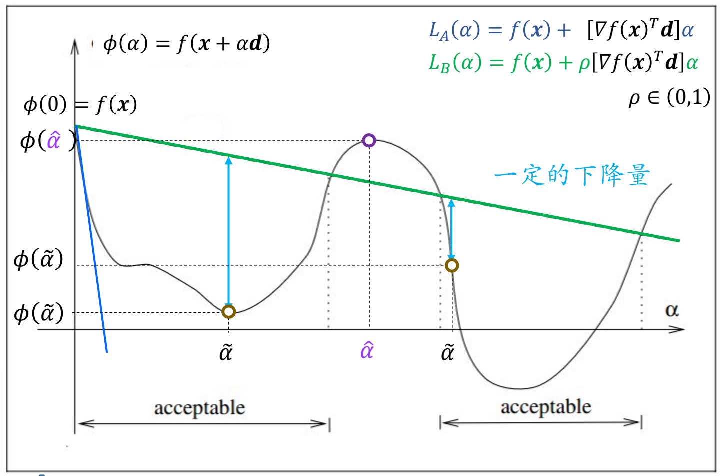


Figure 3.3 Sufficient decrease condition.

1. 引言
2. Armijo条件（充分下降条件）
3. Goldstein条件
4. Wolfe条件

Armijo条件　充分下降条件

$$f(\boldsymbol{x}+\alpha\boldsymbol{d}) \le f(\boldsymbol{x})+[\rho\nabla f(\boldsymbol{x})^T\boldsymbol{d}]\alpha\,,\quad \rho \in (0,1) \qquad \phi(\alpha) \le \phi(0)+[\rho\phi'(0)]\alpha$$

$$L_A(\alpha) \le f(\boldsymbol{x}+\alpha\boldsymbol{d}) \le L_B(\alpha)$$

$\rho \in (0,1)$

一般地， $\rho = 10^{-2} \sim 10^{-4}$

函数值的下降量 $\ge$ $|[\rho\nabla f(\boldsymbol{x})^T\boldsymbol{d}]\alpha|$

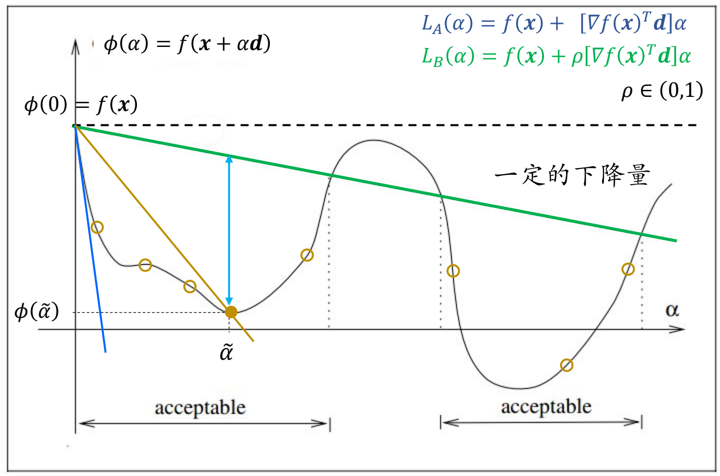


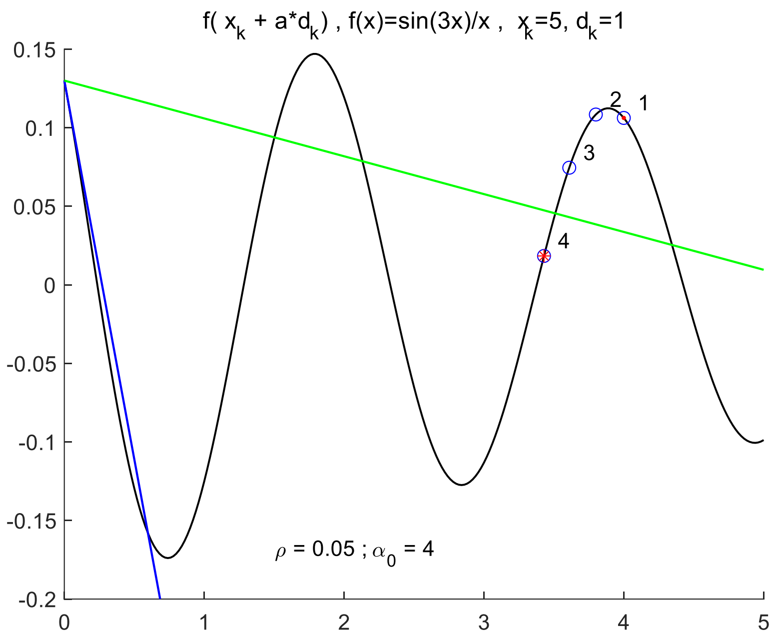
Figure 3.3 Sufficient decrease condition.

Armijo非精确搜索法的Backtracking算法

Given $\alpha_0 = 1, \beta \in (0,1), \rho \in (0,0.5), \boldsymbol{x}, \boldsymbol{d}, f$ with $\nabla f(\boldsymbol{x})^T\boldsymbol{d} < 0$
Set $\alpha \leftarrow \alpha_0$
While $f(\boldsymbol{x}+\alpha\boldsymbol{d}) > f(\boldsymbol{x})+[\rho\nabla f(\boldsymbol{x})^T\boldsymbol{d}]\alpha$ do
　$\alpha \leftarrow \beta\alpha$
End(while)

同时，还通过定义最大迭代次数终止运算

一般地，
$\alpha_0 = 1$
$\sigma = 10^{-4} \sim 10^{-1}$
$\beta = 0.5$
$k_{max} = 20$



MainBacktrack_Armijo.m

Armijo非精确搜索方法的基本算法

Step 1

Input $\boldsymbol{x}^{(k)}$, $\boldsymbol{d}^{(k)}$, and compute $\boldsymbol{g}^{(k)}$.

Given $\beta \in (0,1)$, $\sigma \in (0,0.5)$, $m := 0$

Step 2

If $f\left(\boldsymbol{x}^{(k)} + \beta^m \boldsymbol{d}^{(k)}\right) \leq f\left(\boldsymbol{x}^{(k)}\right) + \sigma\beta^m \boldsymbol{g}^{(k)T}\boldsymbol{d}^{(k)}$

then

$m_k := m, \quad \alpha^* = \beta^{m_k}$

$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \beta^{m_k}\boldsymbol{d}^{(k)}$

STOP

else

$m := m + 1$

endif

$$\boldsymbol{g}^{(k)} = \nabla f\left(\boldsymbol{x}^{(k)}\right)$$

同时，还通过定义最大迭代次数终止运算

一般地，

$\rho = 0.1, \beta = 0.5$

$k_{max} = 20$

Armijo_search.m

例

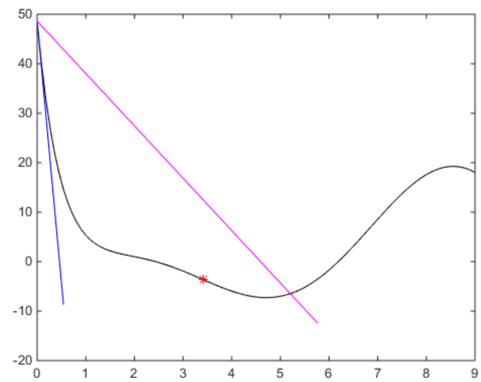
例4.1 已知函数

$$f(x) = -3x\sin(0.75x) + e^{-2x}$$

在当前点$x = -2$处的一个下降方向$d = 1$，用Armijo搜索法获取一个可接受步长（$\rho = 0.1$）。

example_4_1_CH04.m

alpha_acceptable =3.4071
x_next =1.4071
f_next =-3.6129
k =26



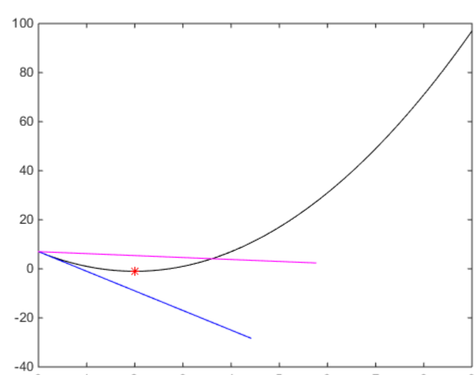
例4.2 已知函数

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

在当前点$x = [2 \quad 2]$处的一个下降方向$d = [-1 \quad -1]^T$，用Armijo搜索法获取一个可接受步长（$\rho = 0.1$）。

example_4_2_CH04.m

alpha_acceptable =2
x_next =[0 0]
f_next =-1
k =2



例4.3 已知函数

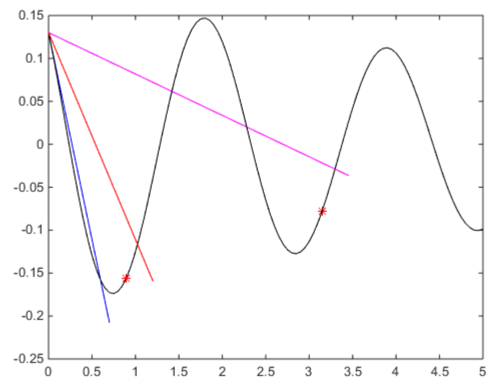
$$f(x) = \frac{\sin(3x)}{x}$$

在当前点$x = 5$处的一个下降方向$d = 1$，用Armijo搜索法获取一个可接受步长（$\rho = 0.1$和$\rho = 0.5$）。

example_4_3_CH04.m

```
alpha_acceptable =3.1471
x_next =8.1471
f_next =-0.0783
k =26


alpha_acceptable = 0.8955
x_next =5.8955
f_next =-0.1557
k =28
```



