Solutions: Homework-2

Friday, September 2, 2022 8:31 PM

MAE 598 - Design Optimization

Prajwal Wanjari

1222575718

1. (20 points) Show that the stationary point (zero gradient) of the function

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

is a saddle (with indefinite Hessian).

Find the directions of downslopes away from the saddle. To do this, use Taylor's expansion at the saddle point to show that

$$f(x_1, x_2) = f(1,1) + (a\partial x_1 - b\partial x_2)(c\partial x_1 - d\partial x_2),$$

with some constants a, b, c, d and $\partial x_i = x_i - 1$ for $i = 1, 2$. Then the directions of downslopes are such $(\partial x_1, \partial x_2)$ that

$$f(x_1, x_2) - f(1,1) = (a\partial x_1 - b\partial x_2)(c\partial x_1 - d\partial x_2) < 0.$$

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

saddle / stationary pt. is where gradient is zero

$$\frac{\partial f}{\partial x_1} = \frac{\partial (2x_1^2 - 4x_1x_2 + 1.5x_2^2 + x_2)}{\partial x_1}$$

$$g = 4x_1 - 4x_2 = 0$$

$$x_1 = x_2 \qquad — (1)$$

$$\frac{\partial f}{\partial x_2} = \frac{\partial (2x_1^2 - 4x_1x_2 + 1.5x_2^2 + x_2)}{\partial x_2}$$

$$g = -4x_1 + 3x_2 + 1 = 0$$

from eq (1)

$$x_1 = 1$$

$\therefore \quad x_1 = x_2 = 1$

$\therefore$ saddle/ Stationary pt. $= (1, 1)$

$$\boxed{g = \begin{bmatrix} 4x_1 - 4x_2 \\ -4x_1 + 3x_2 + 1 \end{bmatrix}}$$

$$\mathbf{H} = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_2 \partial x_1} \\ \\ \dfrac{\partial^2 f}{\partial x_1 \partial x_2} & \dfrac{\partial^2 f}{\partial x_2^2} \end{bmatrix}$$

$$\frac{\partial f}{\partial x_1} = 4x_1 - 4x_2 \qquad \frac{\partial^2 f}{\partial x_1^2} = 4$$

$$\frac{\partial f}{\partial x_2} = -4x_1 + 3x_2 + 1 \qquad \frac{\partial^2 f}{\partial x_2^2} = 3$$

$$\frac{\partial^2 f}{\partial x_2 \partial x_1} = -4$$

$$\frac{\partial^2 f}{\partial x_1 \partial x_2} = -4$$

$$\boxed{\mathbf{H} = \begin{bmatrix} 4 & -4 \\ -4 & 3 \end{bmatrix}}$$

$$|H - \lambda I| = 0$$

$$\left| \begin{bmatrix} 4 & -4 \\ -4 & 3 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} \right| = 0$$

$$\begin{vmatrix} 4-\lambda & -4 \\ -4 & 3-\lambda \end{vmatrix} = 0$$

$$\begin{vmatrix} 4-\lambda & -4 \\ -4 & 3-\lambda \end{vmatrix} = 0$$

$$(4-\lambda)(3-\lambda) - 16 = 0$$

$$12 - 4\lambda - 3\lambda + \lambda^2 - 16 = 0$$

$$\lambda^2 - 7\lambda - 4 = 0$$

$$\lambda = \frac{7}{2} \pm \frac{\sqrt{65}}{2}$$

$$\therefore \lambda_1 = 7.53 \qquad \lambda_2 = -0.53$$

Both eigen values are either +ve or -ve. i.e Hessian is indefinate
∴ The given function is saddle.

To find the direction of downslope away from the saddle :-
using 2^{nd} order taylor's expansion at saddle point

$$f(x_1,x_2) = (2x_1^2 - 4x_1x_2 + 1.5x_2^2 + x_2) + \begin{bmatrix} 4x_1 - 4x_2 \\ -4x_1 + 3x_2 + 1 \end{bmatrix}^T \begin{bmatrix} x_1 - 1 \\ x_2 - 1 \end{bmatrix}$$

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

$$= (2\cdot1^2 - 4\times1\times1 + 1.5\times1^2 + 1) + \begin{bmatrix} 4\cdot1 - 4\cdot1 \\ -4\cdot1 + 3\cdot1 + 1 \end{bmatrix}^T \begin{bmatrix} x_1 - 1 \\ x_2 - 1 \end{bmatrix}$$

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

(at saddle pt.)

