

# Examen 2

May 25, 2022

## 1 Curso de Optimización (DEMAT)

### 1.1 Parcial 2

Descripción:	Fechas
Fecha de publicación del documento:	<b>Mayo 25, 2022</b>
Hora de inicio:	<b>15:00</b>
Hora límite de entrega:	<b>18:00</b>

#### 1.1.1 Indicaciones

Lea con cuidado los ejercicios.

Puede usar las notas de clase y las tareas hechas para resolver el examen.

Al final, entregue el notebook con sus respuestas, junto con los códigos que hagan falta para reproducir los resultados. Si es más de un archivo, genere un archivo ZIP que contenga el notebook y los scripts adicionales.

A partir del notebook genere un archivo PDF con las respuestas y envíelo por separado antes de la hora límite.

### 1.2 Ejercicio 1. (5 puntos)

Sea

$$f(\mathbf{x}) = f(x_1, x_2) = 2x_1^2 + x_2^2 - x_1x_2 - 6.5x_1 + 2.5x_2.$$

Considere el problema

%

$$\min f(\mathbf{x}) \quad \text{sujeto a} \quad -1 \leq x_1 \leq 1, \quad -1 \leq x_2 \leq 1.$$

Encontrar la solución usando un método de barrera logarítmica (Clase 25).

Para esto construimos la función barrera logarítmica

$$b_{\log}(\mathbf{x}; \mu) = f(x_1, x_2) + \frac{1}{\mu}(-\log(1 - x_1) - \log(1 - x_2) - \log(1 + x_1) - \log(1 + x_2))$$

1. Programar las funciones y sus gradientes

$$f(\mathbf{x}) \quad \text{y} \quad b_{log}(\mathbf{x}; \mu)$$

2. Programar el método de barrera logarítmica:

a) Dar un punto inicial  $\mathbf{x}_0$ , la función  $b_{log}(\mathbf{x}; \mu)$ , su gradiente,  $\mu_0$ , una tolerancia  $\tau > 0$  y los parámetros que se necesiten para usar el método BFGS.

b) Hacer  $k = 0$  y repetir los siguientes pasos:

b1) Usando el método BFGS, calcular el mínimo  $\mathbf{x}_{k+1}$  de  $b_{log}(\mathbf{x}; \mu_k)$ . Para el método BFGS use la tolerancia  $\epsilon_m^{1/4}$  porque no necesitamos tener demasiada precisión en el cálculo. Además, para el algoritmo de backtracking use  $\alpha_{ini} = 1$ , para reducir la posibilidad de evaluar la función de barrera en puntos que están fuera de la región factible.

b2) Imprimir si el algoritmo BFGS encuentra la solución, y en ese caso imprimir

- el entero  $k$
- el valor  $\mu_k$ ,
- el número de iteraciones que realizó el algoritmo BFGS,
- el punto  $\mathbf{x}_{k+1}$  y
- el valor  $f(\mathbf{x}_{k+1})$ .

b3) Si  $\|\mathbf{x}_{k+1} - \mathbf{x}_k\| < \tau$ , terminar devolviendo  $\mathbf{x}_{k+1}$

b4) En caso contrario, hacer  $\mu_{k+1} = 10\mu_k$ ,  $k = k + 1$  y volver al paso (b1)

3. Probar el algoritmo usando  $\mathbf{x}_0 = (-0.75, 0.5)$ ,  $\mu_0 = 1$  y  $\tau = 0.001$ .

4. Si  $(x_1^k, x_2^k)$  es el punto que devuelve el algoritmo de barrera logarítmica y si está en frontera, genere la gráfica de la función evaluándola en puntos sobre la arista en donde está punto para visualmente verificar que tiene sentido el resultado.

### 1.2.1 Solución:

```
[ ]: import numpy as np

def f(x):
    return 2*x[0,0]**2 + x[0,1]**2 - x[0,0]*x[0,1]-6.5*x[0,0]+2.5*x[0,1]

def g(x):
    return np.array([[4*x[0,0]-x[0,1]-6.5, 2*x[0,1]-x[0,0]+2.5*x[0,1]]])

def b_log(x, mu):
    if(x[0,0] < -1 or x[0,0] > 1):
        return np.nan
    if(x[0,1] < -1 or x[0,1] > 1):
        return np.nan
    return f(x) - (sum(np.log(1-x.squeeze()))+sum(np.log(1+x.squeeze())))/mu

def g_b_log(x, mu):
```

```

    return g(x) + 1/mu*np.array([[1/(1-x[0,0])+1/(1+x[0,0]),1/(1-x[0,1])+1/
↪(1+x[0,1])]])

def backtracking(f, f_k, gf_k, x_k, p_k, alpha, ro, c):
    ff = f(x_k + alpha*p_k)
    while (ff is np.NAN):
        alpha = alpha / 2
        ff = f(x_k + alpha*p_k)
    while (ff > f_k + c*alpha*gf_k@p_k.T):
        alpha = ro*alpha
        ff = f(x_k + alpha*p_k)
        while (ff is np.NAN):
            alpha = alpha / 2
            ff = f(x_k + alpha*p_k)
    return alpha

def BFGS(x_0, tol, f, g, H_0, MAX_IT):
    k = 0
    x_k = np.copy(x_0)
    H_k = np.copy(H_0)
    N, N = H_k.shape
    I = np.identity(N)
    res = 0

    while k < MAX_IT:
        g_k = g(x_k)

        if np.linalg.norm(g_k) < tol:
            res = 1
            break

        p_k = - g_k @ H_k.T
        alpha_k = backtracking(f, f(x_k), g_k, x_k, p_k, 1, 0.8, 0.001)
        x_k_1 = x_k + alpha_k * p_k
        g_k_1 = g(x_k_1)
        s_k = x_k_1 - x_k
        y_k = g_k_1 - g_k

        rhok = 1 / (y_k@s_k.T)[0,0]
        H_k_1 = ( (I - rhok*(s_k.T@y_k)) @ H_k @ (I - rhok*(y_k.T@s_k)) +
↪rhok*(s_k.T@s_k) )

        k = k + 1
        x_k = x_k_1
        H_k = H_k_1
        g_k = g_k_1

```

```

    return res, k, x_k

def logarithm_barrier(x_0, b_log, g_b_log, mu_0, tol, H_0):
    mu_k = mu_0
    x_k = x_0
    k = 0
    while 1 == 1:
        res, kk, x_k_1 = BFGS(x_0, tol, lambda x : b_log(x,mu_k), lambda x :
↪g_b_log(x,mu_k), H_0, 5000)
        if res == 1:
            print(k, mu_k, kk, x_k_1, b_log(x_k_1,mu_k))
            if np.linalg.norm(x_k_1-x_k) < tol:
                return x_k_1
            mu_k = 10*mu_k
            x_k = x_k_1
            k = k+1

    return 1

x_0 = np.array([[ -0.75,0.5]])
mu_0 = 1
H_0 = np.identity(2)
tol = 0.0001
logarithm_barrier(x_0, b_log, g_b_log, mu_0, tol, H_0)
# gg = lambda x : g_b_log(x,mu_0).squeeze()
# print(gg(x_0))

```

```

C:\Users\batma\AppData\Local\Temp\ipykernel_13440\3700271115.py:54:
RuntimeWarning: divide by zero encountered in double_scalars
    rhok = 1 / (y_k@s_k.T)[0,0]
C:\Users\batma\AppData\Local\Temp\ipykernel_13440\3700271115.py:55:
RuntimeWarning: invalid value encountered in multiply
    H_k_1 = ( (I - rhok*(s_k.T@y_k)) @ H_k @ (I - rhok*(y_k.T@s_k)) +
rhok*(s_k.T@s_k) )
C:\Users\batma\AppData\Local\Temp\ipykernel_13440\3700271115.py:14:
RuntimeWarning: divide by zero encountered in log
    return f(x) - (sum(np.log(1-x.squeeze()))+sum(np.log(1+x.squeeze())))/mu

```

-----  
KeyboardInterrupt

Traceback (most recent call last)

c:

```

↪\Users\batma\OneDrive\Documents\Academic\School\University\Optimization\Exam\2\Examen_
↪2.ipynb Cell 3' in <module>
    <a href='vscode-notebook-cell:/c%3A/Users/batma/OneDrive/Documents/Academi /
↪School/University/Optimization/Exam/2/Examen%202.ipynb#ch0000002?line=81'>82<
↪a> H_0 = np.identity(2)

```

```

    <a href='vscode-notebook-cell:/c%3A/Users/batma/OneDrive/Documents/Academi /
↳School/University/Optimization/Exam/2/Examen%202.ipynb#ch0000002?line=82'>83<
↳a> tol = 0.0001
---> <a href='vscode-notebook-cell:/c%3A/Users/batma/OneDrive/Documents/Academi /
↳School/University/Optimization/Exam/2/Examen%202.ipynb#ch0000002?line=83'>84<
↳a> logarithm_barrier(x_0, b_log, g_b_log, mu_0, tol, H_0)

c:
↳\Users\batma\OneDrive\Documents\Academic\School\University\Optimization\Exam\2\Examen_
↳2.ipynb Cell 3' in logarithm_barrier(x_0, b_log, g_b_log, mu_0, tol, H_0)
    <a href='vscode-notebook-cell:/c%3A/Users/batma/OneDrive/Documents/Academi /
↳School/University/Optimization/Exam/2/Examen%202.ipynb#ch0000002?line=66'>67<
↳a> k = 0
    <a href='vscode-notebook-cell:/c%3A/Users/batma/OneDrive/Documents/Academi /
↳School/University/Optimization/Exam/2/Examen%202.ipynb#ch0000002?line=67'>68<
↳a> while 1 == 1:
---> <a href='vscode-notebook-cell:/c%3A/Users/batma/OneDrive/Documents/Academi /
↳School/University/Optimization/Exam/2/Examen%202.ipynb#ch0000002?line=68'>69<
↳a>     res, kk, x_k_1 = BFGS(x_0, tol, lambda x : b_log(x,mu_k), lambda x :
↳g_b_log(x,mu_k), H_0, 5000)
    <a href='vscode-notebook-cell:/c%3A/Users/batma/OneDrive/Documents/Academi /
↳School/University/Optimization/Exam/2/Examen%202.ipynb#ch0000002?line=69'>70<
↳a>     if res == 1:
    <a href='vscode-notebook-cell:/c%3A/Users/batma/OneDrive/Documents/Academi /
↳School/University/Optimization/Exam/2/Examen%202.ipynb#ch0000002?line=70'>71<
↳a>         print(k, mu_k, kk, x_k_1, b_log(x_k_1,mu_k))

c:
↳\Users\batma\OneDrive\Documents\Academic\School\University\Optimization\Exam\2\Examen_
↳2.ipynb Cell 3' in BFGS(x_0, tol, f, g, H_0, MAX_IT)
    <a href='vscode-notebook-cell:/c%3A/Users/batma/OneDrive/Documents/Academi /
↳School/University/Optimization/Exam/2/Examen%202.ipynb#ch0000002?line=39'>40<
↳a> while k < MAX_IT:
    <a href='vscode-notebook-cell:/c%3A/Users/batma/OneDrive/Documents/Academi /
↳School/University/Optimization/Exam/2/Examen%202.ipynb#ch0000002?line=40'>41<
↳a>     g_k = g(x_k)
---> <a href='vscode-notebook-cell:/c%3A/Users/batma/OneDrive/Documents/Academi /
↳School/University/Optimization/Exam/2/Examen%202.ipynb#ch0000002?line=42'>43<
↳a>     if np.linalg.norm(g_k) < tol:
    <a href='vscode-notebook-cell:/c%3A/Users/batma/OneDrive/Documents/Academi /
↳School/University/Optimization/Exam/2/Examen%202.ipynb#ch0000002?line=43'>44<
↳a>         res = 1
    <a href='vscode-notebook-cell:/c%3A/Users/batma/OneDrive/Documents/Academi /
↳School/University/Optimization/Exam/2/Examen%202.ipynb#ch0000002?line=44'>45<
↳a>         break

