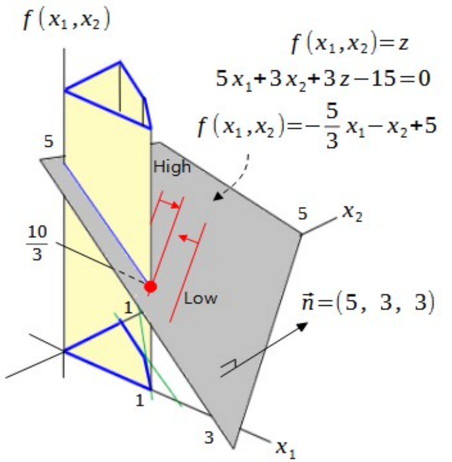
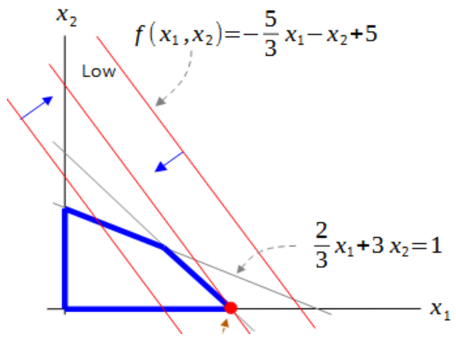
MX-AI





5. Convex Optimization

Quadratic Programming (QP)

Part 1: Overview of constrained QP problem

This video was produced in Korean and translated into English, and the audio was generated by AI (TTS).

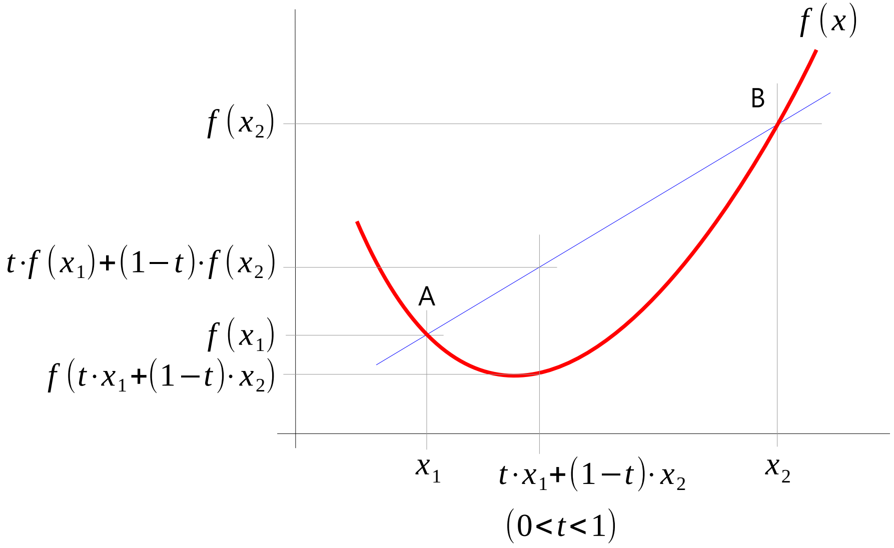
www.youtube.com/@meanxai

MX-AI

MX-AI

Overview: Convex function, linear function, affine function

Mathematical definition of Convex function



$$f(t\cdot x_1+(1-t)\cdot x_2) \le t\cdot f(x_1)+(1-t)\cdot f(x_2)$$

Linear function

We say a function $f : R^m \to R^n$ is linear if

1) for any vectors x and y in R^m, $f(x+y)=f(x)+f(y)$
2) for any vectors x in R^m and scalar a, $f(ax)=af(x)$

Example:

$f(x)=3x$

$f(2x+5y)=3(2x+5y)=2\times 3x+5\times 3y=2f(x)+5f(y)$

Affine function

We say a function $g : R^m \to R^n$ is affine if there is a linear function $f : R^m \to R^n$ and a vector b in R^n such that $g(x)=f(x)+b$ for all x in R^m. In other words, an affine function is just a linear function plus an intercept.

Example: $g(x)=3x+5$ ← affine function

source : http://cfsv.synechism.org/c1/sec15.pdf

MX-AI

■ Overview: Objective, constraint, feasible, solution, active/inactive, binding/non-binding

- Linear constrained optimization

$$\underset{x_1,x_2}{argmin}\left(-\frac{5}{3}x_1-x_2+5\right)$$: objective function

↓ same problem

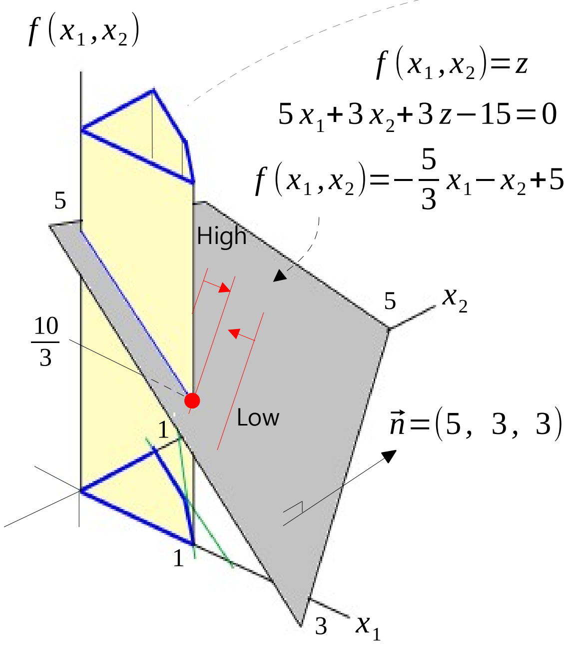
$$\underset{x_1,x_2}{argmin}\left(-\frac{5}{3}x_1-x_2\right)$$: objective function

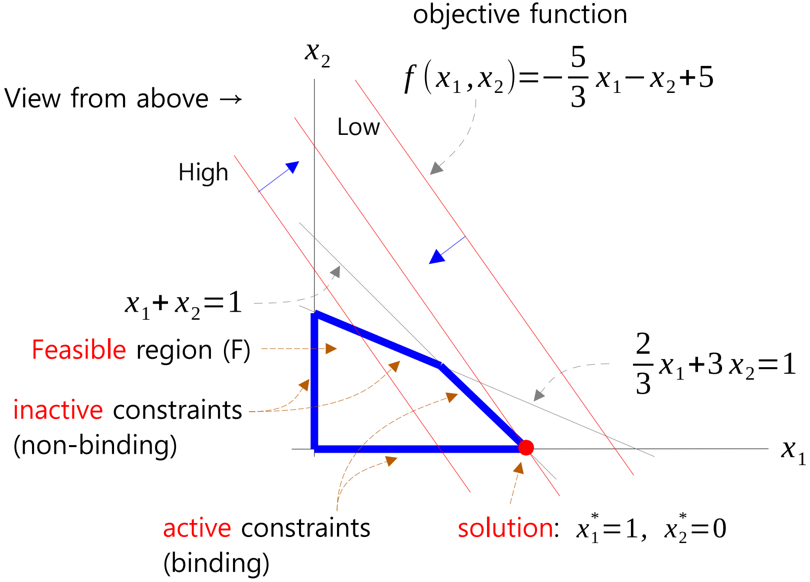
subject to
$$x_1+x_2\le 1,\quad \frac{2}{3}x_1+3x_2\le 1,\quad x_1\ge 0,\quad x_2\ge 0$$
constraints

solution: $x_1^*=1,\ x_2^*=0$

$$\underset{x_1,x_2}{argmin}\left(-\frac{5}{3}x_1-x_2\right)=(1,0)$$

$$\underset{x_1,x_2}{min}\left(-\frac{5}{3}x_1-x_2+5\right)=\frac{10}{3}$$





If optimal solution is on the line for the constraint, it is binding,
Otherwise it is non-binding.

$x_1^*+x_2^*=1$ ← binding $\qquad \frac{2}{3}x_1^*+3x_2^*<1$ ← non-binding

MX-AI

Constrained Linear Programming (LP) & Quadratic Programming (QP)

Linear Programming

- problem

$$\min_{x_1, x_2} 2x_1 + x_2$$

subject to $-x_1 + x_2 \le 1$

$x_1 + x_2 \ge 2$

$x_2 \ge 0$

inequality constraint → $x_1 - 2x_2 \le 4$

equality constraint → $x_1 - 5x_2 = 15$

- standard form

$$\min_{x_1, x_2} c^T \cdot x$$

subject to $G \cdot x \le h$

$A \cdot x = b$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad c = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$G = \begin{bmatrix} -1 & 1 \\ -1 & -1 \\ 0 & -1 \\ 1 & -2 \end{bmatrix} \qquad h = \begin{bmatrix} 1 \\ -2 \\ 0 \\ 4 \end{bmatrix}$$

$$A = [1, 1] \qquad b = 1$$

Quadratic Programming

- problem

$$\min_{x_1, x_2} 2x_1^2 + x_2^2 + x_1 x_2 + x_1 + x_2$$

subject to $x_1 + x_2 = 1$

$x_1 \ge 0$

$x_2 \ge 0$

- standard form

$$\min_{x_1, x_2} \frac{1}{2} x^T \cdot p \cdot x + q^T \cdot x$$

subject to $G \cdot x \le h$

$A \cdot x = b$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad p = \begin{bmatrix} 4 & 1 \\ 1 & 2 \end{bmatrix} \qquad q = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$G = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \qquad h = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$A = [1 \ \ 1] \qquad b = 1$$

MX-AI

■ Constrained Quadratic Programming (QP) problems.