Substituting $x_1 - 1 = dx_1$ & $x_2 - 1 = dx_2$

$$= \frac{1}{2} + \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} dx_1 \\ dx_2 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} dx_1 & dx_2 \end{bmatrix} \begin{bmatrix} 4 & -4 \\ -4 & 3 \end{bmatrix} \begin{bmatrix} \partial x_1 \\ \partial x_2 \end{bmatrix}$$

$$= \frac{1}{2} + \frac{1}{2} \left[(4\partial x_1 - 4dx_2)\partial x_1 + (-4\partial x_1 + 3\partial x_2)\partial x_2 \right)$$

$$= \frac{1}{2} + \frac{1}{2} [(4\partial x_1 - 4 d x_2)\partial x_1 + (-4\partial x_1 + 3\partial x_2)\partial x_2)$$

$$= \frac{1}{2} + \frac{1}{2} (4\partial x_1^2 - 4\partial x_1 \partial x_2 - 4\partial x_1 \partial x_2 + 3\partial x_2^2)$$

$$= \frac{1}{2} + \frac{1}{2} (4\partial x_1^2 - 8\partial x_1 \partial x_2 + 3\partial x_2^2)$$

$$= \frac{1}{2} + \frac{1}{2} (2\partial x_1 (2\partial x_1 (2\partial x_1 - \partial x_2) - 3\partial x_2 (2\partial x_1 - \partial x_2))$$

$$= \frac{1}{2} + \frac{1}{2} (2\partial x_1 - 3\partial x_2)(2\partial x_1 - \partial x_2)$$

$$\boxed{f(x_1, x_2) = \frac{1}{2} + (\partial x_1 - \frac{3}{2}\partial x_2)(2\partial x_1 - \partial x_2)} \quad — (2)$$

(a) $f(x_1, x_2) = f(1,1) + (a\partial x_1 - b\partial x_2)(c\,dx_1 - d\partial x_2) \quad — (3)$

where a, b, c, d are constants and

$$\partial x_1 = x_1 - 1$$
$$\partial x_2 = x_2 - 1$$

$$f(x_1, x_2) = 2x_1^2 - 4x_1x_2 + 1.5x_2^2 + x_2$$
$$f(1,1) = 2(1)^2 - 4(1) + 1.5(1) + 1 = 2 - 4 + 1.5 + 1$$
$$f(1,1) = 0.5 = \frac{1}{2}$$

substituting value of $f(1,1)$ in eq. (3)

$$f(x_1, x_2) = \frac{1}{2} + (a\partial x_1 - b\partial x_2)(c\partial x_1 - d\partial x_2) \quad — (4)$$

comparing eq (2) and (4)

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

$\therefore$ The directions of downslopes are such $(\partial x_1, \partial x_2)$ that

$$\boxed{f(x_1, x_2) - \frac{1}{2} = (\partial x_1 - \frac{3}{2}\partial x_2)(2\partial x_1 - \partial x_2) < 0}$$

2. (a) (10 points) Find the point in the plane $x_1 + 2x_2 + 3x_3 = 1$ in $\mathbb{R}^3$ that is nearest to the point $(-1, 0, 1)^T$. Is this a convex problem? Hint: Convert the problem into an unconstrained problem using $x_1 + 2x_2 + 3x_3 = 1$.

(b) (40 points) Implement the gradient descent and Newton's algorithm for solving the problem. Attach your codes in the report, along with a short summary of your findings. The summary should include: (1) The initial points tested; (2) corresponding solutions; (3) A log-linear convergence plot.

$x_1 + 2x_2 + 3x_3 = 1$ $\mathbb{R}^3$

(a) find a point nearest to $(-1, 0, 1)^T$

convex problm = ?