满足Armijo条件的迭代点列不一定收敛于驻点

例 $f(x) = \frac{1}{2}x^2, \quad x_0 = 1$ 判断迭代是否满足Armijo条件，并判断迭代是否收敛至驻点？

用梯度下降法

$$\nabla f(x) = x, \quad d_k = -\nabla f(x_k) = -x_k$$

$$\alpha_k = \frac{1}{2^{k+2}x_k} = \frac{1}{2^{k+2}\left(\frac{1}{2} + \frac{1}{2^{k+1}}\right)} = \frac{1}{2^{k+1} + 2} < 1$$

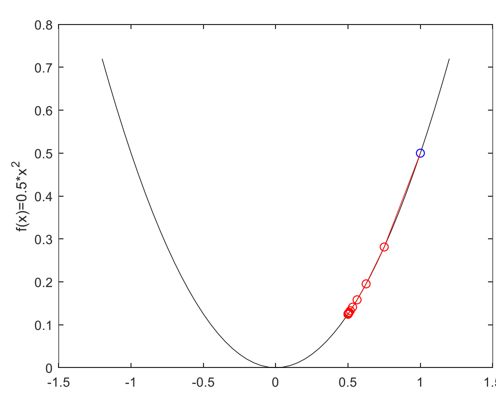
$$x_{k+1} = x_k + \alpha_k d_k = x_k - \frac{1}{2^{k+2}x_k} x_k$$

$$x_k = \frac{1}{2} + \frac{1}{2^{k+1}} \quad \to \frac{1}{2}$$

$$f(x_k) = \frac{1}{2}\left(\frac{1}{2} + \frac{1}{2^{k+1}}\right)^2 \quad \to \frac{1}{8}$$

函数值一直下降

没有收敛到驻点



$$f(x)=\frac{1}{2}x^2, \qquad x_0=1$$

用梯度下降法

$$\nabla f(x)=x, \qquad d_k=-\nabla f(x_k)=-x_k$$

$$\alpha_k=\frac{1}{2^{k+1}+2}<1$$

分析:

$$\phi(\alpha)=f\big(x_0+\alpha\nabla f(x_0)\big)$$
$$=\frac{1}{2}(x_0-\alpha x_0)^2$$
$$=\frac{1}{2}(1-\alpha)^2$$

$$x_k=\frac{1}{2}+\frac{1}{2^{k+1}}$$

$$f(x_k)=\frac{1}{2}\left(\frac{1}{2}+\frac{1}{2^{k+1}}\right)^2$$

$$L_A(\alpha)=f(x_0)+\alpha\nabla f(x_0)^T\big(-\nabla f(x_0)\big)$$
$$=\frac{1}{2}-\alpha x_0^2=\frac{1}{2}-\alpha$$

$$L_B(\alpha)=\frac{1}{2}-\rho\alpha x_0^2=\frac{1}{2}-\rho\alpha \qquad \rho=0.5$$

$$L_A(\alpha)=\frac{1}{2}-\alpha$$

$$L_B(\alpha)=\frac{1}{2}-\frac{1}{2}\alpha$$

$$\phi(\alpha)=\frac{1}{2}(1-\alpha)^2$$

$$L_A(\alpha)=\frac{1}{2}-\alpha$$

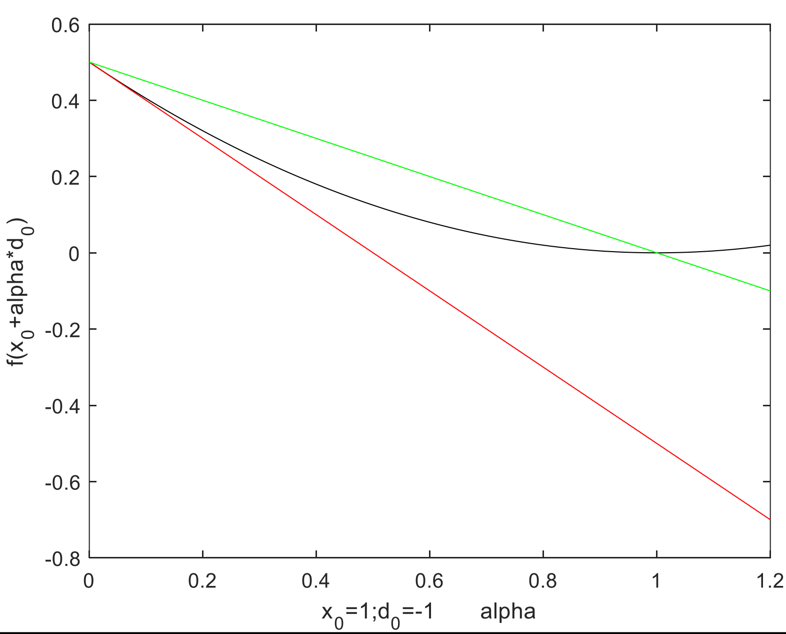
$$L_B(\alpha)=\frac{1}{2}-\frac{1}{2}\alpha$$

$$\alpha\in[0 \quad 1]$$

$$L_A(\alpha)\le\phi(\alpha)\le L_B(\alpha)$$

$$\rho=0.5$$

满足Armijo Rule



$$L_A(\alpha) = \frac{1}{2} - \alpha$$

$$L_B(\alpha) = \frac{1}{2} - \frac{1}{2}\alpha$$

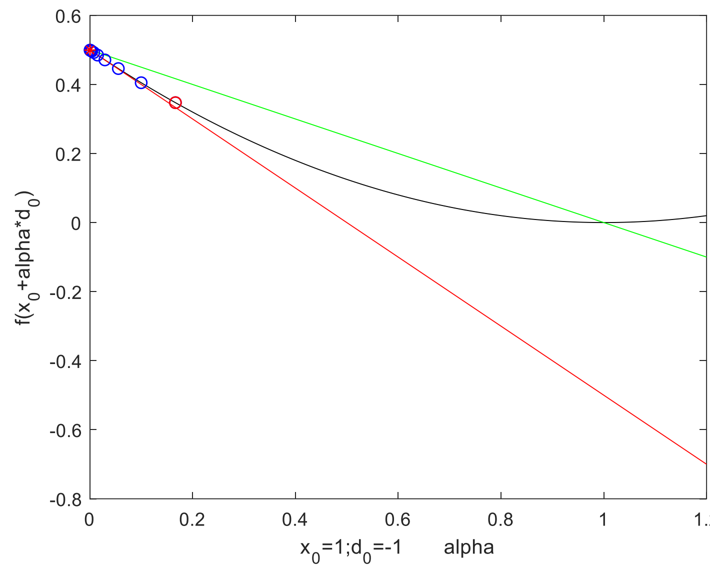
$$\phi(\alpha) = \frac{1}{2}(1-\alpha)^2$$

$$\alpha_k = \frac{1}{2^{k+1}+2} \quad \alpha_k \to 0，太小$$

$$x_k = \frac{1}{2} + \frac{1}{2^{k+1}}$$

$$\phi(\alpha_k) = \frac{1}{2}(1-\alpha_k)^2 \to \frac{1}{2}$$

函数值的下降量越来越小→ 0



此例中，虽然满足Armijo条件，但是，**没有收敛到驻点**

说明：Armijo条件**依然不能够保证**全局收敛性（即迭代收敛到全局最小值）

这是因为步长选取仍然**过于保守**

也就是说，我们的步长选取应该要**大胆一些**

因而，产生了避免选择步长过于小的Goldstein条件

M05M11084 最优化理论、算法与应用

4 非精确一维搜索方法

1. 引言
2. Armijo条件（充分下降条件）
3. Goldstein条件
4. Wolfe条件

Goldstein条件

$L_A(\alpha) = f(\boldsymbol{x}) + \ [\nabla f(\boldsymbol{x})^T \boldsymbol{d}]\alpha$

$L_B(\alpha) = f(\boldsymbol{x}) + \rho[\nabla f(\boldsymbol{x})^T \boldsymbol{d}]\alpha$

$L_C(\alpha) = f(\boldsymbol{x}) + (1-\rho)[\nabla f(\boldsymbol{x})^T \boldsymbol{d}]\alpha$

$\rho \in (0,1)$

为了避免可接受步长过小，增加一个条件

$$f(\boldsymbol{x} + \alpha \boldsymbol{d}) \le f(\boldsymbol{x}) + [\rho \boldsymbol{g}(\boldsymbol{x})^T \boldsymbol{d}]\alpha$$

$$f(\boldsymbol{x} + \alpha \boldsymbol{d}) \ge f(\boldsymbol{x}) + [(1-\rho) \boldsymbol{g}(\boldsymbol{x})^T \boldsymbol{d}]\alpha$$

$$\rho \in (0,0.5)$$

$$f(\boldsymbol{x}) + [(1-\rho)\boldsymbol{g}(\boldsymbol{x})^T \boldsymbol{d}]\alpha \le f(\boldsymbol{x} + \alpha \boldsymbol{d}) \le f(\boldsymbol{x}) + [\rho \boldsymbol{g}(\boldsymbol{x})^T \boldsymbol{d}]\alpha$$