- Equality constraint (A)

$$\min_{x_1,x_2} x_1^2+x_2^2 \quad \text{subject to } x_1+x_2=1$$

- Inequality constraint (B)

$$\min_{x_1,x_2} x_1^2+x_2^2 \quad \text{subject to } x_1+x_2\le 1$$

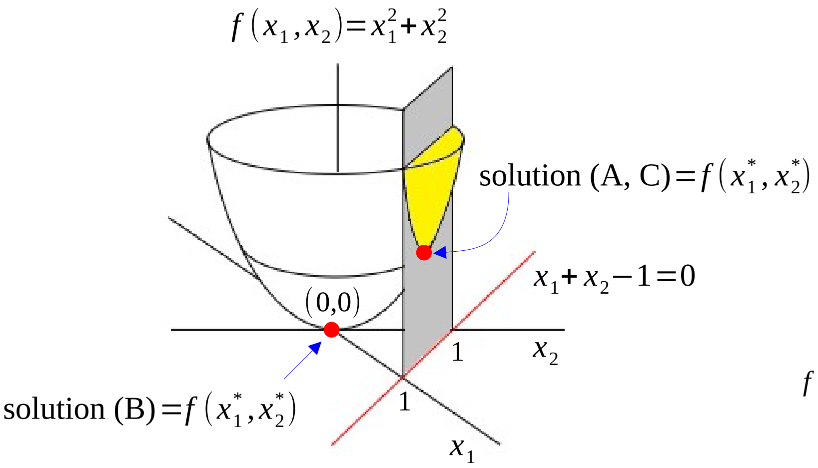
- Inequality constraint (C)

$$\min_{x_1,x_2} x_1^2+x_2^2 \quad \text{subject to } x_1+x_2\ge 1$$

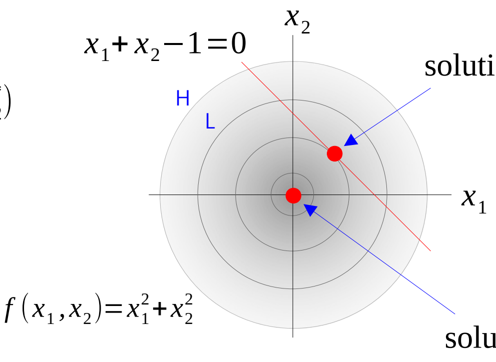
- Standard form

$$\min_{x_1,x_2} \frac{1}{2}x^T\cdot p\cdot x+q^T\cdot x \quad \text{subject to } G\cdot x \le h \quad A\cdot x = b$$

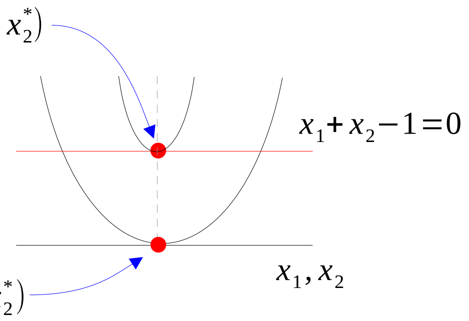
- Graphical solution



- View from above



- View from the right side

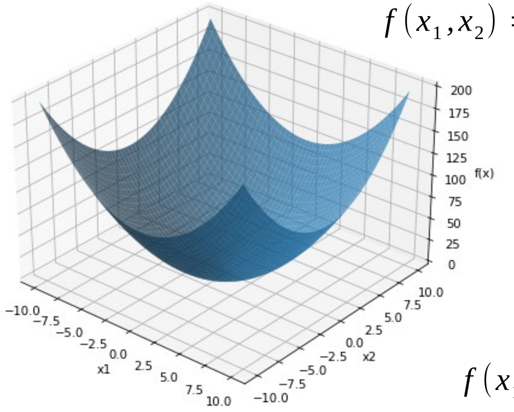


MX-AI

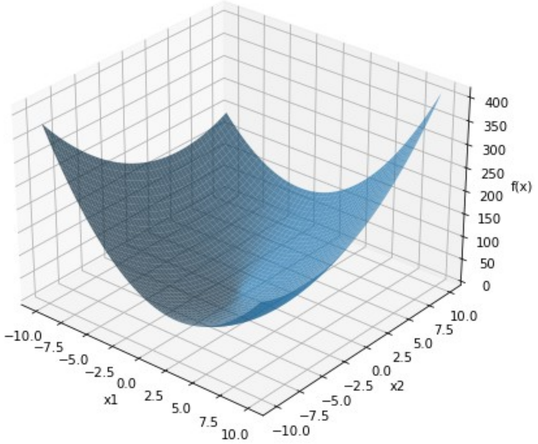
■ 3D plot of the objective function: convex function

```
# [MXML-5-01] 1.plot_convex.py (Plot 3D convex function)
import matplotlib.pyplot as plt
import numpy as np

# f(x)
def f_xy(x1, x2):
    # return (x1 ** 2) + (x2 ** 2)
    # return 3 * x1 + x2
    # return (x1 ** 2) + x2 * (x1 - 1)
    return 2 * (x1 ** 2) + (x2 ** 2) + x1 * x2 + x1 + x2

t = 0.1
x, y = np.meshgrid(np.arange(-10, 10, t), np.arange(-10, 10, t))
zs = np.array([f_xy(a, b) for [a, b] in \
               zip(np.ravel(x), np.ravel(y))])
z = zs.reshape(x.shape)

fig = plt.figure(figsize=(7,7))
ax = fig.add_subplot(111, projection='3d')

# Draw surface
ax.plot_surface(x, y, z, alpha=0.7)

ax.set_xlabel('x1')
ax.set_ylabel('x2')
ax.set_zlabel('f(x)')
ax.azim = -50
ax.elev = 30
plt.show()
```

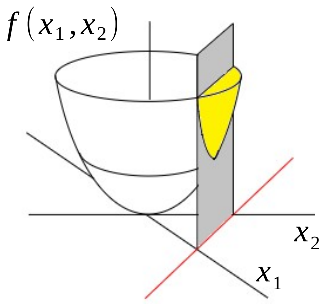
$f(x_1, x_2) = x_1^2 + x_2^2$

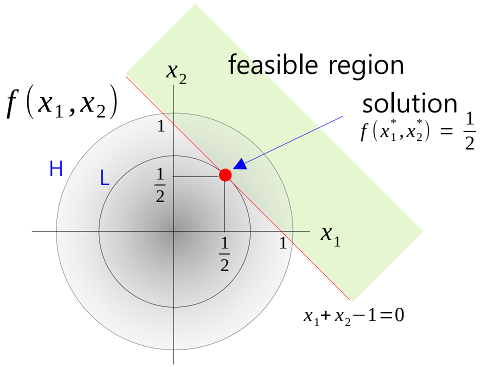


$f(x_1, x_2) = 2x_1^2 + x_2^2 + x_1 x_2 + x_1 + x_2$



MX-AI





5. Convex Optimization

Quadratic Programming (QP)

Part 2: Lagrange method for EQP and IQP

This video was produced in Korean and translated into English, and the audio was generated by AI (TTS).

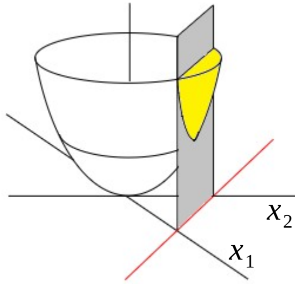
www.youtube.com/@meanxai

■ Equality constrained Quadratic Programming (EQP) : Lagrange method

$$\min_{x_1,x_2} x_1^2+x_2^2$$ ← Convex function (least squares problem)

subject to $x_1+x_2=1$ ← equality constraint

$f(x_1,x_2)$



▪ Lagrange function

$$L(x_1,x_2,\lambda)=x_1^2+x_2^2 + \lambda(x_1+x_2-1), \quad (\lambda\in\mathbb{R})$$

$$\frac{\partial L}{\partial x_1} = 2x_1+\lambda=0 \quad \rightarrow x_1 = -\frac{\lambda}{2}$$

The point where the slope of Lagrange function is 0 (saddle point).

$$\frac{\partial L}{\partial x_2} = 2x_2+\lambda=0 \quad \rightarrow x_2 = -\frac{\lambda}{2}$$

$$\frac{\partial L}{\partial \lambda} = x_1+x_2-1=0 \quad \rightarrow -\lambda-1 = \rightarrow \lambda=-1$$ ← For EQP, λ can be either positive or negative (λ ∈ R). For IQP, λ must be positive.