$x_1 = 1 - 2x_2 - 3x_3$ — (1)

$f(x) = (x_1 + 1)^2 + (x_2 - 0)^2 + (x_3 - 1)^2$ — (2)

Substituting value (1) in eq (2)

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

$= (-2x_2 - 3x_3 + 2)^2 + x_2^2 + x_3^2 - 2x_3 + 1$

$= 4x_2^2 + 12x_2x_3 + 9x_3^2 - 8x_2 - 12x_3 + 4 + x_2^2 + x_3^2 - 2x_3 + 1$

$$\boxed{f(x) = 5x_2^2 + 10x_3^2 + 12x_2x_3 - 8x_2 - 14x_3 + 5}$$

$$g = \frac{\partial f}{\partial x_2} = \frac{\partial(5x_2^2 + 10x_3^2 + 12x_2x_3 - 8x_2 - 14x_3 + 5)}{\partial x_2}$$

$$= 10x_2 + 12x_3 - 8$$

$$\frac{\partial f}{\partial x_3} = \frac{\partial(5x_2^2 + 10x_3^2 + 12x_2x_3 - 8x_2 - 14x_3 + 5)}{\partial x_3}$$

$$= 20x_3 + 12x_2 - 14$$

$$\boxed{g = \begin{bmatrix} 10x_2 + 12x_3 - 8 \\ 20x_3 + 12x_2 - 14 \end{bmatrix}}$$

<u>at saddle pt :- $g = 0$</u>

$\therefore$ $10x_2 + 12x_3 - 8 = 0$

$12x_2 + 20x_3 - 14 = 0$

$$x_2 = \frac{-1}{7} \qquad x_3 = \frac{11}{14}$$

$$x_1 + 2x_2 + 3x_3 = 1$$

$$x_1 - \frac{2}{7} + \frac{33}{14} = 1$$

$$x_1 = 1 + \frac{4 - 33}{14}$$

$$x_1 = \frac{\cancel{18} - 33}{14}$$

$$x_1 = \frac{-15}{14}$$

nearest pt. is $(-1,0,1)$ is $(x_1, x_2, x_3) = \left(\frac{-15}{14}, \frac{-1}{7}, \frac{11}{14}\right)$

$= (-1.07, -0.14, 0.78)$

To find the convexity of $f(x) = 5x_2^2 + 10x_3^2 + 12x_2x_3 - 8x_2 - 14x_3 + 5$

Hessian of $f(x)$ is $\begin{bmatrix} \partial^2 x / \partial x_1^2 & \partial^2 x / \partial x_2 \partial x_1 \\ \frac{\partial^2 x}{\partial x_2 \partial x_1} & \partial^2 x / \partial x_2^2 \end{bmatrix}$

$$\frac{\partial f}{\partial x_2} = 10x_2 + 12x_3 - 8 \qquad \frac{\partial^2 f}{\partial x_2^2} = 10$$

$$\frac{\partial f}{\partial x_3} = 20x_3 + 12x_2 - 14 \qquad \frac{\partial^2 f}{\partial x_3^2} = 20$$

$$\frac{\partial^2 f}{\partial x_3 \partial x_2} = 12$$

$$\frac{\partial^2 f}{\partial x_2 \partial x_3} = 12$$

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

$$|H - \lambda I|$$

$$\begin{vmatrix} 10-\lambda & 12 \\ 12 & 20-\lambda \end{vmatrix}$$

$$(10-\lambda)(20-\lambda) - (12)^2$$

$$200 - 10\lambda - 20\lambda + \lambda^2 - 144$$

$$\lambda^2 - 30\lambda + 56$$

$$\underline{\lambda_1 = 28 \qquad \lambda_2 = 2}$$

Both eigen values are positive, $\therefore$ H is positive definate
Hence, This function is a strictly convex function.

(b) https://colab.research.google.com/drive/1BLSecPETMePB-oYfDL6o9AnC4FI82TXg?usp=sharing

DO_HW2_Q2b

DO_HW2_Q2b.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Design Optimization - HW2

Prajwal Wanjari

1222575718

Q2.(b)

Implement the gradient descent and Newton's algorithm for solving the problem. Attach your codes in the report, along with a short summary of your findings. The summary should include:

(1) The initial points tested;

(2) corresponding solutions;

(3) A log-linear convergence plot.

