ME598/494 Homework 1

1. Solve the following problem using Python SciPy.optimize. Please attach your code and results. Specify your initial guesses of the solution. If you change your initial guess, do you find different solutions? (**30 points**)

$$
\begin{aligned}
\text{minimize:} \quad & (x_1 - x_2)^2 + (x_2 + x_3 - 2)^2 + (x_4 - 1)^2 + (x_5 - 1)^2 \\
\text{subject to:} \quad & x_1 + 3x_2 = 0 \\
& x_3 + x_4 - 2x_5 = 0 \\
& x_2 - x_5 = 0 \\
& -10 \leq x_i \leq 10, \ i = 1, \ldots, 5
\end{aligned}
$$

Note: Please learn how to use **break points** to debug. You can use Python on Google Colab.

Ans Project Link: https://github.com/monarkparekh/MAE-598__Design-Optimization.git

→ I have also attached the PDF of the project below

MAE 598 Design Optimization: Assignment 1, Question 1</h1>

Name: Monark Parekh
ASU ID: 1222179426

$$
\begin{aligned}
\text{minimize:} \quad & (x_1 - x_2)^2 + (x_2 + x_3 - 2)^2 + (x_4 - 1)^2 + (x_5 - 1)^2 \\
\text{subject to:} \quad & x_1 + 3x_2 = 0 \\
& x_3 + x_4 - 2x_5 = 0 \\
& x_2 - x_5 = 0 \\
& -10 \leq x_i \leq 10, \; i = 1, \ldots, 5
\end{aligned}
$$

In [ ]:

```
# Importing Libraries
from scipy.optimize import minimize
import random
```

Defining the Function

In [ ]:

```
function = lambda x: (x[0] - x[1]) ** 2 + (x[1] + x[2] - 2) ** 2 + (x[3] - 1) ** 2 +
```

Defining the Constraints

In [ ]:

```
contraints = ({'type': 'eq', 'fun': lambda x: x[0] + 3 * x[1]},
              {'type': 'eq', 'fun': lambda x: x[2] + x[3] - 2 * x[4]},
              {'type': 'eq', 'fun': lambda x: x[1] - x[4]})
```

Defining the Boundary Conditions

In [ ]:

```
bounds = ((-10, 10), (-10, 10), (-10, 10), (-10, 10), (-10, 10))
```

Defining the Inital Guess

In [ ]:

```
init_guess_1 = (1, 2, 3, 4, 5)
```

Method to determine the minimum value based on the constraints, boundary conditions and the initial guess

In [ ]:

```
result = minimize(function, init_guess_1, method='SLSQP', bounds=bounds, constraints
```

Results

In [ ]:

```
print(f"For the initial guess of {init_guess_1} the minimum values are {result.x}")
```

```
For the initial guess of (1, 2, 3, 4, 5) the minimum values are [-0.76743912  0.2558
1304  0.62791188 -0.1162858   0.25581304]
```

