最优化homework4

姓名	学号
崔璨明	20337025

题目一：使用 KKT 条件，推导出下面问题的解

$$max\ xyz$$

$$s.t.\ x^2+y^2=1, x+z=1$$

$$\begin{aligned} &\max\ xyz \\ &s.t.\ x^2+y^2=1,\ x+z=1 \end{aligned} \iff \begin{aligned} &\min\ f(x,y,z)=-xyz \\ &s.t.\ h_1(x,y,z)=x^2+y^2-1=0 \\ &\qquad h_2(x,y,z)=x+z-1=0 \end{aligned}$$

解：$L(x,y,z)=f(x,y,z)+v_1h_1(x,y,z)+v_2h_2(x,y,z)$

$$=-xyz+v_1(x^2+y^2-1)+v_2(x+z-1)$$

由KKT条件，有：

$$\begin{cases} -yz+2v_1x+v_2=0 & \cdots ① \\ -xz+2v_1y=0 & \cdots ② \\ -xy+v_2=0 & \cdots ③ \\ x^2+y^2-1=0 & \cdots ④ \\ x+z-1=0 & \cdots ⑤ \end{cases}$$

联立，消去 z, v_1, v_2，有：

$$xy - y + \frac{x^2(1-x)}{y} + xy = 0$$

显然 $y \neq 0$（若 $y=0$，则 $xyz=0$，明显不是最大值）

∴ $xy^2 - y^2 + x^2 - x^3 + xy^2 = 0$

代入④，得：$-3x^3 + 2x^2 + 2x - 1 = 0$

即 $(x-1)(3x^2+x-1) = 0$

$y \neq 0$，$x-1 \neq 0$ ∴ $3x^2 + x - 1 = 0$

解得 $\begin{cases} x_1 = \frac{\sqrt{13}-1}{6} \\ x_2 = \frac{-\sqrt{13}-1}{6} \end{cases}$

(1) 当 $x = x_1 = \frac{\sqrt{13}-1}{6}$ 时，$z = 1-x = \frac{7-\sqrt{13}}{6}$，$y^2 = \frac{22+2\sqrt{13}}{36}$

∵ $x>0$ 且 $z>0$，要 $\min -xyz$，则 $y>0 \Rightarrow y = \frac{\sqrt{22+2\sqrt{13}}}{36}$

∴ $xyz_1 = \frac{(\sqrt{13}-1)(7-\sqrt{13})(\sqrt{22+2\sqrt{13}})}{216}$

(2) 当 $x = x_2 = \frac{-\sqrt{13}-1}{6}$ 时，$z = 1-x = \frac{7+\sqrt{13}}{6} > 0$，$y^2 = 1-x^2 = \frac{22-2\sqrt{13}}{36}$

应有 $y<0$，故 $y = \frac{-\sqrt{22+2\sqrt{13}}}{6}$ ∴ $xyz_2 = \frac{(\sqrt{13}+1)(7+\sqrt{13})(\sqrt{22+2\sqrt{13}})}{216}$

$xyz_2 > xyz_1$

∴ 该问题解为 $\begin{cases} x = \frac{-\sqrt{13}-1}{6} \\ y = -\frac{\sqrt{22+2\sqrt{13}}}{6} \\ z = \frac{7+\sqrt{13}}{6} \end{cases}$

题目二：写代码实现 Interior-point methods，找下面约束优化问题的解

$$min\ 2x_1^2 + 2x_1x_2 + x_2^2 - 10x_1 - 10x_2$$

$$s.t.\ x^2 + y^2 <= 1$$

利用障碍函数内点法来寻找该问题的解，将该问题转化为如下子问题：

$$t(2x_1^2 + 2x_1x_2 + x_2^2 - 10x_1 - 10x_2) - log(1 - x^2 - y^2)$$

在 t 的每一步迭代中用拉格朗日乘子法对子问题进行求解。

编写的代码如下：

```
import numpy as np
import time

#计算梯度
def grad(t, x1, x2):
    return np.array([[t * (4*x1 + 2*x2 - 10) + (2*x1)/(1 - pow(x1,2) - pow(x2,2)) ],
                     [t * (2*x2 + 2*x1 - 10) + (2*x2)/(1 - pow(x1,2) - pow(x2,2)) ]])

#计算Hessian矩阵
def Hessian(t, x1, x2):
    return np.array([[4*t + ( 2*(1-pow(x1,2)-pow(x2,2)) + 4*pow(x1,2) ) / pow((1-
pow(x1,2)-pow(x2,2)), 2) , 2*t + (4*x1*x2) / pow((1-pow(x1,2)-pow(x2,2)), 2)],
            [2*t + (4*x1*x2) / pow((1-pow(x1,2)-pow(x2,2)), 2) , 2*t + (2*(1-pow(x1,2)-
pow(x2,2)) + 4*pow(x2,2) ) / pow((1-pow(x1,2)-pow(x2,2)), 2) ]])

def Newton_Raphson(t, x1, x2):
    gf = grad(t, x1, x2)
    Hf = Hessian(t, x1, x2)
    Hf_inv = np.linalg.inv(Hf)
    deltaX = 0.1 * np.matmul(Hf_inv, gf)
    res = np.linalg.norm(deltaX, 2)
    return x1 - deltaX[0, 0], x2 - deltaX[1, 0], res

if __name__ == "__main__":
    time_start = time.time()
    t = 2
    x1 = 0.3
    x2 = 0.4
    while True:
        while True:
            x1, x2, res = Newton_Raphson(t, x1, x2)
            if res < 0.0001:
                break

        if 3.0 / t < 0.0001:
            time_end = time.time()
            print('Result:\ncosting time:', time_end - time_start)
            print("t:{}\nx1:{}\nx2:{}".format(t, x1, x2))
            break
        t = 2 * t
    print("mini value:{}".format(2*pow(x1,2)+2*x1*x2+pow(x2,2)-10*x1-10*x2))
```

程序运行结果如下，最终的求解结果为$min f(x_1,x_2)\approx -11.690, x_1\approx 0.6355, x2\approx 0.7721$:

```
PS F:\VSCODE> & E:/python/python.exe f:/VSCODE/py/temp.py
Result:
costing time: 0.012992858886871875
t:32768
x1:0.6354757808686078
x2:0.7721080589612498
mini value:-11.690716664077843
PS F:\VSCODE>
```