Solution-

```
[17] #importing essential libraries
     import numpy as np
     import matplotlib.pyplot as plt
```

```
# Using Inexact line search / Armijo Line search algorithm

def f_calculator(alpha,x2,x3):
  #t can be any value between 0 & 1; t is reducing slope between approximation lines
  t = 0.7
  obj_func = objective_function(x2,x3)
  grad = gradient(x2,x3)
  # print gradient
  # f(x) = 5*x2^2+10*x3^2+12*x2*x3-8*x2-14*x3+5
  f_alpha_grad= 5*((x2-alpha*grad[0][0])**2) + 10*((x3 - alpha*grad[1][0] )**2) + 12*(x2-alpha*grad[0][0])*(x3 - alpha*grad[1][0]) - 14*(x3 - alpha*grad[1][0]) - 8*(x2-alpha*grad[0][
  # phi = f(x)-t*grad^T*grad*alpha
  phi = obj_func - t*(np.linalg.norm(grad)**2)*alpha
  return (f_alpha_grad-phi)

def gradient(x2,x3):
  return ([10*x2 + 12*x3 - 8],
          [12*x2 + 20*x3 - 14])

def objective_function(x2,x3):
  return 5*(x2**2) + 10*(x3**2) + 12*x2*x3 - 14*x3 - 8*x2 + 5

def inexact_line_search(alpha,x2,x3):
  counter = 0                        #termination criterion/counter
  while counter<100:
    s = f_calculator(alpha,x2,x3)    #calling function
    if s>0:
      alpha = alpha*0.5
      counter = counter+1
    else:
      # print("Solution not converging")
      break
  return alpha
```

✓ 0s completed at 8:40 PM

```
#initializing values/guesses
x2 = 0
x3 = 0
to = 0.0001
grad = gradient(x2,x3)
grad_e = np.linalg.norm(grad)

counter = 0
c = [counter]
error = [grad_e]
while grad_e>to and counter<100:
  alpha = 1                                 #
  alpha = inexact_line_search(alpha,x2,x3)  #
  x2 = x2 - alpha*grad[0][0]
  x3 = x3 - alpha*grad[1][0]
  grad = gradient(x2,x3)
  grad_e = np.linalg.norm(grad)
  error.append(grad_e)
  counter = counter + 1
  c.append(counter)

print(f'Initial guess is: x1:1 , x2:0 , x3:0')
x1 = 1-2*x2-3*x3
print("The point in the plane x1+2*x2+3*x3=1 that is nearest to the point (-1, 0, 1) is: ")
print(f'x1:{x1},x2:{x2},x3:{x3}')

d = (x1+1)**2 + x2**2 + (x3-1)**2
print("Minimum Distance: ",np.sqrt(d))

#log-linear convergence plot
plt.plot(c,error)
```

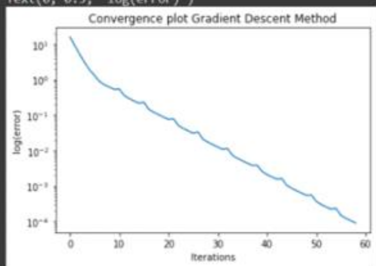
✓ 0s completed at 8:40 PM

+ Code + Text

```
d = (x1+1)**2 + x2**2 + (x3-1)**2
print("Minimum Distance: ",np.sqrt(d))

#log-linear convergence plot
plt.plot(c,error)
plt.yscale("log")
plt.title("Convergence plot Gradient Descent Method")
plt.xlabel("Iterations")
plt.ylabel("log(error)")
```

```
Initial guess is: x1:1 , x2:0 , x3:0
The point in the plane x1+2*x2+3*x3=1 that is nearest to the point (-1, 0, 1) is:
x1:-1.0714325370678943,x2:-0.14282110558427424,x3:0.785691582745481
Minimum Distance:  0.2672612453357342
Text(0, 0.5, 'log(error)')
```



0s completed at 8:40 PM

```
[20] #using Newton's Method

#initializing values/guesses
x2 = 0
x3 = 0
to = 0.0001
grad = gradient(x2,x3)
grad_e = np.linalg.norm(grad)
H = ([10,12],[12,20])          #Hessian matrix
H_inv = np.linalg.inv(H)       #inverse of Hessian matrix

counter = 0
c = [counter]
error = [grad_e]
while grad_e>to and counter<100:
  dx = np.matmul(H_inv,grad)      #
  x2 = x2 - dx[0][0]
  x3 = x3 - dx[1][0]
  grad = gradient(x2,x3)
  grad_e = np.linalg.norm(grad)
  error.append(grad_e)
  counter = counter + 1
  c.append(counter)

print(f'Initial guess is: x1:1 , x2:0 , x3:0')
x1 = 1-2*x2-3*x3
print("The point in the plane x1+2*x2+3*x3=1 that is nearest to the point (-1, 0, 1) is: ")
print(f'x1:{x1},x2:{x2},x3:{x3}')

d = (x1+1)**2 + x2**2 + (x3-1)**2
print("Minimum Distance: ",np.sqrt(d))
```