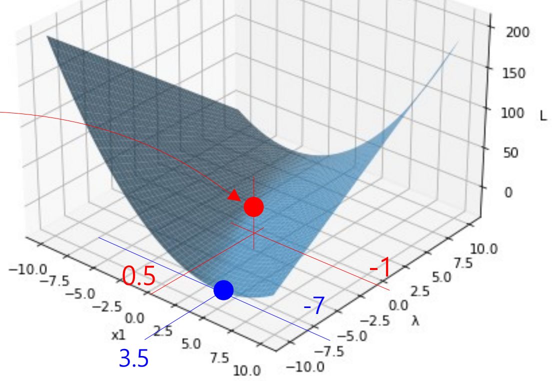
solution : $x_1^* = x_2^* = \frac{1}{2}, \; f(x_1^*,x_2^*) = \frac{1}{2}$

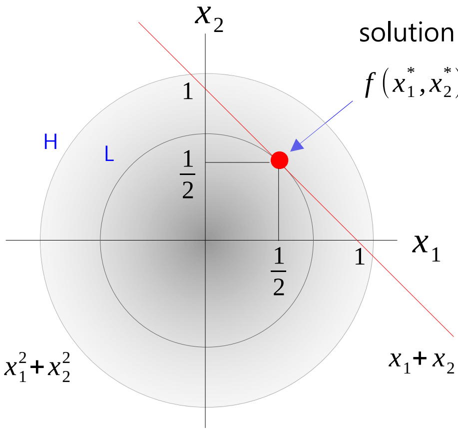
Observe only the x1 and λ axes.

$L(x_1,\lambda)=x_1^2+\lambda(x_1-1)$

saddle point

$L(x_1=0.5,\lambda=-1)$
$= 0.75$





solution

$f(x_1^*,x_2^*) = \frac{1}{2}$

$f(x_1,x_2)=x_1^2+x_2^2$

$x_1+x_2-1=0$

MX-AI

EQP : Standard form and Lagrange method

$$\min_{x_1,x_2} x_1^2+x_2^2$$

subject to $x_1+x_2=1$

Standard form

$$\min_x x^T\cdot p\cdot x$$

subject to $A\cdot x=b$

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

- proof

$$L=\begin{bmatrix}x_1&x_2\end{bmatrix}\cdot\begin{bmatrix}p_{11}&p_{12}\\p_{12}&p_{22}\end{bmatrix}\cdot\begin{bmatrix}x_1\\x_2\end{bmatrix}+\lambda\left(\begin{bmatrix}a_1&a_2\end{bmatrix}\cdot\begin{bmatrix}x_1\\x_2\end{bmatrix}-b\right)$$ (p : assume symmetry)

$$L=x_1^2p_{11}+2x_1x_2p_{12}+x_2^2p_{22}+\lambda a_1x_1+\lambda a_2x_2-\lambda b$$

$$\nabla_{x,\lambda}L=\begin{bmatrix}\frac{\partial L}{\partial x_1}\\\frac{\partial L}{\partial x_2}\\\frac{\partial L}{\partial\lambda}\end{bmatrix}=\begin{bmatrix}2x_1p_{11}+2x_2p_{12}+\lambda a_1\\2x_2p_{22}+2x_1p_{12}+\lambda a_2\\a_1x_1+a_2x_2-b\end{bmatrix}=\begin{bmatrix}0\\0\\0\end{bmatrix}$$

$$\rightarrow\begin{array}{c}\begin{array}{ccc}2p & A^T & x\end{array}\\\begin{bmatrix}2p_{11}&2p_{12}&a_1\\2p_{12}&2p_{22}&a_2\\a_1&a_2&0\end{bmatrix}\cdot\begin{bmatrix}x_1\\x_2\\\lambda\end{bmatrix}=\begin{bmatrix}0\\0\\b\end{bmatrix}\\A\end{array}$$

$$\begin{bmatrix}2p&A^T\\A&0\end{bmatrix}\cdot\begin{bmatrix}x\\\lambda\end{bmatrix}=\begin{bmatrix}0\\b\end{bmatrix}\quad\square$$

- Lagrange function

$$L(x_1,x_2,\lambda)=x^T\cdot p\cdot x+\lambda(A\cdot x-b)$$

$\nabla_x L=2p\cdot x+A^T\cdot\lambda=0$ ← Gradient of Lagrange

$$\nabla_\lambda L=A\cdot x-b=0$$

$$\begin{bmatrix}2p&A^T\\A&0\end{bmatrix}\cdot\begin{bmatrix}x\\\lambda\end{bmatrix}=\begin{bmatrix}0\\b\end{bmatrix}\longrightarrow\begin{bmatrix}2&0&1\\0&2&1\\1&1&0\end{bmatrix}\cdot\begin{bmatrix}x_1\\x_2\\\lambda\end{bmatrix}=\begin{bmatrix}0\\0\\1\end{bmatrix}$$

$$\begin{bmatrix}x_1^*\\x_2^*\\\lambda^*\end{bmatrix}=\begin{bmatrix}2&0&1\\0&2&1\\1&1&0\end{bmatrix}^{-1}\cdot\begin{bmatrix}0\\0\\1\end{bmatrix}=\begin{bmatrix}1/4&-1/4&1/2\\1/4&1/4&1/2\\1/2&1/2&-1\end{bmatrix}\cdot\begin{bmatrix}0\\0\\1\end{bmatrix}=\begin{bmatrix}1/2\\1/2\\-1\end{bmatrix}$$

solution : $x_1^*=x_2^*=\frac{1}{2}\qquad f(x_1^*,x_2^*)=\frac{1}{2}$

MX-AI

■ Inequality constrained Quadratic Programming (IQP) : Lagrange method – slack variable

$$\min_{x_1,x_2} x_1^2+x_2^2$$ ← Convex function (least squares problem)

subject to $x_1+x_2\le 1$ ← inequality constraint

$x_1+x_2-1+\epsilon^2=0$ ← Convert it to an EQP problem by adding a slack variable (ε).

- Lagrange function

$$L(x_1,x_2,\lambda,\epsilon)=x_1^2+x_2^2 + \lambda(x_1+x_2-1+\epsilon^2),\quad (\lambda,\epsilon \in \mathbb{R})$$

$$\frac{\partial L}{\partial x_1} = 2x_1+\lambda=0 \rightarrow x_1 = -\frac{\lambda}{2}$$

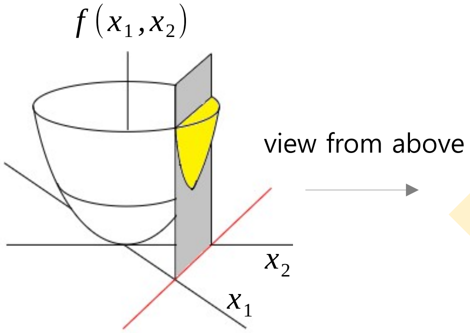
$$\frac{\partial L}{\partial x_2} = 2x_2+\lambda=0 \rightarrow x_2 = -\frac{\lambda}{2}$$

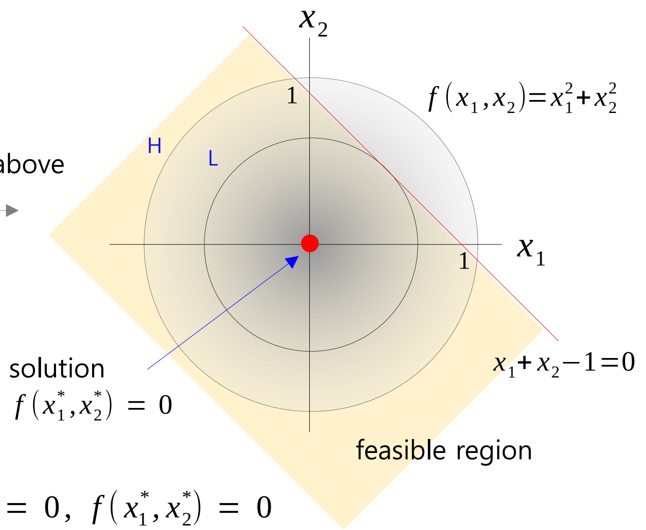
$$\frac{\partial L}{\partial \lambda} = x_1+x_2-1+\epsilon^2=0 \rightarrow -\lambda-1+\epsilon^2=0 \rightarrow \epsilon^2=\lambda+1$$

$$\frac{\partial L}{\partial \epsilon} = 2\lambda\epsilon = \pm 2\lambda\sqrt{\lambda+1}=0 \rightarrow \lambda=0 \text{ or } \lambda=-1$$

$$\begin{cases} \lambda = 0 \rightarrow \epsilon = \pm 1 : x_1=x_2=0,\ f(x_1,x_2) = 0 \\ \lambda = -1 \rightarrow \epsilon = 0 : x_1=x_2=\frac{1}{2},\ f(x_1,x_2) = \frac{1}{2} \end{cases}$$

In both cases, the conditions λ, ε ∈ R are not violated. But we choose λ = 0 because f(x) is smaller when λ = 0. If ε = 0, then this is the solution to EQP. If there exists a case where ε > 0, we choose that ε as the solution to the IQP.





solution : $x_1^* = x_2^* = 0,\ f(x_1^*,x_2^*) = 0$

MX-AI

IQP: Lagrange method – without the slack variable

$$\min_{x_1,x_2} x_1^2+x_2^2$$ ← Convex function (least squares problem)

subject to $x_1+x_2 \le 1$ ← inequality constraint

solution : $x_1^* = x_2^* = 0,\ f(x_1^*, x_2^*) = 0$

- Lagrange function

$L(x_1,x_2,\lambda)=x_1^2+x_2^2 + \lambda(x_1+x_2-1),\ \lambda \ge 0$ ← This condition has been added.