Evaluating the result for different initial guesses

In [ ]:

```
random_init_guess = [1,2,3,4,5,6,7,8,9,10]  # Temporary Defination
result = [1,2,3,4,5,6,7,8,9,10]             # Temporary Defination
for i in range(0,10):
    random_init_guess[i] = random.sample(range(-10, 10), 5)
    result[i] = minimize(function, random_init_guess[i], method='SLSQP', bounds=boun
    print(f"For the initial guess of {random_init_guess[i]}\t the minimum values are
```

```
For the initial guess of [-3, 3, -7, 5, 9]      the minimum values are [-0.76744186
0.25581395  0.62790698 -0.11627907  0.25581395]
For the initial guess of [2, 8, 7, -10, -2]     the minimum values are [-0.76744186
0.25581395  0.62790697 -0.11627906  0.25581395]
For the initial guess of [1, 0, -8, -2, 6]      the minimum values are [-0.76771936
0.25590645  0.62769626 -0.11588336  0.25590645]
For the initial guess of [-6, 6, -10, 3, 1]     the minimum values are [-0.7674424
0.25581413  0.62789394 -0.11626568  0.25581413]
For the initial guess of [4, -5, 6, 7, 0]       the minimum values are [-0.76744185
0.25581395  0.62790697 -0.11627907  0.25581395]
For the initial guess of [-10, 8, 1, -4, 7]     the minimum values are [-0.76744429
0.25581476  0.62790968 -0.11628016  0.25581476]
For the initial guess of [7, 1, 2, -10, 6]      the minimum values are [-0.76744185
0.25581395  0.62790697 -0.11627906  0.25581395]
For the initial guess of [-1, 2, 5, 0, 1]       the minimum values are [-0.76746246
0.25582082  0.62792047 -0.11627883  0.25582082]
For the initial guess of [-4, -5, 0, -10, 3]    the minimum values are [-0.76744187
0.25581396  0.62790697 -0.11627906  0.25581396]
For the initial guess of [0, -2, 5, 8, 6]       the minimum values are [-0.76744186
0.25581395  0.62790698 -0.11627907  0.25581395]
```

Determining the the overall impact of initial guess

In [ ]:

```
mean = [0,0,0,0,0]  # Temporary Defination
for i in range(0,5):
  for j in range(0,10):
    mean[i] = mean[i] + result[j].x[i]
  mean[i] = mean[i]/10

standard_deviation = [0,0,0,0,0]  # Temporary Defination
for i in range(0,5):
  for j in range(0,10):
    standard_deviation[i] = standard_deviation[i] + (result[j].x[i] - mean[i])**2
  standard_deviation[i] = (standard_deviation[i]/10)**(1/2)

for i in range(0,5):
  print(f'The Mean value of x{i+1} is {mean[i]}, whereas the Standard Deviation is {
```

```
The Mean value of x1 is -0.767471966559278, whereas the Standard Deviation is 8.2687
2836963747e-05
The Mean value of x2 is 0.2558239888530927, whereas the Standard Deviation is 2.7562
42789879397e-05
The Mean value of x3 is 0.6278862199165085, whereas the Standard Deviation is 6.3601
85309680369e-05
The Mean value of x4 is -0.1162382422103232, whereas the Standard Deviation is 0.000
11836440374652052
The Mean value of x5 is 0.2558239888530927, whereas the Standard Deviation is 2.7562
42789879397e-05
```

Effect of different Initial Guess

As observed from the results above, the Standard Deviation of the minimum values obtained using 10 random initial guesses is in the range of 10^-5. With such a small standard deviation we can conclude that the effect of initial guess is very small on the results, hence the error produced by an inital guess is negligible.

2. Let x and $b \in \mathbb{R}^n$ be vectors and $A \in \mathbb{R}^{n\times n}$ be a square matrix. Define $f : \mathbb{R}^n \to \mathbb{R}$ as $f(x) = b^T x + x^T Ax$. (**50 points**)

(a) What is the gradient and Hessian of $f(x)$ with respect to x?

(b) Derive the first and second order Taylor's approximations of $f(x)$ at $x = 0$. Are these approximations exact?

(c) What are the necessary and sufficient conditions for A to be positive definite?

(d) What are the necessary and sufficient conditions for A to have full rank?

(e) If there exists $y \in \mathbb{R}^n$ and $y \neq 0$ such that $A^T y = 0$, then what are the conditions on b for $Ax = b$ to have a solution for x?

