Zainab Al-Suwaykit
Daniel González Esparza

AMCS CS 212
Numerical Optimization
Assignment

1. Exercise in R3. Write and solve the first order optimality conditions of the following problem.

minimize x1x2 + x2x3
subjectto x21+x2−2=0
x21 +x23 −2=0,

H5-1

First order optimality conditions.

-For non-linear equality constraints,

1- $\nabla_x L = 0$

2- $\nabla_\nu L = 0$

That is, $Z(x^*)^T \nabla f(x^*) = 0$

$$L(x,\nu) = f(x) + \sum_{i}^{p} \nu_i h_i(x) = f(x) + \nu^T h(x)$$

$$(x_1 x_2 + x_2 x_3) + \nu_1 (x_1^2 + x_2^2 - 2) + \nu_2 (x_1^2 + x_3^2 - 2)$$

1-

$$\nabla_{x_1} L = x_2 + 2\nu_1 x_1 + 2\nu_2 x_1 = 0$$

$$\nabla_{x_2} L = (x_1 + x_3) + 2\nu_1 x_2 = 0$$

$$\nabla_{x_3} L = (x_2) + 2\nu_2 x_3 = 0$$

2-

$$\nabla_{\nu_1} L = x_1^2 + x_2^2 - 2 = 0$$

$$\nabla_{\nu_2} L = x_1^2 + x_3^2 - 2 = 0$$

Zainab Al-Suwaykit
Daniel González Esparza

```
from scipy.optimize import fsolve

def equations(p):
    x1,x2,x3,l1,l2 = p
    return(x2 + 2*l1*x1 + 2*l2*x1, x1+x3 + 2*l1*x2, x2+ 2*l2*x3, x1**2+x2**2 -2,
           x1**2 +x3**2 -2)
x1,x2,x3,l1,l2 = fsolve(equations, (1,1,1,1,1))
print ("x1: ",x1)
print("x2: ",x2)
print("x3: ",x3)
print("lamda 1: ",l1)
print("lamda 2: ",l2)
```

```
x1:  0.5411961003879037
x2:  1.3065629647160462
x3:  1.3065629649003845
lamda 1:  -0.7071067812852867
lamda 2:  -0.4999999998705373
```

2. Significance of the Lagrange Multipliers. Consider the equality-constrained problem:

minimize f (x)

subject to hi(x) = 0, i = 1,...,p

5-2.

$f_1^* = f^* + \sum \nu_i h_i(x^*)$

$f_2^* = f^* + \nu_1 h_1 + \nu_2 h_2 + \nu_3 h_3 + \nu_i(h_i - \varepsilon) + \nu + \dots$

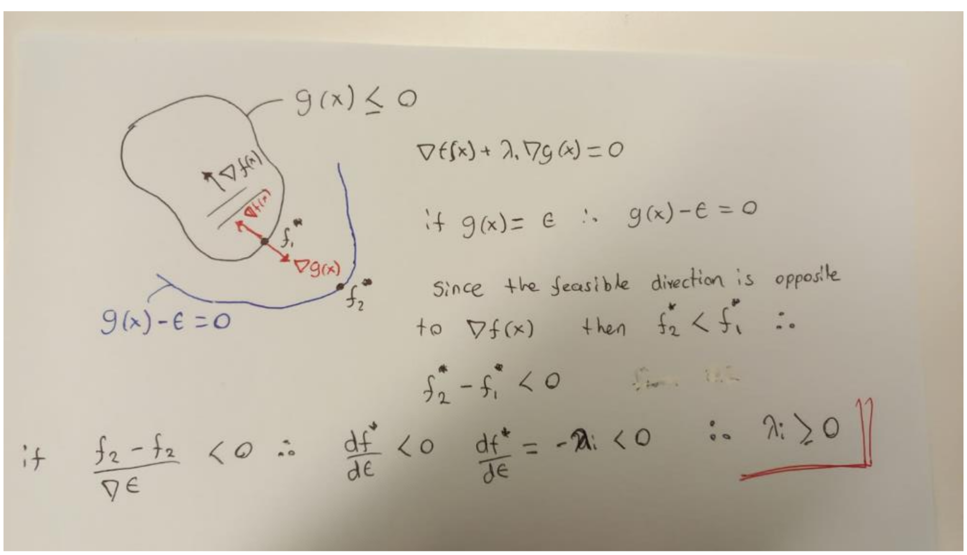
$\Delta f^* = f_2^* - f_1^*$

$= \nu_i(h_i - e) - \nu_i h_i$

$\Delta f^* = \nu_i(h_i - e - h_i)$

$= -\nu_i \varepsilon$

3. Nonnegativity of the Lagrange Multipliers. Explain with the aid of an appropriate diagram why it is not possible for the Lagrange multiplier of an inequality constraint to be negative at an optimal point.



