Numerical Analysis Hw 11 Kevin Ho

HW Use the method of Lagrange multipliers to solve the problem.

$$\min_{x_1, x_2} x_1^2 + x_2^2, \text{ subj. to } \begin{cases} x_1 \geq 1 \\ x_2 \geq 2 \end{cases}$$

Ans.: we rewrite the problem as follows.

$$\mathcal{L}(x_1, x_2, \alpha_1, \alpha_2) = x_1^2 + x_2^2 - \alpha_1(x_1 - 1) - \alpha_2(x_2 - 2) \quad , \quad \alpha_1, \alpha_2$$

$$\max_{\alpha \geq 0} \min_{x} \mathcal{L}(x, \alpha) \qquad \nabla \mathcal{L}(x, \alpha) = \begin{bmatrix} 2x_1 - \alpha_1 \\ 2x_2 - \alpha_2 \end{bmatrix} = 0$$

$$2x_1 - \alpha_1 = 0 \qquad \frac{\alpha_1}{2} = x_1 \qquad \text{Sing}$$

$$2x_2 - \alpha_2 = 0 \qquad \frac{\alpha_2}{2} = x_2$$

$$\mathcal{L}(x_1, x_2, \alpha_1, \alpha_2) = \frac{\alpha_1^2}{4} + \frac{\alpha_2^2}{4} - \alpha_1\left(\frac{\alpha_1}{2} - 1\right) - \alpha_2\left(\frac{\alpha_2}{2} - 2\right)$$

$$\max_{\alpha_1, \alpha_2 \geq 0} = -\frac{\alpha_1^2}{4} + \alpha_1 - \frac{\alpha_2^2}{4} + 2\alpha_2 \Rightarrow \alpha_1 = 2, \alpha_2 = 4$$

$$\Rightarrow x_1 = 1 \quad x_2 = 2$$

So $x = 5$ ✓

.2: min $x_1^2 + x_2^2$ subj to $x_1 + x_2 = 5$.

a) Express the problem in the matrix form as in 11.20 $C = [1\ 1]$

$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ $c = 5$ $b = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

b) $\begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$,

$[1\ 1] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 5$

c) $\begin{bmatrix} A & C^T \\ C & 0 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 1 & 0 \end{bmatrix}$, $\begin{bmatrix} b \\ c \end{bmatrix} = \begin{pmatrix} 0 \\ 0 \\ 5 \end{pmatrix}$

So ~~Constants~~ Solution is $x_1 = x_2 = 2.5$, $\lambda^* = 5$

d) $\begin{bmatrix} 2 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 1 & 0 \end{bmatrix}^{-1} \begin{pmatrix} 0 \\ 0 \\ 5 \end{pmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \lambda \end{bmatrix} = \begin{bmatrix} 2.5 \\ 2.5 \\ -5 \end{bmatrix}$

$\downarrow$

$\begin{bmatrix} \frac{1}{4} & -\frac{1}{4} & \frac{1}{2} \\ -\frac{1}{4} & \frac{1}{4} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & -1 \end{bmatrix}$

$2.5 + 2.5 = 5$ ✓

11.3: d.) So for the constraint line, it is the only tangent line for $c = 12.5$ with $x_1 = x_2 = 2.5$ thus an optimal solution to the problem. ✓

11.4: The given solution you gave is incorrect as it does not satisfy the constraints.