Ans (a) We know that, by using the defination of gradient:

$$\nabla f(x) = \frac{\partial f(x)}{\partial x}$$

$$\nabla f(x) = \frac{\partial (b^T x + x^T A x)}{\partial x}$$

- $$\frac{\partial (b^T x)}{\partial x} = \begin{bmatrix} \frac{\partial b^T x}{\partial x_1} \\ \frac{\partial b^T x}{\partial x_2} \\ \vdots \\ \frac{\partial b^T x}{x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x_1}(b_1 x_1 + b_2 x_2 + b_3 x_3 + \cdots + b_n x_n) \\ \vdots \\ \frac{\partial}{\partial x_n}(b_1 x_1 + b_2 x_2 + b_3 x_3 + \cdots + b_n x_n) \end{bmatrix}$$

$$\therefore \quad \frac{\partial b^T x}{\partial x} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = b \quad \text{———} \quad (1)$$

$\rightarrow$ Simplifying, $x^T A x = [x_1 \cdots x_n] \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$

$$= \left[(a_{11}x_1 + \cdots + a_{n1}x_1) \cdots (a_{1n}x_1 + \cdots + a_{nn}x_n) \right] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$= \left[\sum_{i=1}^{n} a_{i1} x_i \cdots\cdots \sum_{i=1}^{n} a_{in} x_i \right] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$= x_1 \sum_{i=1}^{n} a_{i1} x_i \cdots\cdots x_n \sum_{i=1} a_{in} x_i$$

$$\therefore \quad x^T A x = \sum_{j=1}^{n} \sum_{i=1}^{n} x_i a_{ij} x_j \qquad \text{—— } (2)$$

→ Calculating partial derivative of equation (2)

- $$(x^T A x) = \sum_{i=1}^{n} \left(a_{ii} x_i^2 + \sum_{j \neq i} (x_i a_{ij} x_j) \right)$$

→ Let us take into consideration the k^{th} row and do partial derivative

$$\therefore \quad \frac{\partial x^T A x}{\partial x_k} = \frac{\partial}{\partial x} \sum_{i=1}^{n} \left(a_{ii} x_i^2 + \sum_{j \neq 1} (x_i a_{ij} x_j) \right)$$

$$= 2a_{kk} x_k + \sum_{j \neq k} x_j a_{jk} + \sum_{j \neq k} a_{kj} x_j$$

$$= \sum_{j=1}^{n} x_j a_{jk} + \sum_{j=1}^{n} a_{kj} x_j$$

$$\therefore \quad \frac{\partial x^T A x}{\partial x} = \begin{bmatrix} \sum_{j=1}^{n} x_j a_{j1} \\ \vdots \\ \sum_{j=1}^{n} x_j a_{jn} \end{bmatrix} + \begin{bmatrix} \sum_{j=1}^{n} a_{1j} x_j \\ \vdots \\ \sum_{j=1}^{n} a_{nj} x_j \end{bmatrix}$$

$$= A^T x + A x$$

$$\therefore \quad \frac{\partial (x^T A x)}{\partial x} = (A^T + A) x \qquad \text{———} \ (3)$$

→ Using equations (1) and (3), the gradient of $f(x)$ is:

$$\boxed{\Rightarrow \nabla f(x) = b + (A^T + A) x}$$

→ Here, if A is symmetric matrix, hence $A^T = A$; then

$$(A^T + A) = (A + A) = 2A$$

$$\boxed{\Rightarrow \nabla f(x) = b + 2Ax} \quad (A \text{ is symmetric})$$

→ We know that the Hessian matrix is defined as:

$$\nabla^2 f(x) = \frac{\partial^2 f(x)}{\partial x \, \partial x^T}$$

We know that, $\nabla f(x) = b_k + \sum_{j=1}^{n} x_j a_{jk} + \sum_{j=1}^{n} a_{kj} x_j$