0s completed at 8:40 PM

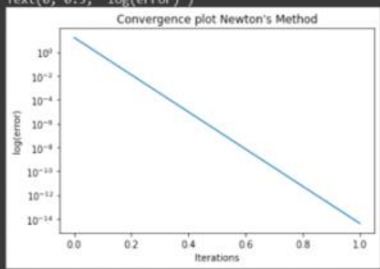
+ Code + Text

```python
print(f'x1:{x1},x2:{x2},x3:{x3}')

d = (x1+1)**2 + x2**2 + (x3-1)**2
print("Minimum Distance: ",np.sqrt(d))

#log-linear convergence plot
plt.plot(c,error)
plt.yscale("log")
plt.title("Convergence plot Newton's Method")
plt.xlabel("Iterations")
plt.ylabel("log(error)")
```

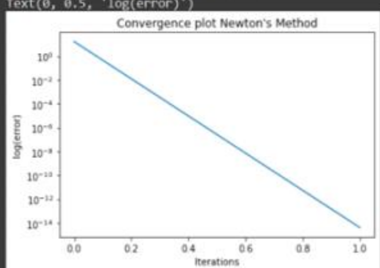
```
Initial guess is: x1:1 , x2:0 , x3:0
The point in the plane x1+2*x2+3*x3=1 that is nearest to the point (-1, 0, 1) is:
x1:-1.071428571428572,x2:-0.1428571428571428,x3:0.7857142857142858
Minimum Distance:  0.26726124191242445
Text(0, 0.5, 'log(error)')
```



0s completed at 8:40 PM

+ Code + Text

```
Minimum Distance:  0.26726124191242445
Text(0, 0.5, 'log(error)')
```



In the inexact gradient decent method, variable t is representing reducing slope between the approximation lines where t can be any value between 0 & 1. In this question, as we move towards value of t= 0.8 & 0.9; we get a more consistent convergence plot, whereas the graph overshoots it values if value of t>=1;

In Newton's method, as the objective is quadratic, it only takes one step to compute convergence plot.

From both methods, the initial guesses are x1:1 , x2:0 , x3:0 ;

the point in the plane x1+2x2+3x3=1 that is nearest to the point (−1, 0, 1) is: x1:-1.071428571428572, x2:-0.1428571428571428, x3:0.7857142857142858

and the minimum distance between both points is 0.26726124191242445.

0s completed at 8:40 PM

3. (5 points) Prove that a hyperplane is a convex set. Hint: A hyperplane in $\mathbb{R}^n$ can be expressed as: $\mathbf{a}^T\mathbf{x} = c$ for $\mathbf{x} \in \mathbb{R}^n$, where $\mathbf{a}$ is the normal direction of the hyperplane and c is some constant.

We need to show that any two points of the hyperplane of $\mathbb{R}^n$ can be joined by a line segment.

Hyperplane H is expressed as

$a^T x = c$ where x is normal & c is constant

$\&$ $a_1 x_1 + a_2 x_2 + \cdots\cdots a_n x_n = c$ where $a_1 \cdots a_n \neq 0$; $c \in \mathbb{R}$
$x \in \mathbb{R}^n$

Let x_1 and x_2 points on hyperplane

$x_1, x_2 \in$ Hyperplane

$\therefore$ $a^T x_1 = c$
$a^T x_2 = c$

For any $\lambda \in [0, 1]$
$= \lambda x_1 + (1-\lambda) x_2$
$= a^T (\lambda x_1 + (1-\lambda) x_2)$
$= \lambda a^T \cancel{x_1} + a^T x_2 - \lambda a^T \cancel{x_2}$
$= a^T x_2$
$= c$

