MAE 598 HW2

Sarasij Banerjee

1) $f(x_1, x_2) = 2x_1^2 - 4x_1x_2 + 1.5\,x_2^2 + x_2$

$\therefore \nabla f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 4x_1 - 4x_2 \\ -4x_1 + 3x_2 + 1 \end{bmatrix}$

for stationary point

$\left.\begin{matrix} 4x_1 - 4x_2 = 0 \\ -4x_1 + 3x_2 + 1 = 0 \end{matrix}\right\} \Rightarrow x_1 = x_2 = 1$

$\therefore$ stationary point $= (1, 1)$

$H(x_1, x_2) = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{bmatrix} = \begin{bmatrix} 4 & -4 \\ -4 & 3 \end{bmatrix}$

$|H| = 12 - 16 = -4$

$|H| < 0$ $\therefore$ product of eigenvalues < 0

$\therefore$ some are positive, while some are -ve

$\therefore$ saddle point (neither max, nor min)

Using Taylor's Theorem

$f(x_1, x_2) = f(1,1) + \cancel{\nabla f^T_{(1,1)}} \begin{bmatrix} x_1 - 1 \\ x_2 - 1 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} x_1 - 1 \\ x_2 - 1 \end{bmatrix}^T \begin{bmatrix} 4 & -4 \\ -4 & 3 \end{bmatrix} \begin{bmatrix} x_1 - 1 \\ x_2 - 1 \end{bmatrix}$ ($\nabla f_{(1,1)} \to 0$)

$\therefore f(x_1, x_2) - f(1,1) = \frac{1}{2} \begin{bmatrix} (x_1 - 1) & (x_2 - 1) \end{bmatrix} \begin{bmatrix} 4 & -4 \\ -4 & 3 \end{bmatrix} \begin{bmatrix} x_1 - 1 \\ x_2 - 1 \end{bmatrix}$

$= \frac{1}{2} \begin{bmatrix} (x_1 - 1) & (x_2 - 1) \end{bmatrix} \begin{bmatrix} 4(x_1 - 1) - 4(x_2 - 1) \\ -4(x_1 - 1) + 3(x_2 - 1) \end{bmatrix}$

$= \frac{1}{2} \begin{bmatrix} \partial x_1 & \partial x_2 \end{bmatrix} \begin{bmatrix} 4\partial x_1 - 4\partial x_2 \\ -4\partial x_1 + 3\partial x_2 \end{bmatrix}$

$f(x_1, x_2) - f(1,1) < 0$ for direction of downslope

$$\Rightarrow \quad [\partial x_1 \quad \partial x_2] \begin{bmatrix} 4\partial x_1 - 4\partial x_2 \\ -4\partial x_1 + 3\partial x_2 \end{bmatrix} < 0$$

$$\Rightarrow \quad \partial x_1(4\partial x_1 - 4\partial x_2) + \partial x_2(-4\partial x_1 + 3\partial x_2) < 0$$

$$\Rightarrow \quad 4\partial x_1^2 - 4\partial x_1\partial x_2 - 4\partial x_1\partial x_2 + 3\partial x_2^2 < 0$$

$$\Rightarrow \quad 4\partial x_1^2 - 8\partial x_1\partial x_2 + 3\partial x_2^2 < 0$$

$$\Rightarrow \quad \cancel{2(2\partial x_1} \quad 4\partial x_1^2 - 6\partial x_1\partial x_2 - 2\partial x_1\partial x_2 + 3\partial x_2^2$$

$$\Rightarrow \quad 2\partial x_1(2\partial x_1 - 3\partial x_2) - \partial x_2(2\partial x_1 - 3\partial x_2) < 0$$

$$\Rightarrow \quad (2\partial x_1 - \partial x_2)(2\partial x_1 - 3\partial x_2) < 0$$

$$\{a, b, c, d\} = \{2, 1, 2, 3\}$$

or

$$\{2, 3, 2, 1\}$$

or

$$\{1, 2, 3, 2\}$$

or

$$\{3, 2, 1, 2\}$$

2(a) find a point (x_1, x_2, x_3) in plane $x_1+2x_2+3x_3=1$ that is nearest to the point $(-1,0,1)^T$