$\therefore$ Second partial derivative for the k^{th} row is:

$$\nabla^2 f(x) = 0 + a_{k'k} + a_{kk'}$$

$$\therefore \nabla^2 f(x) = \begin{bmatrix} a_{11}+a_{11} & a_{21}+a_{12} & \cdots & a_{n1}+a_{n1} \\ \vdots & \ddots & & \vdots \\ a_{1n}+a_{n1} & \cdots & \cdots & a_{nn}+a_{nn} \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & & & \vdots \\ a_{1n} & \cdots & \cdots & a_{nn} \end{bmatrix} + \begin{bmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ \vdots & \ddots & & \vdots \\ a_{n1} & \cdots & \cdots & a_{nn} \end{bmatrix}$$

$$= A + A^T$$

→ Hessian Matrix of $f(x)$ is :

$$\boxed{\Rightarrow \quad \nabla^2 f(x) = A + A^T}$$

→ Here, if A is symmetric matrix, hence $A^T = A$; then

$$(A^T + A) = (A + A) = 2A$$

$$\boxed{\Rightarrow \quad \nabla^2 f(x) = 2A} \quad (A \text{ is Symmetric})$$

(b)

Ans Using the defination of first order taylor approximation :

→ 1^{st} Order Taylor Approximation of $f(x)$:

- $$f(x) = f(x_0) + \left.\frac{\partial f}{\partial x}\right|_{x_0} (x - x_0)$$

$$\therefore \quad f(x) \approx f(x_0) + \nabla f\Big|_{x_0}^{T} (x - x_0)$$

$$\therefore \quad f(x) = \left(b^T x_0 + x_0^T A x_0\right) + \left(b + (A^T + A) x_0\right)^T (x - x_0)$$

$$= \left(b^T x_0 + x_0^T A x_0\right) + \left(x_0^T A + x_0^T A^T + b^T\right)(x - x_0)$$

$$= \left(\cancel{b^T x_0} + x_0^T \cancel{A x_0}\right) + \left(x_0^T A x - x_0^T \cancel{A} x_0 + x_0^T A^T x - x_0^T A^T x_0 + b^T x - \cancel{b^T x_0}\right)$$

$$= x_0^T A x + x_0^T A^T x - x_0^T A^T x_0 + b^T x$$

For $x_0 = 0$

$\longrightarrow$ 1st Order Jaylor Approximation of $f(x)$:

$$\boxed{\Rightarrow f(x) = b^T x}$$

Using the defination of second order taylor approximation:

$\longrightarrow$ 2nd Order Jaylor Approximation of $f(x)$:

- $f(x) \approx f(x_0) + \nabla f\Big|_{x_0}^{T} (x - x_0) + \frac{1}{2}(x - x_0)^T H\Big|_{x_0} (x - x_0)$

$$f(x) = \left(0 + b^T x\right) + \frac{1}{2} x^T (A + A^T) x \qquad \left[\begin{matrix}\text{For} \\ x_0 = 0\end{matrix}\right]$$

$\longrightarrow$ 2nd Order Jaylor Approximation of $f(x)$:

$$\boxed{\Rightarrow f(x) = b^T x + \frac{1}{2} x^T (A + A^T) x}$$

- If A is symmetric matrix:

$$\boxed{\Rightarrow f(x) = b^T x + \frac{1}{2} x^T (2A) x = b^T x + x^T A x}$$

$\Rightarrow$ 1st order Approximation is not exact because the function is quadratic. Thus, the 2nd order taylor approximation gives us the exact value.

(c)

Ans Since, $A \in \mathbb{R}^{n \times n}$, the necessary and sufficient condition for A to be positive definite is

$\longrightarrow$ All the eigen values of A must be positive, i.e $\boxed{\lambda > 0}$

$\longrightarrow$ $\boxed{x^T A x > 0}$ for all x, other than zero vector

$\boxed{\longrightarrow \text{All upper left determinants must be greater than } 0}$

(d)

Ans The necessary and sufficient condition for A to have full rank is:

$\longrightarrow$ The $\boxed{|A| \neq 0}$ {Determinant of $A \neq 0$}

$\Rightarrow$ All rows are linearly independant

(e)