Zainab Al-Suwaykit
Daniel González Esparza

4. Necessary but not sufficient conditions. Verify that the first order optimality conditions of the following problem are satisfied at the points $(-2, 2)$ and $(2, -2)$ but yet neither is a solution.

$$\text{minimize } x_1 - x_2$$
$$\text{subject to } x_1 x_2 + 4 = 0$$

Zainab Al-Suwaykit
Daniel González Esparza

H5-4.

The first order optimality

minimize $x_1 - x_2$

subject to $x_1 x_2 + 4 = 0$

1- $\nabla_x L = 0$

2- $\nabla_\nu L = 0$

Lagrangian:

$L(x, \nu) = f(x) + \sum^{p} \nu_i h_i(x)$

$= f(x) + \nu^T h(x)$

$\rightarrow (x_1 - x_2) + \nu_1 (x_1 x_2 + 4)$

$\nabla_{x_1} L = 1 + \nu_1 x_2 = 0$

$\nabla_{x_2} L = -1 + \nu_1 x_1 = 0$

$\nabla_\nu L = x_1 x_2 + 4 = 0$

At $(-2, 2)$:-

$\left.\begin{array}{l} \nabla_{x_1} L = 1 + 2\nu_1 = 0 \\ \nabla_{x_2} L = -1 + -2\nu_1 = 0 \end{array}\right\}$ satisfied

$\nabla_\nu L = (-2)(2) + 4 = 0$ ✓ satisfied

At $(2, -2)$:-

$\left.\begin{array}{l} \nabla_{x_1} L = 1 - 2\nu_1 = 0 \\ \nabla_{x_2} L = -1 + 2\nu_1 = 0 \end{array}\right\}$ satisfied

$\nabla_\nu L = (2)(-2) + 4 = 0$ satisfied

At the optimal point, $\nabla f(x^*) \perp Z(x^*)$

$\nu = \pm \frac{1}{2}$

$\nabla f(-2, 2) = 0$?

$A(x^*) = \begin{bmatrix} x_2 \\ x_1 \end{bmatrix} = \begin{bmatrix} 2 \\ -2 \end{bmatrix} \times \frac{1}{2} = 0$

$\begin{bmatrix} 1 \\ -1 \end{bmatrix} \neq 0$ does not satisfy the cond.

$\nabla f(2, -2) = 0 + \begin{bmatrix} -2 \\ 2 \end{bmatrix} \times -\frac{1}{2} =$

$\begin{bmatrix} 1 \\ -1 \end{bmatrix} \neq 0$ does not satisfy the condition.

Zainab Al-Suwaykit
Daniel González Esparza

5. Healthy Snack. Consider the problem of purchasing afternoon snacks. Health conscious buyers need at least 6 total ounces of chocolate, 10 ounces of sugar, and 8 ounces of cream cheese. There are 2 choices of snacks: brownies and cheescakes whose ingredients are listed below. Brownies cost 50 cents and mini-cheesecakes cost 80 cents.

