Vyankatesh Chandge

1223508715

MAE: 598 Design Optimization

HW 1: Solution

Problem 1:

Please find the code for problem 1 attached here and also in the git hub repository by the name of HW1.py file, the link to the GitHub repository is here: https://github.com/Goof1999/Design-Optimization

The initial guess was taken to be x1=x2=x3=x4=x5=1 for which we get a minimal value for the given function to be around 4.0930…..

We find that even by changing the initial guess to different values for all 5 variables within the given range, we still obtain the same solution for the minimization function, this proves that we found the true solution for the given problem.

Included below is the code and the results for different initial guess values.

```
## Vyankatesh Change
## 1223508715
## MAE 598: Design Optimization

# HW1 problem 1 Solution

import scipy.optimize as spo


# Define Optimization function
def f(x):
    y = (x[0] - x[1]) ** 2 + (x[1] + x[2] - 2) ** 2 + (x[3] - 1) ** 2 + (x[4] -
1) ** 2
    return y


# Define Constraints
def cons1(x):
    return x[0] + x[1] * 3


def cons2(x):
    return x[2] + x[3] - 2 * x[4]


def cons3(x):
    return x[1] - x[4]

# Give constraint
constr1 = {'type': 'eq', 'fun': cons1}
constr2 = {'type': 'eq', 'fun': cons2}
constr3 = {'type': 'eq', 'fun': cons3}
constr = [constr1,constr2,constr3]

# Bounds limits for all the variables
b = [-10,10]
boundrylimit = (b,b,b,b,b)


# Initial guess
guess=[1,1,1,1,1]

print("Initial guess is",guess)

print("The solution for the initial guess is: ")

# Finally calculating the minimum value of the function

solution = spo.minimize(f,guess,bounds = boundrylimit,constraints = constr)
print(solution)
```

```
C:\Users\vrcch\AppData\Local\Programs\Python\Python310\python.exe D:/ASU/DO/Design-Optimization/HW1.py
Initial guess is [1, 1, 1, 1, 1]
The solution for the initial guess is
     fun: 4.09302326452976
     jac: array([-2.04664832, -0.18578869, -2.23243701, -2.23257673, -1.48833793])
 message: 'Optimization terminated successfully'
    nfev: 38
     nit: 6
    njev: 6
  status: 0
 success: True
       x: array([-0.76749312,  0.25583104,  0.62795044, -0.11628835,  0.25583104])

Process finished with exit code 0

```

```
C:\Users\vrcch\AppData\Local\Programs\Python\Python310\python.exe D:/ASU/DO/Design-Optimization/HW1.py
Initial guess is [2, 2, 2, 2, 2]
The solution for the initial guess is:
     fun: 4.09302328745791
     jac: array([-2.04625618, -0.18625832, -2.2325145 , -2.23279339, -1.48843598])
 message: 'Optimization terminated successfully'
    nfev: 43
     nit: 7
    njev: 7
  status: 0
 success: True
       x: array([-0.76734607,  0.25578202,  0.62796073, -0.11639669,  0.25578202])

Process finished with exit code 0

```

```
C:\Users\vrcch\AppData\Local\Programs\Python\Python310\python.exe D:/ASU/DO/Design-Optimization/HW1.py
Initial guess is [5, 4, 3, 2, 1]
The solution for the initial guess is:
     fun: 4.093023364708078
     jac: array([-2.04703307, -0.18514216, -2.23217535, -2.23254979, -1.48824167])
 message: 'Optimization terminated successfully'
    nfev: 37
     nit: 6
    njev: 6
  status: 0
 success: True
       x: array([-0.76763743,  0.25587914,  0.62803318, -0.11627489,  0.25587914])

Process finished with exit code 0

```

Problem 2

a) $f(x) = b^T x + x^T A x$

gradient of $f(x) = (\nabla_x f)$

So in linear algebraic terms $b^T x$ is same as bx

so $\frac{d(bx)}{dx} = b$; so $\frac{d(b^T x)}{dx}$ also $= b$

Similarly $x^T A x$ is in linear algebraic terms is same as Ax^2, so $\frac{d(Ax^2)}{dx} = 2Ax$,

so $\frac{d(x^T A x)}{dx} = 2Ax$

Thus $\nabla_x f = \left[\frac{\partial f}{\partial x}\right] = \frac{\partial(b^T x + x^T A x)}{\partial x} = \underline{\underline{b + 2Ax}}$