特别适合牛顿法，但，不适合拟牛顿法

Goldstein_search.m

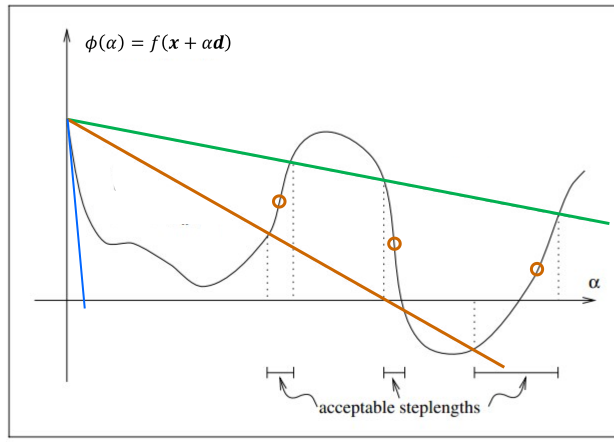
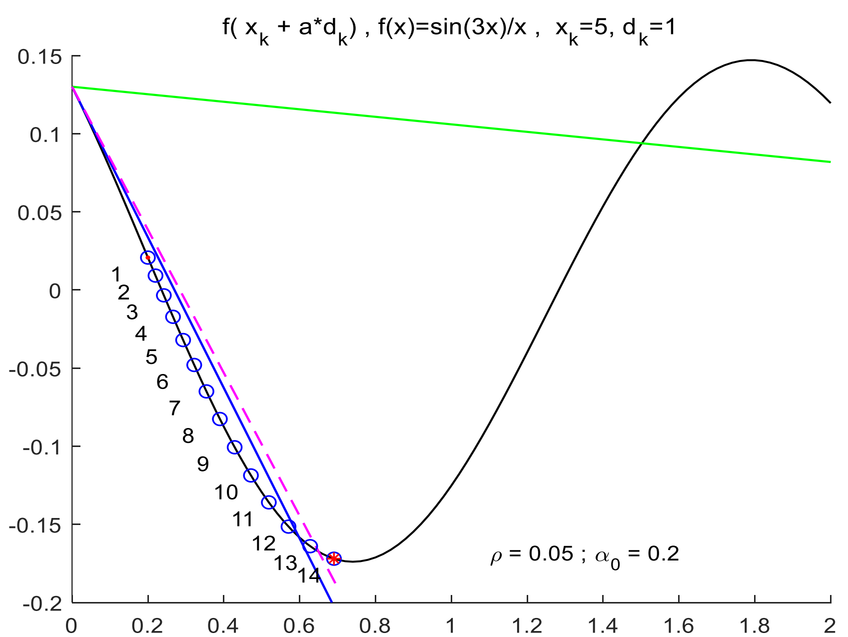


Figure 3.6 The Goldstein conditions.



MainGoldstein.m

例

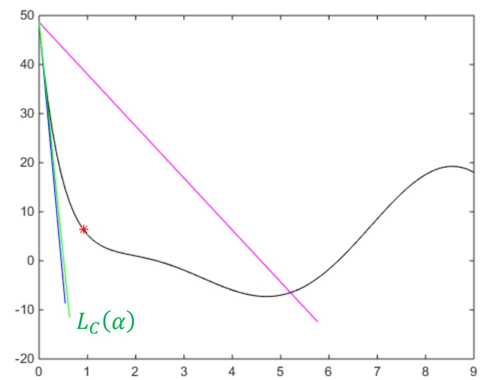
例4.4 已知函数

$$f(x) = -3x\sin(0.75x) + e^{-2x}$$

在当前点$x = -2$处的一个下降方向$d = 1$，用Goldstein搜索法获取一个可接受步长（$\rho = 0.1$）。

example_4_4_CH04.m

alpha_acceptable = 0.9182
x_next = -1.0818
f_next =6.3485
k =1



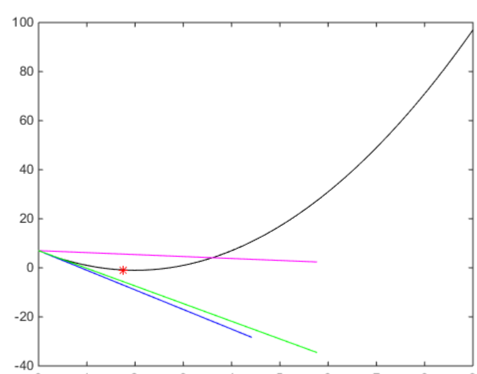
例4.5 已知函数

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

在当前点$x = [2 \quad 2]$处的一个下降方向$d = [-1 \quad -1]^T$，用Goldstein搜索法获取一个可接受步长（$\rho = 0.1$）。

example_4_5_CH04.m

alpha_acceptable = 1.7500
x_next =[0.2500   0.2500]
f_next = -0.8750
k = 1



例4.6 已知函数

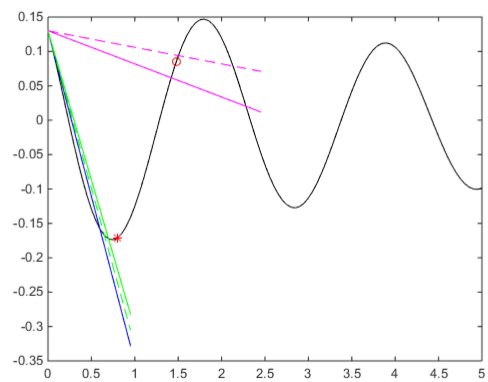
$$f(x)=\frac{\sin(3x)}{x}$$

在当前点$x=5$处的一个下降方向$d=1$，用Goldstein搜索法获取一个可接受步长（$\rho=0.1$和$\rho=0.5$）。

example_4_6_CH04.m

alpha_acceptable = 0.8015
x_next = 5.8015
f_next = -0.1710
k = 3

alpha_acceptable = 1.4787
x_next = 6.4787
f_next = 0.0854
k = 2



Goldstein条件的局限

$L_A(\alpha)=f(\boldsymbol{x})+\ [\nabla f(\boldsymbol{x})^T\boldsymbol{d}]\alpha$

$L_B(\alpha)=f(\boldsymbol{x})+\rho[\nabla f(\boldsymbol{x})^T\boldsymbol{d}]\alpha$

$L_C(\alpha)=f(\boldsymbol{x})+(1-\rho)[\nabla f(\boldsymbol{x})^T\boldsymbol{d}]\alpha$

$\rho\in(0,1)$

类似于Armijo条件，但是需要步长减少的不能太少

参数$\rho\in(0,0.5)$

满足该条件的步长$\tilde{\alpha}$被两个射线$L_B(\alpha)$和$L_C(\alpha)$包围着

使用该方法可能会错过最优解，如图

Wolfe条件
- Armijo条件
- 曲率条件

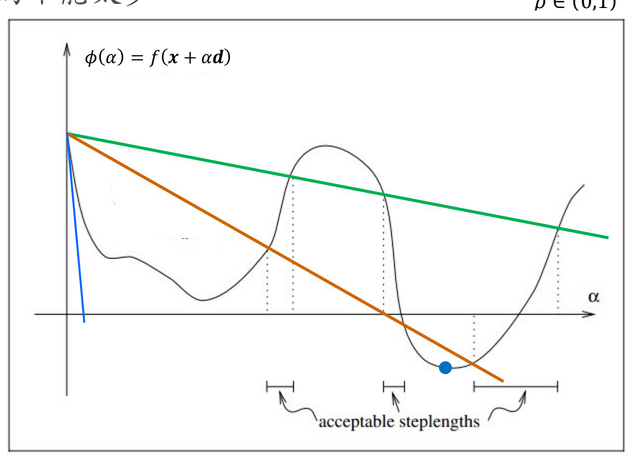


Figure 3.6 The Goldstein conditions.

斜率变化与极小点之间的关系

极小点α^*满足 $\phi'(\alpha^*) = 0$

① 斜率负且较小

如果$\phi'(\alpha) < 0$，且$|\phi'(\alpha)|$较大，即，斜率$\phi'(\alpha)$“强烈”为负，“离”$\phi'(\alpha^*) = 0$“较远”，说明沿方向$\boldsymbol{d}$移动（增大步长）可更多地降低f

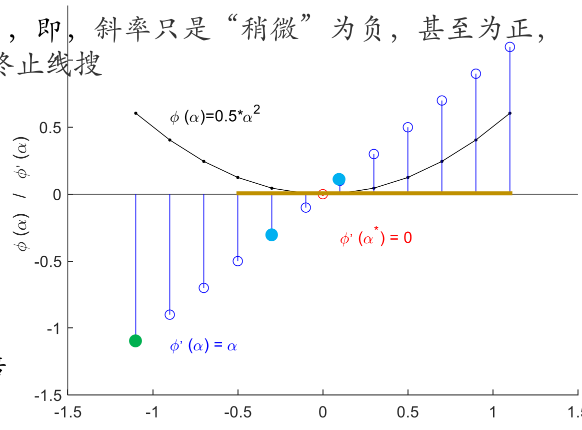
② 斜率负且较大 或者 斜率正

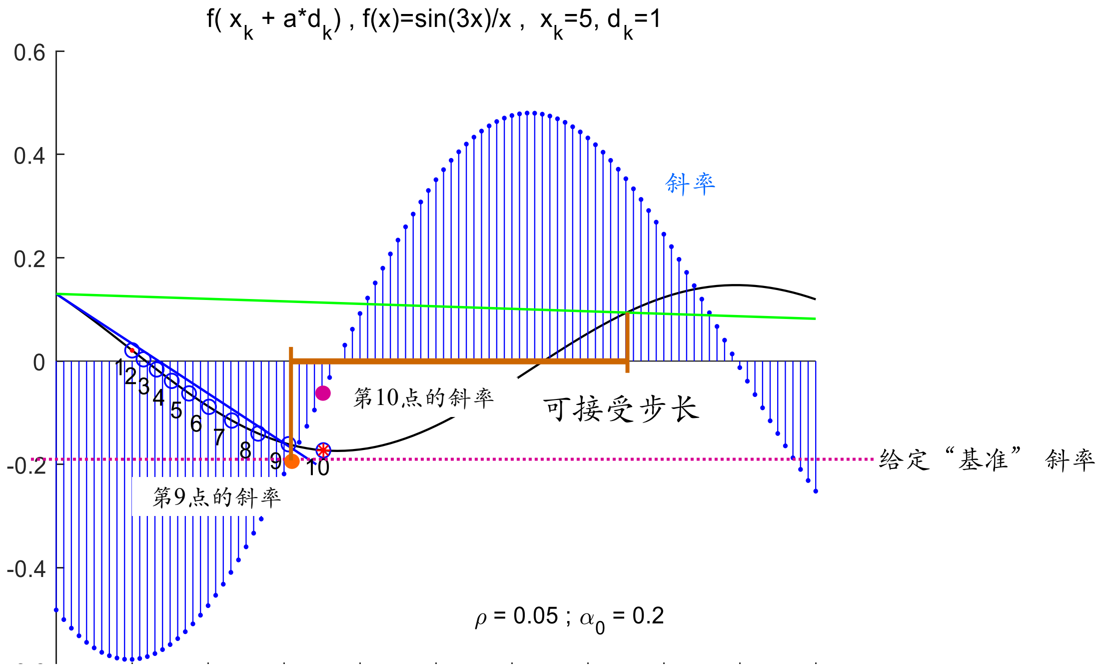
如果$\phi'(\alpha) < 0$ 且 $|\phi'(\alpha)|$较小，或 $\phi'(\alpha) > 0$，即，斜率只是“稍微”为负，甚至为正，那么，沿方向$\boldsymbol{d}$不能再使f有更大的下降量，终止线搜

