Numerical Optimization, 2020 Fall

Homework 7

Due on 14:59 NOV 26, 2020

请尽量使用提供的 tex 模板, 若手写作答请标清题号并拍照加入文档.

1 收敛速率

分别构造具有次线性，线性，超线性和二阶收敛速率的序列的例子。[10 pts]

1. 次线性: $\{\frac{1}{k} : k \in N^*\}$ 1° $\lim_{k\to\infty} \frac{1}{k} = 0$

2° $\lim_{k\to\infty} \frac{\frac{1}{k+1}-0}{\frac{1}{k}-0} = \lim_{k\to\infty} \frac{k}{k+1} = 1$

线性: $\{\frac{1}{2^k} : k \in N^*\}$ 1° $\lim_{k\to\infty} \frac{1}{2^k} = 0$

2° $\lim_{k\to\infty} \frac{\frac{1}{2^{k+1}}-0}{\frac{1}{2^k}-0} = \frac{1}{2}$

超线性: $\{\frac{1}{k!} : k \in N^*\}$ 1° $0 \le \frac{1}{k!} \le \frac{1}{k}$ $\forall k \in N^*$

$0 = \lim_{k\to\infty} 0 \le \lim_{k\to\infty} \frac{1}{k!} \le \lim_{k\to\infty} \frac{1}{k} = 0$

$\therefore \lim_{k\to\infty} \frac{1}{k!} = 0$

2° $\lim_{k\to\infty} \frac{\frac{1}{(k+1)!}-0}{\frac{1}{k!}-0} = \lim_{k\to\infty} \frac{1}{k+1} = 0$

二阶: $\{\frac{1}{2^{2^k}} : k \in N^*\}$ 1° $\lim_{k\to 0} \frac{1}{2^{2^k}} = 0$

2° $\lim_{k\to 0} \frac{2^{2^{(k+1)}}-0}{(2^{2^k}-0)^2} = \lim_{k\to\infty} \frac{2^{2^{k+1}}}{2^{(2^k+2^k)}} = \lim_{k\to\infty} \frac{2^{2^{k+1}}}{2^{2^{k+1}}} = 1$

2 梯度下降法的收敛性分析

考虑如下优化问题:

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad f(\boldsymbol{x}), \tag{1}$$

其中目标函数 f 满足一下性质:

- 对任意 $\boldsymbol{x}$, $f(\boldsymbol{x}) \geq \underline{f}$。
- ∇f 是 Lipschitz 连续的，即对于任意的 $\boldsymbol{x}, \boldsymbol{y}$，存在 $L > 0$ 使得

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leq L\|\boldsymbol{x} - \boldsymbol{y}\|_2.$$

若采用梯度下降法求解问题(1)，记所产生的迭代点序列为 $\{\boldsymbol{x}^k\}$。迭代点的更新为 $\boldsymbol{x}^{k+1} \leftarrow \boldsymbol{x}^k + \alpha^k \boldsymbol{d}^k$。试证明以下问题。

(i) 在一点 $\boldsymbol{x}^k$ 处给定一个下降方向 $\boldsymbol{d}^k$，即 $\boldsymbol{d}^k$ 满足 $\langle \nabla f(\boldsymbol{x}^k), \boldsymbol{d}^k \rangle < 0$。试证明：对于充分小的 $\alpha > 0$，有 $f(\boldsymbol{x} + \alpha \boldsymbol{d}^k) < f(\boldsymbol{x}^k)$ 成立。[10 pts]

(ii) 假设存在 $\delta > 0$ 使得 $-\frac{\langle \nabla f(\boldsymbol{x}^k), \boldsymbol{d}^k \rangle}{\|\nabla f(\boldsymbol{x}^k)\|_2 \|\boldsymbol{d}^k\|_2} > \delta$。证明回溯线搜索会有限步终止，并给出对应步长 α^k 的下界。[10 pts]