```

File <\_\_array\_function\_\_ internals>:180, in norm(\*args, \*\*kwargs)

File ~\AppData\Roaming\Python\Python310\site-packages\numpy\linalg\linalg.py:  
↳2516, in norm(x, ord, axis, keepdims)

```

    <a href='file:///c%3A/Users/batma/AppData/Roaming/Python/Python310/
↪site-packages/numpy/linalg/linalg.py?line=2513'>2514</a>      sqnorm = dot(x.
↪real, x.real) + dot(x.imag, x.imag)
    <a href='file:///c%3A/Users/batma/AppData/Roaming/Python/Python310/
↪site-packages/numpy/linalg/linalg.py?line=2514'>2515</a> else:
-> <a href='file:///c%3A/Users/batma/AppData/Roaming/Python/Python310/
↪site-packages/numpy/linalg/linalg.py?line=2515'>2516</a>      sqnorm = dot(x, : )
    <a href='file:///c%3A/Users/batma/AppData/Roaming/Python/Python310/
↪site-packages/numpy/linalg/linalg.py?line=2516'>2517</a> ret = sqrt(sqnorm)
    <a href='file:///c%3A/Users/batma/AppData/Roaming/Python/Python310/
↪site-packages/numpy/linalg/linalg.py?line=2517'>2518</a> if keepdims:

```

File <\_\_array\_function\_\_ internals>:180, in dot(\*args, \*\*kwargs)

KeyboardInterrupt:

### 1.3 Ejercicio 2. (5 puntos)

Consideremos la función  $f(z; x_1, \dots, x_{rs})$  definida por una combinación lineal de funciones trigonométricas:

$$\begin{aligned}
 f(z; x_1, \dots, x_{rs}) = & x_1 \sin(\omega_1 z + \phi_1) + \dots + x_r \sin(\omega_1 z + \phi_r) \\
 & + x_{r+1} \sin(\omega_2 z + \phi_1) + \dots + x_{2r} \sin(\omega_2 z + \phi_r) \\
 & + \dots \\
 & + x_{r(s-1)+1} \sin(\omega_s z + \phi_1) + \dots + x_{rs} \sin(\omega_s z + \phi_r)
 \end{aligned}$$

Para  $x_1, \dots, x_{rs}$  fijos y unos puntos  $z_1, z_2, \dots, z_m$  datos, calculamos  $b_i = f(z_i; x_1, \dots, x_{rs})$  para  $i = 1, 2, \dots, m$ . Es decir, si construimos la matriz

$$\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} = \begin{bmatrix} \sin(\omega_1 z_1 + \phi_1) & \cdots & \sin(\omega_1 z_1 + \phi_r) & \sin(\omega_2 z_1 + \phi_1) & \cdots & \sin(\omega_s z_1 + \phi_r) \\ \sin(\omega_1 z_2 + \phi_1) & \cdots & \sin(\omega_1 z_2 + \phi_r) & \sin(\omega_2 z_2 + \phi_1) & \cdots & \sin(\omega_s z_2 + \phi_r) \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ \sin(\omega_1 z_m + \phi_1) & \cdots & \sin(\omega_1 z_m + \phi_r) & \sin(\omega_2 z_m + \phi_1) & \cdots & \sin(\omega_s z_m + \phi_r) \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{rs} \end{pmatrix}$$

Es decir,

$$\mathbf{b} = \mathbf{A}\mathbf{x}$$

El siguiente código contruye la matriz  $\mathbf{A}$ , propone un vector  $\mathbf{x}_{real}$  y genera el vector  $\mathbf{b} = \mathbf{A}\mathbf{x}_{real}$ .

```
[ ]: import numpy as np
import matplotlib.pyplot as plt

# Evalua cada función trigonométrica.
def fnctrig(x, fparam):
    omega = fparam['omega']
    phi    = fparam['phi']
    return np.sin(omega*x + phi)

m    = 15    # Número de puntos z_i
r    = 10    # Número de ángulos phi
s    = 20    # Número de frecuencias omega

z     = np.linspace(0, 9, m)
phi   = np.linspace(0, np.pi, r)
omega = np.linspace(0.5, 2.0, s)

print('Vector de frecuencias omega:\n', omega)
print('\nVector de ángulos de fase phi:\n', phi)

n    = r*s

# Se crea la matriz A
A    = np.zeros((m, n))
for i,z_i in enumerate(z):
    mc = 0
    for j,phi_j in enumerate(phi):
        for l,omega_l in enumerate(omega):
            fparam = {'omega':omega_l, 'phi':phi_j}
            A[i,mc] = fnctrig(z_i, fparam)
            mc += 1

x_real = np.zeros(n)
nc = 1
for i in range(n):
    if i%r==3 and (i//s)%2==0:
        x_real[i] = (-1)**nc
        nc += 1

b    = A @ x_real

print('\nDimensiones de la matriz A:', A.shape)

plt.figure()
```

```

plt.bar(np.arange(n), np.squeeze(x_real))
plt.xlabel('Índice i')
plt.ylabel(r'$x_i$')
plt.title(r'Componentes del vector $x_{real}$ que genera los datos',
         fontsize=14)

plt.figure()
plt.plot(np.arange(m), b, 'r', linewidth=1)
plt.xlabel('Índice i')
plt.ylabel(r'$b_i$')
plt.title(r'Datos generados $b_i$', fontsize=14)

```

Vector de frecuencias omega:

```

[0.5          0.57894737 0.65789474 0.73684211 0.81578947 0.89473684
 0.97368421 1.05263158 1.13157895 1.21052632 1.28947368 1.36842105
 1.44736842 1.52631579 1.60526316 1.68421053 1.76315789 1.84210526
 1.92105263 2.          ]

```

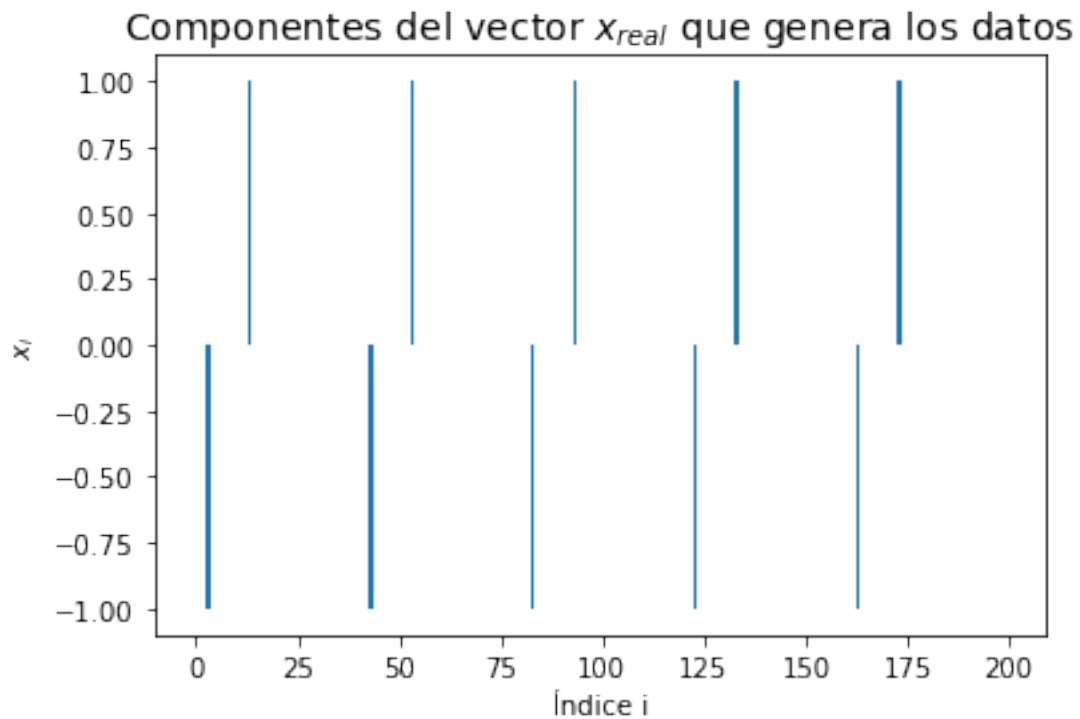
Vector de ángulos de fase phi:

```

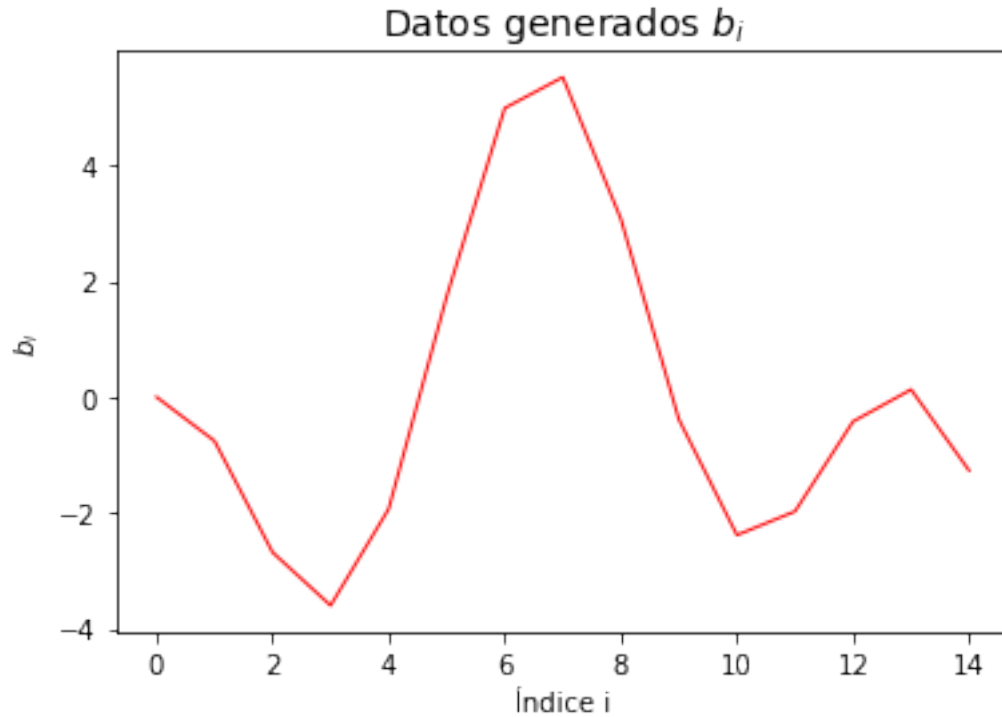
[0.          0.34906585 0.6981317  1.04719755 1.3962634  1.74532925
 2.0943951  2.44346095 2.7925268  3.14159265]

```

Dimensiones de la matriz A: (15, 200)







Dados la matriz  $\mathbf{A}$  y el vector  $\mathbf{b}$ , queremos calcular un vector  $\mathbf{x}$  tal que

$$\mathbf{b} = \mathbf{A}\mathbf{x}.$$

Como  $\mathbf{A}$  es de tamaño  $m \times n$ , con  $m < n$ , no se puede resolver el sistema de ecuaciones o aplicar el método de mínimos cuadrados.