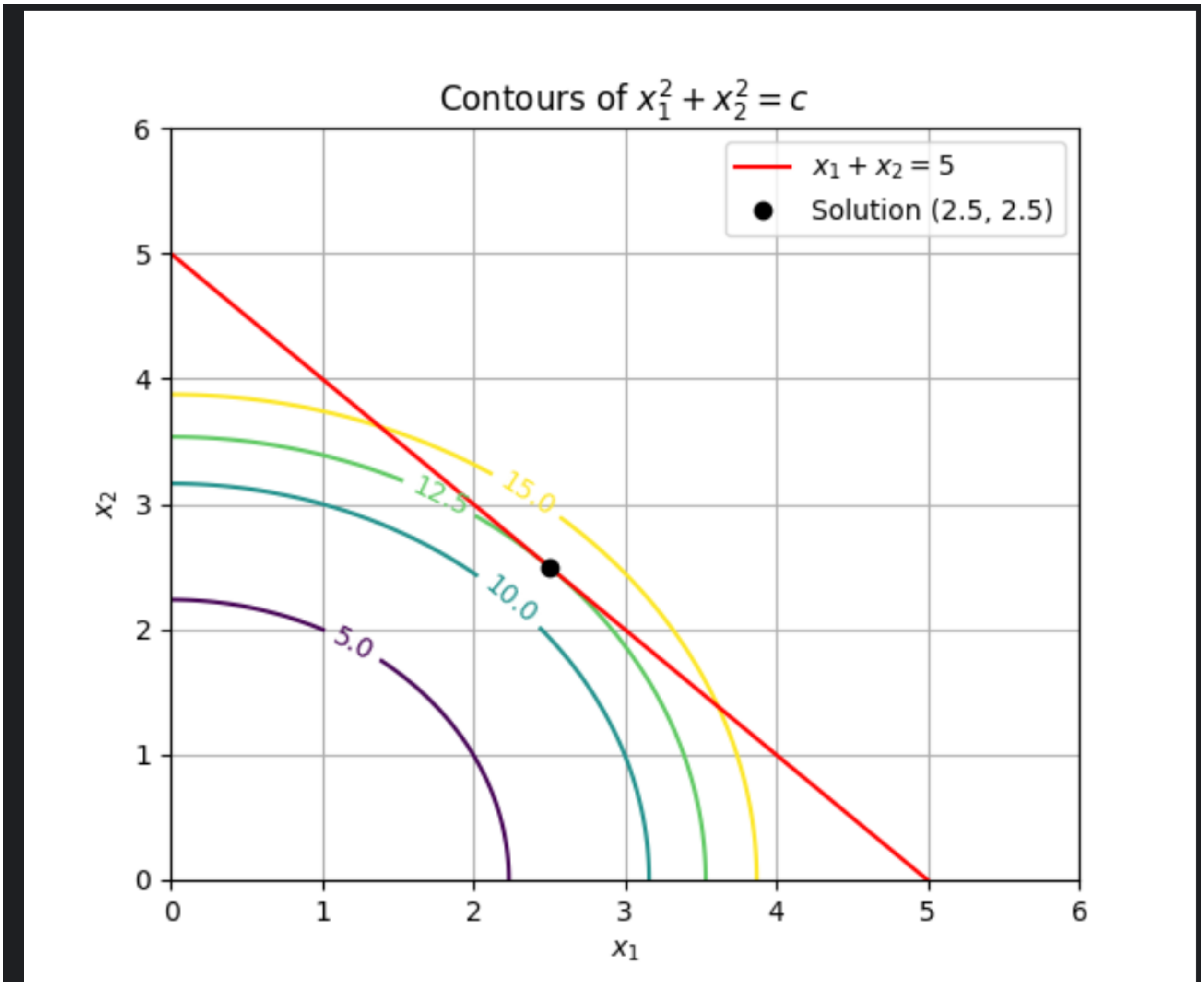
a) diag is $\begin{matrix} 4 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 2 \end{matrix}$ $\rightarrow A = \begin{bmatrix} 8 & -3 & 2 \\ -3 & 10 & 3 \\ 2 & 3 & 4 \end{bmatrix}$ $C = \begin{bmatrix} 2 & -1 & -3 \\ 1 & -3 & 3 \end{bmatrix}$ Rest is done in code.

b) $\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} 4 & -1.5 & 1 \\ -1.5 & 5 & 1.5 \\ 1 & 1.5 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + [x_1\ x_2\ x_3] \begin{bmatrix} -2 \\ 0 \\ 0 \end{bmatrix}$, $c = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

$\begin{bmatrix} 2 & -1 & -3 \\ 1 & -3 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$