Ans Given, $A^T y = 0$ — (1) $\Rightarrow$ y belongs to the Null Space of A^T

$$\therefore \quad y \in N(A^T)$$

It is also given that, $Ax = b$

$$\therefore \quad (Ax)^T = b^T$$

Multiplying by y on both sides

$$(Ax)^T y = b^T y$$

$$\therefore \quad b^T y = (Ax)^T y = (x^T A^T) y$$

$$b^T y = x^T (A^T y)$$

Using (1) we know that $A^T y = 0$

$$\therefore \quad b^T y = 0$$

Or y is perpendicular to b

& since $y \in N(A^T)$, we can conclude that for $Ax = b$ to have a solution for x

$\Rightarrow b \in C(A)$ [b must belong to column space of A]

3. Due to the recent inflation, let's reconsider the Stigler diet problem proposed by Nobel laureate George Stigler after the second World War: Consider that the grocery store offers N types of food of your interest, and each food contains the same M types of nutrition. Denote a_{ij} as the quantity of nutrition type j of food type i for $i = 1, 2, ..., N$ and $j = 1, 2, ..., M$, c_i the unit price of food type i, and b_j the necessary

quantity of nutrition type j for a month. Formulate an optimization problem to determine the minimum grocery cost to satisfy the nutrition needs. (**20 points**)

Ans Problem Defination :

→ Food Types = N

→ Each Food Contains = M types of nutrition

→ a_{ij} → Quantity of Nutrition type j of food type i

→ c_i → Unit price of food type i

→ b_j → Necessary quantity of nutrition type j for one month

→ Determine minimum grocery cost to satisfy nutrition needs

Let;

→ $x = [x_1 \ x_2 \ \cdots \ x_N]^T$ → Quantity of each food item that is bought for a month

→ Total Grocery Price for the month $= \sum_{i=1}^{i=N} (c_i x_i)$

→ Constraints : $\left\{ \left[\sum_{i=1}^{N} (x_i a_{ij}) - b_j \right] \geq 0 \right\} \forall j = 1 \text{ to } M$

and $x_i \geq 0$

- $x_1 a_{11} + x_2 a_{21} + x_3 a_{31} + \cdots + x_N a_{N1} \geq b_1$
- $x_1 a_{12} + x_2 a_{22} + x_3 a_{32} + \cdots + x_N a_{N2} \geq b_2$
- $x_1 a_{13} + x_2 a_{23} + x_3 a_{33} + \cdots + x_N a_{N3} \geq b_3$
- $x_1 a_{1M} + x_2 a_{2M} + x_3 a_{3M} + \cdots + x_N a_{NM} \geq b_M$

$$\therefore \begin{bmatrix} a_{11} & a_{21} & a_{31} & \cdots & a_{N1} \\ a_{12} & a_{22} & a_{32} & \cdots & a_{N2} \\ a_{13} & a_{23} & a_{33} & \cdots & a_{N3} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{1M} & a_{2M} & a_{3M} & \cdots & a_{NM} \end{bmatrix}_{M\times N} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}_{N\times 1} - \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_M \end{bmatrix}_{M\times 1} \geq 0$$

$\longleftrightarrow$ Optimization Problem:

$$\Rightarrow \min_{\{x_i\}} \sum_{i=1}^{i=N} (c_i x_i)$$

$$\text{s.t} \left\{ \left[\sum_{i=1}^{N} (x_i a_{ij}) - b_j \right] \geq 0 \right\} \forall j = 1 \text{ to } M, \; x_i \geq 0 \; \forall i = 1 \ldots N$$

OR

$$\Rightarrow \min_{\{x_i\}} \sum_{i=1}^{i=N} (c_i x_i)$$

$$\text{s.t} \begin{bmatrix} a_{11} & a_{21} & a_{31} & \cdots & a_{N1} \\ a_{12} & a_{22} & a_{32} & \cdots & a_{N2} \\ a_{13} & a_{23} & a_{33} & \cdots & a_{N3} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{1M} & a_{2M} & a_{3M} & \cdots & a_{NM} \end{bmatrix}_{M\times N} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}_{N\times 1} - \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_M \end{bmatrix}_{M\times 1} \geq 0$$

$$, \; x_i \geq 0, \; \forall i = 1 \text{ to } N$$