$\therefore$ $a^T x (\lambda x_1 + (1-\lambda) x_2) = c$

$\boxed{\therefore \text{ This hyperplane is a convex set}}$

4. (15 points) Consider the following illumination problem:

$$\min_{\mathbf{p}} \quad \max_k \{h(\mathbf{a}_k^T \mathbf{p}, I_t)\}$$
$$\text{subject to:} \quad 0 \leq p_i \leq p_{\max},$$

where $\mathbf{p} := [p_1, ..., p_n]^T$ are the power output of the n lamps, $\mathbf{a}_k$ for $k = 1, ..., , m$ are fixed parameters for the m mirrors, I_t the target intensity level. $h(I, I_t)$ is defined as follows:

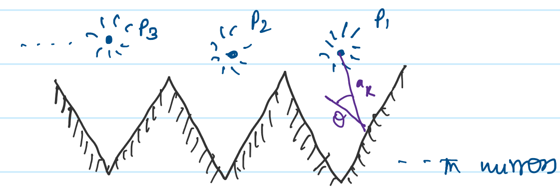
$$h(I, I_t) = \begin{cases} I_t/I & \text{if } I \leq I_t \\ \underset{\text{linear}}{I/I_t} & \text{if } I_t \leq I \end{cases} \Big\} \text{ p.s.d}$$

(b) not possible

(a) (5 points) Show that the problem is convex.

(b) (5 points) If we require the overall power output of any of the 10 lamps to be less than p^*, will the problem have a unique solution?

(c) (5 points) If we require no more than 10 lamps to be switched on ($p > 0$), will the problem have a unique solution?



Illumination problem

$$\min_{p} \max_{k} \{ h(a_k^T p, I_t) \}$$

s.t. $0 \leq p_i \leq p_{max}$

$$h(I, I_t) = \begin{cases} I_t / I & \text{if } I < I_t \\ I / I_t & \text{if } I_t \leq I \end{cases}$$

→ power output $p = [p_1 \; ...... \; p_n]^T$ for lamps $n[1, n]$.

→ constant ~ a_k wher $k = [1, m]$

→ I_t is target intensity level. ~ constant

→ I is variable

(a) - so, for $I < I_t$, $h = I_t / I$

The graph of the above function is parabola till $I = I_t$

$$H(h) = \frac{\partial}{\partial I} \left(\frac{\partial (I_t / I)}{\partial I} \right) = \frac{2 I_t}{I^3}$$

$I = pa$

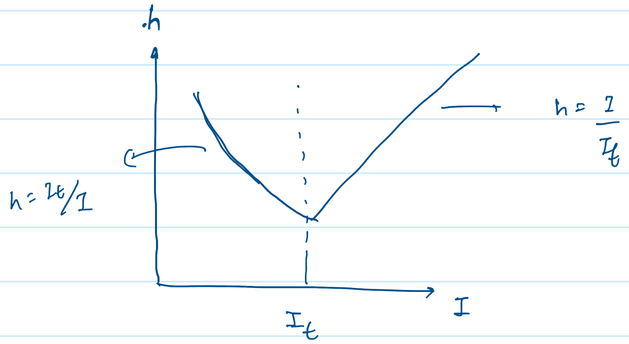
$\because p_i > 0$ & $a > 0$ $\therefore I > 0$

$\therefore h = I_t / I$ is strictly convex as the Hessian of $h(I, I_t)$ at I_t / I condition is positive definite.

- For, $I > I_t$, $h = I / I_t$

The graph of the above funct^n is similar to $y = mx$ hence a linear graph after $I = I_t$.

$\therefore$ $\boxed{\text{The graph is +ve throughout, } h(I, I_t) \text{ is convex.}}$



(b) Overall output of any 10 lamp $\leq p^*$

unique solution = ?

- For $h = I_t/I$, the graph is strictly convex as H of the function is positive definite. But if we look into the other part of the function i.e $h = I/I_t$, the graph is not strictly convex as the function is linear.

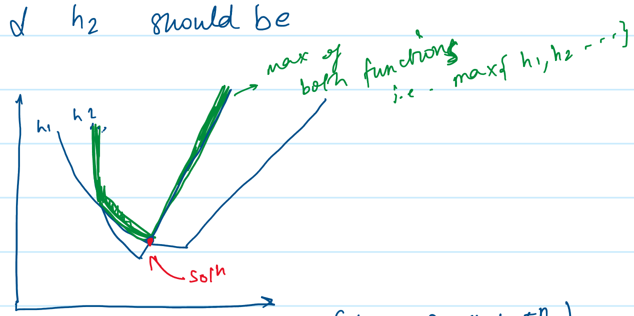
So, to prove that the function has a unique solution :-

$$h = \begin{cases} I_t/I & I \leq I_t \\ I/I_t & I \geq I_t \end{cases}$$

$$I = ap$$

for $\max\{h_1, h_2, \ldots\}$,

the graph of h_1 & h_2 should be



(to representation)

The solution of $\max\{h_1, h_2\}$ must lie on the intersection of both functions and it will be always the case as the intersection point would be last where the function stops decreasing and gives the best function value.

