**Tutorial- 5:**

**Non linear Programming using Python**

Min x1x4 (x1+x2+x3)+x3

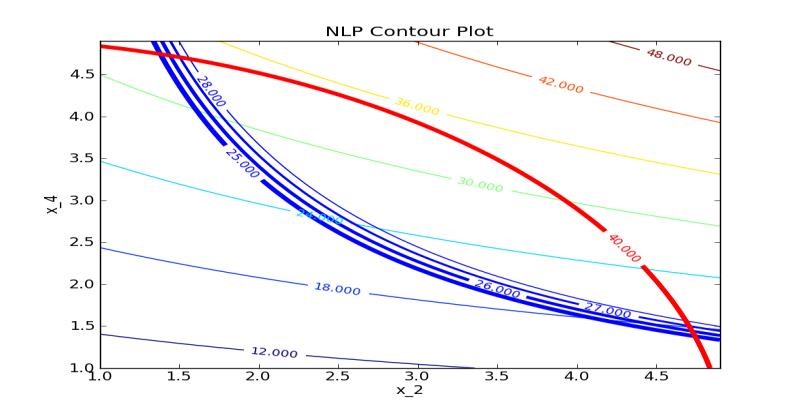
s.t. x1x2x3x4 ≥25

x21+x22 +x23+x24 =40

1≤x1,x2,x3,x4≤5

x0=(1,5,5,1)

This problem has a nonlinear objective that the optimizer attempts to minimize. The variable values at the optimal solution are subject to (s.t) both equality(=40) and inequality (>25) constraints. The product of the four variables must be greater than 25 while the sum of squares of the variables must also equal 40. In addition, all variables must be between 1 and 5 and the initial guess is x1 = 1, x2 = 5, x3 = 5, and x4 = 1.



For this problem determine:

1. A potential feasible solution
2. Identify the constraints on the contour plot
3. Mark the set of feasible solutions on the contour plot
4. Identify the minimum objective feasible solution
5. Identify the maximum objective feasible solution
6. Use a nonlinear programming solver to find a solution

Example 2:

maximise 2x1 + x2 - 5loge(x1)sin(x2)

subject to

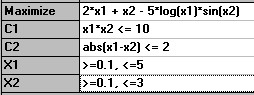
x1x2 <= 10

| x1 - x2 | <= 2

0.1 <= x1 <= 5

0.1 <= x2 <= 3

Here we have a nonlinear objective and nonlinear constraints. Using the package we have the input:



and it clear from this there there are values of x1 and x2 that exceed the supposed maximum objective function value of 8.8166 given by the package.

**CODE**

import numpy as np

from scipy.optimize import minimize

def objective(x):

  return x[0]\*x[3]\*(x[0]+x[1]+x[2])+x[2]

def constraint1(x,sign=1.0):

  return x[0]\*x[1]\*x[2]\*x[3]-25.0

def constraint2(x,sign=1.0):

  sum\_eq = 40.0

  for i in range(4):

      sum\_eq = sum\_eq - x[i]\*\*2

  return sum\_eq

#initial guess

n = 4

x0 = np.zeros(n)

x0[0] = 1.0

x0[1] = 5.0

x0[2] = 5.0

x0[3] = 1.0

print('Initial SSE Objective:'+str(objective(x0)))

**#output:-** Initial SSE Objective:16.0

b=(1.0,5.0)

bnds=(b,b,b,b)

con1={'type':'ineq','fun': constraint1}

con2={'type':'eq','fun': constraint2}

cons=([con1,con2])

solution=minimize(objective,x0,method='SLSQP',bounds=bnds,constraints=cons)

x=solution.x

print('Final SSE Objective:' + str(objective(x)))

**#output:-** Final SSE Objective:17.01401724563517

print('Solution')

print('x1='+str(x[0]))

print('x2='+str(x[1]))

print('x3='+str(x[2]))

print('x4='+str(x[3]))

**#output:-**

Solution

x1=1.0

x2=4.742996096883977

x3=3.8211546234095715

x4=1.379407645075325