This can be reformulated as

$$\min_{(x_1, x_2, x_3)} f(x) = \|x_1+1\|^2 + \|x_2\|^2 + \|x_3-1\|^2 \quad -(1)$$

subject to $x_1+2x_2+3x_3=1 \quad -(2)$

Substituting the value of x_1 from (2) into (1)

$$\min f(x) = (2-2x_2-3x_3)^2 + x_2^2 + (x_3-1)^2$$

Now, $\nabla f(x) = \begin{bmatrix} 10x_2+12x_3-8 \\ 12x_2+20x_3-14 \end{bmatrix}$

$\therefore$ Stationary point $\Rightarrow \nabla f = 0$

$\therefore$ Solving the 2 eqns:

$$\begin{bmatrix} x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -\frac{1}{7} \\ \frac{11}{14} \end{bmatrix}$$

$\therefore$ from (2) $x_1 = -\frac{15}{14}$

$$H = \begin{bmatrix} 10 & 12 \\ 12 & 20 \end{bmatrix}$$

$\lambda_i(H) = 2, 28$

$\therefore$ positive definite

$\therefore$ Convex

Gradient Descent

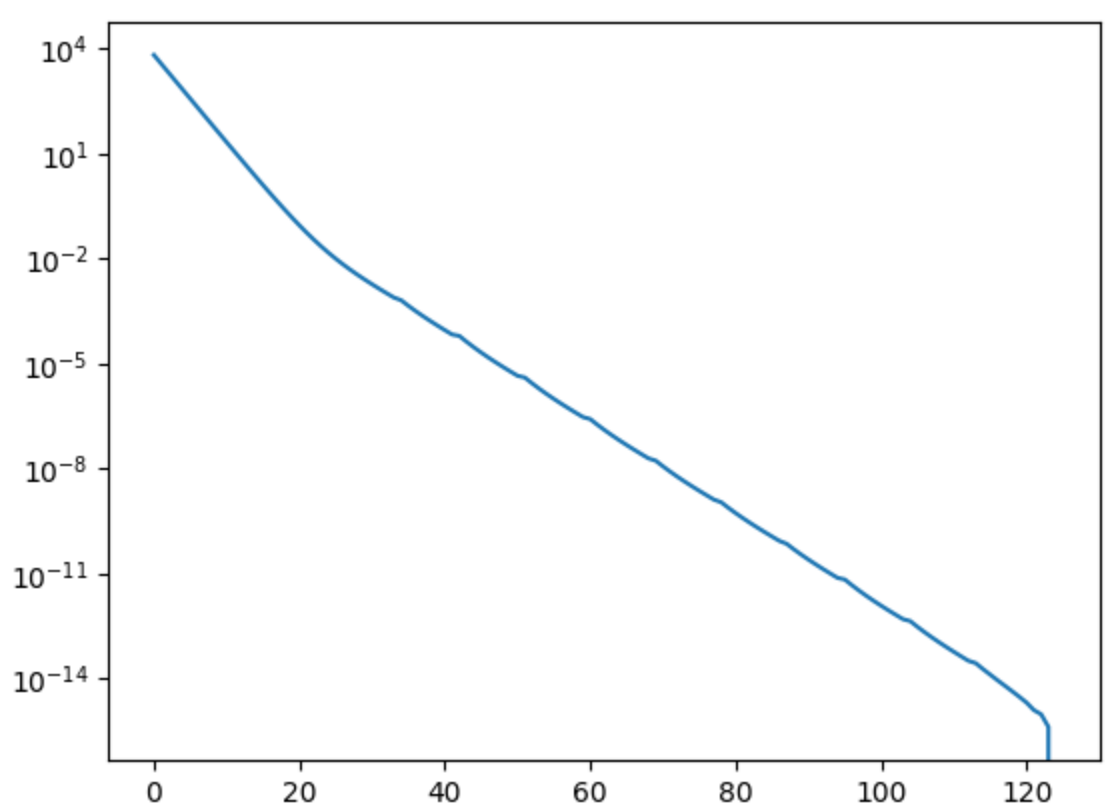
In [ ]:
```
import numpy as np

def objective(x):
    return (2 - 2*x[0] - 3*x[1])**2 + (x[0])**2 + (x[1] - 1)**2  # Objective function for the unconstrained problem
def gradient(x):
    return np.array([ 10*x[0]+12*x[1]-8, 12*x[0]+20*x[1]-14])
def hessian(x):
    return np.array([[10,12],[12,20]])
```

