

OPTI- Lab 2

Marcelino Sánchez Rodríguez 191654

2024-02-14

Definimos función

```
import numpy as np

def mi_esp_nulo(Q, A, c, b) :

    #-----
    # Método del espacio nulo para el problema cuadrático convexo
    # Min  $(1/2) * x.T * Q * x + c.T * x$ 
    # s. a  $A * x = b$ 
    #-----
    (m, n) = np.shape(A)
    #-----
    # Descomposición en valores singulares
    (U, S, Vh) = np.linalg.svd(A, full_matrices = True)
    V = Vh.T
    V1 = V[:, 0 : m]
    #--Base del espacio nulo-----
    Z = V[:, m : n]
    #-----
    # Solución Particular /  $A * xpar = b$ 
    xpar = np.dot(U.T, b)
    Sinv = 1/S
    xpar = Sinv * xpar
    xpar = np.dot(V1, xpar)
    #-----
    # matriz del problema cuadrático convexo sin restricciones
    QZ = np.dot(Z.T, Q)
    QZ = np.dot(QZ, Z)
```

```

#-----
# Lado derecho
ld = np.dot(Q, xpar) + c
ld = -np.dot(Z.T, ld)
#---soluci3n del problema cuadr3tico sin restricciones
xz = np.linalg.solve(QZ, ld)
#-----
# Soluci3n del problema original
xstar = xpar + np.dot(Z, xz)
return xstar

```

Ejemplo

```

import numpy as np
m = 5
n = 9
A = np.random.randn(5, 9)
b = np.ones(m)
c = 10 * np.random.rand(n)
vd = np.arange(1, n + 1)
Q = np.diag(vd)
#
xstar = mi_esp_nulo(Q, A, c, b)
#
print("Soluci3n del problema cuadr3tico ---")
for i in range(len(xstar)) :
    print(f"x[{i}] = ", xstar[i])

```

Soluci3n del problema cuadr3tico ---

```

x[0] = 0.8094596461009143
x[1] = 3.1835910757249186
x[2] = -0.173868194769748
x[3] = -0.9174534466922641
x[4] = -0.9308115045956147
x[5] = -0.8303587167581118
x[6] = 0.6952277122409277
x[7] = 0.49488879298102656
x[8] = -0.6528541159035254

```