Numerical Analysis HW 11

11.3)




```
import numpy as np
import matplotlib.pyplot as plt

# (a) Define a grid for x1, x2
x1_vals = np.linspace(0, 6, 200)
x2_vals = np.linspace(0, 6, 200)
X1, X2 = np.meshgrid(x1_vals, x2_vals)
Z = X1**2 + X2**2  # This is x1^2 + x2^2

# We'll plot contours for c in [5, 10, 12.5, 15]
contour_levels = [5, 10, 12.5, 15]

plt.figure(figsize=(6,5))
CS = plt.contour(X1, X2, Z, levels=contour_levels, cmap='viridis')
plt.clabel(CS, inline=True, fontsize=10)
plt.title(r"Contours of $x_1^2 + x_2^2 = c$")
```

```python
# (b) Constraint line: x1 + x2 = 5 => x2 = 5 - x1
x1_line = np.linspace(0, 5, 200)
x2_line = 5 - x1_line
plt.plot(x1_line, x2_line, 'r-', label=r"$x_1 + x_2 = 5$")

# (c) The solution to minimize x1^2 + x2^2 subject to x1 + x2 = 5
# occurs at x1 = x2 = 2.5 (by symmetry or by Lagrange multipliers).
x_star, y_star = 2.5, 2.5
plt.plot(x_star, y_star, 'ko', label=f"Solution ({x_star}, {y_star})")

plt.xlabel(r"$x_1$")
plt.ylabel(r"$x_2$")
plt.legend()
plt.grid(True)
plt.show()
```

D). So the minimum is (x1,x2) = (2.5,2.5) for a c=12.5. Geometrically, the smallest cirecle centered at the origin that touches the line x1 + x2 = 5 is tangent at (2.5,2.5) which is the solution.

11.4:

```
A:
 [[ 4.  -1.5  1. ]
 [-1.5  5.   1.5]
 [ 1.   1.5  2. ]]
b:
 [-2  0  0]
C:
 [[ 2 -1 -3]
 [ 1 -3  3]]
c:
 [ 1 -1]
**(b) Lagrangian Formulation On Paper
KKT Matrix =
 [[ 8. -3.  2.  2.  1.]
 [-3. 10.  3. -1. -3.]
 [ 2.  3.  4. -3.  3.]
 [ 2. -1. -3.  0.  0.]
 [ 1. -3.  3.  0.  0.]]
Right-hand side (RHS) =
 [ 2  0  0  1 -1]

CONSTRAINT CHECK:  1.0
CONSTRAINT CHECK:  -1.0

Optimal x: [ 0.2972973   0.22297297 -0.20945946]
Lagrange multipliers λ: [0.28378378 0.14189189]
Objective value f(x): 0.2263513513513514
```

```
import numpy as np

# Step 1: Define A, c
Q = np.array([
```

```python
    [ 4.0, -1.5,  1.0],
    [-1.5,  5.0,  1.5],
    [ 1.0,  1.5,  2.0]
])
A = 2 * Q  # so that 1/2 x^T A x = x^T Q x
c = np.array([-2,0,0])

# Step 2: Define C, d
C = np.array([
    [ 2, -1, -3],
    [ 1, -3,  3]
])
d = np.array([1, -1])

# Dimensions
n = 3  # number of variables
m = 2  # number of constraints

# Step 3: Build the KKT matrix
# K = [ A   C^T ]
#     [ C    0  ]
top_block = np.hstack([A, C.T])
bottom_block = np.hstack([C, np.zeros((m, m))])
K = np.vstack([top_block, bottom_block])

# Step 4: Build the RHS = [ -c, d ]^T = [0, d]^T since c=0
rhs = np.concatenate([ -c, d ])

# Step 5: Solve for (x, lambda)
solution = np.linalg.solve(K, rhs)
x_opt = solution[:n]
lambda_opt = solution[n:]

# Step 6: Compute the objective f(x) = x^T Q x = 1/2 x^T A x
f_opt = x_opt @ Q @ x_opt  # or 0.5 * x_opt @ A @ x_opt



print("\nA:\n", Q)
print("b:\n", c)
print("C:\n", C)
print("c:\n", d)



print("**(b) Lagrangian Formulation On Paper")

print("KKT Matrix =\n", K)
print("Right-hand side (RHS) =\n", rhs)

# E: Check if your solution satisfies the equality constraints
print("\nCONSTRAINT CHECK: ", np.dot(C[0], x_opt))
print("CONSTRAINT CHECK: ", np.dot(C[1], x_opt))
```

```python
# Print results
print("\nOptimal x:", x_opt)
print("Lagrange multipliers λ:", lambda_opt)
print("Objective value f(x):", f_opt)
```