Como hay es un sistema subdeterminado, puede haber una infinidad de soluciones. Una manera de obtener un vector  $\mathbf{x}$  es seleccionar aquel que satisface el sistema y tiene norma mínima (para que no ocurra como en las soluciones de mínimos cuadrados en los que las componentes toman valores en el orden de decenas de miles mientras que las componentes de  $\mathbf{b}$  están en orden de unidades). Así se puede plantear el problema

$$\min \|\mathbf{x}\|_2 \quad \text{sujeto a} \quad \mathbf{A}\mathbf{x} = \mathbf{b}.$$

Se puede ver que la solución de este problema está dada por (medio punto adicional si hacen la deducción)

$$\mathbf{x}_{mn2} = \mathbf{A}^\top (\mathbf{A}\mathbf{A}^\top)^{-1} \mathbf{b}.$$

El problema es que esta solución no es mala, es decir, que la mayoría de sus componentes son cero, tal como el vector  $\mathbf{x}_{real}$  que generó los datos  $\mathbf{b}$ .

El método “basis pursuit” propone calcular  $\mathbf{x}$  resolviendo el problema

$$\min \|\mathbf{x}\|_1 \quad \text{sujeto a} \quad \mathbf{Ax} = \mathbf{b}.$$

Al minimizar la suma de valores absolutos de las variables  $x_i$ , se espera que la mayoría de éstas se hagan cero para reducir la suma, en especial si usamos un método como el método simplex que calcula soluciones básicas factibles (todas las variables no básicas son cero).

Este problema lo podemos plantear como un problema de programación lineal, expresando el vector  $\mathbf{x}$  como la resta de dos nuevos vectores de variables  $\mathbf{u}$  y  $\mathbf{v}$  no negativos:

$$\mathbf{x} = \mathbf{u} - \mathbf{v}.$$

Así, se plantea el problema de programación lineal

$$\min \mathbf{c}^\top \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} \quad \text{sujeto a} \quad [\mathbf{A} \quad -\mathbf{A}] \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} = \mathbf{b}, \quad (\mathbf{u}, \mathbf{v}) \geq \mathbf{0},$$

donde  $\mathbf{c}$  es el vector con todas sus componentes iguales a 1.

1. Calcule la solución de mínima norma 2,

$$\mathbf{x}_{mn2}.$$

- Imprima el valor del error  $\|\mathbf{Ax} - \mathbf{b}\|$
  - Grafique las entradas del vector mínima norma 2 para que pueda constatar que no es vector raro y que no tiene parecido con  $\mathbf{x}_{real}$ .
2. Resuelva el problema de programación lineal con alguna librería de Python.
  3. Construya y resuelva el problema dual para obtener el vector  $\lambda$  y defina  $\mathbf{s}$  como las variables de holgura del problema dual.
  4. Verifique si se cumplen las condiciones KKT para la variable primal  $(\mathbf{u}, \mathbf{v})$  y las variables duales  $(\lambda, \mathbf{s})$  (pueden que no se cumplan si la librería envió warnings sobre el mal condicionamiento de una matriz).
  5. Imprima y compare los valores de la función objetivo del problema primal y del problema dual.
  6. Calcule  $\mathbf{x} = \mathbf{u} - \mathbf{v}$ . Imprima el error  $\|\mathbf{Ax} - \mathbf{b}\|$  y grafique las entradas del vector  $\mathbf{x}$ . Comparado con la gráfica de  $\mathbf{x}_{real}$ , describa las similitudes o diferencias, y si menos se cumple que es un vector raro, que era el propósito original de cambiar la norma 2 por la norma 1 en la función objetivo del problema.
  7. Repita los pasos 2 al 6, cambiando  $m = 50$ . Vuelva a ejecutar el código que genera a la matriz  $\mathbf{A}$  y al vector  $\mathbf{b}$ . Describa si hay algún cambio en los resultados por el hecho de tener más ecuaciones.

### 1.3.1 Solución:

```
[ ]: x_mn2 = A.T@np.linalg.inv(A@A.T)@b
print("x_(mn2) =", x_mn2)
print("||Ax-b|| = ", np.linalg.norm(A@x_mn2-b))
plt.figure()
plt.bar(np.arange(n), np.squeeze(x_mn2))
plt.xlabel('Índice i')
plt.ylabel(r'$x_i$')
plt.title(r'Componentes del vector $x_{mn2}$ que genera los datos', fontsize=14)

from scipy.optimize import linprog

def report(AAA, b, c, e_i, tol, bounds = [], dual = 0):

    if e_i == 1:
        if(len(bounds) > 0):
            status = linprog(c, A_ub=AAA, b_ub=b, bounds = bounds)
        else:
            status = linprog(c, A_ub=AAA, b_ub=b)
    else:
        if(len(bounds) > 0):
            status = linprog(c, A_eq=AAA, b_eq=b, bounds = bounds)
        else:
            status = linprog(c, A_eq=AAA, b_eq=b)

    print(status.message)
    if status.success == 1:
        print("x =", status.x)
        print("c^Tx =", c@np.array([status.x]).T)
        dif = (status.slack if e_i == 1 else status.con)
        print("s =", dif)
        if(dual == 0):
            print("||Ax-b|| =", np.linalg.norm(dif))
            e_x = sum(np.abs(status.x[status.x < 0]))
            e_bAx = sum(np.abs(dif[dif < 0]))
            print("E_x =", e_x, ", la condición de no negatividad" + (" no" if
↪ e_x >= tol else "") + " se cumple")
            if e_i == 1:
                print("E_(b-Ax) =", e_bAx, ", la restricciones de desigualdad"
↪ + ("no" if e_bAx >= tol else "") + " se cumplen")
            return status
        else:
            return -1
```

N = n

```

c = np.ones(2*N)

AA = np.concatenate((A,-A),axis=1)

print("PROMBLEMA LINEAL")
xx = report(AA,b,c,0,np.finfo(float).eps**(1/2))
u,v = xx.x[:N],xx.x[N:]
lamb_bounds = [(None, None)]*len(b)
print("PROMBLEMA DUAL")
tlambda = report(AA.T, c, -b, 1, np.finfo(float).eps**(1/2), lamb_bounds, 1)

def dual_report(A, b, c, status_x, status_lambda, tol):

    print("|c^Tx - b^t(lambda)|/|c^Tx| =",np.abs(c@np.array([status_x.x]).T -
↳b@np.array([status_lambda.x]).T)/np.abs(c@np.array([status_x.x]).T))
    print("||A^T+s-c|| = 0")
    print("||Ax-b|| =", np.linalg.norm(status_x.con))
    e_x = sum(np.abs(status_x.x[status_x.x < 0]))
    e_s = sum(np.abs(status_lambda.slack[status_lambda.slack < 0]))
    print("E_x =", e_x, ", la condición de no negatividad" + (" no" if e_x >=
↳tol else "") + " se cumple")
    print("E_s =", e_s, ", la condición de no negatividad" + (" no" if e_s >=
↳tol else "") + " se cumple")
    e_xs = sum(np.abs(status_x.x*status_lambda.slack))
    print("Sum_i |x_is_i| =", e_xs, ", la condición de complementaridad" + ("
↳no" if e_xs >= tol else "") + " se cumple")

dual_report(AA, b, c, xx, tlambda, 0.0001)

x = u - v
print("||Ax-b|| =", np.linalg.norm(A@x.T-b))
plt.figure()
plt.bar(np.arange(n), np.squeeze(x))
plt.xlabel('Índice i')
plt.ylabel(r'$x_i$')
plt.title(r'Componentes del vector $x$ que genera los datos', fontsize=14)

```

```

x_(mn2) = [ 0.1407288  0.20800105  0.07497326  0.03897606 -0.14984309
-0.1579133
-0.1796099 -0.32392657 -0.49514563 -0.41524146 -0.54904685 -0.66279104
-0.66929479 -0.60472257 -0.40688549 -0.33549669 -0.23224746  0.01883244
 0.31250404  0.46481071  0.17686542  0.16827353  0.21460675  0.05604343
-0.08874239 -0.10747505 -0.09041844 -0.08467594 -0.18223434 -0.30712033
-0.35791697 -0.46314906 -0.54802554 -0.53516847 -0.49815201 -0.50089823
-0.48165445 -0.23117705  0.11606785  0.46251445  0.32982741  0.25004827
 0.11476302  0.03843283 -0.0405359 -0.00491708  0.04923279 -0.01640039
-0.03382106 -0.11902186 -0.23626288 -0.30265632 -0.39229428 -0.46440437

```

```

-0.4253816 -0.54986464 -0.63165384 -0.47331046 -0.18744615 0.15639637
0.36361205 0.19929291 0.16648612 0.06309553 0.07674158 0.15179307
0.10528184 0.18787162 0.16024781 0.0491207 -0.01299018 -0.1164364
-0.07871922 -0.25297185 -0.43483683 -0.56263773 -0.76619538 -0.66876021
-0.25510258 0.21571405 0.29698893 0.21019732 0.07306469 0.03196856
0.05472307 0.14139095 0.22241176 0.27083178 0.22086135 0.22278729
0.12120207 0.11682152 0.10549107 0.00228804 -0.25175704 -0.55689634
-0.73343226 -0.77711478 -0.50639871 0.10311031 0.31976821 0.14420268
0.06045883 0.02546605 0.10874761 0.24332876 0.25716894 0.34049805
0.3782693 0.36469702 0.47446948 0.37057783 0.30517722 0.15711642
-0.07684049 -0.44381447 -0.69521005 -0.82858284 -0.58849775 -0.07524455
0.23883729 0.03272258 -0.03073313 0.01367665 0.19058859 0.24170797
0.3143505 0.36364738 0.39563688 0.46595466 0.50448023 0.51497323
0.56798874 0.38451725 0.14351049 -0.20342918 -0.5115622 -0.65890942
-0.68658244 -0.10850816 0.12503735 -0.0747393 -0.0436021 -0.00109287
0.10749595 0.2384782 0.33346809 0.35150971 0.43893963 0.50663435
0.61259965 0.68900314 0.63951734 0.55298019 0.30645746 -0.0860056
-0.29693893 -0.45574445 -0.56303086 -0.31768753 -0.00978274 -0.12855001
-0.12749811 -0.00334564 0.14105792 0.18898828 0.27838493 0.28466239
0.37369757 0.46390734 0.5855403 0.67869952 0.684962 0.57230816
0.36164277 0.17738629 -0.01833194 -0.27434823 -0.47169185 -0.38460583
-0.16465446 -0.19312152 -0.16407967 0.02039952 0.10203899 0.16737916
0.19507964 0.27134852 0.30666552 0.40054809 0.56748621 0.63146311
0.6367913 0.62705311 0.47891179 0.35125237 0.22198944 -0.0552524
-0.37021863 -0.4511818 ]
||Ax-b|| = 35.305181537242234
PROMBLEMA LINEAL

C:\Users\batma\AppData\Local\Temp\ipykernel_13440\665437207.py:23:
OptimizeWarning: A_eq does not appear to be of full row rank. To improve
performance, check the problem formulation for redundant equality constraints.
    status = linprog(c, A_eq=AAA, b_eq=b)
C:\Users\batma\AppData\Local\Temp\ipykernel_13440\665437207.py:23:
OptimizeWarning: Solving system with option 'cholesky':True failed. It is normal
for this to happen occasionally, especially as the solution is approached.
However, if you see this frequently, consider setting option 'cholesky' to
False.
    status = linprog(c, A_eq=AAA, b_eq=b)
C:\Users\batma\AppData\Local\Temp\ipykernel_13440\665437207.py:23:
OptimizeWarning: Solving system with option 'sym_pos':True failed. It is normal
for this to happen occasionally, especially as the solution is approached.
However, if you see this frequently, consider setting option 'sym_pos' to False.
    status = linprog(c, A_eq=AAA, b_eq=b)
c:\Program Files\Python310\lib\site-packages\scipy\optimize\_linprog_ip.py:117:
LinAlgWarning: Ill-conditioned matrix (rcond=1.45476e-18): result may not be
accurate.
    return sp.linalg.solve(M, r, sym_pos=sym_pos)
c:\Program Files\Python310\lib\site-packages\scipy\optimize\_linprog_ip.py:117:

