OPTI- Lab 4

Marcelino Sánchez Rodríguez 191654

2024-03-06

Método de puntos interiores

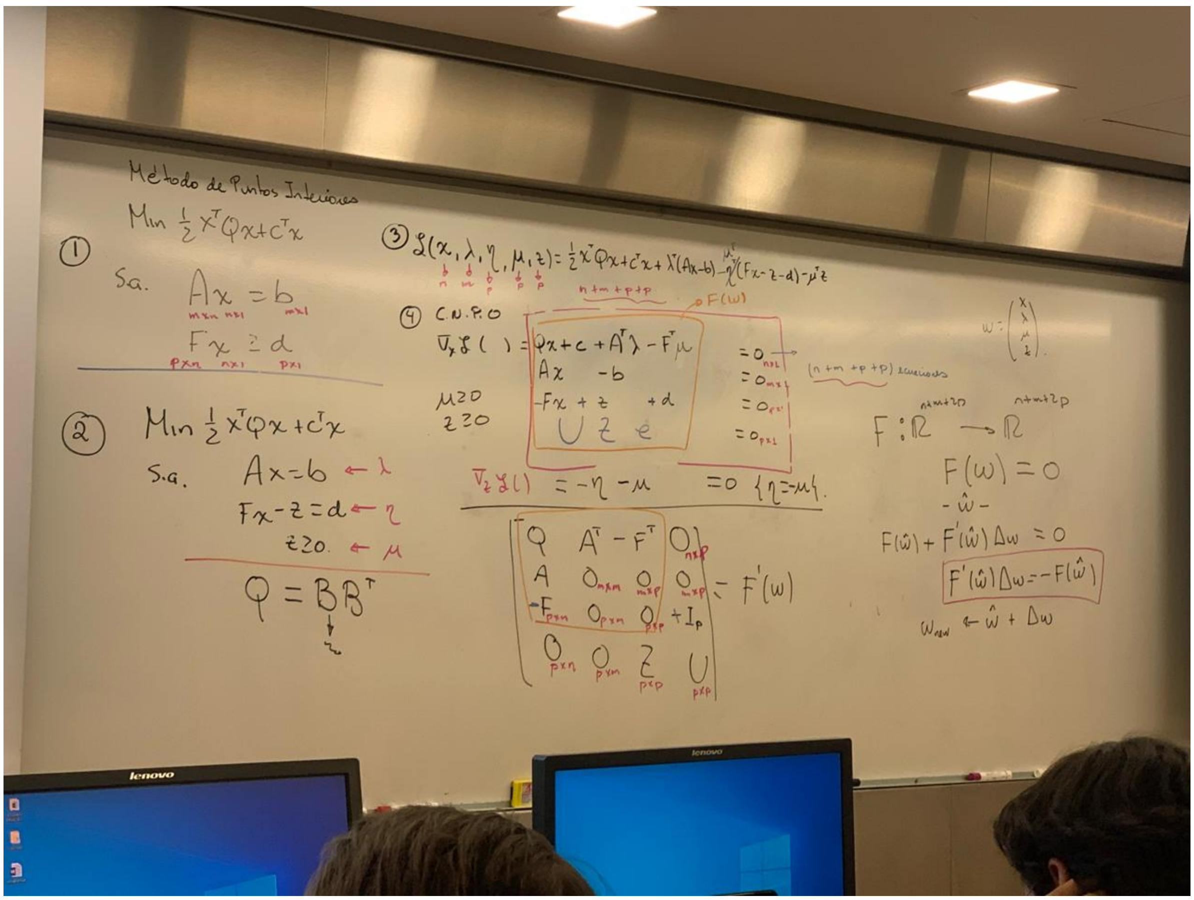

Figura 1: Anotaciones de clase

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

np.random.seed(0)  # Fija la semilla aleatoria a 0

# 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)


# ----------------------------------
# Lado derecho del sistema lineal
tau = 1/2
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("Determinante de M")
print(dd)

# Solución del sistema lineal

dw = np.linalg.solve(M,-ld)
dx= dw[0:n]
dy = dw[n:n+m]
dmu = dw[n+m:n+m+p]
dz = dw[n+m+p:dim]
```

```
Determinante de M
6.454798934012037
```

¿Cómo se va mu?

```
for i in range(len(mu)):
  print([mu[i], dmu[i]])
```

```
[1.0, 1.3732410102093633]
[1.0, 0.848400498082851]
[1.0, 0.22187741090303592]
[1.0, 0.2864042288997859]
[1.0, -0.1622224980056548]
[1.0, 0.2639956813119433]
[1.0, 1.7504222784302133]
[1.0, -1.8905405299739138]
[1.0, 1.2005100930883916]
[1.0, -1.8471297019516129]
```