在满足Armijo条件中，剔除了部分小斜率，保证选择的步长略大一些

比如，

取α_k，使得$\phi(\alpha_k)$的斜率$\phi'(\alpha_k)$比$\phi'(0)$大σ倍



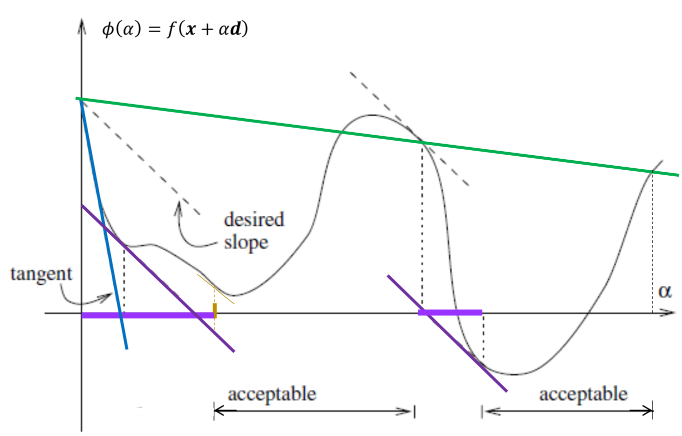


曲率条件

$$\boldsymbol{g}(\boldsymbol{x}+\alpha\boldsymbol{d})^T\boldsymbol{d} \geq \sigma\boldsymbol{g}(\boldsymbol{x})^T\boldsymbol{d}, \qquad \sigma \in (\rho, 1)$$

$\phi'(\alpha_k) = \boldsymbol{g}(\boldsymbol{x}+\alpha_k\boldsymbol{d})^T\boldsymbol{d} \qquad \boldsymbol{g}(\boldsymbol{x})^T\boldsymbol{d} = \phi'(0)$

$\phi(\alpha_k)$的斜率比$\phi'(0)$大σ倍



The curvature condition.

1. 引言
2. Armijo条件（充分下降条件）
3. Goldstein条件
4. Wolfe条件
   - 弱Wolfe条件
   - 强Wolfe条件
   - 满足Wolfe条件的线性搜索算法

Wolfe条件

$$f(\boldsymbol{x}+\alpha \boldsymbol{d}) \le f(\boldsymbol{x}) + [\rho \boldsymbol{g}(\boldsymbol{x})^T \boldsymbol{d}]\alpha, \qquad \rho \in (0,1)$$ Armijo条件

$$\boldsymbol{g}(\boldsymbol{x}+\alpha \boldsymbol{d})^T \boldsymbol{d} \ge \sigma \boldsymbol{g}(\boldsymbol{x})^T \boldsymbol{d}, \qquad \sigma \in (\rho, 1)$$ Curvature条件

- ✓ 若采用牛顿或拟牛顿方法确定搜索方向， 取$\sigma = 0.9$
- ✓ 若采用非线性共轭梯度方法获得搜索方向， 取$\sigma = 0.1$

Wolfe_search.m

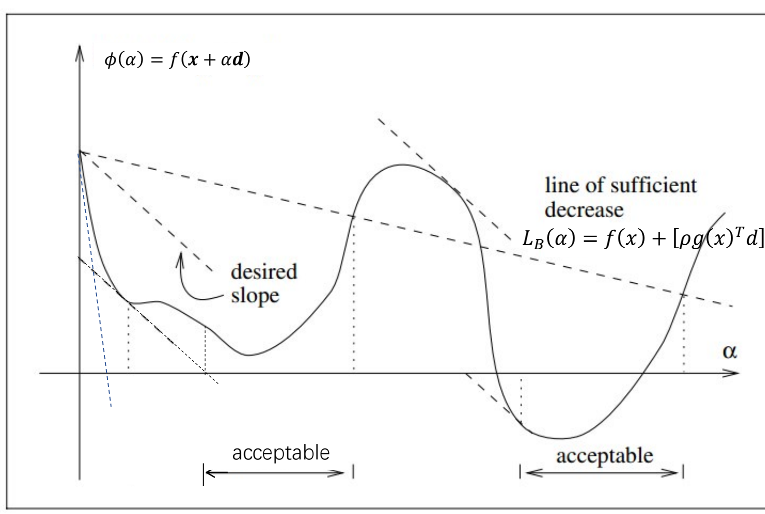


Figure 3.5 Step lengths satisfying the Wolfe conditions.

例

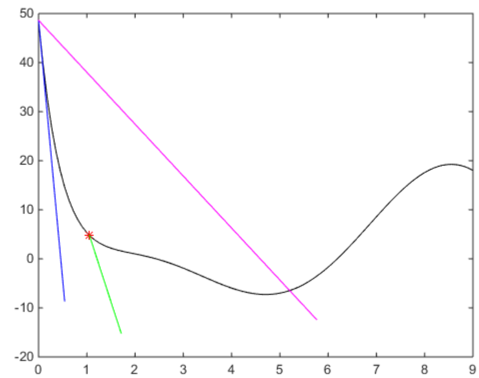
例4.7 已知函数

$$f(x) = -3x\sin(0.75x) + e^{-2x}$$

在当前点$x = -2$处的一个下降方向$d = 1$，用Wolfe搜索法获取一个可接受步长（$\rho = 0.1$，$\sigma = 0.11$）。

example_4_7_CH04.m

alpha_acceptable =1.0530
x_next =-0.9470
f_next =4.7935
k =2



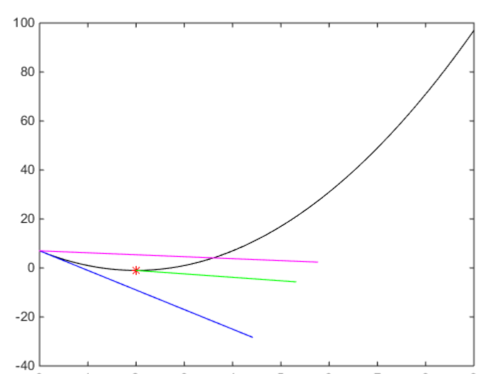
例4.8 已知函数

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

在当前点$x = [2 \quad 2]$处的一个下降方向$d = [-1 \quad -1]^T$，用Wolfe搜索法获取一个可接受步长（$\rho = 0.1$，$\sigma = 0.11$）。

example_4_8_CH04.m

alpha_acceptable = 2
x_next =[0   0]
f_next = -1
k = 2



example_4_9_CH04.m

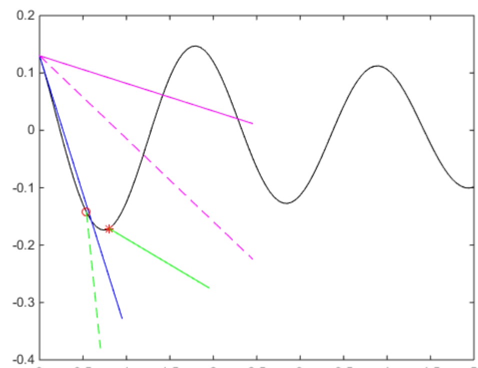
例4.9 已知函数

$$f(x)=\frac{\sin(3x)}{x}$$

在当前点$x=5$处的一个下降方向$d=1$，用Wolfe搜索法获取一个可接受步长（$\rho=0.1$，$\sigma=0.11$和$\rho=0.3$，$\sigma=0.7$）。

alpha_acceptable =0.8015
x_next =5.8015
f_next = -0.1710
k = 3

alpha_acceptable = 0.5399
x_next = 5.5399
f_next = -0.1427
k =1



1. 引言
2. Armijo条件（充分下降条件）
3. Goldstein条件
4. Wolfe条件
- 弱Wolfe条件
- 强Wolfe条件
- 满足Wolfe条件的线性搜索算法

斜率变化与极小点之间的关系　　极小点α^*满足 $\phi'(\alpha^*) = 0$

① 斜率负且较小 距离 极小点 “远” $\Longrightarrow$ Curvature条件

满足Wolfe条件的步长α_k一般还不够接近ϕ的极小值

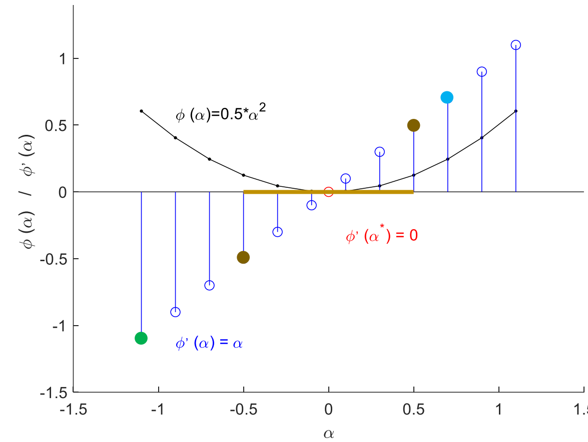
② 斜率正且较大 距离 极小点 “远” $\Longrightarrow$ 增加与Curvature相关的条件

修改曲率条件使$|\phi'(\alpha)|$较小

不允许导数$\phi'(\alpha_k)$太“正”