- Formulate the minimum-cost healthy purchase snack problem as a linear optimization problem,

```
c = [50,80]
A = [[-3,0],[-2, -4], [-2, -5]]
b = [-6,-10, -8]
x0_bounds = (2, None)
x1_bounds = (0, None)
```

```
from scipy.optimize import linprog, minimize
```

```
res = linprog(c, A_ub=A, b_ub=b, bounds=(x0_bounds, x1_bounds), options={"disp":
 ↪ True})
```

```
Primal Feasibility  Dual Feasibility    Duality Gap         Step
Path Parameter      Objective
1.0                 1.0                 1.0                 -                 1.0
230.0
0.1451752904024     0.1451752904024     0.1451752904024     0.8621486037553
0.1451752904024     212.1498194428
0.007313214478175   0.007313214478174   0.007313214478174   0.9838194432084
0.007313214478173   219.6076486779
3.559976278034e-06  3.559976266184e-06  3.559976266143e-06  0.9995460886433
3.559976274529e-06  220.0000629544
1.779656660641e-10  1.779816031243e-10  1.779815718046e-10  0.9999500048381
1.779995253049e-10  220.0000000031
Optimization terminated successfully.
         Current function value: 220.000000
         Iterations: 4
```

```
print(res)
```

```
     con: array([], dtype=float64)
     fun: 220.00000000314566
 message: 'Optimization terminated successfully.'
     nit: 4
   slack: array([1.99070982e-10, 1.24105171e-10, 3.50000000e+00])
  status: 0
 success: True
       x: array([2. , 1.5])
```

- Write out explicitly the Lagrangian for this problem, as well as the optimality conditions.
- Compute the values of the Lagrange multipliers from the optimality conditions above. What is their physical interpretation in this problem? Comment on their values.

Zainab Al-Suwaykit
Daniel González Esparza

H5-5.

- Formulate the minimum-cost problem.

    Python code.

- Lagrangian, optimality conditions.

Brownie $\rightarrow x_1$

Cheesecake $\rightarrow x_2$

$50x_1 + 80x_2$ ← minimize $x = [x_1, x_2]$

subject to:

$3x_1 \geq 6 \quad \rightarrow \quad -3x_1 + 6 \leq 0$

$2x_1 + 4x_2 \geq 10 \quad \rightarrow \quad -2x_1 - 4x_2 + 10 \leq 0$

$2x_1 + 5x_2 \geq 8 \quad \rightarrow \quad -2x_1 - 5x_2 + 8 \leq 0$

$x_1, x_2 \geq 0$

- Lagrangian: (no (=) constraints)

$$L(x, \lambda) = f_0(x) + \sum_{i}^{m} \lambda_i f_i(x)$$

$$= 50x_1 + 8x_2 + \lambda_1(-3x_1 + 6) + \lambda_2(-2x_1 - 4x_2 + 10) + \lambda_3(-2x_1 - 5x_2 + 8)$$

- optimal conditions: (KKT) I

1. $f_i(x) \leq 0$ ← done
2. $\lambda_i \geq 0$
3. $\lambda_i f_i(x) = 0$
4. $\nabla f_0(x) + \sum_{i}^{m} \lambda_i \nabla f_i(x) = 0$

Zainab Al-Suwaykit
Daniel González Esparza

$$= [50, 80] + \lambda_1([-3, 0]) + \lambda_2([-2, -4]) + \lambda_3([-2, -5]) = 0$$

- Lagrangian multipliers:

$\lambda_1(-3x_1 + 6) = 0$

$\lambda_2(-2x_1 - 4x_2 + 10) = 0$

$\lambda_3(-2x_1 - 5x_2 + 8) = 0$

$x_1 = 2, \quad x_2 = 1.5$

→ code.

$\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 0$

Thus, we have two active constraints and the rest can be removed.

the first two constraints can be turned to equality.

```
#Lagrangian multipliers
from scipy.optimize import fsolve

def equations(p):
    x1,x2 = 2, 1.5
    l1,l2,l3 = p
    return(l1*(-3*x1 + 6), l2*(-2*x1 -4*x2 +10), l3*(-2*x1 - 5*x2 +8))
l1,l2,l3 = fsolve(equations, (1,1,1))
print (l1,l2,l3)
```

```
1.0 1.0 0.0
```

- Convert the problem to canonical form (i.e., equality constraints with zero-lower bound inequality constraints). This is the form used internally in the solution.