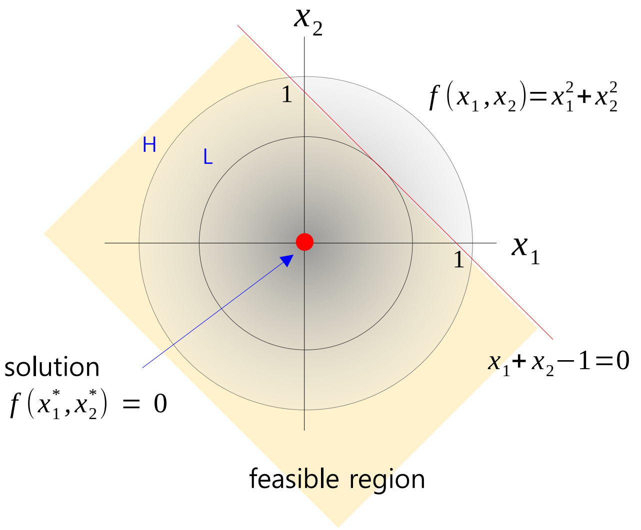
$\frac{\partial L}{\partial x_1} = 2x_1+\lambda=0 \rightarrow x_1 = -\frac{\lambda}{2}$

$\frac{\partial L}{\partial x_2} = 2x_2+\lambda=0 \rightarrow x_2 = -\frac{\lambda}{2}$

The point where the slope of Lagrange function is 0 (saddle point).

$\frac{\partial L}{\partial \lambda} = x_1+x_2-1=0 \rightarrow -\lambda-1=0 \rightarrow \lambda=-1$ ← This violates the constraint. $\rightarrow \lambda=0$

The negative value of λ indicates that the constraint does not affect the optimal solution, and λ should therefore be set to zero. λ=0. This constraint is called a non-binding or inactive.



reference : Constrained Optimization Using Lagrange Multipliers CEE 201L. Uncertainty, Design, and Optimization. Henri P. Gavin and Jeffrey T. Scruggs, Spring 2020

MX-AI

IQP: Lagrange method – slack variable

$$\min_{x_1,x_2} x_1^2+x_2^2$$ ← Convex function (least squares problem)

subject to $x_1+x_2 \geq 1$

$$-x_1-x_2+1+\epsilon^2=0$$ ← Convert it to an EQP problem by adding a slack variable (ε).

- Lagrange function

$$L(x_1,x_2,\lambda,\epsilon)=x_1^2+x_2^2 + \lambda(-x_1-x_2+1+\epsilon^2), \quad (\lambda,\epsilon \in \mathbb{R})$$

$$\frac{\partial L}{\partial x_1} = 2x_1-\lambda=0 \quad \rightarrow x_1 = \frac{\lambda}{2}$$

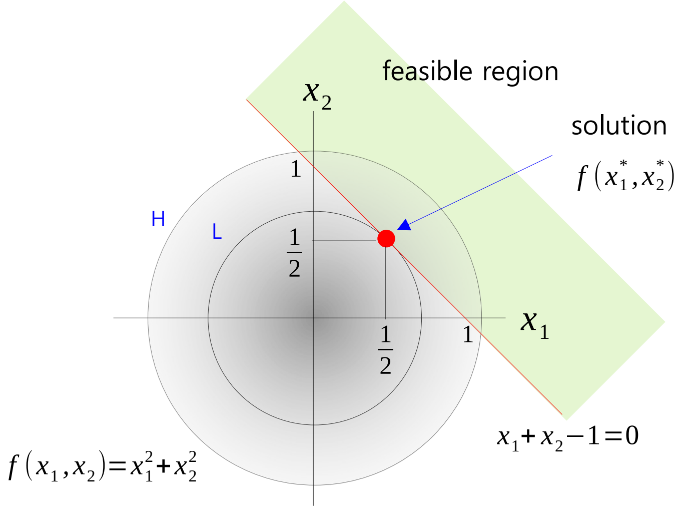
$$\frac{\partial L}{\partial x_2} = 2x_2-\lambda=0 \quad \rightarrow x_2 = \frac{\lambda}{2}$$

The point where the slope of Lagrange function is 0 (saddle point).

$$\frac{\partial L}{\partial \lambda} = -x_1-x_2+1+\epsilon^2=0 \quad \rightarrow -\lambda+1+\epsilon^2=0 \quad \rightarrow \epsilon^2=\lambda-1$$

$$\frac{\partial L}{\partial \epsilon} = 2\lambda\epsilon = \pm 2\lambda\sqrt{\lambda-1}=0 \quad \rightarrow \lambda=0 \quad \text{or } \lambda=1$$

$$\begin{cases} \lambda = 0 \rightarrow \epsilon^2 = -1 : \text{This violates the constraint, } \varepsilon \in R \\ \lambda = 1 \rightarrow \epsilon = 0 : x_1=x_2=\frac{1}{2},\ f(x_1,x_2) = \frac{1}{2} \end{cases}$$



solution : $x_1^* = x_2^* = \frac{1}{2},\ f(x_1^*,x_2^*) = \frac{1}{2}$

MX-AI

IQP: Lagrange method – no slack variable

$$\min_{x_1, x_2} x_1^2 + x_2^2$$ ← Convex function (least squares problem)

subject to $x_1 + x_2 \geq 1$

solution : $x_1^* = x_2^* = \frac{1}{2}$, $f(x_1^*, x_2^*) = \frac{1}{2}$

- Lagrange function

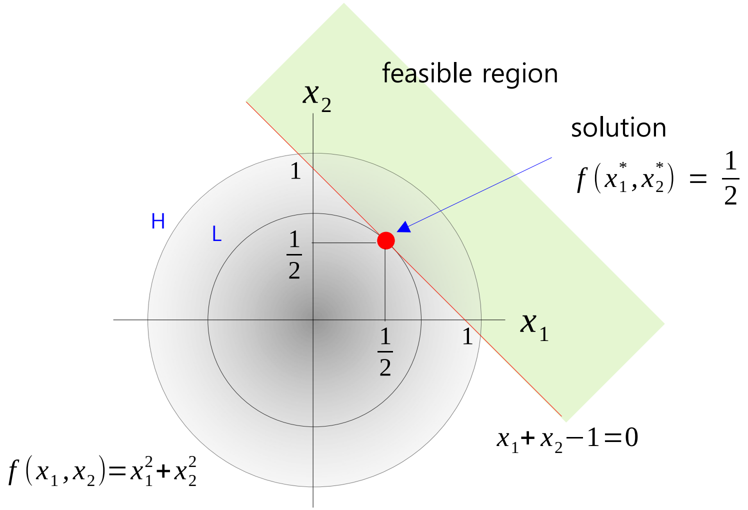
$L(x_1, x_2, \lambda) = x_1^2 + x_2^2 + \lambda(-x_1 - x_2 + 1)$, $\lambda \geq 0$ ← This condition has been added.

$$\frac{\partial L}{\partial x_1} = 2x_1 - \lambda = 0 \rightarrow x_1 = \frac{\lambda}{2}$$

$$\frac{\partial L}{\partial x_2} = 2x_2 - \lambda = 0 \rightarrow x_2 = \frac{\lambda}{2}$$

The point where the slope of Lagrange function is 0 (saddle point).

$$\frac{\partial L}{\partial \lambda} = -x_1 - x_2 + 1 = 0 \rightarrow -\lambda + 1 = 0 \rightarrow \lambda = 1$$



MX-AI

- Lagrange primal function

$$L_p(x_1, x_2, \lambda) = x_1^2 + x_2^2 + \lambda(-x_1 - x_2 + 1)$$

$$\frac{\partial L_p}{\partial x_1} = 2x_1 - \lambda = 0 \quad \rightarrow x_1 = \frac{\lambda}{2}$$

$$\frac{\partial L_p}{\partial x_2} = 2x_2 - \lambda = 0 \quad \rightarrow x_2 = \frac{\lambda}{2}$$

- Lagrange dual function

$$L_d(\lambda) = \left(-\frac{\lambda}{2}\right)^2 + \left(-\frac{\lambda}{2}\right)^2 + \lambda\left(-\frac{\lambda}{2} - \frac{\lambda}{2} - 1\right)$$

$$L_d(\lambda) = -\frac{1}{2}\lambda^2 - \lambda, \quad (\lambda \in \mathbb{R})$$

5. Convex Optimization

Quadratic Programming (QP)

Part 3: Lagrangian Dual Method

This video was produced in Korean and translated into English, and the audio was generated by AI (TTS).

www.youtube.com/@meanxai

MX-AI

EQP: Lagrangian Dual Method

- When it is difficult to solve a problem using the Lagrange method alone, the Lagrange dual method is used.

$$\min_{x_1,x_2} x_1^2+x_2^2$$

subject to $x_1+x_2=1$

- Lagrange primal function

$$L_p(x_1,x_2,\lambda)=x_1^2+x_2^2 + \lambda(x_1+x_2-1), \quad (\lambda\in\mathbb{R})$$

$$\frac{\partial L_p}{\partial x_1} = 2x_1+\lambda=0 \quad \rightarrow x_1 = -\frac{\lambda}{2}$$

$$\frac{\partial L_p}{\partial x_2} = 2x_2+\lambda=0 \quad \rightarrow x_2 = -\frac{\lambda}{2}$$

$$\frac{\partial L_p}{\partial \lambda} = x_1+x_2-1= \quad \rightarrow -\lambda-1 = \rightarrow \lambda=-1$$

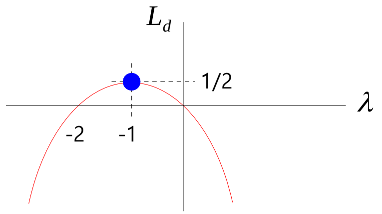
- Lagrange dual function
  - Make it a function of λ only.