(iii) 根据上一问结果证明 $\lim_{k\to\infty} \|\nabla f(\boldsymbol{x}^k)\|_2 = \boldsymbol{0}$。[10 pts]

(iv) 令 $\boldsymbol{d}^k = -\nabla f(\boldsymbol{x}^k)$，采用固定步长 $\alpha^k \equiv \alpha = \frac{1}{L}$。试证明该设定下梯度下降法的全局收敛性。[20 pts]

2. (1) $f(x+\alpha d^k) = f(x^k) + \alpha \nabla f(x^k)^T d_k + \frac{\alpha^2}{2} d_k^T H(\xi) d_k$, ξ between x and x^k

1° $d_k^T H(\xi) d_k \leq 0$: $\forall \alpha \in (0, +\infty)$ $\alpha \nabla f(x^k)^T d_k + \frac{\alpha^2}{2} d_k^T H(\xi) d_k < \alpha \nabla f(x^k)^T d_k < 0$

2° $d_k^T H(\xi) d_k > 0$: $\forall \alpha \in (0, \frac{-2\nabla f(x^k)^T d_k}{d_k^T H(\xi) d_k})$ $\alpha \nabla f(x^k)^T d_k + \frac{\alpha^2}{2} d_k^T H(\xi) d_k < \alpha \nabla f(x^k) d_k - \alpha \nabla f(x^k) d_k = 0$

$\therefore \forall \alpha \in (0, \min\{1, \frac{-2\nabla f(x^k)^T d_k}{d_k^T H(\xi) d_k}\})$, $f(x+\alpha d^k) = f(x^k) + \alpha \nabla f(x^k)^T d_k + \frac{\alpha^2}{2} d_k^T H(\xi) d_k < f(x^k)$, ξ between x and x^k

[if $d_k^T H(\xi) d_k = 0$, we define $\frac{-2\nabla f(x^k)^T d_k}{d_k^T H(\xi) d_k} = +\infty$]

(2) $\because \exists L > 0$, $\forall x, y \in \mathbb{R}^n$: $\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x-y\|_2$ $\qquad \because -\frac{\langle \nabla f(x^k), d^k \rangle}{\|\nabla f(x^k)\|_2 \|d^k\|_2} = \delta > 0$

$\Updownarrow$

$\forall x \in \mathbb{R}^n$, $\exists L > 0$: $\|H(x)\| \leq L$ $\qquad \therefore \nabla f(x^k)^T d^k < 0$

取 L = Lipschitz constant

$f(x^k + \alpha^k d^k) = f(x^k) + \alpha^k \nabla f(x^k)^T d^k + \frac{\alpha^{k2}}{2} d^{kT} H(x^k) d^k < f(x^k) + c_1 \alpha^k \nabla f(x^k)^T d^k$

$< f(x^k) + \alpha^k \nabla f(x^k)^T d^k + \frac{\alpha^{k2}}{2} \cdot L\|d^k\|^2$

令 $f(x^k) + \alpha^k \nabla f(x^k)^T d^k + \frac{\alpha^{k2}}{2} \cdot L\|d^k\|^2 < f(x^k) + c_1 \alpha^k \nabla f(x^k)^T d^k$

$\alpha^k < \frac{2(1-c_1)}{L}$

$\therefore$ 当 $\alpha^k \in (0, \frac{2(1-c_1)}{L})$, $f(x^k + \alpha^k d^k) < f(x^k) + c_1 \alpha^k \nabla f(x^k)^T d^k$ ①

而回溯搜索为选定 $\{\gamma^0, \gamma^1, \cdots\}$, $\gamma \in (0,1)$ 中最大值使得 $x^{k+1} = x^k + \gamma^t d^k$ $t \in N$

满足 $f(x^{k+1}) < f(x^k) + c_1 \alpha^k \nabla f(x^k)^T d^k$ 成立

$\therefore \frac{2(1-c_1)}{L} \cdot \gamma \in (0, \frac{2(1-c_1)}{L})$ 为步 α^k 的一个下界