- Canonical form.

minimize $50x_1 + 80x_2$

subject to: $-3x_1 + 6 + S_1 = 0$

$-2x_1 - 4x_2 + 10 + S_2 = 0$

$S_1 = 1.99070982e-10$

$S_2 = 1.24105171e-10$

6. Center of Polyhedron.

Zainab Al-Suwaykit
Daniel González Esparza

- Formulate the problem as a linear programming problem. Hint. A point x is at a distance R from thelineiifaTi x−bi =−R‖ai‖.
- Use the data sample below and solve the problem.

```
import numpy as np
from scipy.optimize import linprog, minimize
import cvxpy as cp
from numpy import linalg as la
#Using linprog
A = np.array( [[0, -1], [2, -1], [1, 1], [-1/3, 1], [-1, 0], [-1, -1]] )
b = np.array( [0, 8, 7, 3, 0, -1])
rowNrm = np.sqrt(np.sum(A**2, axis=1))
matA = np.vstack((rowNrm, A.T)).T
C = np.array([-1, 0, 0])
res = linprog(C, A_ub=matA, b_ub=b)
print(res)
```

```
     con: array([], dtype=float64)
     fun: -1.8655647843427605
 message: 'Optimization terminated successfully.'
     nit: 5
   slack: array([-1.88109084e-10,  7.01778719e-01,  1.05820686e-09,  5.65354874e-10,
        6.30563411e-01,  7.23385960e-01])
  status: 0
 success: True
       x: array([1.86556478, 2.4961282 , 1.86556478])
```

- what is the significance of the Lagrange multipliers in your solution.

```
# using cvxpy
# variables
radius = Variable(1)
center = Variable(2)
# constraints
a1 = [0, -1]
a2 = [2, -1]
a3 = [1, 1]
a4 = [-1/3, 1]
a5 = [-1, 0]
a6 = [-1, -1]
b = [0, 8, 7, 3, 0, -1]
constraints = [ np.transpose(a1)*center + np.linalg.norm(a1)*radius <= b[0],
                np.transpose(a2)*center + np.linalg.norm(a2)*radius <= b[1],
                np.transpose(a3)*center + np.linalg.norm(a3)*radius <= b[2],
                np.transpose(a4)*center + np.linalg.norm(a4)*radius <= b[3],
                np.transpose(a5)*center + np.linalg.norm(a5)*radius <= b[4],
                np.transpose(a6)*center + np.linalg.norm(a6)*radius <= b[5]]
objective = cp.Minimize(-1*radius)
p = Problem(objective, constraints)
result = p.solve()
print("radius: ",radius.value)
print("center: ",center.value)
l = np.zeros((6,1))
for i in range (0,6):
    l[i] =constraints[i].dual_value
    print("lamda ", i+1, " " ,constraints[i].dual_value)
print("Lamda 2,5,6 are almost zeros so they are not active and does not affect the solution, hence can be removed")
```

```
radius:  [1.86556479]
center:  [2.4961282  1.86556478]
lamda  1   [0.46639119]
lamda  2   [6.16655976e-09]
lamda  3   [0.11659779]
lamda  4   [0.3497934]
lamda  5   [1.70309948e-09]
lamda  6   [3.27651278e-10]
Lamda 2,5,6 are almost zeros so they are not active and does not affect the solution, hence can be removed
```

- Write out the Lagrangian for this problem as well as the optimality conditions. Verify that the solution satisfies the optimality conditions.

Zainab Al-Suwaykit
Daniel González Esparza

Q6:

Lagrangian:-

$$L(x, \nu, \lambda) = f_0(x) + \sum_{i}^{p} \nu_i h_i(x) + \sum_{i}^{m} \lambda_i f_i(x)$$

minimize: $-r$

s.t. $[0,-1]^T x + \|[0,1]\| * r \leq 0$

$[2,-1]^T x + \|[2,-1]\| * r \leq 8$

$[1,1]^T x + \|[1,1]\| * r \leq 7$

$[-\frac{1}{3},1]^T x + \|[-\frac{1}{3},1]\| * r \leq 3$

$[-1,0]^T x + \|[-1,0]\| * r \leq 0$

$[-1,-1]^T x + \|[-1,-1]\| * r \leq -1$

optimality conditions:-

$\lambda_i f_i(x) = 0$

*in the code

$$\nabla f_0(x) + \sum_{1}^{m} \lambda_i \nabla f_i(x) = 0$$

$$\rightarrow L(r, x_1, x_2, \lambda) = -r + \lambda_1(-x_2 + r)$$
$$+ \lambda_2(2x_1 - x_2 + \sqrt{5}\, r - 8)$$
$$+ \lambda_3(x_1 + x_2 + \sqrt{2}\, r - 7)$$
$$+ \lambda_4(-\tfrac{1}{3}x_1 + x_2 + \tfrac{\sqrt{10}}{3} r - 3)$$
$$+ \lambda_5(-x_1 + r)$$
$$+ \lambda_6(-x_1 - x_2 + \sqrt{2}\, r)$$