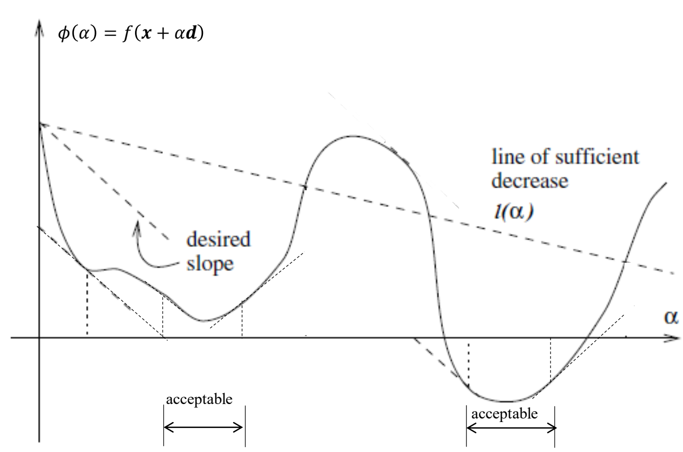
从而，排除远离ϕ的驻点的点

即，保持步长在驻点附近



修正曲率条件　$|\boldsymbol{g}(\boldsymbol{x}+\alpha_k\boldsymbol{d})^T\boldsymbol{d}| \geq \sigma|\boldsymbol{g}(\boldsymbol{x})^T\boldsymbol{d}|$

使α_k位于ϕ的某局部极小值或驻点 附近的小邻域内



Strong Wolfe Conditions

强Wolfe条件

$$f(\boldsymbol{x}+\alpha \boldsymbol{d}) \le f(\boldsymbol{x}) + [\rho \boldsymbol{g}(\boldsymbol{x})^T \boldsymbol{d}]\alpha \qquad \rho \in (0,1)$$
$$|\boldsymbol{g}(\boldsymbol{x}+\alpha \boldsymbol{d})^T \boldsymbol{d}| \ge \sigma |\boldsymbol{g}(\boldsymbol{x})^T \boldsymbol{d}| \qquad \sigma \in (\rho, 1)$$

以得到步长接近于极小值

用于大多数线搜索方法，
且，特别适合拟牛顿法

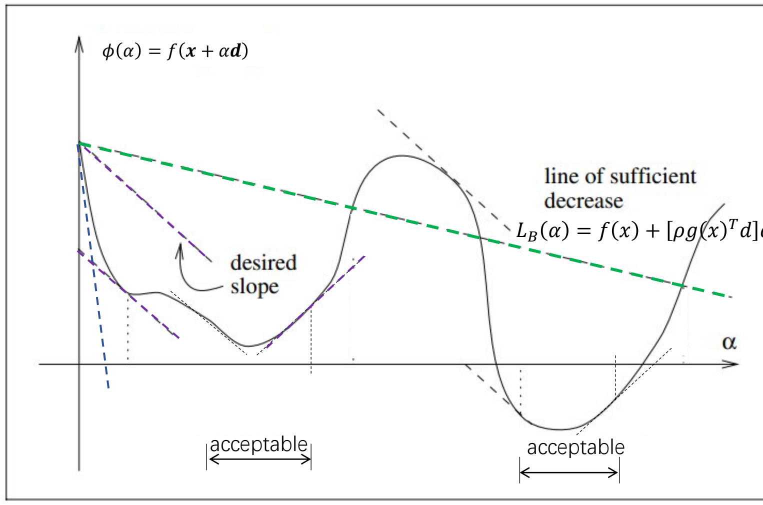


Figure 3.5 Step lengths satisfying the Wolfe conditions.

强Wolfe条件

$$f(\boldsymbol{x}+\alpha \boldsymbol{d}) \le f(\boldsymbol{x}) + [\rho \boldsymbol{g}(\boldsymbol{x})^T \boldsymbol{d}]\alpha$$
$$|\boldsymbol{g}(\boldsymbol{x}+\alpha \boldsymbol{d})^T \boldsymbol{d}| \ge \sigma |\boldsymbol{g}(\boldsymbol{x})^T \boldsymbol{d}| \qquad \rho \in (0,1), \sigma \in (\rho, 1)$$

几何意义：

可接受步长在曲线$f(\boldsymbol{x}+\alpha \boldsymbol{d})$上的对应部分必须位于直线$L_B$以下，

且可接受步长处切线的斜率绝对值不小于σ倍L_A的斜率　可接受步长靠近波谷

$\sigma \to 0$时，避免$\alpha = 0$附近的步长成为可接受步长，且靠近波谷

此方法的效果接近于精确搜索法的，但，计算量增大

$\boxed{\rho = 0.1, \sigma = 0.11}$

线搜索 强Wolfe准则

A LINE SEARCH ALGORITHM FOR THE WOLFE CONDITIONS

Algorithm 3.2 (Line Search Algorithm).
Set $\alpha_0 \leftarrow 0$, choose $\alpha_1 > 0$ and $\alpha_{\max}$;
$i \leftarrow 1$;
repeat
Evaluate $\phi(\alpha_i)$;
If $\phi(\alpha_i) > \phi(0) + c_1\alpha_i\phi'(0)$ or $[\phi(\alpha_i) \geq \phi(\alpha_{i-1})$ and $i > 1]$
$\alpha^* \leftarrow \text{zoom}(\alpha_{i-1}, \alpha_i)$ and **stop**;
Evaluate $\phi'(\alpha_i)$;
if $|\phi'(\alpha_i)| \leq -c_2\phi'(0)$
set $\alpha^* \leftarrow \alpha_i$ and **stop**;
If $\phi'(\alpha_i) \geq 0$
set $\alpha^* \leftarrow \text{zoom}(\alpha_i, \alpha_{i-1})$ and **stop**;
Choose $\alpha_{i+1} \in (\alpha_i, \alpha_{\max})$
$i \leftarrow i+1$;
end (repeat)

Algorithm 3.3 (zoom).
repeat
Interpolate (using quadratic, cubic, or bisection) to find a trial step length α_j between α_{lo} and α_{hi};
Evaluate $\phi(\alpha_j)$;
If $\phi(\alpha_j) > \phi(0) + c_1\alpha_j\phi'(0)$ or $\phi(\alpha_j) \geq \phi(\alpha_{lo})$
$\alpha_{hi} \leftarrow \alpha_j$;
else
Evaluate $\phi'(\alpha_j)$;
if $|\phi'(\alpha_j)| \leq -c_2\phi'(0)$
Set $\alpha^* \leftarrow \alpha_j$ and **stop**;
If $\phi'(\alpha_j)(\alpha_{hi} - \alpha_{lo}) \geq 0$
$\alpha_{hi} \leftarrow \alpha_{lo}$;
$\alpha_{lo} \leftarrow \alpha_j$;
end (repeat)

存在性定理

设函数$f: \mathcal{R}^n \to \mathcal{R}$连续可微，$p_k$为$x_k$处的下降方向，假定函数沿射线$\{x_k + \alpha p_k | \alpha > 0\}$有下界，那么，当$0 < c_1 < c_2 < 1$时，存在满足Wolfe条件和强Wolfe条件的步长区间

因 $\phi(\alpha) = f(x_k + \alpha p_k)$ 有下界，$\forall \alpha > 0$

又$0 < c_1 < 1$，直线$l(\alpha) = f(x_k) + c_1\alpha\nabla f_k^T p_k$ 一定与曲线$\phi(\alpha)$至少相交一次

$\nabla f_k^T p_k < 0$，p_k是下降方向

设$\tilde{\alpha} > 0$是沿α坐标最小的交点

$$f(x_k + \tilde{\alpha} p_k) = f(x_k) + \tilde{\alpha} c_1 \nabla f_k^T p_k$$

$\forall 0 < \alpha < \tilde{\alpha}$，有

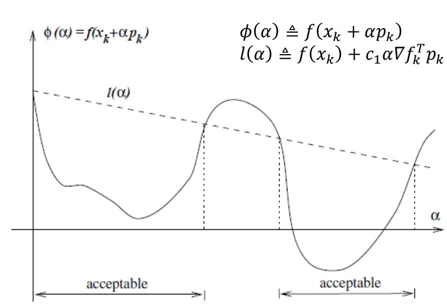
$$f(x_k + \alpha p_k) \leq f(x_k) + \alpha c_1 \nabla f_k^T p_k$$

由中值定理，存在$\hat{\alpha} \in (0, \tilde{\alpha})$，使

$$f(x_k + \tilde{\alpha} p_k) - f(x_k) = \tilde{\alpha} c_1 \nabla f(x_k + \hat{\alpha} p_k)^T p_k$$

$$\Longrightarrow \quad \nabla f(x_k + \hat{\alpha} p_k)^T p_k = c_1 \nabla f_k^T p_k > c_2 \nabla f_k^T p_k$$

由的连续性知，在$\hat{\alpha}$附近的α都满足弱\强Wolfe条件



1. 引言
2. Armijo条件（充分下降条件）
3. Goldstein条件
4. Wolfe条件
    - 弱Wolfe条件
    - 强Wolfe条件
    - 满足Wolfe条件的线性搜索算法