(3). 令 $d_k = -\nabla f(x^k)$ 满足 $-\|\nabla f(x^k)\|^2 < 0$

则 $\exists K > 0, \forall k > K, k \in N^*$: $f(x^{k+1}) < f(x^k) - c_1 \cdot \frac{2(1-c_1)\gamma}{L} \|\nabla f(x^k)\|^2$

记第一个满足该不等式成立的 k 为 K

$$f(x^{K+2}) < f(x^{K+1}) - c_1 \cdot \frac{2(1-c_1)\gamma}{L} \|\nabla f(x^{K+1})\|^2$$

$$c_1 \cdot \frac{2(1-c_1)\gamma}{L} \sum_{i=K}^{\infty} \|\nabla f(x^i)\|^2 < \sum_{i=K}^{\infty} f(x^i) - f(x^{i+1})$$

$$= f(x^K) - \lim_{i \to \infty} f(x^{i+1})$$

$$\because \forall x \in \mathbb{R}: f(x) \geq \underline{f}$$

$$\therefore < f(x^K) - \underline{f}$$

$$< +\infty$$

$$\therefore \sum_{i=K}^{\infty} \|\nabla f(x^i)\|^2 < +\infty$$

$$\therefore \lim_{k \to \infty} \|\nabla f(x^k)\| = 0$$

(4) $x^{k+1} = x^k + \alpha^k d^k = x^k - \frac{1}{L}\nabla f(x^k) \quad \forall k \in N$

$$f(x^{k+1}) = f(x^k - \frac{1}{L}\nabla f(x^k)) = f(x^k) - \frac{1}{L}\|\nabla f(x^k)\|^2 + \frac{1}{2L^2}\nabla f(x^k)^T H(x^k) \nabla f(x^k)$$

$$< f(x^k) - \frac{1}{L}\|\nabla f(x^k)\|^2 + \frac{1}{2L}\|\nabla f(x^k)\|^2$$

$$= f(x^k) - \frac{1}{2L}\|\nabla f(x^k)\|^2$$

$$\frac{1}{2L}\|\nabla f(x^k)\|^2 < f(x^k) - f(x^{k+1})$$

$$\therefore \frac{1}{2L}\sum_{k=0}^{\infty}\|\nabla f(x^k)\|^2 < \sum_{k=0}^{\infty}[f(x^k) - f(x^{k+1})] = f(x^0) - \lim_{k \to \infty} f(x^k) \leq f(x^0) - \underline{f} < +\infty$$

$$\therefore \sum_{k=0}^{\infty}\|\nabla f(x^k)\|^2 < +\infty$$

$$\lim_{k \to +\infty}\|\nabla f(x^k)\| = 0$$

$\therefore$ 偷设定下梯度下降法全局收敛

3 编程题

考虑求解如下优化问题：

$$\min_{x_1, x_2} \quad 100(x_2 - x_1^2)^2 + (1 - x_1)^2. \tag{2}$$

分别用**梯度下降法**和**牛顿法**结合 Armijo 回溯搜索编程求解该问题。分别考虑用 $\boldsymbol{x}^0 = [1.2, 1.2]^T$ 和 $\boldsymbol{x}^0 = [-1.2, 1]^T$(较困难) 作为初始点启动算法。

要求: 对于两种初始点，分别画出两种算法步长 α^k 和 $\|\nabla f(\boldsymbol{x}^k)\|_\infty$ 随迭代步数 k 变化的曲线。(编程可使用 matlab 或 python 完成，请将代码截图贴在该文档中。) [40pts]

(Hint: 步长初始值 $\alpha_0 = 1$, 参数 c_1 可选为 10^{-4}，终止条件为 $\|\nabla f(\boldsymbol{x}^k)\|_\infty \leq 10^{-4}$.) 1. 梯度下降法：