$$\nabla_{x_1} = 2\lambda_2 + \lambda_3 + \tfrac{-1}{3}\lambda_4 - \lambda_5 - \lambda_6 = 0$$
$$\nabla_{x_2} = -\lambda_1 - \lambda_2 + \lambda_3 + \lambda_4 - \lambda_6 = 0$$
$$\nabla_r = -1 + \lambda_1 + \sqrt{5}\lambda_2 + \sqrt{2}\lambda_3 + \tfrac{\sqrt{10}}{3}\lambda_4 + \lambda_5 + \sqrt{2}\lambda_6 = 0$$

Zainab Al-Suwaykit
Daniel González Esparza

```
#the soultion satisfies the optimality conditions.
#optimality conditions: lamda_i f_i(x)=0
print(l[0]* (np.transpose(a1)@center.value + np.linalg.norm(a1)*radius.value ) )
#print(l[1]* (np.transpose(a2)*center.value + np.linalg.norm(a2)*radius.value - b[1]) )
print(l[2]* (np.transpose(a3)@center.value + np.linalg.norm(a3)*radius.value - b[2]) )
print(l[3]* (np.transpose(a4)@center.value + np.linalg.norm(a4)*radius.value - b[3]) )
#print(l[4]* (np.transpose(a5)*center.value + np.linalg.norm(a5)*radius.value ) )
#print(l[5]* (np.transpose(a6)*center.value + np.linalg.norm(a6)*radius.value + b[5]) )
```

```
[1.29228531e-09]
[1.12252997e-09]
[-5.20904991e-10]
```

```
#gradient L=0
gr = -1+ l[0]  + np.sqrt(2)*l[2] + (np.sqrt(10)/3)*l[3]
gx1 =  l[2] - (1/3)*l[3]  #+ 2*l[2] + (2/3)*l[3]
gx2 = -1* l[0]  + l[2] + l[3]
print("the gradient of the lagrangian is almost zero, thus the optimality conditions are satisfied")
print("the gradient with respect to r: ", gr)
print("the gradient with respect to x: ", gx1)
print("the gradient with respect to y: ", gx2)
print(np.sqrt(gr**2 + gx1**2 + gx2**2))
```

```
the gradient of the lagrangian is almost zero, thus the optimality conditions are satisfied
the gradient with respect to r:  [-1.51146274e-08]
the gradient with respect to x:  [-1.03324936e-08]
the gradient with respect to y:  [6.4188197e-09]
[1.94013822e-08]
```

7. Minimum Cost Flow.

- Solve the problem for the graph below where we wish to send 4 units from node 1 to node 3.

```
from scipy.optimize import linprog, minimize
import numpy as np
```

```
q=4
c = np.array([5,2,1,2,4])
A = np.array([[1,1,0,0,0],[-1, 0,1,1,0], [0, 0,0,-1,-1], [0, -1,-1,0,1]])
p = np.array([2,5, 3,7,1])
b = np.array([q,0,-q,0])
I=np.identity(5)
x0_bounds = (0, None)
x1_bounds = (0, None)
```

```
res = linprog(p, A_ub=I, b_ub=c, A_eq=A, b_eq=b, bounds=(x0_bounds, x1_bounds, x1_bounds, x1_bounds, x1_bounds

print(res)
```

```
<ipython-input-153-81fd8457739a>:1: OptimizeWarning: A_eq does not appear to be of full row rank. To improve
performance, check the problem formulation for redundant equality constraints.
  res = linprog(p, A_ub=I, b_ub=c, A_eq=A, b_eq=b, bounds=(x0_bounds, x1_bounds, x1_bounds, x1_bounds, x1_bou
nds),options={"disp": True})
Primal Feasibility  Dual Feasibility    Duality Gap         Step             Path Parameter      Objective
1.0                 1.0                 1.0                 -                1.0                 18.0
0.1562396059839     0.1562396059839     0.1562396059839     0.8522676758958  0.1562396059839     24.448897564
01
0.02131673041496    0.02131673041479    0.02131673041479    0.8758665804189  0.02131673041538    26.478537692
59
3.52784270894e-05   3.527842709009e-05  3.527842709002e-05  0.998468337561   3.527842709536e-05  26.999407512
84
1.763995374892e-09  1.763995700236e-09  1.763995639617e-09  0.999949997893   1.763996479459e-09  26.999999970
37
Optimization terminated successfully.
         Current function value: 27.000000
         Iterations: 4
     con: array([ 6.73525191e-09, -3.36760997e-09, -6.73525857e-09,  3.36761641e-09])
     fun: 26.999999970373853
 message: 'Optimization terminated successfully.'
     nit: 4
   slack: array([ 3.00000000e+00,  6.80593004e-09, -3.21077431e-09,  1.00000000e+00,
        1.00000001e+00])
  status: 0
 success: True
       x: array([2.        , 1.99999999, 1.        , 1.        , 2.99999999])
```

```
res.x
```

```
array([2.50000003, 1.99999993, 1.00000001, 1.50000003, 2.99999993])
```

