

Design Optimization: HW1

Seyed Yousef Soltanian

you can find the file at

https://github.com/YousefSoltanian/MAE598_Design_Optimization.

Problem 1:

```
In [2]: import numpy as np
        from scipy.optimize import minimize, LinearConstraint
```

Here we define the given objective function:

```
In [27]: def objective_function(x):
        return ((x[0]-x[1])**2 + (x[1]+x[2]-2)**2 + (x[3]-1)**2 + (x[4]-1)**2)
```

Defining all the constraints:

```
In [39]: np.random.seed(2)
        n_x = 5

        # generating the initial guess:
        x = np.random.random(n_x)

        # creating matrix for equality conditions
        A = np.array([[1,2,0,0,0],[0,0,1,1,-2],[0,1,0,0,-1]])
        lb=np.zeros((3,))
        ub=np.zeros((3,))
        test = np.ones((5,))

        # creating constraints
        constraint = LinearConstraint(A, lb=lb, ub=ub)

        # creating bounds
        bounds = [(-10, 10) for n in range(n_x)]

        #Solving the minimization problem
        ans = minimize(
            objective_function,
            x0=10 * np.random.random(n_x),
            constraints=constraint,
            bounds=bounds,
        )
        print(ans)
```

```

message: Optimization terminated successfully
success: True
status: 0
  fun: 3.4137934799966376
    x: [-7.583e-01  3.792e-01  6.898e-01  6.853e-02  3.792e-01]
  nit: 7
  jac: [-2.275e+00  4.130e-01 -1.862e+00 -1.863e+00 -1.242e+00]
 nfev: 43
 njev: 7

```

Obviously, by keep running the above cell and changing the initial guess, we can see that the final result is the same. We were expecting this result, because by looking at the objective function, we can easily see that the the function is convex, and we know that a convex function always has a single global optimum.

Problem 2:

a)

For the gradient and Hessian we can write:

$$\begin{aligned}
 f(x) &= b^T x + x^T A x \\
 \nabla f(x) &= b + (A^T + A)x \\
 Hf(x) &= A^T + A
 \end{aligned}$$

b)

Based on the previous part we can write:

$$Firstorderapproximation = f(0) + \nabla f(0)^T x = 0 + (b + 0)^T (x + 0) = b^T x$$

$$\begin{aligned}
 Secondorderapproximation &= f(0) + \nabla f(0)^T x + \frac{1}{2} x^T Hf(0) x \\
 &= 0 + (b + 0)^T (x + 0) + \frac{1}{2} x^T (A^T + A) x = b^T x + \frac{1}{2} x^T (A^T + A) x \\
 &= b^T x + x^T A x
 \end{aligned}$$

Obviously, the first order approximation is not exact and missing the second term. But the second order approximation is exactly equal to the original function, this results is trivial as we now that a second order function only has 2 term in its Teylor series approximation and all other higher order terms are zero!

Problem 3:

a)

The necessary and sufficient condition for a square matrix A to be positive definite are as below:

1. The matrix must be symmetric.
2. All eigenvalues are strictly positive.

b)

The necessary and sufficient condition for a square matrix A to be Full Rank is that all the eigenvalues of the A must be non-zero, basically meaning that all of columns of the A are linearly independent.

c)

if $Ay = 0$, it means that y is in null space of the matrix A, now if we want the equation $Ax = b$ to have an answer, b should not be in the null space of A and it should be in the column space of A. In another word, b should be orthogonal to the null space of A, which means that b is orthogonal to y, which results : $b^T y = 0$.

Problem 4

For this problem, we can think of a vector x with the length of N. x_i represent the amount of food i that we buy. so the cost function to minimize can be written as:

$$\text{Minimize}_x : f(x) = \sum_{i=1}^N x_i c_i$$

But we also do have conditions and constraints, the amount of nutrition that we get from all the food that we buy should be more than the necessary amount of each nutrition that we need for a month. this condition can be written as below:

$$\text{Subjected to : } \sum_{i=1}^N a_{ij} x_i \geq b_j$$