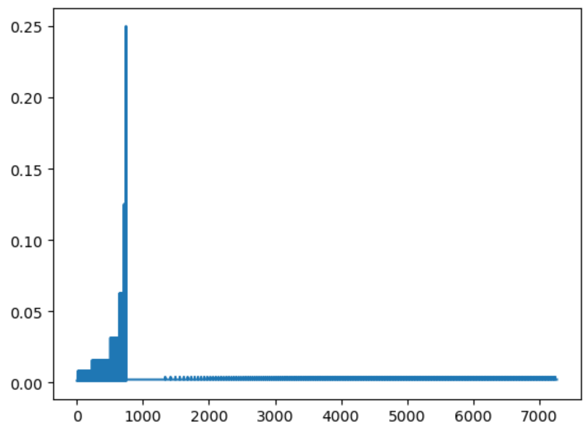
代码:

```python
from sympy import *
import numpy as np
import matplotlib.pyplot as plt
import mpl_toolkits.axisartist as axisartist\

def function(x1,x2):
    return 100*(x2-x1**2)**2+(1-x1)**2

def gradient(x1,x2):
    return 200*(x2-x1**2)*(-2*x1)+2*(1-x1)*(-1),200*(x2-x1**2)

def Armijobtls(x,y):
    alpha=1
    c1=10**(-4)
    theta=0.5
    x1,x2=x,y
    xk1=x1-alpha*gradient(x1,x2)[0]
    xk2=x2-alpha*gradient(x1,x2)[1]
    while function(xk1,xk2)>function(x1,x2)-c1*alpha*(gradient(x1,x2)[0]**2+gradient(x1, x2)[1]**2):
        alpha*=theta
        xk1=x1-alpha*gradient(x1,x2)[0]
        xk2=x2-alpha*gradient(x1,x2)[1]
    return alpha

def gradient_descent_method(x,y):
    Alpha=[]
    Gradient=[]
    x1,x2=x,y
    grad=gradient(x1,x2)
    c=max(abs(grad[0]),abs(grad[1]))
    while c>10**(-4):
        alpha=Armijobtls(x1,x2)
        Alpha.append(alpha)
        Gradient.append(c)
        x1-=alpha*grad[0]
        x2-=alpha*grad[1]
        grad=gradient(x1,x2)
        c=max(abs(grad[0]),abs(grad[1]))
    return (x1,x2),Alpha,Gradient
c,Alpha,Gradient=gradient_descent_method(-1.2,1)
n=[]
for i in range(len(Alpha)):
    n.append(i+1)
#plt.plot(n,Alpha)
plt.plot(n,Gradient)
plt.show()
```

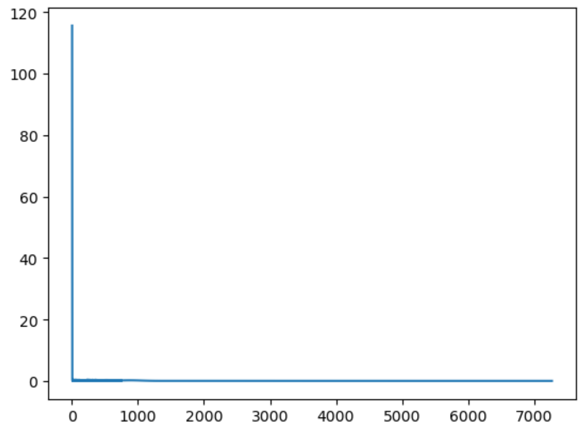
初始点为 (1.2,1.2)

求得解为:(1.0, 0.9999997501898542)

