OPTI- Lab 4

Marcelino Sánchez Rodríguez 191654

2024-03-06

Método de puntos interiores

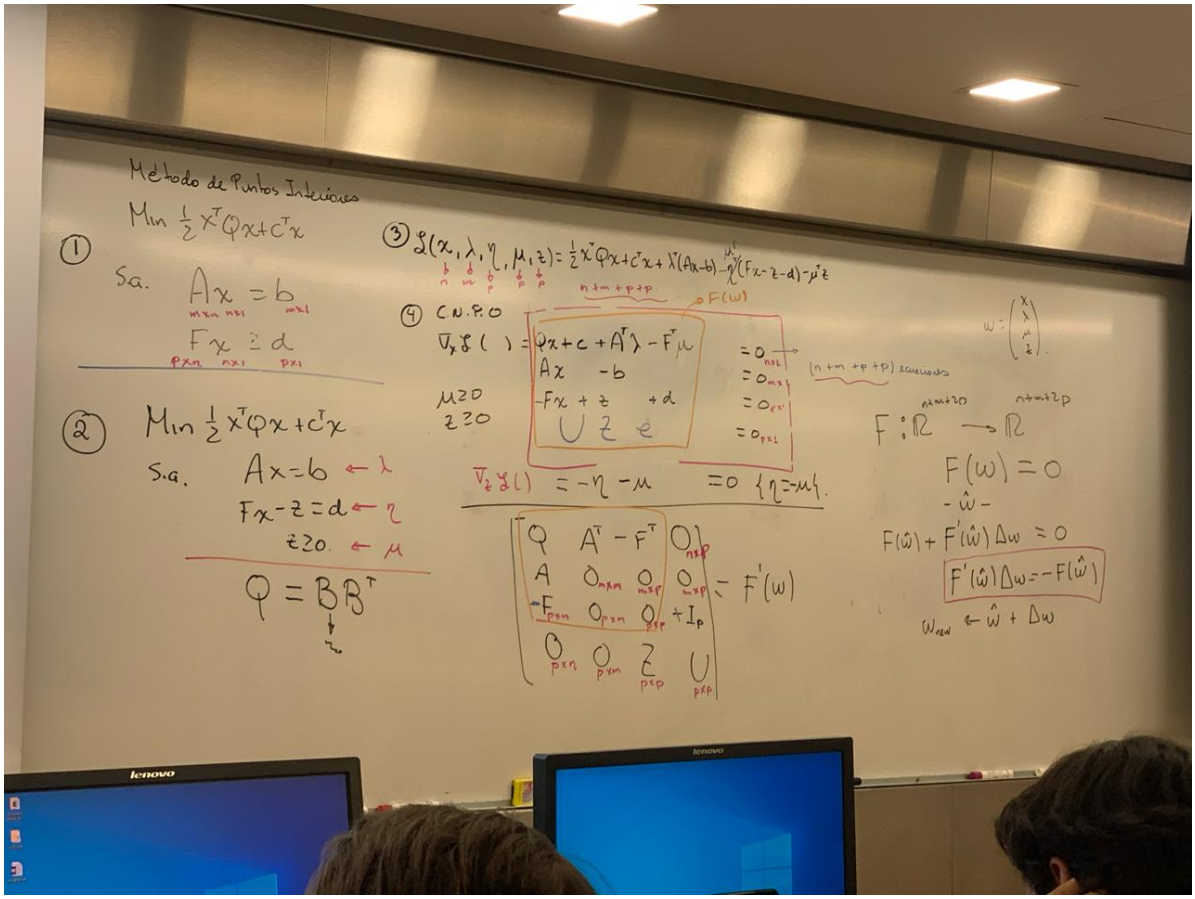

Figura 1: Anotaciones de clase

```
import numpy as np
n=8
m=4
p=10

# Creamos matriz Q para el problema cuadrático
Q=np.random.rand(n,n)
Q = np.tril(Q)
Q = np.dot(Q,Q.T)

A = np.random.rand(m,n)
b = np.ones(m)
F = np.random.rand(p,n)
d = np.random.rand(p)
c = np.ones(n)

# Variables iniciales

x = np.ones(n)
y= np.ones(m)
mu = np.ones(p)
z = np.ones(p)


# ----------------------------------

v1 = np.dot(Q,x) + np.dot(A.T,y) + np.dot(F.T,mu) + c

v2 = np.dot(A,x) - b

v3 = -np.dot(F,x)

v4 = np.multiply(mu,z)

ld = np.concatenate((v1,v2,v3,v4),0)

# ----------------------------------
# Matriz de newton

dim = n+m+p+p
```

```
M = np.zeros((dim,dim))

M[0:n,0:n] = Q

M[0:n,n:n+m] = A.T
M[0:n,n+m:n+m+p] = -F.T

M[n:n+m,0:n] = A

M[n+m:n+m+p,0:n] = -F

M[n+m:n+m+p,n+m+p:n+m+p+p] = np.identity(p)

M[n+m+p:dim,n+m:n+m+p] = np.diag(z)

M[n+m+p:dim,n+m+p:dim] = np.diag(mu)

dd = np.linalg.det(M)
print(dd)
```

```
5.166390877209776
```