Now $H = \frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right) = \frac{\partial}{\partial x}\left(b + 2Ax\right)$

so,

$\Rightarrow H = \frac{\partial}{\partial x}(b + 2Ax) \Rightarrow \underline{\underline{H = 2A}}$, as $\frac{\partial(b)}{\partial x} = 0$

b) First order taylor approximation

$f(x) = f(x_0) + \nabla_x f\Big|_{x_0}^T (x - x_0)$

$x_0 = 0$, $f(x_0) = f(0) = b(0) + (0)A(0) = \underline{\underline{0}}$

$\nabla_x f\Big|_{x_0} = b + 2Ax_0$

$= b + 2A(0) = b$

so $\nabla f\Big|_{x_0}^T = b^T$

So first order approximation is

$f(x) = 0 + \cancel{\ } b^T(x-0) = b^T x$ ~~...~~,

so, $f(x) = b^T x$

Which is not the same as $f(x)$ so, it is not exact

Similarly for second order

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

when $x_0 = 0$

so, $x_0 = 0$, $\frac{1}{2}(x-0)^T(2A)(x-0)$

$= \frac{1}{2}(x^T)(2A)(x) = x^T A x$

so $f(x) = b^T x + x^T A x$

So as the second order approximation is exactly the same as the original function, the second order approximation is exact.

c) for A to be positive definite, the necessary & sufficient conditions are that the eigen values of A are all positive and greater than 0,

i.e. $\lambda > 0$, where λ are the eigenvalues.

d) for A to be full Rank, this means as A is a square matrix, then it should be strictly non-singular, ie, the $\det(A) \neq 0$, all of the columns in A are linearly independant

e) $y \in \mathbb{R}^n$, $y \neq 0$ such that $A^T y = 0$, then what are the conditions for b so that $Ax = b$ has a solution for x.

so as $A^T y = 0$, this means that y is perpendicular to the plane of A^T, which means that A matrix is also singular, and y is also perpendicular to all the columns of A

and if $Ax = b$ then b also lies in the same space of A, as b is the linear transformation of ~~x~~, so, ~~A~~ A space, so A & b are in the same space,

So this means b is also orthogonal to y, so
$b^T y = 0$

Problem 3, Stigler Diet.

N - types of food items
M - types of nutrient in each food item.

a_{ij} - nutrition of food type i & nutrient j
so $1 \leq i \leq N$ & $1 \leq j \leq M$

c_i - unit price of food type i
b_j - minimum necessary value per month of nutrition type j

Now we have to ~~g~~ formulate the optimization problem,

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1M} \\ \vdots & & & \\ a_{N1} & & & a_{NM} \end{bmatrix}, \quad A \in \mathbb{R}^{N \times M}$$

so A is the matrix of nutrition information of each food.

We need to find minimum grocery cost to satisfy the nutrition requirement;

to satisfy nutrition, the total nutrition should be at least b_j = $B = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_M \end{bmatrix}$.

let say X is the list of grocery items for each item, of food type i, so $X = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$

When x_i is the amount of ~~a~~ food items of i

so total nutrition is ~~$A^T x$~~ $A^T X$

so, ~~$A^T x$~~ $A^T x$ should at least be B

so $A^T X \geq B$ is the first constraint for nutrition requirement.

Which then gives us the cost too,

~~So~~ ~~to~~ the cost is ~~AB~~ $\cdot C^T X$, where

$C = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{bmatrix}$, ~~?~~ vector of cost of each grocery item.

so $C^T X$ is the total cost of all grocery item.

so we have to minimize $C^T X$ for which X has a solution for $A^T x \geq B$

so if $A^T x \geq B$ has a solution then this ~~m~~ means

$A^T x = B$ has a solution too, so ~~?~~ for $A^T x = B$ we get the minimum x value, which then ~~cor~~ corresponds to minimum cost value in $C^T X$

so $A^T x = B$

$\Rightarrow (A^T)^{-1} A^T x = (A^T)^{-1} B$

given A is full rank, $(A^T)^{-1} A^T = I$

and ~~?~~ $(A^T)^{-1} = (A^{-1})^T$

$\Rightarrow X = (A^T)^{-1} B$

so substituting x in $C^T x$.

$\Rightarrow$ ~~?~~ minimum cost is $C^T ((A^T)^{-1} B)$

$\Rightarrow C^T (A^{-1})^T B \Rightarrow (A^{-1} C)^T B$

so ~~AB~~ $(A^{-1} C)^T B$ is the minimum cost.