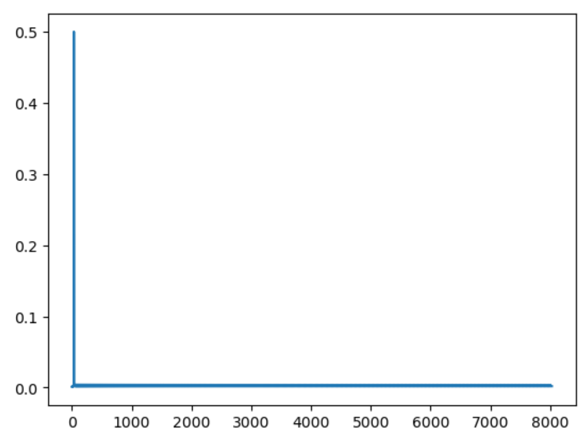
1.1 Alpha:



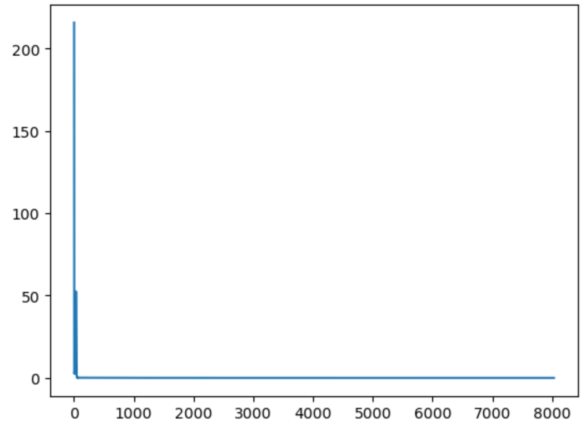
1.1 Gradient:



初始点为 (-1.2,1)
求得解为 (1.0, 0.9999997500407729)
1.2 Alpha:



1.2 Gradient:



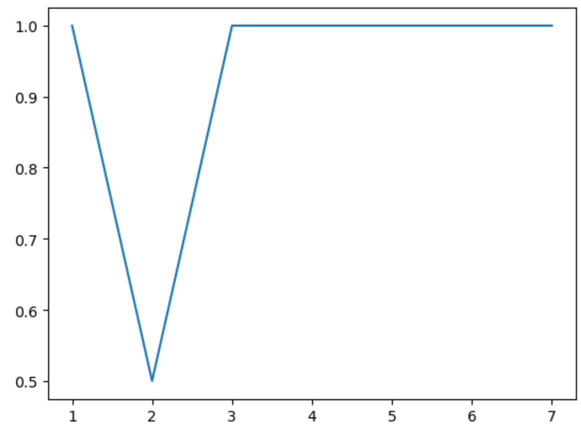
2. 牛顿法: 代码:

```python
from sympy import *
import numpy as np
import matplotlib.pyplot as plt
import mpl_toolkits.axisartist as axisartist\

def function(x1,x2):
    return 100*(x2-x1**2)**2+(1-x1)**2

def gradient(x1,x2):
    return 200*(x2-x1**2)*(-2*x1)+2*(1-x1)*(-1),200*(x2-x1**2)

def hessian(x1,x2):
    return [[400*(3*x1**2-x2)+2,-400*x1],[-400*x1,200]]

def Armijobtls(x,y):
    alpha=1
    c1=10**(-4)
    theta=0.5
    x1,x2=x,y
    grad=gradient(x1,x2)
    hessi=np.mat(hessian(x1,x2)).I.tolist()
    xk1=x1-alpha*(hessi[0][0]*grad[0]+hessi[0][1]*grad[1])
    xk2=x2-alpha*(hessi[1][0]*grad[0]+hessi[1][1]*grad[1])
    while function(xk1,xk2)>function(x1,x2)-c1*alpha*((hessi[0][0]*grad[0]+hessi[0][1]*grad[1])*grad[0]+(hessi[1][0]*grad[0]+hessi[1][1]*grad[1])*grad[1]):
        alpha*=theta
        xk1=x1-alpha*(hessi[0][0]*grad[0]+hessi[0][1]*grad[1])
        xk2=x2-alpha*(hessi[1][0]*grad[0]+hessi[1][1]*grad[1])
    return alpha

def Newton_method(x,y):
    Alpha=[]
    Gradient=[]
    x1,x2=x,y
    grad=gradient(x1,x2)
    hessi=np.mat(hessian(x1,x2)).I.tolist()
    c=max(abs(grad[0]),abs(grad[1]))
    while c>10**(-4):
        alpha=Armijobtls(x1,x2)
        Alpha.append(alpha)
        Gradient.append(c)
        x1-=alpha*(hessi[0][0]*grad[0]+hessi[0][1]*grad[1])
        x2-=alpha*(hessi[1][0]*grad[0]+hessi[1][1]*grad[1])
        grad=gradient(x1,x2)
        c=max(abs(grad[0]),abs(grad[1]))
        hessi=np.mat(hessian(x1,x2)).I.tolist()
    return (x1,x2),Alpha,Gradient
c,Alpha,Gradient=Newton_method(-1.2,1)
n=[]
for i in range(len(Alpha)):
    n.append(i+1)
#plt.plot(n,Alpha)
plt.plot(n,Gradient)
plt.show()
```