In [ ]:
```
eps = 1e-7  # tolerance
x0 = [10 , 20]  # initial value

k_max=50
t=0.1
x_val=[]
obj_val=[]
grad_val=[]
```

In [ ]:
```
def phi(x,alpha):
   return objective(x) - (t * np.matmul(gradient(x).T,gradient(x)) * alpha)

def alpha_update(x):
    alpha = 1
    k = 0
    while objective(x - alpha*gradient(x)) > phi(x,alpha) and k<k_max: # Gradient descent algorithm
        alpha = 0.5 * alpha
        k=k+1
    return alpha
#

while np.linalg.norm(gradient(x0))>eps :
    x_val.append(x0)
    grad_val.append(np.linalg.norm(gradient(x0)))
    obj_val.append(objective(x0))

    alpha=alpha_update(x0)
    x0=x0-alpha*gradient(x0)
```

Plot for convergence

In [ ]:
```
import matplotlib.pyplot as plt
plt.plot(np.arange(len(grad_val)),np.abs(obj_val-obj_val[-1]))
plt.yscale('log')
```



In [ ]:
```
x0
```

Out[ ]:
```
array([-0.14285711,  0.78571426])
```

Newton's Method

In [ ]:
```
eps = 1e-7  # tolerance
x0 = np.array([10 , 20])
x1=np.array([0 , 0])
k=0
obj_value=[]
grad_value=[]
```

In [ ]:
```
while np.linalg.norm(x1-x0)>eps and k<k_max:
    x1=x0-np.matmul(np.linalg.inv(hessian(x0)),gradient(x0))
    grad_value.append(np.linalg.norm(gradient(x0)))
    obj_value.append(objective(x1))
    k+=1
```

In [ ]:
```
x1
```

Out[ ]:
```
array([-0.14285714,  0.78571429])
```

Despite varying the initial conditions, the newton's method converges in a single iteration due to its quadratic convergence rate and therefore does not generate a convergence plot. However the Gradient descent method takes multiple iterations as shown from the plot above. This is due to the linear convergence, and varies on the initial condition, and error tolerance, but does not depend on the value of t. The plot is not an exact straight line due to the fact that the line search method is inexact in nature.

3) $a^T x = c$ ~~$\in$~~ H [H = Hyperplane]

let $x_1, x_2 \in H$

$a^T x_1 = c$

$\therefore$ for some $\lambda \in [0,1]$

$\lambda (a^T x_1) = \lambda c$

$\Rightarrow a^T (\lambda x_1) = \lambda (c)$ — 1

Similarly

$(1-\lambda)(a^T x_2) = (1-\lambda) c$

$\Rightarrow a^T [(1-\lambda) x_2] = (1-\lambda) c$ — 2

Adding 1 & 2

$a^T (\lambda x_1) + a^T [(1-\lambda) x_2] = \lambda c + (1-\lambda) c$

$\Rightarrow a^T [\lambda x_1 + (1-\lambda) x_2] = c$

clearly $\lambda x_1 + (1-\lambda) x_2 \equiv x_3$ is a point lying between x_1 & x_2

$\therefore a^T x_3 = c$

$\therefore x_1, x_2 \in H \Rightarrow x_3 \subseteq H$ for any $\lambda \in [0,1]$ $\therefore$ The hyperplane is a convex set.

4) (a) $\min_{p} \max_{k} \{ h(a_k^T p, I_t) \}$ | $h(I, I_t) = \begin{cases} I_t / I & I \le I_t \\ I / I_t & I_t \le I \end{cases}$

subject to $0 \le p_i \le p_{max}$

This can be written as

$$\min_{p} \max \{ h(a_1^T p, I_t), h(a_2^T p, I_t), \dots, h(a_m^T p, I_t) \}$$

(a) we know if $f_i \; \forall \; i = 1, 2, \dots m$ is convex in p

then $\max\{f_i\}$ would also be convex in p.