```

```

LinAlgWarning: Ill-conditioned matrix (rcond=5.43869e-19): result may not be
accurate.
    return sp.linalg.solve(M, r, sym_pos=sym_pos)
c:\Program Files\Python310\lib\site-packages\scipy\optimize\_linprog_ip.py:117:
LinAlgWarning: Ill-conditioned matrix (rcond=3.10799e-21): result may not be
accurate.
    return sp.linalg.solve(M, r, sym_pos=sym_pos)
c:\Program Files\Python310\lib\site-packages\scipy\optimize\_linprog_ip.py:117:
LinAlgWarning: Ill-conditioned matrix (rcond=7.97324e-20): result may not be
accurate.
    return sp.linalg.solve(M, r, sym_pos=sym_pos)
c:\Program Files\Python310\lib\site-packages\scipy\optimize\_linprog_ip.py:117:
LinAlgWarning: Ill-conditioned matrix (rcond=1.76113e-18): result may not be
accurate.
    return sp.linalg.solve(M, r, sym_pos=sym_pos)
c:\Program Files\Python310\lib\site-packages\scipy\optimize\_linprog_ip.py:117:
LinAlgWarning: Ill-conditioned matrix (rcond=1.58417e-18): result may not be
accurate.
    return sp.linalg.solve(M, r, sym_pos=sym_pos)

```

Optimization terminated successfully.

```

x = [9.09831851e-11 1.04824663e-10 9.70226188e-11 8.68146920e-11
8.24001677e-11 8.63847573e-11 1.01069195e-10 1.22143004e-10
1.21528925e-10 1.09675113e-10 9.83376466e-11 9.08138103e-11
9.74280493e-11 1.25777369e-10 1.59453688e-10 1.43289607e-10
9.64393174e-11 8.28976944e-11 1.06626928e-10 1.06236026e-10
9.55633253e-11 9.99891035e-11 8.77366946e-11 7.75495699e-11
7.24561213e-11 8.24402020e-11 1.00523216e-10 1.19985505e-10
1.16592882e-10 1.08527270e-10 1.01778539e-10 1.03860830e-10
1.30054670e-10 1.82241526e-10 2.09085155e-10 1.63148398e-10
9.60896482e-11 8.23885504e-11 1.02784870e-10 1.07363582e-10
1.01707291e-10 9.47477124e-11 7.96559446e-11 6.29972477e-11
6.05701325e-11 8.04489954e-11 1.00370728e-10 1.15465178e-10
1.11784065e-10 1.06680360e-10 1.05762562e-10 1.22350258e-10
1.71858821e-10 2.59750908e-10 2.41021010e-10 1.75416294e-10
9.65348055e-11 8.39461909e-11 9.99832982e-11 1.08102759e-10
1.09498206e-10 9.05340230e-11 6.96565748e-11 4.50810563e-11
5.39479801e-11 8.17140301e-11 1.00626857e-10 1.09815520e-10
1.06364037e-10 1.03569690e-10 1.09875369e-10 1.43562632e-10
1.90561596e-10 1.27515067e-10 2.45702058e-10 1.75257793e-10
9.78469931e-11 8.76718889e-11 9.78533433e-11 1.08312824e-10
1.18969035e-10 8.75430746e-11 5.89863507e-11 3.64109163e-11
5.85194394e-11 8.62260394e-11 1.01235480e-10 1.04202568e-10
1.00313359e-10 9.97094861e-11 1.13535729e-10 1.59789294e-10
2.72600600e-10 2.87938524e+00 1.75880120e-10 1.62682966e-10
9.99915222e-11 9.36864553e-11 9.60234184e-11 1.07909529e-10
1.29762999e-10 8.59229686e-11 5.50330799e-11 4.50892400e-11
7.19969850e-11 9.28962027e-11 1.02085548e-10 9.93647712e-11]

```

9.47105920e-11 9.60589996e-11 1.16083054e-10 1.65799128e-10  
 3.13580407e-10 1.53038278e-09 1.73234079e-10 1.42571499e-10  
 1.02777997e-10 1.01962701e-10 9.44548067e-11 1.06905731e-10  
 1.39997508e-10 8.62125408e-11 6.20614332e-11 6.56996965e-11  
 8.71546875e-11 1.01317513e-10 1.03031842e-10 9.56506304e-11  
 9.03916481e-11 9.33162014e-11 1.16970738e-10 1.63377790e-10  
 1.12595247e-10 2.93559815e-11 1.67450549e-10 1.21340311e-10  
 1.05787012e-10 1.11978119e-10 9.33308923e-11 1.05433322e-10  
 1.45601832e-10 8.87266148e-11 7.61876825e-11 8.60873257e-11  
 1.02211053e-10 1.11811236e-10 1.03908412e-10 9.31622189e-11  
 8.76008530e-11 9.18104849e-11 1.15981349e-10 1.54870607e-10  
 1.18903467e-10 1.40533272e-10 1.28436249e-10 1.03470047e-10  
 1.08340230e-10 1.22359739e-10 9.28754735e-11 1.03725557e-10  
 1.42811845e-10 9.31856063e-11 9.07321336e-11 1.04287095e-10  
 1.23352625e-10 1.21551502e-10 1.04554776e-10 9.18907298e-11  
 8.64384634e-11 9.16735433e-11 1.13325988e-10 1.39907536e-10  
 1.36764743e-10 1.21075632e-10 9.55377083e-11 9.03843257e-11  
 1.09653538e-10 1.30963417e-10 9.32698399e-11 1.02058494e-10  
 1.32263772e-10 9.92265771e-11 1.06320712e-10 1.31274039e-10  
 1.46125176e-10 1.22082201e-10 1.04881892e-10 9.18081470e-11  
 8.70376826e-11 9.29478575e-11 1.09554906e-10 1.20708960e-10  
 1.10576280e-10 8.92828924e-11 7.68314117e-11 8.15309015e-11  
 1.09268167e-10 1.35711886e-10 9.46406167e-11 1.00677599e-10  
 1.32263772e-10 9.92265774e-11 1.06320712e-10 1.31274039e-10  
 1.46125175e-10 1.22082201e-10 1.04881892e-10 9.18081472e-11  
 8.70376826e-11 9.29478570e-11 1.09554906e-10 1.20708960e-10  
 1.10576280e-10 8.92828923e-11 7.68314120e-11 8.15309016e-11  
 1.09268167e-10 1.35711886e-10 9.46406162e-11 1.00677599e-10  
 1.18595780e-10 1.06842387e-10 1.28823746e-10 1.62416427e-10  
 1.65689274e-10 1.07649364e-10 1.04941747e-10 9.28989557e-11  
 8.94587719e-11 9.56013428e-11 1.05345193e-10 1.03024002e-10  
 8.74074530e-11 7.04834120e-11 6.34903919e-11 7.59809055e-11  
 1.07505888e-10 1.35673333e-10 9.70572148e-11 9.97455011e-11  
 1.06136707e-10 1.15369023e-10 1.49688313e-10 3.52505023e-10  
 4.00116253e-10 8.65794201e-11 1.04901048e-10 9.51558677e-11  
 9.37234449e-11 9.94746213e-11 1.01303593e-10 9.02447488e-11  
 7.23474610e-11 5.14122519e-11 5.18661270e-11 7.32546137e-11  
 1.05346113e-10 1.31283686e-10 1.00504160e-10 9.93295043e-11  
 9.67754623e-11 1.21881457e-10 1.82041675e-10 3.13244847e-09  
 1.12876845e-09 7.62669689e-11 1.04882083e-10 9.85526413e-11  
 9.98999968e-11 1.04129362e-10 9.78708371e-11 8.15783356e-11  
 5.74823389e-11 3.19772405e-11 4.53588549e-11 7.32790645e-11  
 1.03629356e-10 1.23718856e-10 1.04799154e-10 9.94206917e-11  
 9.06445287e-11 1.22504078e-10 5.24522067e-10 2.87938523e+00  
 3.13997783e-10 8.43487243e-11 1.04817364e-10 1.02971461e-10  
 1.07758030e-10 1.08696703e-10 9.53200205e-11 7.54158480e-11  
 4.42264641e-11 2.25887162e-11 4.74949476e-11 7.60909468e-11  
 1.02458829e-10 1.14433502e-10 1.09437707e-10 9.99633935e-11

```

8.71982719e-11 1.17074700e-10 1.04830718e-09 5.89697617e-09
3.32702937e-11 9.74988307e-11 1.04511572e-10 1.08103351e-10
1.16113245e-10 1.12035982e-10 9.37967104e-11 7.17638063e-11
3.97191175e-11 2.96350000e-11 5.72508041e-11 8.18130578e-11
1.01427075e-10 1.04965184e-10 1.13439038e-10 1.00878316e-10
8.58309989e-11 1.10705906e-10 3.65744892e-10 1.47619227e-10
7.56376231e-11 1.01240152e-10 1.03851822e-10 1.13408303e-10
1.22909166e-10 1.13326417e-10 9.33596849e-11 7.15906056e-11
4.71618714e-11 4.86873097e-11 7.05852493e-11 9.09100844e-11
1.00197231e-10 9.65820400e-11 1.15470066e-10 1.02073684e-10
8.61215461e-11 1.07925393e-10 9.27337445e-11 7.75986832e-11
9.60410665e-11 9.69878891e-11 1.02919167e-10 1.18144475e-10
1.26310000e-10 1.12663651e-10 9.40083037e-11 7.51604272e-11
6.22439790e-11 6.95019746e-11 8.69653448e-11 1.04218713e-10
9.87762811e-11 8.99952777e-11 1.14563532e-10 1.03446576e-10
8.78568539e-11 1.07252783e-10 9.81745234e-11 9.46048979e-11
9.04712137e-11 9.12754976e-11 1.01920430e-10 1.21379668e-10
1.25546107e-10 1.11073176e-10 9.56971911e-11 8.16971563e-11
7.84755189e-11 9.01505139e-11 1.14117098e-10 1.22170871e-10
9.74195923e-11 8.54306447e-11 1.11072221e-10 1.04879341e-10
9.09831847e-11 1.04824663e-10 9.70226185e-11 8.68146919e-11
8.24001682e-11 8.63847573e-11 1.01069196e-10 1.22143003e-10
1.21528925e-10 1.09675113e-10 9.83376468e-11 9.08138103e-11
9.74280494e-11 1.25777369e-10 1.59453687e-10 1.43289608e-10
9.64393175e-11 8.28976946e-11 1.06626927e-10 1.06236027e-10]
c^Tx = [5.75877052]
s = [ 4.04674092e-14  7.13022306e-14 -1.23897004e-11 -3.67568753e-11
-7.15908444e-11 -1.14726895e-10 -1.63367542e-10 -2.14267271e-10
-2.63919553e-10 -3.08758352e-10 -3.45360185e-10 -3.70646180e-10
-3.82064158e-10 -3.77721410e-10 -3.56535246e-10 -3.18271187e-10
-2.63591371e-10 -1.94037675e-10 -1.11946674e-10 -2.03629891e-11
 7.71143149e-11 1.76510584e-10 2.73671752e-10 3.64464459e-10
 4.44995152e-10 5.11801268e-10 5.62026869e-10 5.93572302e-10
 6.05192341e-10 5.96571681e-10 5.68332048e-10 5.21986898e-10
 4.59893457e-10 3.85081300e-10 3.01134229e-10 2.11976658e-10
 1.21672672e-10 3.42203488e-11 -4.66627847e-11 -1.17715171e-10
-1.76314296e-10 -2.20585328e-10 -2.49487098e-10 -2.62855071e-10
-2.61381139e-10 -2.46573872e-10 -2.20622631e-10 -1.86297644e-10
-1.46761270e-10 -1.05389808e-10 -6.55876464e-11 -3.05891423e-11
-3.29139493e-12 1.39066536e-11 1.92310334e-11 1.16246318e-11
-9.20442889e-12 -4.27603508e-11 -8.77707906e-11 -1.42286849e-10]
||Ax-b|| = 2.204822481797681e-09
E_x = 0 , la condición de no negatividad se cumple
PROMBLEMA DUAL
Optimization terminated successfully.
x = [-4.57680914  0.82163807  4.90005464  1.33829308  0.47032874 -0.29644776
-1.00572451 -1.69776748 -1.65413437 -0.99974064 -0.43321571  0.04632941
 0.49243543  0.90181822  1.14324926  1.04241133  0.70043729  0.31451619