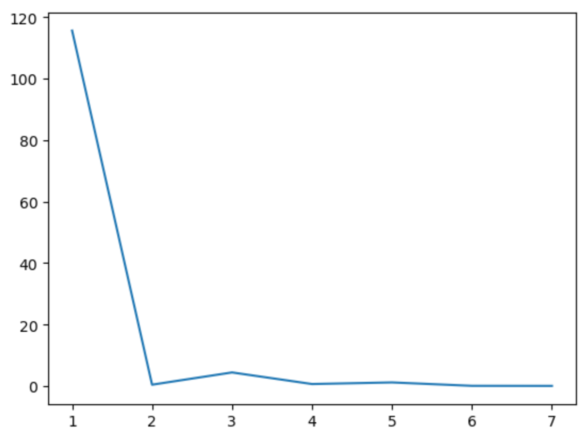
初始点为 (1.2,1.2)

求得解为:(1.0000001778592191, 1.0000003532027988)

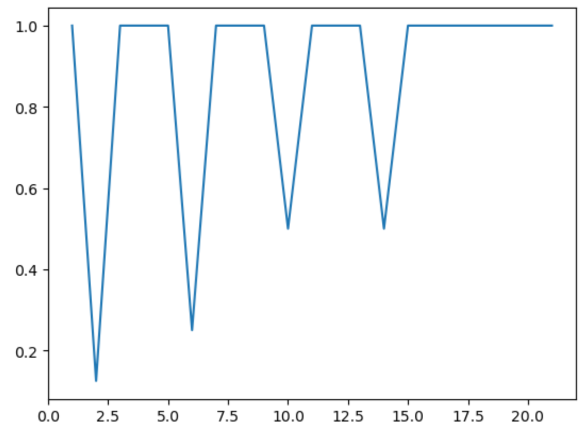
2.1 Alpha:



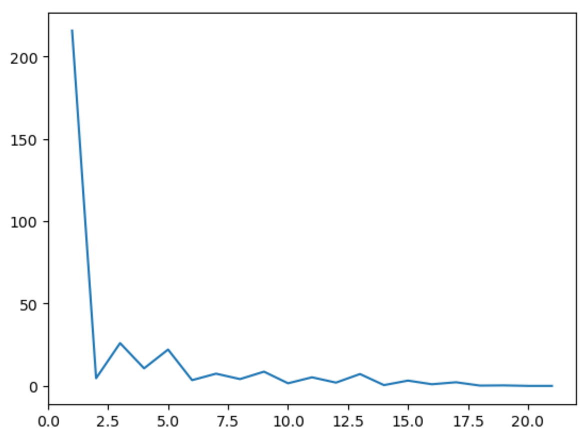
2.1 Gradient:



初始点为 (-1.2,1)
求得解为:(0.9999999999400667, 0.9999999998789006)
2.2 Alpha:


1.0
0.8
0.6
0.4
0.2
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

2.2 Gradient:


200
150
100
50
0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0