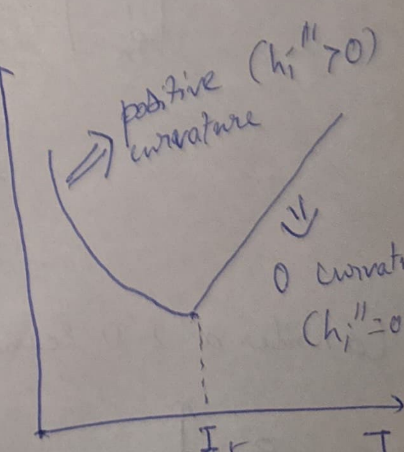
clearly the constraints are linear in nature.

$\therefore$ the feasible space is convex.

now looking at the function:

$$\frac{dh_i}{dp} = \frac{\partial h_i}{\partial I_i} \cdot \frac{\partial I_i}{\partial p} \quad [\text{where } I = a_i^T p]$$

$$= h_i' \, a$$



$$H = \frac{d^2 h_i}{dp} = \frac{\partial h_i'}{\partial I_i} \frac{\partial I_i}{\partial p} a^T$$

$$= h_i'' (a a^T)$$

from the plot $h_i'' \geq 0$ & $a a^T \succeq 0$

$\therefore H \succeq 0 \quad \therefore$ problem is convex in p.

4)

(b) Considering the original problem having a unique solution, we need to make sure that imposing the constraint shall not alter the convexity of the feasible space.

The given constraint says that the overall power output of any 10 lamps should be less than p^*

The constraints are $p_1 + p_2 + \dots + p_{10} \le p^*$
or
$p_2 + p_4 + p_6 + \dots$ 10 terms $< p^*$

This can be reformulated as $\Sigma a_i p_i \le p^*$

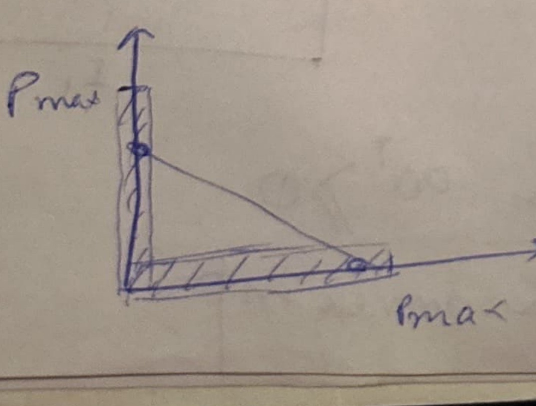
or $$[a_1\ a_2 \dots a_n] \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} \le p^*$$

or $a^T p \le p^*$ where $a_i \in \{0, 1\}$ & $\Sigma a_i = 10$

These are linear constraints which are convex. The feasible set is convex. Therefore there can be a unique solution

(c) $p = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix}$ Think of these as n-dimensions with a constraint on the number of lamps to be switched on we are restricting our feasible set to be a union of certain axes only. ~~However~~ ~~However a linear~~ which makes it non-convex.

Consider a 2-D Scenario. The feasible set would be the shaded region for the constraint that only 1 of 2 lamps can be turned on at a time. However the line joining them does not belong to the feasible set. Hence non-convex. No unique soln



5) $c^*(y) = \max\{xy - c(x)\}$

$f(y_1) = xy_1 - c(x)$

$f(y_2) = xy_2 - c(x)$

$\therefore \lambda f(y_1) = \lambda(xy_1) - \lambda c(x)$

$(1-\lambda) f(y_2) = (1-\lambda) xy_2 - (1-\lambda) c(x)$

$\Rightarrow \lambda f(y_1) + (1-\lambda) f(y_2) = x[\lambda y_1 + (1-\lambda)y_2] - c(x)$

$\Rightarrow f(\lambda y_1 + (1-\lambda) y_2) = \lambda f(y_1) + (1-\lambda) f(y_2)$

$\Rightarrow f[\lambda y_1 + (1-\lambda) y_2] \leq \lambda f(y_1) + (1-\lambda) f(y_2)$

$\therefore$ Convex function [affine in y]

Also if f_i is convex $\forall\, i \in [1, N]$

then $\max\{f_i\}$ is convex

$\therefore c^*(y)$ is convex w.r.t y