- What is the significance of the multipliers of the capacity constraints? What is the significance of the multipliers of the node constraints?

Zainab Al-Suwaykit
Daniel González Esparza

5-7.

C_i is max capacity

P_i cost

$p^T x$ ← minimize

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & -1 & -1 \\ 0 & -1 & -1 & 0 & 1 \end{bmatrix}$$

$Ax = [9\ 0\ 9\ 0]^T$

$Ix \le C = [5\quad 2\quad 1\quad 2\quad 4]^T$

$P = [2\quad 5\quad 3\quad 7\quad 1]^T$

Since slack variables for $\lambda_1, \lambda_4, \lambda_5$ are zero, thus are not active.

if $f_2(x) = \varepsilon$

$\Delta f^* = -0.3 \approx -\varepsilon \lambda_2 = -(0.1)(\lambda_2) \to \lambda_2 \approx 3$

if $\varepsilon = 0.1$ and $f_3(x) = \varepsilon$

$\Delta p^*(x) = -0.3 \approx -\varepsilon \lambda_3 \to \lambda_3 = 3$

$\therefore \lambda = [0\quad 3\quad 3\quad 0\quad 0]^T$

$h_1(x) = \varepsilon = 0.5$

$h_3(x) = \varepsilon = 0.5$

$\Delta \varphi^* \approx \varepsilon(\nu_1 + \nu_3) \approx 31.5 - 27$

$\nu_1 + \nu_3 \approx 9$

Zainab Al-Suwaykit
Daniel González Esparza

$$2 + \nu_1 - \nu_2 = 0$$
$$5 + \nu_1 - \nu_4 + 8 = 0$$
$$3 + \nu_2 - \nu_4 + 3 = 0$$
$$7 + \nu_2 + \nu_3 = 0$$
$$1 - \nu_3 + \nu_4 = 0$$

$$\nu_2 = \nu_1 + 2$$
$$\nu_4 = \nu_1 + 8$$
$$\nu_3 = \nu_1 + 9$$

$$7 + (2 + \nu_1) - \nu_3 = 0$$
$$9 + \nu_1 - \nu_3 = 0$$

$$\nu_2 = \nu_1 + 2$$
$$\nu_3 = \nu_1 + 9$$
$$\nu_4 = \nu_1 + 8$$

$$\nu_1 + \nu_3 \approx 9$$
$$\nu_3 = \approx 9 - \nu_1$$
$$\cancel{9} - \nu_1 \approx \nu_1 + \cancel{9}$$
$$\nu_1 \approx 0$$
$$\nu_2 \approx 2$$
$$\nu_3 \approx 9$$
$$\nu_4 \approx 8$$

Considering the results for the inequality constrains we realize that lambda 1, 4 and 5 are equal zero, what means that they are not active and, therefore, do not affect the solution because of the the high price per amount of good shipped. Lambda 2 and 3 are equal three and the solutions depend on those variables due to the limitation of capacity and low cost through the edges.

For the equality constrains We observe that the multiplier 3 affects more the solution because has a larger multiplier and all units sent depends directly of it. On the other hand the second node more important is the number 4, since one of its edges has the minimum price, therefor, the units sent will tend to cross through this edge to minimize the objective function.