- This intersection is a unique solution as there exists the set of linear functions which are equal. These functions are

$$\frac{a_1^T p}{I_t} = \frac{a_2^T p}{I_t} = \cdots = \frac{a_n^T p}{I_t}$$

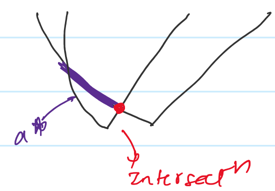
Furthermore, at intersection

$$\frac{I_t}{a_1'^T p} = \frac{I_t}{a_2'^T p} = \cdots$$

$(a_1 - a_2)^T p = 0$

$$\& \quad (a_1 - a_2)^T p = 0$$
$$\vdots$$
$$(a_n - a_{n+1})^T p = 0$$

so, if we take a function representing the intersection conditions, that would be



$$\min_{p} \quad \frac{I_t}{a^{*T} p} = \frac{I_t}{I^*}$$

$$\text{s.t.} \quad (a_1 - a_2)^T p = 0$$
$$\vdots$$
$$(a_n - a_{n+1})^T p = 0$$

$\therefore$ Hessian of above funct^n is $= \dfrac{\partial}{\partial I^*}\left(\dfrac{\partial [I_t / I]}{\partial I^*}\right) = \dfrac{2 I_t}{I^{*3}}$

so for $P_i > 0$, $I^* \geq 0$

$$\therefore \quad \frac{2 I_t}{I^3} > 0$$

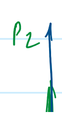
$\therefore$ The above function is always convex is the hessian is P.D.

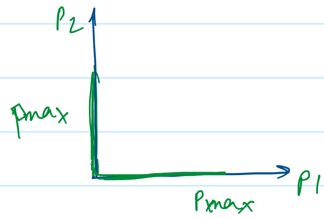
Hence, we can say that this problem has a strictly convex function with a unique solution at its intersection. This case will be similar for c_n^{10} lamps as well, with a power output of p. $(p \leq p^*)$

c) If we require no more than 10 lamps to be switched on $(p > 0)$, then we can't say if the function will have a unique sol^n or not.

for example,

if we are using only 2 lamps, i.e P_1 & P_2. Either of those lamps can give power output between 0 & P_{max}. It is possible that the power output would not lie in between.





$\boxed{\therefore \text{ Not always a convex function, we can't say anything in certain.}}$

5. (10 points) Let $c(x)$ be the cost of producing x amount of product A and assume that $c(x)$ is differentiable everywhere. Let y be the price set for the product. Assuming that the product is sold out. The total profit is defined as

$$c^*(y) = \max_x \{xy - c(x)\}.$$

Show that $c^*(y)$ is a convex function with respect to y.

Assumptions :- ① $c(x)$ is differentiable everywhere.
② product A is all sold out with price y.

$c(x) \Rightarrow$ cost of producing x amount of product A

$y \Rightarrow$ selling price of A

total profit ; $c^*(y) = \max_x \{xy - c(x)\}$

$c(y) = xy - c(x)$

gradient of $c(y)$ is

$$g = \frac{\partial c(y)}{\partial y} = \frac{\partial(xy - c(x))}{\partial y}$$

$$= x$$

Hessian of $c(y)$ is

$$\mathbf{H} = \frac{\partial^2 c(y)}{\partial y^2} = \frac{\partial (x)}{\partial y}$$

$$\mathbf{H} = 0$$

$\because$ $\mathbf{H} = 0 \Rightarrow \lambda_1, \lambda_2 \geq 0$

$\therefore$ Hessian is positive semidefinate everywhere
So, $c(y)$ is a convex function w.r.t y, furthermore, if $c(y)$ is a convex function, the maximum of multiple convex functions is also convex.

$\therefore$ $\boxed{\max\{c^{*}(y)\} \text{ is a convex function}}$