```



```

-0.05834662 -0.42355853 -0.76955155 -1.01121785 -1.01424992 -0.78058046
-0.44819107 -0.10307125 0.24132204 0.58558719 0.89078766 1.04088521
0.93945469 0.65643453 0.31581794 -0.03222202 -0.38765275 -0.74739259
-1.04241009 -1.115933 -0.91149637 -0.56451189 -0.18870329 0.19565323
0.60356844 1.01627365 1.26428663 1.13297644 0.73951771 0.28588286
-0.19274192 -0.74717874 -1.47852223 -2.10717917 -1.64513171 -0.83945148
-0.12120936 0.76542639 1.90390796 6.35196853 0.94508197 -6.61495317]
c^Tx = [-5.75877048]
s = [ 5.47811010e-01 5.20460073e-01 8.07122583e-01 1.22100628e+00
1.54963321e+00 1.67249508e+00 1.61222197e+00 1.49427059e+00
1.44573375e+00 1.50051929e+00 1.57366707e+00 1.52368017e+00
1.26399730e+00 8.48521686e-01 4.66076100e-01 3.32944495e-01
5.41716426e-01 9.66183258e-01 1.30512605e+00 1.27301343e+00
6.14640842e-01 7.51889894e-01 1.13572748e+00 1.54164602e+00
1.7776589e+00 1.78171372e+00 1.63223407e+00 1.47521633e+00
1.41492733e+00 1.44133314e+00 1.44107590e+00 1.28607704e+00
9.41790856e-01 5.19495851e-01 2.27530589e-01 2.40285247e-01
5.64006593e-01 9.95872162e-01 1.22944495e+00 1.07245033e+00
7.27950675e-01 1.01324546e+00 1.44796164e+00 1.79695525e+00
1.91208853e+00 1.79664615e+00 1.57598941e+00 1.39884395e+00
1.33407456e+00 1.32891570e+00 1.25528448e+00 1.01396880e+00
6.26605290e-01 2.48425908e-01 8.21562901e-02 2.39258810e-01
6.38883999e-01 1.02605894e+00 1.12608940e+00 8.63148644e-01
8.74073672e-01 1.27300342e+00 1.70616502e+00 1.95613992e+00
1.93639983e+00 1.71549130e+00 1.45027193e+00 1.27436512e+00
1.21292746e+00 1.17682617e+00 1.03870198e+00 7.40175708e-01
3.56456637e-01 6.80068923e-02 4.74874883e-02 3.29988988e-01
7.57317325e-01 1.05310263e+00 1.00752562e+00 6.70353256e-01
1.03538524e+00 1.49983314e+00 1.87919447e+00 2.00000000e+00
1.84776749e+00 1.54803764e+00 1.27024500e+00 1.11679380e+00
1.06609816e+00 1.00340880e+00 8.17451445e-01 4.97721265e-01
1.63928811e-01 5.43778356e-11 1.27705753e-01 5.01532382e-01
9.05021762e-01 1.07374136e+00 8.88054128e-01 5.17318130e-01
1.19242883e+00 1.66637561e+00 1.94618010e+00 1.92324532e+00
1.65688188e+00 1.31448255e+00 1.05762255e+00 9.45135420e-01
9.11296458e-01 8.29580272e-01 6.18218964e-01 3.15849050e-01
7.22435095e-02 5.26078662e-02 3.13135578e-01 7.33198328e-01
1.06418198e+00 1.08548579e+00 7.82084964e-01 4.22501562e-01
1.32626266e+00 1.75254335e+00 1.89904244e+00 1.73513364e+00
1.38676662e+00 1.04299623e+00 8.38049958e-01 7.80094522e-01
7.67193708e-01 6.76306880e-01 4.65034910e-01 2.16495537e-01
9.24593332e-02 2.19485206e-01 5.81411389e-01 9.97044493e-01
1.21560090e+00 1.08691937e+00 7.02399570e-01 3.97339828e-01
1.42074440e+00 1.74794325e+00 1.74346699e+00 1.45835398e+00
1.07000160e+00 7.66323921e-01 6.38010935e-01 6.41577470e-01
6.51170832e-01 5.62075656e-01 3.76375541e-01 2.11644225e-01
2.22137955e-01 4.80504149e-01 9.00175164e-01 1.26124714e+00
1.34101517e+00 1.07786920e+00 6.58609180e-01 4.44867805e-01

```

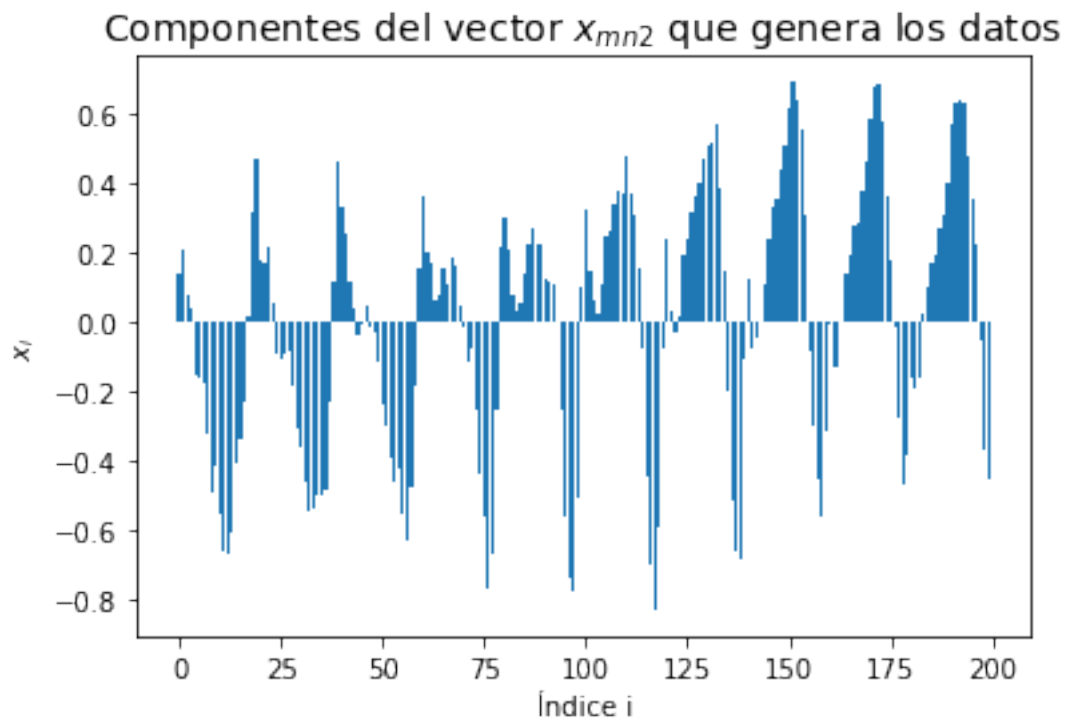
1.46447816e+00	1.65313016e+00	1.49821845e+00	1.12629008e+00
7.44793349e-01	5.17836399e-01	4.81633136e-01	5.46291465e-01
5.77221903e-01	5.00664571e-01	3.62934485e-01	3.01880255e-01
4.45638220e-01	8.04181958e-01	1.23097929e+00	1.49393952e+00
1.42529798e+00	1.05942685e+00	6.55995561e-01	5.59352918e-01
1.45218899e+00	1.47953993e+00	1.19287742e+00	7.78993721e-01
4.50366789e-01	3.27504923e-01	3.87778031e-01	5.05729406e-01
5.54266251e-01	4.99480708e-01	4.26332933e-01	4.76319829e-01
7.36002697e-01	1.15147831e+00	1.53392390e+00	1.66705551e+00
1.45828357e+00	1.03381674e+00	6.94873955e-01	7.26986572e-01
1.45218899e+00	1.47953993e+00	1.19287742e+00	7.78993721e-01
4.50366789e-01	3.27504923e-01	3.87778031e-01	5.05729406e-01
5.54266251e-01	4.99480708e-01	4.26332933e-01	4.76319829e-01
7.36002697e-01	1.15147831e+00	1.53392390e+00	1.66705551e+00
1.45828357e+00	1.03381674e+00	6.94873955e-01	7.26986572e-01
1.38535916e+00	1.24811011e+00	8.64272518e-01	4.58353985e-01
2.22234106e-01	2.18286278e-01	3.67765931e-01	5.24783675e-01
5.85072668e-01	5.58666858e-01	5.58924095e-01	7.13922960e-01
1.05820914e+00	1.48050415e+00	1.77246941e+00	1.75971475e+00
1.43599341e+00	1.00412784e+00	7.70555052e-01	9.27549675e-01
1.27204932e+00	9.86754544e-01	5.52038357e-01	2.03044752e-01
8.79114688e-02	2.03353845e-01	4.24010591e-01	6.01156046e-01
6.65925445e-01	6.71084299e-01	7.44715521e-01	9.86031205e-01
1.37339471e+00	1.75157409e+00	1.91784371e+00	1.76074119e+00
1.36111600e+00	9.73941056e-01	8.73910597e-01	1.13685136e+00
1.12592633e+00	7.26996580e-01	2.93834981e-01	4.38600832e-02
6.36001691e-02	2.84508695e-01	5.49728074e-01	7.25634884e-01
7.87072543e-01	8.23173827e-01	9.61298023e-01	1.25982429e+00
1.64354336e+00	1.93199311e+00	1.95251251e+00	1.67001101e+00
1.24268268e+00	9.46897368e-01	9.92474385e-01	1.32964674e+00
9.64614757e-01	5.00166857e-01	1.20805528e-01	-5.89701621e-11
1.52232509e-01	4.51962357e-01	7.29754996e-01	8.83206204e-01
9.33901835e-01	9.96591202e-01	1.18254855e+00	1.50227874e+00
1.83607119e+00	2.00000000e+00	1.87229425e+00	1.49846762e+00
1.09497824e+00	9.26258640e-01	1.11194587e+00	1.48268187e+00
8.07571169e-01	3.33624388e-01	5.38199039e-02	7.67546751e-02
3.43118120e-01	6.85517446e-01	9.42377454e-01	1.05486458e+00
1.08870354e+00	1.17041973e+00	1.38178104e+00	1.68415095e+00
1.92775649e+00	1.94739213e+00	1.68686442e+00	1.26680167e+00
9.35818023e-01	9.14514209e-01	1.21791504e+00	1.57749844e+00
6.73737338e-01	2.47456652e-01	1.00957564e-01	2.64866362e-01
6.13233380e-01	9.57003772e-01	1.16195004e+00	1.21990548e+00
1.23280629e+00	1.32369312e+00	1.53496509e+00	1.78350446e+00
1.90754067e+00	1.78051479e+00	1.41858861e+00	1.00295551e+00
7.84399102e-01	9.13080626e-01	1.29760043e+00	1.60266017e+00
5.79255599e-01	2.52056751e-01	2.56533009e-01	5.41646015e-01
9.29998402e-01	1.23367608e+00	1.36198906e+00	1.35842253e+00
1.34882917e+00	1.43792434e+00	1.62362446e+00	1.78835577e+00