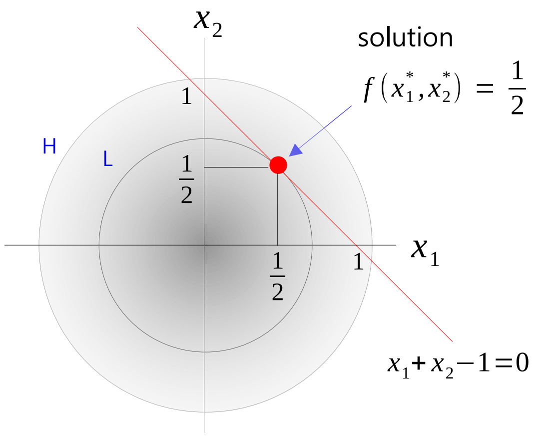
$$L_d(\lambda)=(-\frac{\lambda}{2})^2+(-\frac{\lambda}{2})^2+\lambda(-\frac{\lambda}{2}-\frac{\lambda}{2}-1)$$

$$L_d(\lambda) = -\frac{1}{2}\lambda^2 - \lambda, \quad (\lambda\in\mathbb{R})$$

  - Since it is a concave function, it has a maximum value.

$$\frac{\partial L_d}{\partial \lambda} = -\lambda-1 = \rightarrow \lambda=-1$$





solution : $x_1^* = x_2^* = \frac{1}{2}, \; f(x_1^*,x_2^*) = \frac{1}{2}$

Primal solution : $p^*=1/2$

Dual solution : $d^*=1/2$

In general, p* ≥ d* holds true. However, when special conditions are met, p* = d* holds true. These conditions are called constraint qualifications, and a representative example of these is Slater's condition, which we will discuss in the next video.

MX-AI

EQP: cvxopt

standard form

$$\min_{x_1,x_2} x_1^2+x_2^2 \quad \text{subject to} \quad x_1+x_2=1 \longrightarrow \min_x \frac{1}{2}x^T\cdot P\cdot x+q^T x \quad \text{subject to} \quad \begin{matrix} G\cdot x\le h \\ A\cdot x=b \end{matrix}$$

$$x=\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad p=\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad q=\begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad A=\begin{bmatrix} 1 & 1 \end{bmatrix} \quad b=1$$

```
# [MXML-5-03] 2.EQP.py
# Equality constrained QP (EQP)
#
# Least squares problem:
# minimize   x1^2 + x2^2  subject to x1 + x2 = 1
#
# QP standard form:
# minimize   1/2 * xᵀ.P.x + qT.x
# subject to G.x <= h
#            A.x = b
#
# min. 1/2 * [x1 x2][2 0][x1] + [0 0][x1]
#                   [0 2][x2]        [x2]
#
# s.t. [1 1][x1] = 1
#           [x2]
#
# x = [x1]  P = [2 0]  q = [0]  A = [1 1]  b = 1
#     [x2]      [0 2]      [0]
from cvxopt import matrix, solvers
import numpy as np

P = matrix(np.array([[2, 0], [0, 2]]), tc='d')
q = matrix(np.array([[0], [0]]), tc='d')
A = matrix(np.array([[1, 1]]), tc='d')
b = matrix(1, tc='d')
```

```
sol = solvers.qp(P, q, A=A, b=b)

p_star = sol['primal objective']
x1, x2 = sol['x']
y = sol['y'][0]      # Lagrange multiplier for A.x = b
gap = sol['gap']     # duality gap

# z and y are Lagrange multipliers. z is not used here.
# L = (1/2) * xᵀ.P.x + qT.x + zᵀ(G.x - h) + yᵀ(A.x - b)
# zᵀ = z-transpose, yᵀ = y-transpose
print('\nx1 = {:.3f}'.format(x1))
print('x2 = {:.3f}'.format(x2))
print('λ = {:.3f}'.format(y))
print('p* = {:.3f}'.format(p_star))
print('duality gap = {:.3f}'.format(gap))
```

```
Results:
x1 = 0.500      x2 = 0.500
λ = -1.000      p* = 0.500      duality gap = 0.000
```

MX-AI

IQP-1: Lagrangian Dual Method

$$\min_{x_1, x_2} x_1^2 + x_2^2$$

subject to $x_1 + x_2 \le 1$

- Lagrange primal function

$$L_p(x_1, x_2, \lambda) = x_1^2 + x_2^2 + \lambda(x_1 + x_2 - 1), \ (\lambda \ge 0)$$

$$\frac{\partial L_p}{\partial x_1} = 2x_1 + \lambda = 0 \ \rightarrow x_1 = -\frac{\lambda}{2}$$

$$\frac{\partial L_p}{\partial x_2} = 2x_2 + \lambda = 0 \ \rightarrow x_2 = -\frac{\lambda}{2}$$

$$\frac{\partial L_p}{\partial \lambda} = x_1 + x_2 - 1 = 0 \ \rightarrow -\lambda - 1 = 0$$

$$\lambda = -1 \rightarrow \lambda = 0$$

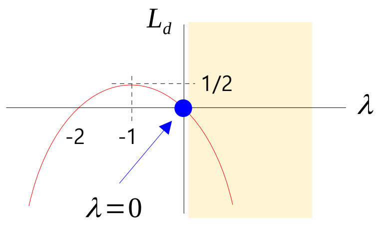
- Lagrange dual function
  - Make it a function of λ only.

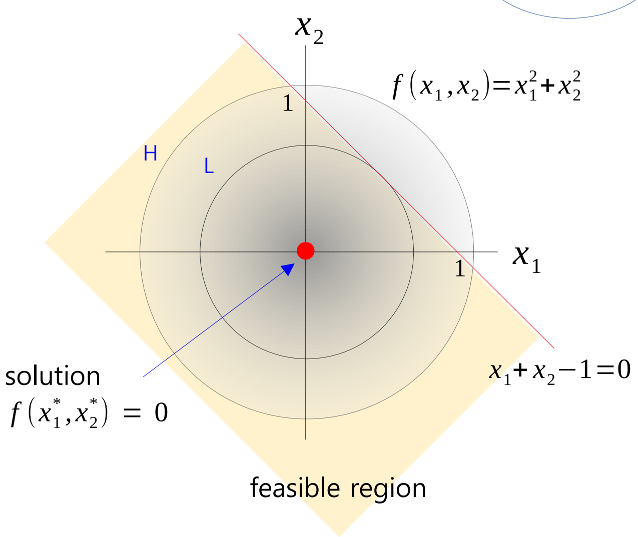
$$L_d(\lambda) = \left(-\frac{\lambda}{2}\right)^2 + \left(-\frac{\lambda}{2}\right)^2 + \lambda\left(-\frac{\lambda}{2} - \frac{\lambda}{2} - 1\right)$$

$$L_d(\lambda) = -\frac{1}{2}\lambda^2 - \lambda, \ (\lambda \ge 0)$$

  - Since it is a concave function, it has a maximum value.

$$\frac{\partial L_d}{\partial \lambda} = -\lambda - 1 = \rightarrow \lambda = -1 \rightarrow \lambda = 0$$





solution : $x_1^* = x_2^* = 0, \ f(x_1^*, x_2^*) = 0$

Primal solution : $p^* = 0$

Dual solution : $d^* = 0$

In general, p* ≥ d* holds true. However, when special conditions are met, p* = d* holds true. These conditions are called constraint qualifications, and a representative example of these is Slater's condition, which we will discuss in the next video.

MX-AI

IQP-1: cvxopt

standard form

$$\min_{x_1,x_2} x_1^2+x_2^2 \quad \text{subject to} \quad x_1+x_2 \le 1 \longrightarrow \min_x \frac{1}{2} x^T \cdot P \cdot x + q^T x \quad \text{subject to} \quad \begin{matrix} G\cdot x \le h \\ A \cdot x = b \end{matrix}$$

$$x=\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad p=\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad q=\begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad G=\begin{bmatrix} 1 & 1 \end{bmatrix} \quad h=1$$

```python
# [MXML-5-03] 3.IQP_1.py: Inequality constrained QP (IQP-1)
#
# Least squares problem:
# minimize    x1^2 + x2^2
# subject to x1 + x2 <= 1
#
# QP standard form
# minimize    1/2 * xᵀ.P.x + qT.x
# subject to G.x <= h,   A.x = b
#
# min. 1/2 [x1 x2][2 0][x1] + [0 0][x1]
#                 [0 2][x2]        [x2]
#
# s.t. [1 1][x1] <= 1
#           [x2]
#
# x = [x1]   P = [2 0]   q = [0]   G = [1 1]   h = 1
#     [x2]       [0 2]       [0]
from cvxopt import matrix, solvers
import numpy as np

P = matrix(np.array([[2, 0], [0, 2]]), tc='d')
q = matrix(np.array([[0], [0]]), tc='d')
G = matrix(np.array([[1, 1]]), tc='d')
h = matrix(1, tc='d')
sol = solvers.qp(P, q, G, h)
```

```python
p_star = sol['primal objective']
x1, x2 = sol['x']
z = sol['z'][0]     # Lagrange multiplier for G.x <= h
gap = sol['gap']    # duality gap
s = sol['s'][0]     # slack variable

# z and λ are Lagrange multipliers. y is not used here.
# L = (1/2) * xᵀ.P.x + qT.x + zᵀ(G.x - h) + yᵀ(A.x - b)
# zᵀ = z-transpose, yᵀ = y-transpose
print('\nx1 = {:.3f}'.format(x1))
print('x2 = {:.3f}'.format(x2))
print('z = {:.3f}'.format(z))
print('s = {:.3f}'.format(s))
print('p* = {:.3f}'.format(p_star))
print('duality gap = {:.3f}'.format(gap))
```

```
Results:
     pcost       dcost       gap    pres   dres
 0:  1.2500e-01 -3.7500e-01  5e-01  0e+00  2e+00
 1:  2.8092e-02  2.0462e-02  8e-03  2e-16  3e-01
 2:  3.9472e-07 -8.8890e-04  9e-04  3e-17  4e-17
 3:  3.9472e-11 -8.8851e-06  9e-06  2e-17  7e-19
 4:  3.9472e-15 -8.8851e-08  9e-08  1e-16  9e-21
Optimal solution found.

x1 = -0.000    x2 = -0.000
z = 0.000      s = 1.000
P* = 0.000
duality gap = 0.000
```