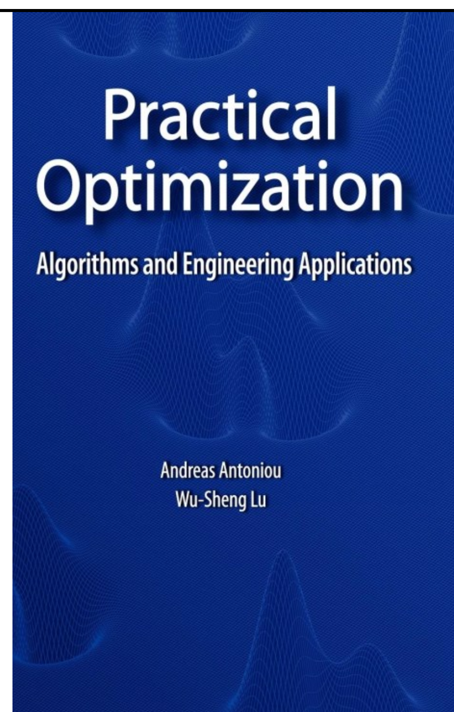
Algorithm 4.6 Inexact line search

Wolfe条件

Fletcher算法，实用，稳定性高

inex_lsearch.m

Practical Optimization: Algorithms and Engineering applications,
Wu-Sheng LU



REVIEW

两点抛物线插值法——内插法

1. 已知函数一点的函数值和导数及另一点的函数值，可以用内插法

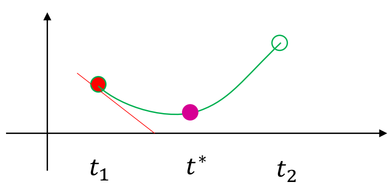
设 函数$f(t)$，已知$f_i = f(t_i)$，$i = 1,2$，$f_1' = f'(t_1)$

二次多项式$P(t) = a + bt + ct^2$，满足$P(t_i) = f_i$，$i = 1,2$；$f_1' = P'(t_1)$

$$P(t_1) = a + bt_1 + ct_1^2 = f_1$$
$$P(t_2) = a + bt_2 + ct_2^2 = f_2$$
$$P'(t_1) = b + 2ct_1 = f_1'$$

$$P''(t) = c > 0$$
$$P'(t) = b + 2ct = 0 \rightarrow \quad t^* = -\frac{b}{2c}$$

$$t^* = t_1 - \frac{f_1'(t_2 - t_1)^2}{2[f_2 - f_1 - f_1'(t_2 - t_1)]}$$



t^*位于t_1切线的右侧，t_2的左侧

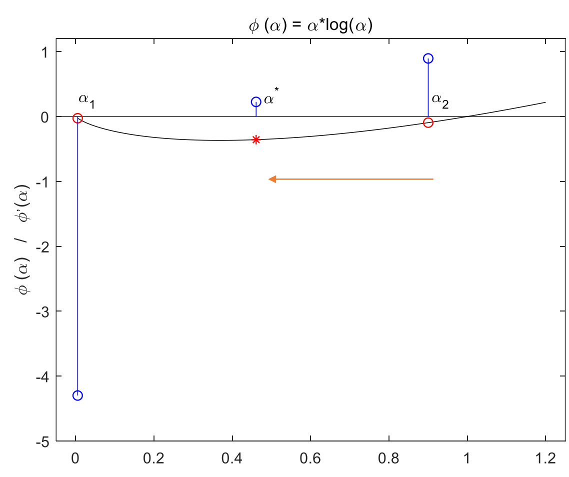
$$t_1 < t^* < t_2$$

$$\phi(\alpha) = \alpha \log \alpha \qquad [0.05,1.2]$$

$$t^* = t_1 - \frac{f_1'(t_2 - t_1)^2}{2[f_2 - f_1 - f_1'(t_2 - t_1)]}$$

$$\phi'(\alpha) = 1 + \frac{1}{\alpha}$$

$$\alpha_1 = 0.05$$
$$\alpha_2 = 0.9 \quad \longrightarrow \alpha^* = 0.4606$$



α^*位于α_1的右侧，α_2的左侧

$$\alpha_1 < \alpha^* < \alpha_2$$

$$\phi'(\alpha_1) = -4.30 < \phi'(\alpha^*) = 0.22 < \phi'(\alpha_2) = 0.89$$

Interpolation_Extrapolation.m

$$t^* = t_1 - \frac{\phi_1'(t_2-t_1)^2}{2[\phi_2 - \phi_1 - \phi_1'(t_2-t_1)]}$$

条件：$\phi_2 > \phi_1 + \phi_1'(t_2-t_1)$

可接受步长区间$[\alpha_L, \alpha_U]$，

测试步长$\alpha_M \in [\alpha_L, \alpha_U]$，求 $\breve{\alpha}_M < \alpha_M$

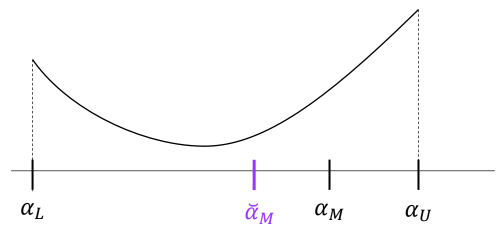
$$\phi(\alpha) = f(\boldsymbol{x} + \alpha\boldsymbol{d})$$

$$\phi'(\alpha) = \frac{df(\boldsymbol{x}+\alpha\boldsymbol{d})}{d\alpha} = \boldsymbol{g}(\boldsymbol{x}+\alpha\boldsymbol{d})^T\boldsymbol{d}$$

$$\phi_1' = \boldsymbol{g}(\boldsymbol{x}+\alpha_L\boldsymbol{d})^T\boldsymbol{d}$$

$$\phi_1 = f(\boldsymbol{x}+\alpha_L\boldsymbol{d})$$

$$\phi_2 = f(\boldsymbol{x}+\alpha_M\boldsymbol{d})$$



$$\alpha_{new\_i} = \alpha_L - \frac{(\alpha_M-\alpha_L)^2\boldsymbol{g}(\boldsymbol{x}+\alpha_L\boldsymbol{d})^T\boldsymbol{d}}{2[f(\boldsymbol{x}+\alpha_M\boldsymbol{d}) - f(\boldsymbol{x}+\alpha_L\boldsymbol{d}) - (\alpha_M-\alpha_L)\boldsymbol{g}(\boldsymbol{x}+\alpha_L\boldsymbol{d})^T\boldsymbol{d}\,]}$$

$$\alpha_{new\_i} = \breve{\alpha}_M \Longrightarrow \alpha_M$$

两点抛物线插值法——外差法

已知函数二点的导数 及第二个点的函数值，可以用外插法

设 函数$f(t)$，已知$f_2 = f(t_2)$，$f_i' = f'(t_i)$，$i = 1,2$

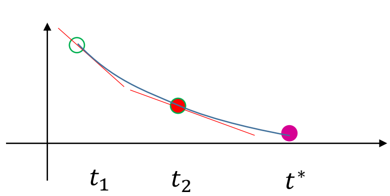
二次多项式$P(t) = a + bt + ct^2$，满足$P(t_2) = f_2$； $f_i' = P'(t_i)$，$i = 1,2$

$$P(t_2) = a + bt_2 + ct_2^2 = f_2$$
$$P'(t_1) = b + 2ct_1 = f_1'$$
$$P'(t_2) = b + 2ct_2 = f_2'$$

$P"(t) = c > 0$ $\quad$ $f_2' > f_1'$

$$P'(t) = b + 2ct = 0 \rightarrow \quad t^* = -\frac{b}{2c}$$

$$t^* = t_2 - \frac{f_2'(t_2-t_1)}{f_2' - f_1'}$$



t^*位于t_2右侧，比t_2更大

$$t^* = t_2 - \frac{\phi_2'(t_2 - t_1)}{\phi_2' - \phi_1'}$$

条件：$\phi_2' > \phi_1'$

可接受步长区间$[\alpha_L, \alpha_U]$

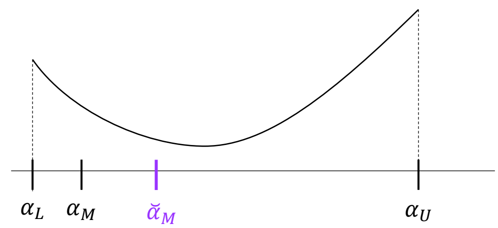
测试步长$\alpha_M \in [\alpha_L, \alpha_U]$，求 $\breve{\alpha}_M > \alpha_M$

$$\phi(\alpha) = f(\boldsymbol{x} + \alpha \boldsymbol{d})$$

$$\phi'(\alpha) = \frac{df(\boldsymbol{x} + \alpha \boldsymbol{d})}{d\alpha} = \boldsymbol{g}(\boldsymbol{x} + \alpha \boldsymbol{d})^T \boldsymbol{d}$$

$$\phi_1' = \boldsymbol{g}(\boldsymbol{x} + \alpha_L \boldsymbol{d})^T \boldsymbol{d}$$

$$\phi_2' = \boldsymbol{g}(\boldsymbol{x} + \alpha_M \boldsymbol{d})^T \boldsymbol{d}$$



$$\boxed{\alpha_{new\_e} = \alpha_M - \frac{(\alpha_M - \alpha_L)\boldsymbol{g}(\boldsymbol{x} + \alpha_M \boldsymbol{d})^T \boldsymbol{d}}{\boldsymbol{g}(\boldsymbol{x} + \alpha_M \boldsymbol{d})^T \boldsymbol{d} - \boldsymbol{g}(\boldsymbol{x} + \alpha_L \boldsymbol{d})^T \boldsymbol{d}}}$$

$$\alpha_{new\_e} = \breve{\alpha}_M \Longrightarrow \alpha_M$$

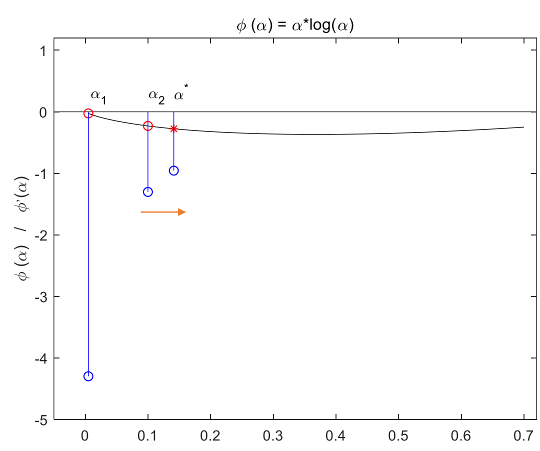
$\phi(\alpha) = \alpha \log \alpha \quad [0.05,0.7]$