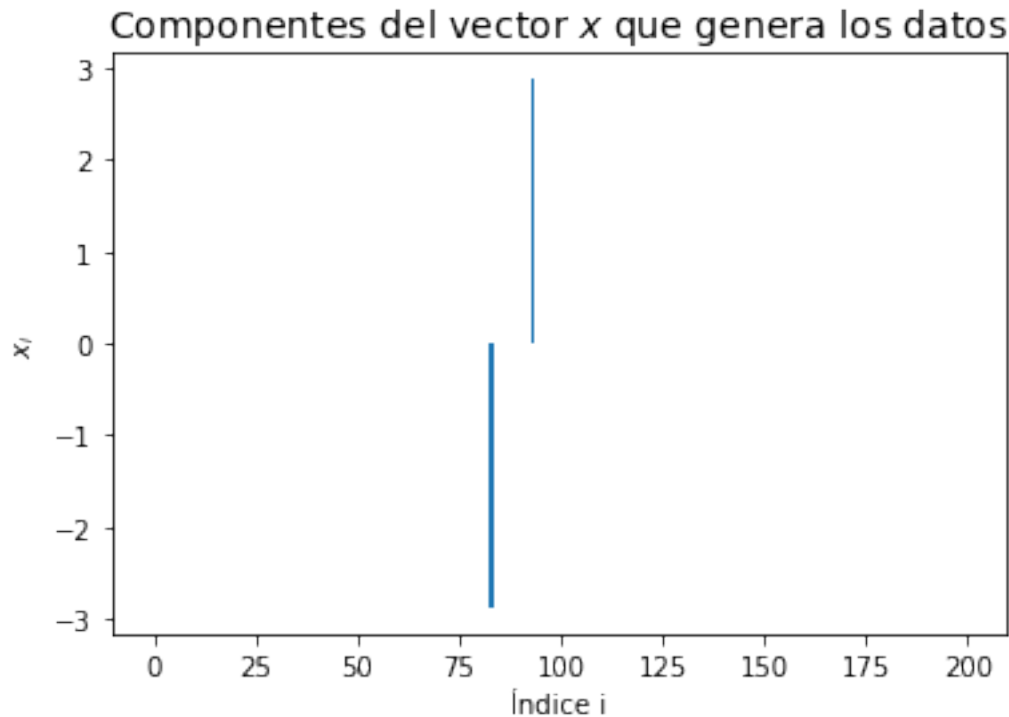
```

1.77786204e+00  1.51949585e+00  1.09982484e+00  7.38752864e-01
6.58984831e-01  9.22130802e-01  1.34139082e+00  1.55513220e+00
5.35521845e-01  3.46869844e-01  5.01781547e-01  8.73709924e-01
1.25520665e+00  1.48216360e+00  1.51836686e+00  1.45370853e+00
1.42277810e+00  1.49933543e+00  1.63706551e+00  1.69811975e+00
1.55436178e+00  1.19581804e+00  7.69020712e-01  5.06060482e-01
5.74702022e-01  9.40573153e-01  1.34400444e+00  1.44064708e+00
5.47811010e-01  5.20460073e-01  8.07122583e-01  1.22100628e+00
1.54963321e+00  1.67249508e+00  1.61222197e+00  1.49427059e+00
1.44573375e+00  1.50051929e+00  1.57366707e+00  1.52368017e+00
1.26399730e+00  8.48521686e-01  4.66076100e-01  3.32944495e-01
5.41716426e-01  9.66183258e-01  1.30512605e+00  1.27301343e+00]
|c^Tx - b^t(lambda)|/|c^Tx| = [6.70911031e-09]
||A^T+s-c|| = 0
||Ax-b|| = 2.204822481797681e-09
E_x = 0 , la condición de no negatividad se cumple
E_s = 5.897016208677996e-11 , la condición de no negatividad se cumple
Sum_i |x_is_i| = 3.962670517409115e-08 , la condición de complementaridad se
cumple
||Ax-b|| = 2.204824055250458e-09

```

`[' ']: Text(0.5, 1.0, 'Componentes del vector  $x_{mn2}$  que genera los datos')`





Los picos de las gráficas  $x_{real}$  y los de  $x_{mn2}$  parecen coincidir, así como donde se concentran los datos de  $x$  coinciden con los de  $x_{mn2}$ . Ahora podemos ver que las gráficas de  $x$  y  $x_{real}$  se parecen aún mas, ambas tiene picos más “agresivos” pero puesto que  $x$  se cálculo minimizando la norma 1 esto máxima los 0 por lo que perdimos algunos picos.

```
[ ]: m = 60 # Número de puntos z_i
r = 10 # Número de ángulos phi
s = 20 # Número de frecuencias omega

z = np.linspace(0, 9, m)
phi = np.linspace(0, np.pi, r)
omega = np.linspace(0.5, 2.0, s)

print('Vector de frecuencias omega:\n', omega)
print('\nVector de ángulos de fase phi:\n', phi)

n = r*s

# Se crea la matriz A
A = np.zeros((m, n))
for i, z_i in enumerate(z):
    mc = 0
    for j, phi_j in enumerate(phi):
```

```

        for l, omega_l in enumerate(omega):
            fparam = {'omega':omega_l, 'phi':phi_j}
            A[i,mc] = fnctrig(z_i, fparam)
            mc += 1

x_real = np.zeros(n)
nc = 1
for i in range(n):
    if i%r==3 and (i//s)%2==0:
        x_real[i] = (-1)**nc
        nc += 1

b = A @ x_real

print('\nDimensiones de la matriz A:', A.shape)

plt.figure()
plt.bar(np.arange(n), np.squeeze(x_real))
plt.xlabel('Índice i')
plt.ylabel(r'$x_i$')
plt.title(r'Componentes del vector $x_{real}$ que genera los datos',
    ↪fontsize=14)

plt.figure()
plt.plot(np.arange(m), b, 'r', linewidth=1)
plt.xlabel('Índice i')
plt.ylabel(r'$b_i$')
_=plt.title(r'Datos generados $b_i$', fontsize=14)

```

Vector de frecuencias omega:

```

[0.5          0.57894737 0.65789474 0.73684211 0.81578947 0.89473684
 0.97368421 1.05263158 1.13157895 1.21052632 1.28947368 1.36842105
 1.44736842 1.52631579 1.60526316 1.68421053 1.76315789 1.84210526
 1.92105263 2.          ]

```

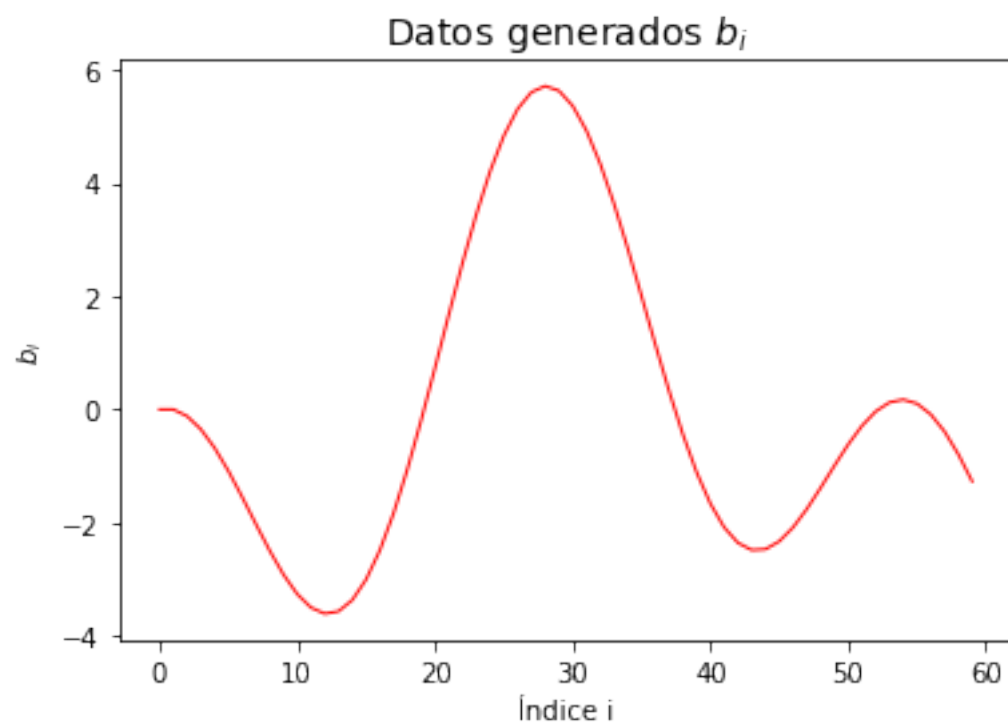
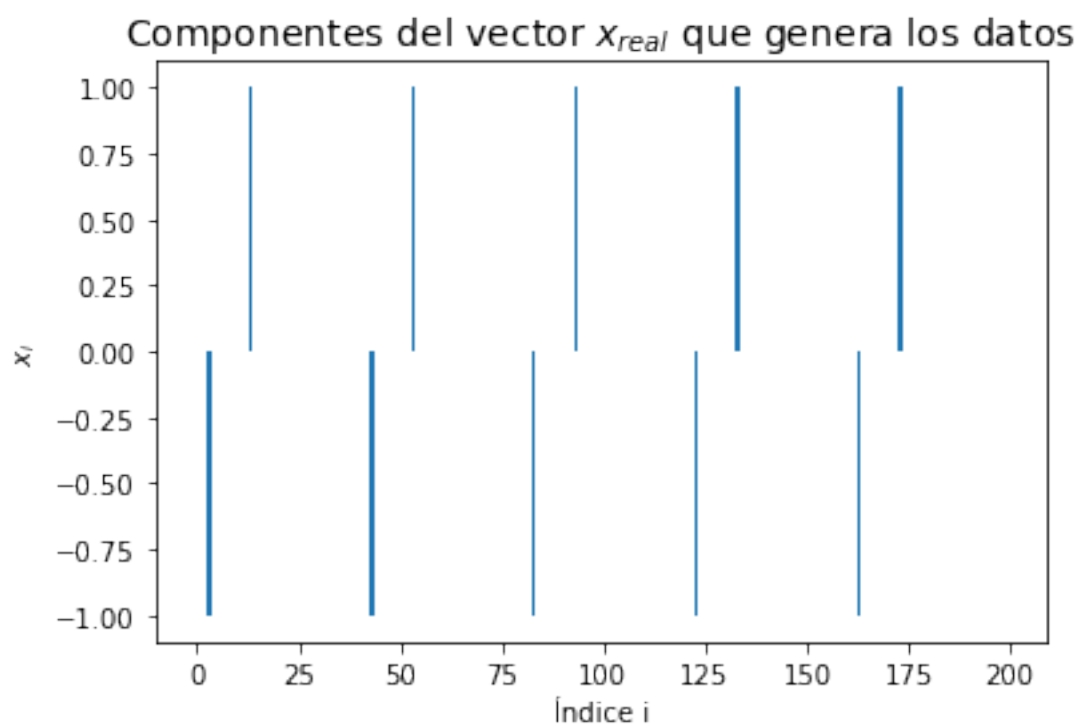
Vector de ángulos de fase phi:

```

[0.          0.34906585 0.6981317  1.04719755 1.3962634  1.74532925
 2.0943951  2.44346095 2.7925268  3.14159265]