■ IQP-2: Lagrangian Dual Method

$$\min_{x_1, x_2} x_1^2 + x_2^2$$

subject to $x_1 + x_2 \geq 1$

- Lagrange primal function

$$L_p(x_1, x_2, \lambda) = x_1^2 + x_2^2 + \lambda(-x_1 - x_2 + 1), \ (\lambda \geq 0)$$

$$\frac{\partial L_p}{\partial x_1} = 2x_1 - \lambda = 0 \ \rightarrow x_1 = \frac{\lambda}{2}$$

$$\frac{\partial L_p}{\partial x_2} = 2x_2 - \lambda = 0 \ \rightarrow x_2 = \frac{\lambda}{2}$$

$$\frac{\partial L_p}{\partial \lambda} = -x_1 - x_2 + 1 = 0 \ \rightarrow -\lambda + 1 = 0$$

$$\lambda = 1$$

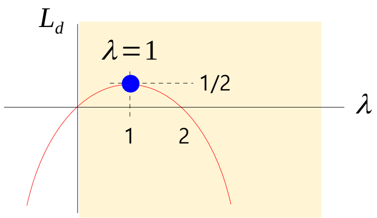
- Lagrange dual function
  - Make it a function of λ only.

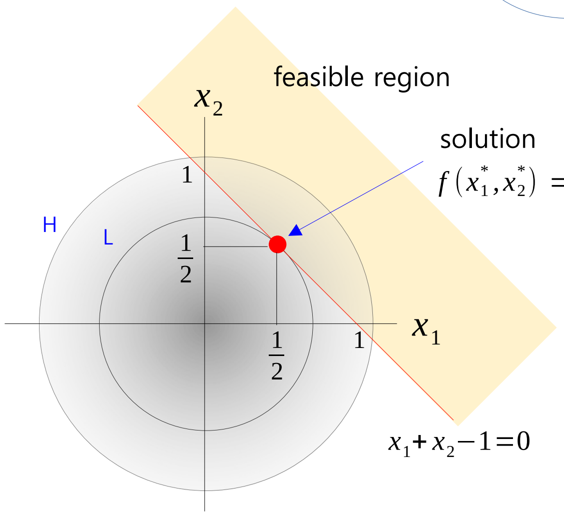
$$L_d(\lambda) = \left(\frac{\lambda}{2}\right)^2 + \left(\frac{\lambda}{2}\right)^2 + \lambda\left(-\frac{\lambda}{2} - \frac{\lambda}{2} + 1\right)$$

$$L_d(\lambda) = -\frac{1}{2}\lambda^2 + \lambda, \ (\lambda \geq 0)$$

  - Since it is a concave function, it has a maximum value.

$$\frac{\partial L_d}{\partial \lambda} = -\lambda + 1 = \rightarrow \lambda = 1$$





solution : $x_1^* = x_2^* = \frac{1}{2}, \ f(x_1^*, x_2^*) = \frac{1}{2}$

Primal solution : $p^* = 1/2$

Dual solution : $d^* = 1/2$

In general, p* ≥ d* holds true. However, when special conditions are met, p* = d* holds true. These conditions are called constraint qualifications, and a representative example of these is Slater’s condition, which we will discuss in the next video.

MX-AI

■ IQP-2: cvxopt

standard form

$$\min_{x_1,x_2} x_1^2+x_2^2 \quad \text{subject to } x_1+x_2 \geq 1 \longrightarrow \min_x \frac{1}{2} x^T \cdot P \cdot x + q^T x \quad \text{subject to } \begin{matrix} G\cdot x \leq h \\ A\cdot x = b \end{matrix} \quad x=\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad p=\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \quad q=\begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad G=\begin{bmatrix} -1 & -1 \end{bmatrix} \quad h=-1$$

```python
# [MXML-5-03] 4.IQP_2.py: Inequality constrained QP (IQP-2)
#
# Least squares problem:
# minimize   x1^2 + x2^2
# subject to x1 + x2 >= 1 --> -x1 - x2 <= -1로 변환.
#
# QP standard form
# minimize   (1/2) * xT.P.x + qT.x
# subject to G.x <= h
#            A.x = b
#
# min. (1/2) * [x1 x2][2 0][x1] + [0 0][x1]
#                     [0 2][x2]        [x2]
#
# s.t. [-1 -1][x1] <= -1
#             [x2]
#
# x = [x1]  P = [2 0]  q = [0]  G = [-1 -1]  h = -1
#     [x2]      [0 2]      [0]
from cvxopt import matrix, solvers
import numpy as np

P = matrix(np.array([[2, 0], [0, 2]]), tc='d')
q = matrix(np.array([[0], [0]]), tc='d')
G = matrix(np.array([[-1, -1]]), tc='d')
h = matrix(-1, tc='d')
sol = solvers.qp(P, q, G, h)
```

```python
p_star = sol['primal objective']
x1, x2 = sol['x']
z = sol['z'][0]     # Lagrange multiplier for G.x <= h
s = sol['s'][0]     # slack variable
gap = sol['gap']    # duality gap

# z and λ are Lagrange multipliers. y is not used here.
# L = (1/2) * xᵀ.P.x + qT.x + zᵀ(G.x - h) + yᵀ(A.x - b)
# zᵀ = z-transpose, yᵀ = y-transpose
print('\nx1 = {:.3f}'.format(x1))
print('x2 = {:.3f}'.format(x2))
print('z = {:.3f}'.format(z))
print('s = {:.3f}'.format(s))
print('p* = {:.3f}'.format(p_star))
print('duality gap = {:.3f}'.format(gap))
```

```
Results:
     pcost       dcost       gap    pres   dres
 0:  1.2500e-01  3.7500e-01  5e-01  2e+00  2e-16
 1:  2.9106e-01  4.7191e-01  8e-03  2e-01  0e+00
 2:  5.0089e-01  5.0000e-01  9e-04  0e+00  2e-15
 3:  5.0001e-01  5.0000e-01  9e-06  0e+00  3e-16
 4:  5.0000e-01  5.0000e-01  9e-08  0e+00  3e-16
Optimal solution found.

X1 = 0.500      x2 = 0.500
z = 1.000       s = 0.000
p* = 0.500
duality gap = 0.000
```

- KKT condition

$$\min_x f(x) \quad \leftarrow \text{f(x): Convex function}$$

$$\text{subject to} \quad g_i(x) \le 0, \quad (i=1,2,\ldots,m)$$

$$h_j(x)=0, \quad (j=1,2,\ldots,n)$$

$$L(x,\lambda,\mu) = f(x)+\sum_{i=1}^{m}\lambda_i g_i(x)+\sum_{j=1}^{n}\mu_i h_i(x), \quad (\lambda_i \ge 0)$$

$$\left.\begin{array}{lll} \text{1) Stationality} & \nabla_x L(x^*,\lambda^*,\mu^*) = 0 & \\ \text{2) Complementary slackness} & \lambda_i g_i(x^*) = 0 & \\ \text{3) Primal feasibility} & g_i(x^*) \le 0, \quad h_j(x^*)=0 & \\ \text{4) Dual feasibility} & \lambda_i \ge 0 & \end{array}\right\} \text{for all i, j}$$

5. Convex Optimization

Quadratic Programming (QP)

Part 4: Slater's condition, KKT condition

This video was produced in Korean and translated into English, and the audio was generated by AI (TTS).