$$t^* = t_2 - \frac{f_2'(t_2 - t_1)}{f_2' - f_1'}$$

$$\phi'(\alpha) = 1 + \frac{1}{\alpha}$$

$\alpha_1 = 0.05$

$\alpha_2 = 0.1 \quad \longrightarrow \alpha^* = 0.1413$



α^*位于α_2的右侧

$$\alpha_1 < \alpha_2 < \alpha^*$$

$$\phi'(\alpha_1) = -4.30 < \phi'(\alpha_2) = -0.96 < \phi'(\alpha^*) = -1.30$$

Interpolation_Extrapolation.m

内插与外插示例对比

$\phi(\alpha) = \alpha \log \alpha \quad [0.05,1.2]$

内插

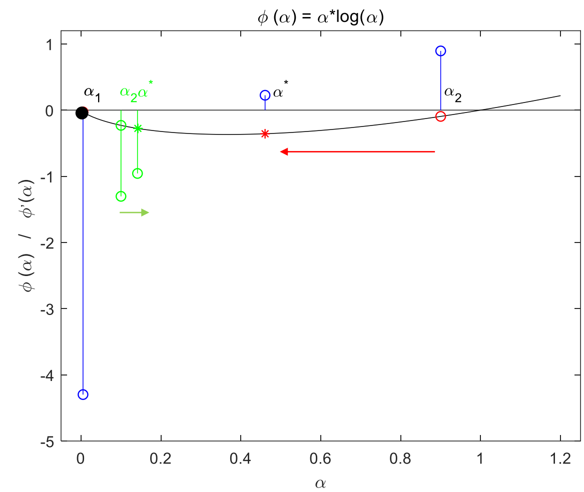
$\alpha_1 = 0.05$

$\alpha_2 = 0.9 \quad \longrightarrow \alpha^* = 0.4606$

外插

$\alpha_1 = 0.05$

$\alpha_2 = 0.1 \quad \longrightarrow \alpha^* = 0.1413$



Interpolation_Extrapolation.m

运用内插法和外插法调整测试步长的大小

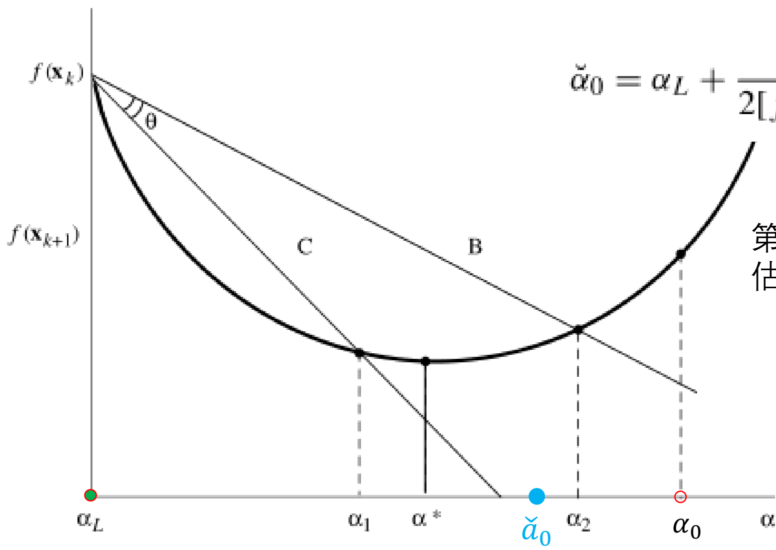
假定希望所求步长$\alpha_0 \in [\alpha_1 \quad \alpha_2]$

1. 当估算值$\alpha_0 > \alpha_2$，采用内插法，得改进点$\breve{\alpha}_0 < \alpha_0 < \cdots < \alpha_2$

2. 当估算值$\alpha_0 < \alpha_1$，采用外插法，得改进点$\underbrace{\breve{\alpha}_0 > \alpha_0 > \cdots > \alpha_1}_{\text{可能经历多次迭代}}$

1. 当估算值$\alpha_0 > \alpha_2$，采用内插法，得改进点$\breve{\alpha}_0 \in [\alpha_L \quad \alpha_2]$

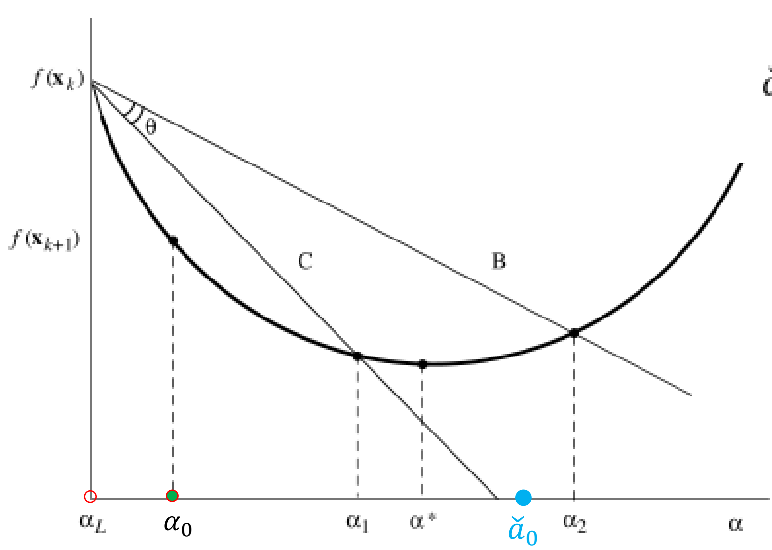
$$\breve{\alpha}_0 = \alpha_L + \frac{(\alpha_0 - \alpha_L)^2 f_L'}{2[f_L - f_0 + (\alpha_0 - \alpha_L) f_L']}$$



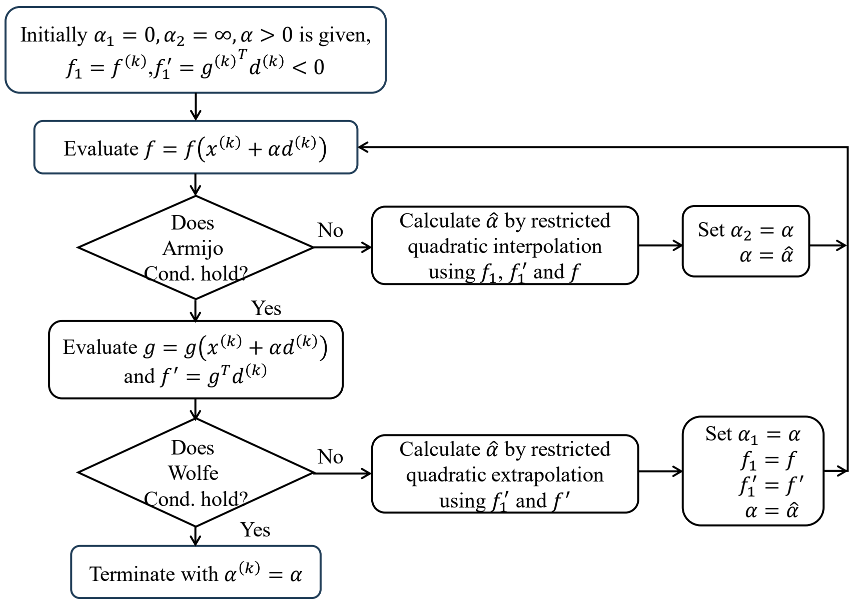
第 1 点α_L的函数值和导数值
估计点α_0的函数值

2. 当估算值$\alpha_0 < \alpha_1$，采用外插法，得改进点$\breve{\alpha}_0 \in [\alpha_L \quad \alpha_2]$

$$\breve{\alpha}_0 = \alpha_0 + \frac{(\alpha_0 - \alpha_L) f_0'}{(f_L' - f_0')}$$



第 1 点α_L的导数值
估计点α_0的函数值和导数值



初值 $\alpha = \min\{1, -2\Delta f / f_1'\}, \Delta f > 0$
或 $\alpha = c > 0$

Armijo Cond.

$$f^{(k)} - f^{(k+1)} \geq -\rho {g^{(k)}}^T d^{(k)}$$

Wolfe Cond.

$$ {g^{(k+1)}}^T d^{(k)} \geq \sigma {g^{(k)}}^T d^{(k)}$$

Interpolation

$$\hat{\alpha} - \alpha_1 = \frac{1}{2} \frac{\alpha_2 - \alpha_1}{1 + \frac{f_1 - f_2}{(\alpha_2 - \alpha_1) f_1'}}$$

Extrapolation

$$\hat{\alpha} - \alpha = \frac{(\alpha - \alpha_1) f'}{f_1' - f'}$$

Algorithm 4.6 Inexact line search

Step 1

Input $\boldsymbol{x}^{(k)}$, $\boldsymbol{d}^{(k)}$, and compute $\boldsymbol{g}^{(k)}$.

Initialize algorithm parameters ρ, σ, τ, and χ.

Set $\alpha_L = 0$ and $\alpha_U = 10^{99}$.

$\tau = 0.1, \quad \chi = 0.9$

$\rho = 0.1, \quad \sigma = 0.7$

Step 2

Compute $f_L = f\left(\boldsymbol{x}^{(k)} + \alpha_L \boldsymbol{d}^{(k)}\right)$

$f_L' < 0$，否则，初始点非下降方向

Compute $f_L' = \boldsymbol{g}\left(\boldsymbol{x}^{(k)} + \alpha_L \boldsymbol{d}^{(k)}\right)^T \boldsymbol{d}^{(k)}$

Step 3

Estimate α_0. $\quad \alpha_0 = -\frac{f_L}{f_L'}$

$$\alpha_0 = \begin{cases} 1 & |f_L'| < \varepsilon \\ -\frac{2 f_L}{f_L'} & |f_L'| \geq \varepsilon \end{cases}$$

$\alpha_0 \leftarrow 1 \quad \alpha_0 < 10^{-12}$ or $\alpha_0 > 1$

Step 4

Compute $f_0 = f\left(\boldsymbol{x}^{(k)} + \alpha_0 \boldsymbol{d}^{(k)}\right)$.