```

Dimensiones de la matriz A: (60, 200)



```

[ ]: x_mn2 = A.T@np.linalg.inv(A@A.T)@b
print("x_(mn2) =", x_mn2)
print("||Ax-b|| = ", np.linalg.norm(A@x_mn2-b))
plt.figure()
plt.bar(np.arange(n), np.squeeze(x_mn2))
plt.xlabel('Índice i')
plt.ylabel(r'$x_i$')
plt.title(r'Componentes del vector $x_{mn2}$ que genera los datos', fontsize=14)

from scipy.optimize import linprog

def report(AAA, b, c, e_i, tol, bounds = [], dual = 0):

    if e_i == 1:
        if(len(bounds) > 0):
            status = linprog(c, A_ub=AAA, b_ub=b, bounds = bounds)
        else:
            status = linprog(c, A_ub=AAA, b_ub=b)
    else:
        if(len(bounds) > 0):
            status = linprog(c, A_eq=AAA, b_eq=b, bounds = bounds)
        else:
            status = linprog(c, A_eq=AAA, b_eq=b)

    print(status.message)
    if status.success == 1:
        print("x =", status.x)
        print("c^Tx =", c@np.array([status.x]).T)
        dif = (status.slack if e_i == 1 else status.con)
        print("s =", dif)
        if(dual == 0):
            print("||Ax-b|| =", np.linalg.norm(dif))
            e_x = sum(np.abs(status.x[status.x < 0]))
            e_bAx = sum(np.abs(dif[dif < 0]))
            print("E_x =", e_x, ", la condición de no negatividad" + (" no" if
↪e_x >= tol else "") + " se cumple")
            if e_i == 1:
                print("E_(b-Ax) =", e_bAx, ", la restricciones de desigualdad"
↪+ ("no" if e_bAx >= tol else "") + " se cumplen")
            return status
        else:
            return -1

N = n
c = np.ones(2*N)

```

```

AA = np.concatenate((A,-A),axis=1)

print("PROMBLEMA LINEAL")
xx = report(AA,b,c,0,np.finfo(float).eps**(1/2))
u,v = xx.x[:N],xx.x[N:]
lamb_bounds = [(None, None)]*len(b)
print("PROMBLEMA DUAL")
tlambda = report(AA.T, c, -b, 1, np.finfo(float).eps**(1/2), lamb_bounds, 1)

def dual_report(A, b, c, status_x, status_lambda, tol):

    print("|c^Tx - b^t(lambda)|/|c^Tx| =",np.abs(c@np.array([status_x.x]).T -
    ↪b@np.array([status_lambda.x]).T)/np.abs(c@np.array([status_x.x]).T))
    print("||A^T+s-c|| = 0")
    print("||Ax-b|| =", np.linalg.norm(status_x.con))
    e_x = sum(np.abs(status_x.x[status_x.x < 0]))
    e_s = sum(np.abs(status_lambda.slack[status_lambda.slack < 0]))
    print("E_x =", e_x, ", la condición de no negatividad" + (" no" if e_x >=
    ↪tol else "") + " se cumple")
    print("E_s =", e_s, ", la condición de no negatividad" + (" no" if e_s >=
    ↪tol else "") + " se cumple")
    e_xs = sum(np.abs(status_x.x*status_lambda.slack))
    print("Sum_i |x_is_i| = ", e_xs, ", la condición de complementaridad" + ("
    ↪no" if e_xs >= tol else "") + " se cumple")

dual_report(AA, b, c, xx, tlambda, 0.0001)

x = u - v
print("||Ax-b|| =", np.linalg.norm(A@x.T-b))
plt.figure()
plt.bar(np.arange(n), np.squeeze(x))
plt.xlabel('Índice i')
plt.ylabel(r'$x_i$')
plt.title(r'Componentes del vector $x$ que genera los datos', fontsize=14)

```

```

x_(mn2) = [ 0.1407288  0.20800105  0.07497326  0.03897606 -0.14984309
-0.1579133
-0.1796099 -0.32392657 -0.49514563 -0.41524146 -0.54904685 -0.66279104
-0.66929479 -0.60472257 -0.40688549 -0.33549669 -0.23224746  0.01883244
 0.31250404  0.46481071  0.17686542  0.16827353  0.21460675  0.05604343
-0.08874239 -0.10747505 -0.09041844 -0.08467594 -0.18223434 -0.30712033
-0.35791697 -0.46314906 -0.54802554 -0.53516847 -0.49815201 -0.50089823
-0.48165445 -0.23117705  0.11606785  0.46251445  0.32982741  0.25004827
 0.11476302  0.03843283 -0.0405359 -0.00491708  0.04923279 -0.01640039
-0.03382106 -0.11902186 -0.23626288 -0.30265632 -0.39229428 -0.46440437
-0.4253816 -0.54986464 -0.63165384 -0.47331046 -0.18744615  0.15639637
 0.36361205  0.19929291  0.16648612  0.06309553  0.07674158  0.15179307

```



```

0.10528184 0.18787162 0.16024781 0.0491207 -0.01299018 -0.1164364
-0.07871922 -0.25297185 -0.43483683 -0.56263773 -0.76619538 -0.66876021
-0.25510258 0.21571405 0.29698893 0.21019732 0.07306469 0.03196856
0.05472307 0.14139095 0.22241176 0.27083178 0.22086135 0.22278729
0.12120207 0.11682152 0.10549107 0.00228804 -0.25175704 -0.55689634
-0.73343226 -0.77711478 -0.50639871 0.10311031 0.31976821 0.14420268
0.06045883 0.02546605 0.10874761 0.24332876 0.25716894 0.34049805
0.3782693 0.36469702 0.47446948 0.37057783 0.30517722 0.15711642
-0.07684049 -0.44381447 -0.69521005 -0.82858284 -0.58849775 -0.07524455
0.23883729 0.03272258 -0.03073313 0.01367665 0.19058859 0.24170797
0.3143505 0.36364738 0.39563688 0.46595466 0.50448023 0.51497323
0.56798874 0.38451725 0.14351049 -0.20342918 -0.5115622 -0.65890942
-0.68658244 -0.10850816 0.12503735 -0.0747393 -0.0436021 -0.00109287
0.10749595 0.2384782 0.33346809 0.35150971 0.43893963 0.50663435
0.61259965 0.68900314 0.63951734 0.55298019 0.30645746 -0.0860056
-0.29693893 -0.45574445 -0.56303086 -0.31768753 -0.00978274 -0.12855001
-0.12749811 -0.00334564 0.14105792 0.18898828 0.27838493 0.28466239
0.37369757 0.46390734 0.5855403 0.67869952 0.684962 0.57230816
0.36164277 0.17738629 -0.01833194 -0.27434823 -0.47169185 -0.38460583
-0.16465446 -0.19312152 -0.16407967 0.02039952 0.10203899 0.16737916
0.19507964 0.27134852 0.30666552 0.40054809 0.56748621 0.63146311
0.6367913 0.62705311 0.47891179 0.35125237 0.22198944 -0.0552524
-0.37021863 -0.4511818 ]

```

||Ax-b|| = 35.305181537242234

PROMBLEMA LINEAL

Optimization terminated successfully.

```

x = [9.09831851e-11 1.04824663e-10 9.70226188e-11 8.68146920e-11
8.24001677e-11 8.63847573e-11 1.01069195e-10 1.22143004e-10
1.21528925e-10 1.09675113e-10 9.83376466e-11 9.08138103e-11
9.74280493e-11 1.25777369e-10 1.59453688e-10 1.43289607e-10
9.64393174e-11 8.28976944e-11 1.06626928e-10 1.06236026e-10
9.55633253e-11 9.99891035e-11 8.77366946e-11 7.75495699e-11
7.24561213e-11 8.24402020e-11 1.00523216e-10 1.19985505e-10
1.16592882e-10 1.08527270e-10 1.01778539e-10 1.03860830e-10
1.30054670e-10 1.82241526e-10 2.09085155e-10 1.63148398e-10
9.60896482e-11 8.23885504e-11 1.02784870e-10 1.07363582e-10
1.01707291e-10 9.47477124e-11 7.96559446e-11 6.29972477e-11
6.05701325e-11 8.04489954e-11 1.00370728e-10 1.15465178e-10
1.11784065e-10 1.06680360e-10 1.05762562e-10 1.22350258e-10
1.71858821e-10 2.59750908e-10 2.41021010e-10 1.75416294e-10
9.65348055e-11 8.39461909e-11 9.99832982e-11 1.08102759e-10
1.09498206e-10 9.05340230e-11 6.96565748e-11 4.50810563e-11
5.39479801e-11 8.17140301e-11 1.00626857e-10 1.09815520e-10
1.06364037e-10 1.03569690e-10 1.09875369e-10 1.43562632e-10
1.90561596e-10 1.27515067e-10 2.45702058e-10 1.75257793e-10
9.78469931e-11 8.76718889e-11 9.78533433e-11 1.08312824e-10
1.18969035e-10 8.75430746e-11 5.89863507e-11 3.64109163e-11
5.85194394e-11 8.62260394e-11 1.01235480e-10 1.04202568e-10