www.youtube.com/@meanxai

MX-AI

Complementary slackness

$$\min_{x_1,x_2} x_1^2+x_2^2$$

subject to $x_1+x_2 \le 1 \rightarrow x_1+x_2-1+\epsilon = 0, (\epsilon \ge 0)$

$$L_p(x_1,x_2,\lambda) = x_1^2+x_2^2 + \lambda(x_1+x_2-1),\ (\lambda \ge 0)$$

$$L_d(\lambda) = -\frac{1}{2}\lambda^2 - \lambda$$

solution : $\lambda^* = 0,\ \epsilon^* = 1,\ x_1^* = x_2^* = 0,\ f(x_1^*,x_2^*) = 0$

$$\min_{x_1,x_2} x_1^2+x_2^2$$

subject to $x_1+x_2 \ge 1 \rightarrow -x_1-x_2+1+\epsilon = 0, (\epsilon \ge 0)$

$$L_p(x_1,x_2,\lambda) = x_1^2+x_2^2 + \lambda(-x_1-x_2+1),\ (\lambda \ge 0)$$

$$L_d(\lambda) = -\frac{1}{2}\lambda^2 + \lambda,\ (\lambda \ge 0)$$

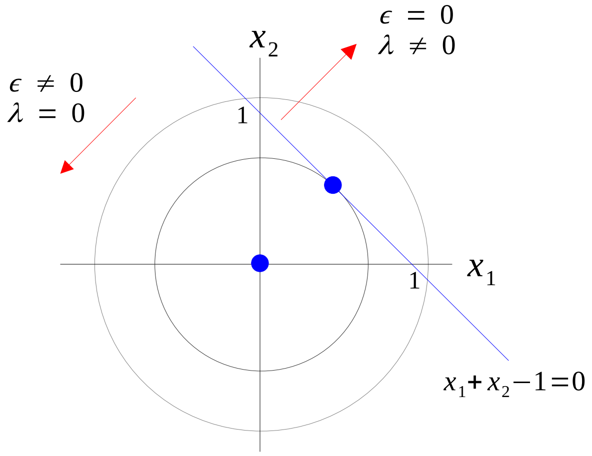
solution : $\lambda^* = \frac{1}{2},\ \epsilon^* = 0,\ x_1^* = x_2^* = \frac{1}{2},\ f(x_1^*,x_2^*) = \frac{1}{2}$

- If ε is not zero, then λ is zero. If ε is 0, λ does not need to be 0.
- ε and λ are complementary. (λε = 0)
- This is called complementary slackness, and following expression hods true.

$$\lambda(x_1+x_2-1) = 0 \quad \begin{cases} \epsilon \ne 0 \rightarrow 0 \times (0+0-1) = 0 \\ \epsilon = 0 \rightarrow \frac{1}{2} \times (\frac{1}{2}+\frac{1}{2}-1) = 0 \end{cases}$$

- Complementary slackness is one of the KKT conditions, which we will cover later.
- Example for λ = 0 & ε =0

subject to $x_1-x_2 \le 0$, or $x_1-x_2 \ge 0$



MX-AI

Slater's condition

Strong duality and duality gap

~ Primal solution (p*) ≥ dual solution (d*) always holds, duality gap = p* - d*

~ Strong duality: p* = d*, (duality gap = 0)

~ If primal problem is convex, strong duality generally holds.

~ Conditions that guarantee strong duality in convex problems are called constraint qualifications.

Terminology

- affine set, affine combination, affine hull, interior, relative interior, etc.

$^{\exists}x \in relint\ D$ D : domain (feasible region)
relint : relative interior of the convex set (non-empty interior)

$x \le y$ - inequality, $x < y$ - strictly inequality

$g(x) = ax$ - linear

$g(x) = ax+b$ - affine

example for p* = d*

$$\min_x f(x)$$

s.t. $x^2 \le 1, \quad 5x+1 \le 2$

Note that since second constraint is affine, we only need to check the first condition. Since x∈R, $^{\exists}$x s.t. x^2 < 1. Hence Slater's condition holds and we have strong duality for this problem.

https://bpb-us-e1.wpmucdn.com/sites.usc.edu/dist/3/137/files/2017/02/lec9-20hn5d7.pdf

Slater's condition

~ This is a sufficient condition for strong duality to hold for a convex optimization problem.

$$\min_x f(x)$$ ← f(x): convex function

subject to $g_i(x) \le 0, \quad (i=1,2,\dots,m)$

$A \cdot x = b$

1) Strong duality holds, if $^{\exists}x \in relint\ D$ such that

$g_i(x) < 0$ - strictly feasible

$A \cdot x = b$

~ If there is at least one x that satisfies these conditions in the feasible region, strong duality holds.

2) For affine g(x), "feasible x" is only required.

$g_i(x) \le 0, \quad A \cdot x = b$

~ If f(x) is convex and g(x) is affine, strong duality holds.

MX-AI

■ Karush–Kuhn–Tucker (KKT) method : KKT condition

▪ KKT condition

$$\min_x f(x) \quad \leftarrow \text{f(x): Convex function}$$

$$\text{subject to } \; g_i(x) \le 0, \quad (i=1,2,\dots,m)$$

$$h_j(x)=0, \quad (j=1,2,\dots,n)$$

$$L(x,\lambda,\mu) = f(x)+\sum_{i=1}^{m}\lambda_i g_i(x)+\sum_{j=1}^{n}\mu_i h_i(x), \quad (\lambda_i \ge 0)$$

1) Stationality $\nabla_x L(x^*,\lambda^*,\mu^*) = 0$

2) Complementary slackness $\lambda_i g_i(x^*) = 0$

3) Primal feasibility $g_i(x^*)\le 0, \quad h_j(x^*)=0$

4) Dual feasibility $\lambda_i \ge 0$

for all i, j

1. KKT conditions for non-convex problems.

For any optimization problem with differentiable objective and constraint functions for which strong duality obtains, any pair of primal and dual optimal points must satisfy the KKT conditions.

2. KKT conditions for convex problems

When the primal problem is convex, the KKT conditions are also sufficient for the points to be primal and dual optimal. In other words, if f(x) are convex and h(x) are affine, and x*, λ*, μ* are any points that satisfy the KKT conditions then x* and (λ*, μ*) are primal and dual optimal, with zero duality gap.

source : "Convex optimization (Stephen Boyd & Lieven Vandenberghe)" p.243

non-convex f(x) : KKT condition ← strong duality
KKT is a necessary condition for strong duality.

convex f(x) : KKT condition ↔ strong duality
KKT is a necessary and sufficient condition for strong duality.

MX-AI

■ KKT method : EQP

▪ KKT condition

$$\min_x f(x) \quad \leftarrow \text{f(x): Convex function}$$

$$\text{subject to } \; g_i(x) \le 0, \quad (i=1,2,\dots,m)$$

$$h_j(x)=0, \quad (j=1,2,\dots,n)$$

$$L(x,\lambda,\mu) = f(x)+\sum_{i=1}^{m}\lambda_i g_i(x)+\sum_{j=1}^{n}\mu_i h_i(x), \quad (\lambda_i \ge 0)$$

1) Stationality $\quad \nabla_x L(x^*,\lambda^*,\mu^*) = 0$

2) Complementary slackness $\quad \lambda_i g_i(x^*) = 0$

3) Primal feasibility $\quad g_i(x^*) \le 0, \; h_j(x^*)=0$

4) Dual feasibility $\quad \lambda_i \ge 0$

for all i, j

1. Equality constrained problem (EQP)

$$\min_{x_1,x_2} x_1^2+x_2^2 \quad \leftarrow \text{Convex function}$$

$$\text{subject to } \; x_1+x_2=1$$

~ Since there is no g(x), only (1) and the h(x*) = 0 of (2) of the KKT conditions are used.

$$L(x_1,x_2,\mu)=x_1^2+x_2^2 + \mu(x_1+x_2-1), \quad \mu \in \mathbb{R}$$

1) $\nabla_{x_1} L = 2x_1+\mu = 0 \rightarrow x_1 = -\dfrac{\mu}{2}$

$\nabla_{x_2} L = 2x_2+\mu = 0 \rightarrow x_2 = -\dfrac{\mu}{2}$

3) $x_1+x_2-1=0$

$-\mu-1=0 \rightarrow \mu=-1$

$$\text{solution : } x_1^* = x_2^* = \frac{1}{2}, \quad p^* = f(x_1^*,x_2^*) = \frac{1}{2}$$

MX-AI

■ KKT method : IQP-1

▪ KKT condition

$$\min_{x} f(x) \quad \leftarrow \text{f(x): Convex function}$$

$$\text{subject to } g_i(x) \le 0, \quad (i=1,2,\dots,m)$$

$$h_j(x)=0, \quad (j=1,2,\dots,n)$$

$$L(x,\lambda,\mu) = f(x)+\sum_{i=1}^{m}\lambda_i g_i(x)+\sum_{j=1}^{n}\mu_i h_i(x), \quad (\lambda_i \ge 0)$$

1) Stationality $\nabla_x L(x^*,\lambda^*,\mu^*) = 0$

2) Complementary slackness $\lambda_i g_i(x^*) = 0$

3) Primal feasibility $g_i(x^*)\le 0, \quad h_j(x^*)=0$

4) Dual feasibility $\lambda_i \ge 0$

for all i, j

2. Inequality constrained problem (IQP) - 1

$$\min_{x_1,x_2} x_1^2+x_2^2 \quad \leftarrow \text{Convex function}$$

$$\text{subject to } x_1+x_2\le 1$$

$$L(x_1,x_2,\lambda)=x_1^2+x_2^2 + \lambda(x_1+x_2-1)$$

1) $\nabla_{x_1} L = 2x_1+\lambda = 0 \rightarrow x_1 = -\dfrac{\lambda}{2}$

$\nabla_{x_2} L = 2x_2+\lambda = 0 \rightarrow x_2 = -\dfrac{\lambda}{2}$

It is discarded since it violates the condition (4).

