HW 6

(1). Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$.

Show exactly one of following Condition is TRUE:

(i). $\exists\, x \in \mathbb{R}^n$ that $Ax \leq b$ and $x \geq 0$.

(ii) $\exists\, y \in \mathbb{R}^m$ that $A^T y \geq 0$, $y \geq 0$, and $b^T y < 0$

Define $B = [A \mid I] \in \mathbb{R}^{m \times (m+n)}$, Let $\alpha = \begin{bmatrix} x \\ y \end{bmatrix}$

such that $K = \{B\alpha\}$ where $\alpha \in \mathbb{R}^{m+n}$, $\alpha \geq 0$

By Farkas's Lemma, for $b \in \mathbb{R}^m$,

Either $b \in K$:

$\exists\, x, y, \alpha = \begin{bmatrix} x \\ y \end{bmatrix}$, $B\alpha = b$,

$$Ax + y = b, \quad x, y \geq 0$$

so $Ax \leq b$, $x \geq 0$, (i) satisfies.

Or $\exists\, d$ such that $b^T d < 0$, $B^T d \geq 0$,

which implies $\begin{bmatrix} A^T d \\ d \end{bmatrix} \geq 0$ so $A^T d \geq 0$ and $d \geq 0$

Let $y = d$, (ii) satisfies. $\square$

2. Exercise 12.28

For $\min f(x,y) = (x-2)^2 + (y-2)^2$

$$\text{s.t} \quad (x-1)^2 = 5y$$

(a) Find all KKT points,

aim to find all (x,y), λ such that

$$\begin{cases} \mathcal{L}((x,y),\lambda) = (x-1)^2 + (y-2)^2 - \lambda((x-1)^2 - 5y) \\ ① \ \nabla \mathcal{L} = \begin{pmatrix} 2(x-1) - 2\lambda(x-1) \\ 2(y-2) + 5\lambda \end{pmatrix} = 0 \\ ② \ (x-1)^2 + \cancel{(y-2)^2} = 0 \\ \quad \cancel{x-1>y} \ -5y \\ ③ \ \lambda((x-1)^2 - 5y) = 0 \end{cases}$$

From ① ② we get that the only possible solution is $\begin{cases} x = 1 \\ y = 0 \\ \lambda = \frac{4}{5} \end{cases}$

In this case, $(1,0)$ is the KKT-point.

Since $\nabla((x-1)^2-5y) = \begin{pmatrix} 2(x-1) \\ -5 \end{pmatrix}$

when $x=1, y=0$, it is $\begin{pmatrix} 0 \\ -5 \end{pmatrix}$, which is linearly independent, so $(1,0)$ satisfies LICQ condition.

(b). which of the point is solution.

Note that given any minimizer x^*, ~~$\nabla_x \mathcal{L}(x^*)$~~

Let $c(x^*) = (x-1)^2-5y$, ~~$x^* = (x_1, x_2)$~~

$\nabla c(x^*) = \begin{pmatrix} 2x-2 \\ -5 \end{pmatrix}$, which must be LICQ.

So x^* must be one KKT point,

so it could only be $(1,0)$.

Now show $(1,0)$ is one solution.

Since $\lambda = \frac{4}{5}$, $\nabla_x^2 \mathcal{L} = \begin{pmatrix} 2-2\lambda & 0 \\ 0 & 2 \end{pmatrix}$

$\nabla_x^2 \mathcal{L}((1,0), \lambda) = \begin{pmatrix} \frac{2}{5} & 0 \\ 0 & 2 \end{pmatrix}$, which is positive definite, so by 2nd order sufficient condition, $(1,0)$ is one strict solution.

Thus all possible solution is only $(1,0)$.

(c). substitute $y = \frac{1}{5}(x-1)^2$ to $f(x,y)$,

so that ~~$f(x,y) = (x-1)^2 + \ldots$~~

$$f(x,y) = 5y + (y-2)^2$$

try to solve $\min_{y} f(x,y)$

let $g(y) = f(x,y)$,

$$g(y) = y^2 + y + 4,$$

$$g'(y) = 2y + 1$$

$$g''(y) = 2 > 0$$

So $y = -\frac{1}{2}$ is one minimizer

then x can not be one real number.

So the solution is different from the original $(0,2)$.

3. Exercise 13.4

Construct one Lp with its dual Lp' such that they are all infeasible.

Define Lp:

$$\begin{aligned} \min \quad & x+2y \\ \text{s.t} \quad & x+y=1 \\ & x+y=3 \end{aligned}$$

its Lp:

$$\begin{aligned} \max \quad & z_1+3z_2, \\ \text{s.t} \quad & z_1+z_2=1 \\ & z_1+z_2=2 \end{aligned}$$

Clearly both of them have no solutions.

Problem 4

Exercise 17.3

Use Quatratic penalty method to calculate the constrained minimized problem:

$$ \min x_1 + x_2\\ \qquad \qquad s.t \ x_1^2 + x_2^2 -2=0 $$

The penalty equation is defined like:

$$Q(x;\mu) = x_1 + x_2 + \frac{\mu}{2}\left(x_1^2 + x_2^2 - 2\right)^2$$

```
import numpy as np

def norm(x):
    return np.linalg.norm(x)

def Q(x,mu):
    x_1 = x[0]
    x_2 = x[1]
    return x_1 + x_2 + (mu/2)*(x_1**2 + x_2**2 -2)**2

def gradient(x,mu):
    x_1 = x[0]
    x_2 = x[1]
    y_1 = 1 + 2*mu*x_1*(x_1**2 + x_2**2 -2)
    y_2 = 1 + 2*mu*x_2*(x_1**2 + x_2**2 -2)
    return np.array([y_1, y_2])


def step_length(x,p,mu):
    alpha = 1
    rho = 0.8
    c_1 = 0.6
    while(Q(x + alpha*p,mu) > (Q(x,mu) +
c_1*alpha*np.dot(gradient(x,mu),p))):
        alpha = rho*alpha
    return alpha



def LineSearch(x_0,mu,tao):



    x = x_0
    while(norm(gradient(x,mu)) > tao):
```

```
        p = (-1)*gradient(x,mu)
        alpha = step_length(x,p,mu)
        x = x + alpha*p
    return x

def Quatratic_Penalty(mu_list):
    x = np.zeros(2)
    for mu in mu_list:
        tao = 1/mu
        x = LineSearch(x,mu,tao)
        print(f"{x} is the aprroximated minimizer for Q when mu is
{mu}")


mu_list = [1,10,100,1000]
Quatratic_Penalty(mu_list)
```

```
[-1.08589935 -1.08589935] is the aprroximated minimizer for Q when mu
is 1
[-1.01272563 -1.01272563] is the aprroximated minimizer for Q when mu
is 10
[-1.0012502 -1.0012502] is the aprroximated minimizer for Q when mu is
100
[-1.00012502 -1.00012502] is the aprroximated minimizer for Q when mu
is 1000
```