Step 5 (Interpolation) Armijo条件

If $f_0 > f_L + \rho(\alpha_0 - \alpha_L)f_L'$, then do:

a. If $\alpha_0 < \alpha_U$, then set $\alpha_U = \alpha_0$.

b. Compute $\breve{\alpha}_0$ using Eq. (4.57).

c. If $\breve{\alpha}_0 < \alpha_L + \tau(\alpha_U - \alpha_L)$ then set $\breve{\alpha}_0 = \alpha_L + \tau(\alpha_U - \alpha_L)$.

d. If $\breve{\alpha}_0 > \alpha_U - \tau(\alpha_U - \alpha_L)$ then set $\breve{\alpha}_0 = \alpha_U - \tau(\alpha_U - \alpha_L)$.

e. Set $\alpha_0 = \breve{\alpha}_0$and go to Step 4.

$$\breve{\alpha}_0 = \alpha_L + \frac{(\alpha_0 - \alpha_L)^2 f_L'}{2[f_L - f_0 + (\alpha_0 - \alpha_L)f_L']}$$

保证步长既不太靠近左端点，也不太靠近右端点

Step 6

Compute $f_0' = \boldsymbol{g}\left(\boldsymbol{x}^{(k)} + \alpha_0 \boldsymbol{d}^{(k)}\right)^T \boldsymbol{d}^{(k)}$.

① In Step 5, $\breve{\alpha}_0$ is checked and if necessary it is adjusted through a series of interpolations to ensure that $\alpha_L < \breve{\alpha}_0 < \alpha_U$.

A suitable value for τ is 0.1.

This assures that $\breve{\alpha}_0$ is no closer to α_L or α_U than 10 percent of the permissible range.

Step 7 (Extrapolation)

If $f_0' < \sigma f_L'$, then do: 曲率条件

a. Compute $\Delta\alpha_0 = (\alpha_0 - \alpha_L)f_0'/(f_L' - f_0')$ (see Eq. (4.58)).

b. If $\Delta\alpha_0 < \tau(\alpha_0 - \alpha_L)$, then set $\Delta\alpha_0 = \tau(\alpha_0 - \alpha_L)$.

c. If $\Delta\alpha_0 > \chi(\alpha_0 - \alpha_L)$, then set $\Delta\alpha_0 = \chi(\alpha_0 - \alpha_L)$.

d. Compute $\breve{\alpha}_0 = \alpha_0 + \Delta\alpha_0$.

e. Set $\alpha_L = \alpha_0$, $\alpha_0 = \breve{\alpha}_0$, $f_L = f_0$, $f_L' = f_0'$, and go to Step 4.

$$\breve{\alpha}_0 = \alpha_0 + \frac{(\alpha_0 - \alpha_L)f_0'}{(f_L' - f_0')}$$

Step 8

Output α_0 and $f_0 = f\left(\boldsymbol{x}^{(k)} + \alpha_0 \boldsymbol{d}^{(k)}\right)$, and stop.

① In Step 5, $\breve{\alpha}_0$ is checked and if necessary it is adjusted through a series of interpolations to ensure that $\alpha_L < \breve{\alpha}_0 < \alpha_U$.

A suitable value for τ is 0.1.

$\tau = 0.1$，$\chi = 0.9$

This assures that $\breve{\alpha}_0$ is no closer to α_L or α_U than 10 percent of the permissible range.

② A similar check is applied in the case of extrapolation, as can be seen in Step 7.

The value for χ suggested by Fletcher is 0.9.

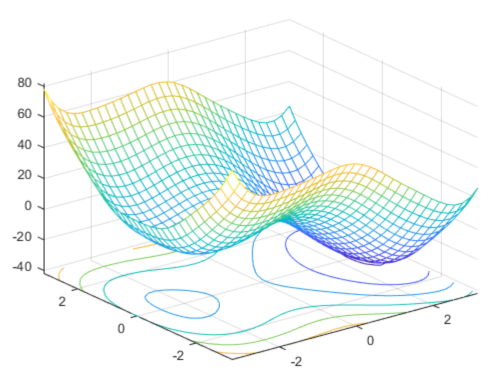
The precision to which the minimizer is determined depends on the values of ρ and σ.

Small values like $\rho = \sigma = 0.1$ will yield a relatively precise line search whereas values like $\rho = 0.3$ and $\sigma = 0.9$ will yield a somewhat imprecise line search.

The values $\rho = 0.1$ and $\sigma = 0.7$ give good results.

$$f(\boldsymbol{x}) = 0.7x_1^4 - 8x_1^2 + 6x_2^2 + \cos(x_1x_2) - 8x_1, \qquad -\pi \le x_1, x_2 \le \pi$$

$$\boldsymbol{x}^{(0)} = \begin{bmatrix} -\pi \\ \pi \end{bmatrix}, \boldsymbol{d}^{(0)} = \begin{bmatrix} 1.0 \\ -1.3 \end{bmatrix}$$



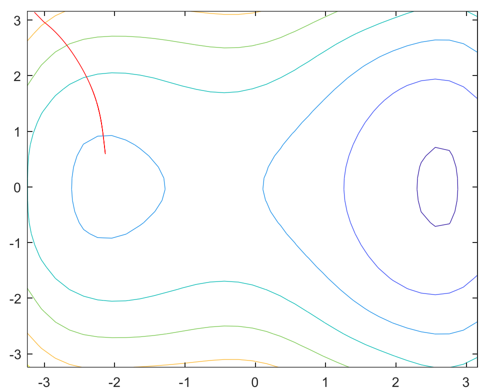
Exercise_4_11_WushengLU.m

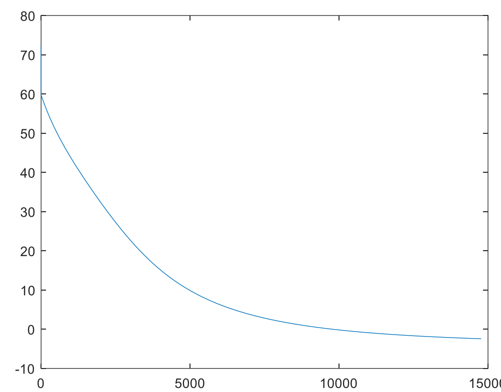
梯度下降法，搜索方向是负梯度方向
迭代次数14770

迭代产生的步长非常小

x_star =[-2.1327   0.5917]
f_star = -2.4395

[0, 4.8332].

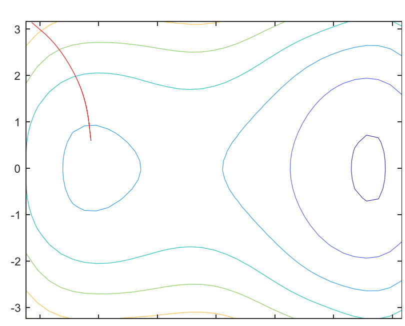


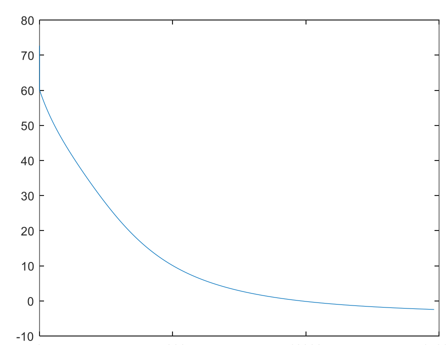


$$x^{(0)} = \begin{bmatrix} -\pi \\ \pi \end{bmatrix}, d^{(0)} = \begin{bmatrix} 1.0 \\ -1.1 \end{bmatrix}$$

x_star =[-2.1324 0.5920]
f_star = -2.4383

梯度下降法，搜索方向是负梯度方向
迭代次数14815





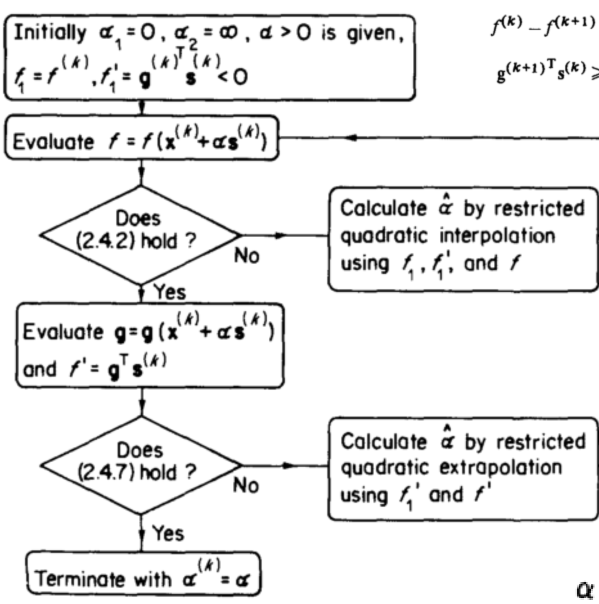
s [0, 5.7120].

作业

4-2

4-3

4-6



$$f^{(k)} - f^{(k+1)} \geqslant -\rho g^{(k)T} \delta^{(k)} \tag{2.4.2}$$

$$g^{(k+1)T} s^{(k)} \geqslant \sigma g^{(k)T} s^{(k)} \tag{2.4.7}$$

$$\hat{\alpha} - \alpha_1 = \tfrac{1}{2}(\alpha_2 - \alpha_1)/(1 + (f_1 - f_2)/((\alpha_2 - \alpha_1) f_1')).$$

$\alpha = c$

$$\hat{\alpha} - \alpha = (\alpha - \alpha_1) f'/(f_1' - f')$$

$\alpha = \min(1, -2\Delta f / f_1')$ 初值 $\Delta f > 0$