2) $\lambda(x_1+x_2-1)=0 \rightarrow \lambda(-\lambda-1)=0 \rightarrow \lambda=0, \text{ or } \lambda=-1$

3) $x_1+x_2-1\le 0 \rightarrow -\lambda-1\le 0$

4) $\lambda\ge 0$

solution : $x_1^* = x_2^* = 0, \quad p^* = f(x_1^*,x_2^*) = 0$

MX-AI

■ KKT method : IQP-2

▪ KKT condition

$$\min_{x} f(x) \quad \leftarrow \text{f(x): Convex function}$$

$$\text{subject to } \; g_i(x) \le 0, \quad (i=1,2,\dots,m)$$

$$h_j(x)=0, \quad (j=1,2,\dots,n)$$

$$L(x,\lambda,\mu) = f(x)+\sum_{i=1}^{m} \lambda_i g_i(x)+\sum_{j=1}^{n} \mu_i h_i(x), \quad (\lambda_i \ge 0)$$

1) Stationality $\quad \nabla_x L(x^*,\lambda^*,\mu^*) = 0$

2) Complementary slackness $\quad \lambda_i g_i(x^*) = 0$

3) Primal feasibility $\quad g_i(x^*) \le 0, \;\; h_j(x^*)=0$

4) Dual feasibility $\quad \lambda_i \ge 0$

for all i, j

3. Inequality constrained problem (IQP) - 2

$$\min_{x_1,x_2} x_1^2+x_2^2 \quad \leftarrow \text{Convex function}$$

$$\text{subject to } \; x_1+x_2 \ge 1$$

$$L(x_1,x_2,\lambda)=x_1^2+x_2^2 + \lambda(-x_1-x_2+1)$$

1) $\nabla_{x_1} L = 2x_1-\lambda = 0 \rightarrow x_1 = \dfrac{\lambda}{2}$

$\quad \nabla_{x_2} L = 2x_2-\lambda = 0 \rightarrow x_2 = \dfrac{\lambda}{2}$

It is discarded since it violates the condition (3).

2) $\lambda(-x_1-x_2+1)=0 \rightarrow \lambda(-\lambda+1)=0 \rightarrow \lambda=0, \text{ or } \lambda=1$

3) $-x_1-x_2+1 \le 0 \rightarrow -\lambda+1 \le 0$

4) $\lambda \ge 0$

$$\text{solution : } x_1^* = x_2^* = \frac{1}{2}, \quad p^* = f(x_1^*,x_2^*) = \frac{1}{2}$$

MX-AI

■ Quadratic Programming (QP) example : CVXOPT

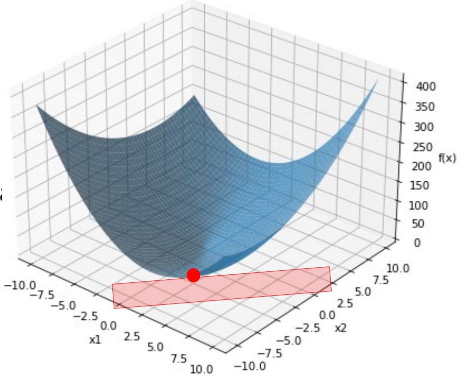
$$\min_{x_1, x_2}\left(2x_1^2 + x_2^2 + x_1 x_2 + x_1 + x_2\right) \quad \text{subject to} \quad x_1 \geq 0, \quad x_2 \geq 0, \quad x_1 + x_2 = 1$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad p = \begin{bmatrix} 4 & 1 \\ 1 & 2 \end{bmatrix} \quad q = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad G = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \quad h = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 1 \end{bmatrix} \quad b = 1$$

```
# [MXML-5-04] 5.QP.py
# QP problem with an equality and an inequality constraints.
# https://cvxopt.org/examples/tutorial/qp.html
#
# min. 2 * x1^2 + x2^2 + x1 * x2 + x1 + x2
# s.t. x1 >= 0, x2 >= 0, x1 + x2 = 1
#
# QP standard form
# minimize   (1/2) * xT.P.x + qT.x
# subject to G.x <= h, A.x = b
#
# min. 1/2 [x1 x2][4 1][x1] + [1 1][x1]
#                 [1 2][x2]        [x2]
#
# s.t. [-1  0][x1] <= [0]
#      [ 0 -1][x2]    [0]
#
#      [1 1][x1] = 1
#           [x2]
#
# x = [x1] P = [4 1] q = [1] G = [-1  0] h = [0] A = [1 1] b = 1
#     [x2]     [1 2]     [1]     [ 0 -1]     [0]
from cvxopt import matrix, solvers
import numpy as np

P = matrix(np.array([[4, 1], [1, 2]]), tc='d')
q = matrix(np.array([[1], [1]]), tc='d')
G = matrix(np.array([[-1, 0],[0, -1]]), tc='d')
h = matrix(np.array([[0], [0]]), tc='d')
```

$x_1 \geq 0, \ x_2 \geq 0$ are inactive

```
A = matrix(np.array([[1, 1]]), tc='d')
b = matrix(1, tc='d')

sol = solvers.qp(P, q, G, h, A, b)

p_star = sol['primal objective']
x1, x2 = sol['x']
y = sol['y'][0]      # Lagrange multiplier for x1 + x2 = 1
z1 = sol['z'][0]     # Lagrange multiplier for -x1 <= 0
z2 = sol['z'][1]     # Lagrange multiplier for -x2 <= 0
gap = sol['gap']     # duality gap

# z and y are Lagrange multipliers.
# L = (1/2) * xT.P.x + qT.x + zT(G.x - h) + yT(A.x - b)
# zT = z-transpose, yT = y-transpose
print('\nx1 = {:.3f}'.format(x1))
print('x2 = {:.3f}'.format(x2))
print('y = {:.3f}'.format(y))
print('z1 = {:.3f}'.format(z1))
print('z2 = {:.3f}'.format(z2))
print('p* = {:.3f}'.format(p_star))
print('duality gap = {:.3f}'.format(g
```




```
Results :

x1 = 0.250      x2 = 0.750
y = -2.750
z1 = 0.000      z2 = 0.000
P* = 1.875      duality gap = 0.000
```

MX-AI

Linear Programming (LP) example : CVXOPT

$$\min_{x_1, x_2}(2x_1 + x_2) \quad \text{subject to} \; -x_1 + x_2 \le 1, \quad x_1 + x_2 \ge 2, \quad x_2 \ge 0, \quad x_1 - 2x_2 \le 4, \quad x_1 - 5x_2 = 15$$

```
# [MXML-5-04] 6.LP.py
# LP problem (https://cvxopt.org/examples/tutorial/lp.html)
#
# min. 2 * x1 + x2
# s.t. -x1 + x2 <= 1
#       x1 + x2 >= 2    --> -x1 - x2 <= -2
#            x2 >= 0    --> -x2 <= 0
#       x1 - 2 * x2 <= 4
#       x1 - 5 * x2 = 15
#
# standard form : minimize cT.x, subject to G.x <= h, A.x = b
#
# min. [2 1][x1]
#           [x2]
#
# s.t. G.x <= h   [-1  1][x1] <= [ 1]
#                 [-1 -1][x2]    [-2]
#                 [ 0 -1]        [ 0]
#                 [ 1 -2]        [ 4]
#      A.x = b    [1 -5][x1] = 15
#                       [x2]
# x = [x1] c = [2] G = [-1  1] h = [ 1] A = [1 -5] b = 1
#     [x2]     [1]     [-1 -1]     [-2]
#                      [ 0 -1]     [ 0]
#                      [ 1 -2]     [ 4]
from cvxopt import matrix, solvers
import numpy as np
c = matrix(np.array([[2], [1]), tc='d')
G = matrix(np.array([[-1, 1],[-1, -1],[0, -1],[1, -2]], tc='d')
h = matrix(np.array([[1], [-2], [0], [4]]), tc='d')
A = matrix(np.array([[1, -5]]), tc='d')
b = matrix(1, tc='d')
sol = solvers.lp(c, G, h, A, b)

p_star = sol['primal objective']
x1, x2 = sol['x']
y = sol['y'][0]     # Lagrange multiplier for A.x = b
z1 = sol['z'][0]    # Lagrange multiplier for G1.x <= h1
z2 = sol['z'][1]    # Lagrange multiplier for G2.x <= h2
z3 = sol['z'][2]    # Lagrange multiplier for G3.x <= h3
z4 = sol['z'][3]    # Lagrange multiplier for G4.x <= h4
gap = sol['gap']    # duality gap

# z and y are Lagrange multipliers.
# L = cT.x + zT(G.x - h) + yT(A.x - b)
print('\nx1 = {:.3f}'.format(x1))
print('x2 = {:.3f}'.format(x2))
print('y = {:.3f}'.format(y))
print('z1 = {:.3f}'.format(z1))
print('z2 = {:.3f}'.format(z2))
print('z3 = {:.3f}'.format(z3))
print('z4 = {:.3f}'.format(z4))
print('p* = {:.3f}'.format(p_star))
print('duality gap = {:.3f}'.format(gap))
```

```
Results: x1 = 1.833   x2 = 0.167
         y = -0.167
         z1 = 0.000   z2 = 1.833   z3 = 0.000   z4 = 0.000
         p* = 3.833
         duality gap = 0.000
```