```

1.00313359e-10 9.97094861e-11 1.13535729e-10 1.59789294e-10  
 2.72600600e-10 2.87938524e+00 1.75880120e-10 1.62682966e-10  
 9.99915222e-11 9.36864553e-11 9.60234184e-11 1.07909529e-10  
 1.29762999e-10 8.59229686e-11 5.50330799e-11 4.50892400e-11  
 7.19969850e-11 9.28962027e-11 1.02085548e-10 9.93647712e-11  
 9.47105920e-11 9.60589996e-11 1.16083054e-10 1.65799128e-10  
 3.13580407e-10 1.53038278e-09 1.73234079e-10 1.42571499e-10  
 1.02777997e-10 1.01962701e-10 9.44548067e-11 1.06905731e-10  
 1.39997508e-10 8.62125408e-11 6.20614332e-11 6.56996965e-11  
 8.71546875e-11 1.01317513e-10 1.03031842e-10 9.56506304e-11  
 9.03916481e-11 9.33162014e-11 1.16970738e-10 1.63377790e-10  
 1.12595247e-10 2.93559815e-11 1.67450549e-10 1.21340311e-10  
 1.05787012e-10 1.11978119e-10 9.33308923e-11 1.05433322e-10  
 1.45601832e-10 8.87266148e-11 7.61876825e-11 8.60873257e-11  
 1.02211053e-10 1.11811236e-10 1.03908412e-10 9.31622189e-11  
 8.76008530e-11 9.18104849e-11 1.15981349e-10 1.54870607e-10  
 1.18903467e-10 1.40533272e-10 1.28436249e-10 1.03470047e-10  
 1.08340230e-10 1.22359739e-10 9.28754735e-11 1.03725557e-10  
 1.42811845e-10 9.31856063e-11 9.07321336e-11 1.04287095e-10  
 1.23352625e-10 1.21551502e-10 1.04554776e-10 9.18907298e-11  
 8.64384634e-11 9.16735433e-11 1.13325988e-10 1.39907536e-10  
 1.36764743e-10 1.21075632e-10 9.55377083e-11 9.03843257e-11  
 1.09653538e-10 1.30963417e-10 9.32698399e-11 1.02058494e-10  
 1.32263772e-10 9.92265771e-11 1.06320712e-10 1.31274039e-10  
 1.46125176e-10 1.22082201e-10 1.04881892e-10 9.18081470e-11  
 8.70376826e-11 9.29478575e-11 1.09554906e-10 1.20708960e-10  
 1.10576280e-10 8.92828924e-11 7.68314117e-11 8.15309015e-11  
 1.09268167e-10 1.35711886e-10 9.46406167e-11 1.00677599e-10  
 1.32263772e-10 9.92265774e-11 1.06320712e-10 1.31274039e-10  
 1.46125175e-10 1.22082201e-10 1.04881892e-10 9.18081472e-11  
 8.70376826e-11 9.29478570e-11 1.09554906e-10 1.20708960e-10  
 1.10576280e-10 8.92828923e-11 7.68314120e-11 8.15309016e-11  
 1.09268167e-10 1.35711886e-10 9.46406162e-11 1.00677599e-10  
 1.18595780e-10 1.06842387e-10 1.28823746e-10 1.62416427e-10  
 1.65689274e-10 1.07649364e-10 1.04941747e-10 9.28989557e-11  
 8.94587719e-11 9.56013428e-11 1.05345193e-10 1.03024002e-10  
 8.74074530e-11 7.04834120e-11 6.34903919e-11 7.59809055e-11  
 1.07505888e-10 1.35673333e-10 9.70572148e-11 9.97455011e-11  
 1.06136707e-10 1.15369023e-10 1.49688313e-10 3.52505023e-10  
 4.00116253e-10 8.65794201e-11 1.04901048e-10 9.51558677e-11  
 9.37234449e-11 9.94746213e-11 1.01303593e-10 9.02447488e-11  
 7.23474610e-11 5.14122519e-11 5.18661270e-11 7.32546137e-11  
 1.05346113e-10 1.31283686e-10 1.00504160e-10 9.93295043e-11  
 9.67754623e-11 1.21881457e-10 1.82041675e-10 3.13244847e-09  
 1.12876845e-09 7.62669689e-11 1.04882083e-10 9.85526413e-11  
 9.98999968e-11 1.04129362e-10 9.78708371e-11 8.15783356e-11  
 5.74823389e-11 3.19772405e-11 4.53588549e-11 7.32790645e-11  
 1.03629356e-10 1.23718856e-10 1.04799154e-10 9.94206917e-11

```

9.06445287e-11 1.22504078e-10 5.24522067e-10 2.87938523e+00
3.13997783e-10 8.43487243e-11 1.04817364e-10 1.02971461e-10
1.07758030e-10 1.08696703e-10 9.53200205e-11 7.54158480e-11
4.42264641e-11 2.25887162e-11 4.74949476e-11 7.60909468e-11
1.02458829e-10 1.14433502e-10 1.09437707e-10 9.99633935e-11
8.71982719e-11 1.17074700e-10 1.04830718e-09 5.89697617e-09
3.32702937e-11 9.74988307e-11 1.04511572e-10 1.08103351e-10
1.16113245e-10 1.12035982e-10 9.37967104e-11 7.17638063e-11
3.97191175e-11 2.96350000e-11 5.72508041e-11 8.18130578e-11
1.01427075e-10 1.04965184e-10 1.13439038e-10 1.00878316e-10
8.58309989e-11 1.10705906e-10 3.65744892e-10 1.47619227e-10
7.56376231e-11 1.01240152e-10 1.03851822e-10 1.13408303e-10
1.22909166e-10 1.13326417e-10 9.33596849e-11 7.15906056e-11
4.71618714e-11 4.86873097e-11 7.05852493e-11 9.09100844e-11
1.00197231e-10 9.65820400e-11 1.15470066e-10 1.02073684e-10
8.61215461e-11 1.07925393e-10 9.27337445e-11 7.75986832e-11
9.60410665e-11 9.69878891e-11 1.02919167e-10 1.18144475e-10
1.26310000e-10 1.12663651e-10 9.40083037e-11 7.51604272e-11
6.22439790e-11 6.95019746e-11 8.69653448e-11 1.04218713e-10
9.87762811e-11 8.99952777e-11 1.14563532e-10 1.03446576e-10
8.78568539e-11 1.07252783e-10 9.81745234e-11 9.46048979e-11
9.04712137e-11 9.12754976e-11 1.01920430e-10 1.21379668e-10
1.25546107e-10 1.11073176e-10 9.56971911e-11 8.16971563e-11
7.84755189e-11 9.01505139e-11 1.14117098e-10 1.22170871e-10
9.74195923e-11 8.54306447e-11 1.11072221e-10 1.04879341e-10
9.09831847e-11 1.04824663e-10 9.70226185e-11 8.68146919e-11
8.24001682e-11 8.63847573e-11 1.01069196e-10 1.22143003e-10
1.21528925e-10 1.09675113e-10 9.83376468e-11 9.08138103e-11
9.74280494e-11 1.25777369e-10 1.59453687e-10 1.43289608e-10
9.64393175e-11 8.28976946e-11 1.06626927e-10 1.06236027e-10]
c^Tx = [5.75877052]
s = [ 4.04674092e-14  7.13022306e-14 -1.23897004e-11 -3.67568753e-11
-7.15908444e-11 -1.14726895e-10 -1.63367542e-10 -2.14267271e-10
-2.63919553e-10 -3.08758352e-10 -3.45360185e-10 -3.70646180e-10
-3.82064158e-10 -3.77721410e-10 -3.56535246e-10 -3.18271187e-10
-2.63591371e-10 -1.94037675e-10 -1.11946674e-10 -2.03629891e-11
 7.71143149e-11  1.76510584e-10  2.73671752e-10  3.64464459e-10
 4.44995152e-10  5.11801268e-10  5.62026869e-10  5.93572302e-10
 6.05192341e-10  5.96571681e-10  5.68332048e-10  5.21986898e-10
 4.59893457e-10  3.85081300e-10  3.01134229e-10  2.11976658e-10
 1.21672672e-10  3.42203488e-11 -4.66627847e-11 -1.17715171e-10
-1.76314296e-10 -2.20585328e-10 -2.49487098e-10 -2.62855071e-10
-2.61381139e-10 -2.46573872e-10 -2.20622631e-10 -1.86297644e-10
-1.46761270e-10 -1.05389808e-10 -6.55876464e-11 -3.05891423e-11
-3.29139493e-12  1.39066536e-11  1.92310334e-11  1.16246318e-11
-9.20442889e-12 -4.27603508e-11 -8.77707906e-11 -1.42286849e-10]
||Ax-b|| = 2.204822481797681e-09
E_x = 0 , la condición de no negatividad se cumple

```

# PROMBLEMA DUAL

Optimization terminated successfully.

x = [-4.57680914 0.82163807 4.90005464 1.33829308 0.47032874 -0.29644776

-1.00572451 -1.69776748 -1.65413437 -0.99974064 -0.43321571 0.04632941

0.49243543 0.90181822 1.14324926 1.04241133 0.70043729 0.31451619

-0.05834662 -0.42355853 -0.76955155 -1.01121785 -1.01424992 -0.78058046

-0.44819107 -0.10307125 0.24132204 0.58558719 0.89078766 1.04088521

0.93945469 0.65643453 0.31581794 -0.03222202 -0.38765275 -0.74739259

-1.04241009 -1.115933 -0.91149637 -0.56451189 -0.18870329 0.19565323

0.60356844 1.01627365 1.26428663 1.13297644 0.73951771 0.28588286

-0.19274192 -0.74717874 -1.47852223 -2.10717917 -1.64513171 -0.83945148

-0.12120936 0.76542639 1.90390796 6.35196853 0.94508197 -6.61495317]

c^Tx = [-5.75877048]

s = [ 5.47811010e-01 5.20460073e-01 8.07122583e-01 1.22100628e+00

1.54963321e+00 1.67249508e+00 1.61222197e+00 1.49427059e+00

1.44573375e+00 1.50051929e+00 1.57366707e+00 1.52368017e+00

1.26399730e+00 8.48521686e-01 4.66076100e-01 3.32944495e-01

5.41716426e-01 9.66183258e-01 1.30512605e+00 1.27301343e+00

6.14640842e-01 7.51889894e-01 1.13572748e+00 1.54164602e+00

1.77776589e+00 1.78171372e+00 1.63223407e+00 1.47521633e+00

1.41492733e+00 1.44133314e+00 1.44107590e+00 1.28607704e+00

9.41790856e-01 5.19495851e-01 2.27530589e-01 2.40285247e-01

5.64006593e-01 9.95872162e-01 1.22944495e+00 1.07245033e+00

7.27950675e-01 1.01324546e+00 1.44796164e+00 1.79695525e+00

1.91208853e+00 1.79664615e+00 1.57598941e+00 1.39884395e+00

1.33407456e+00 1.32891570e+00 1.25528448e+00 1.01396880e+00

6.26605290e-01 2.48425908e-01 8.21562901e-02 2.39258810e-01

6.38883999e-01 1.02605894e+00 1.12608940e+00 8.63148644e-01

8.74073672e-01 1.27300342e+00 1.70616502e+00 1.95613992e+00

1.93639983e+00 1.71549130e+00 1.45027193e+00 1.27436512e+00

1.21292746e+00 1.17682617e+00 1.03870198e+00 7.40175708e-01

3.56456637e-01 6.80068923e-02 4.74874883e-02 3.29988988e-01

7.57317325e-01 1.05310263e+00 1.00752562e+00 6.70353256e-01

1.03538524e+00 1.49983314e+00 1.87919447e+00 2.00000000e+00

1.84776749e+00 1.54803764e+00 1.27024500e+00 1.11679380e+00

1.06609816e+00 1.00340880e+00 8.17451445e-01 4.97721265e-01

1.63928811e-01 5.43778356e-11 1.27705753e-01 5.01532382e-01

9.05021762e-01 1.07374136e+00 8.88054128e-01 5.17318130e-01

1.19242883e+00 1.66637561e+00 1.94618010e+00 1.92324532e+00

1.65688188e+00 1.31448255e+00 1.05762255e+00 9.45135420e-01

9.11296458e-01 8.29580272e-01 6.18218964e-01 3.15849050e-01

7.22435095e-02 5.26078662e-02 3.13135578e-01 7.33198328e-01

1.06418198e+00 1.08548579e+00 7.82084964e-01 4.22501562e-01

1.32626266e+00 1.75254335e+00 1.89904244e+00 1.73513364e+00

1.38676662e+00 1.04299623e+00 8.38049958e-01 7.80094522e-01

7.67193708e-01 6.76306880e-01 4.65034910e-01 2.16495537e-01

9.24593332e-02 2.19485206e-01 5.81411389e-01 9.97044493e-01

1.21560090e+00 1.08691937e+00 7.02399570e-01 3.97339828e-01

1.42074440e+00	1.74794325e+00	1.74346699e+00	1.45835398e+00
1.07000160e+00	7.66323921e-01	6.38010935e-01	6.41577470e-01
6.51170832e-01	5.62075656e-01	3.76375541e-01	2.11644225e-01
2.22137955e-01	4.80504149e-01	9.00175164e-01	1.26124714e+00
1.34101517e+00	1.07786920e+00	6.58609180e-01	4.44867805e-01
1.46447816e+00	1.65313016e+00	1.49821845e+00	1.12629008e+00
7.44793349e-01	5.17836399e-01	4.81633136e-01	5.46291465e-01
5.77221903e-01	5.00664571e-01	3.62934485e-01	3.01880255e-01
4.45638220e-01	8.04181958e-01	1.23097929e+00	1.49393952e+00
1.42529798e+00	1.05942685e+00	6.55995561e-01	5.59352918e-01
1.45218899e+00	1.47953993e+00	1.19287742e+00	7.78993721e-01
4.50366789e-01	3.27504923e-01	3.87778031e-01	5.05729406e-01
5.54266251e-01	4.99480708e-01	4.26332933e-01	4.76319829e-01
7.36002697e-01	1.15147831e+00	1.53392390e+00	1.66705551e+00
1.45828357e+00	1.03381674e+00	6.94873955e-01	7.26986572e-01
1.45218899e+00	1.47953993e+00	1.19287742e+00	7.78993721e-01
4.50366789e-01	3.27504923e-01	3.87778031e-01	5.05729406e-01
5.54266251e-01	4.99480708e-01	4.26332933e-01	4.76319829e-01
7.36002697e-01	1.15147831e+00	1.53392390e+00	1.66705551e+00
1.45828357e+00	1.03381674e+00	6.94873955e-01	7.26986572e-01
1.38535916e+00	1.24811011e+00	8.64272518e-01	4.58353985e-01
2.22234106e-01	2.18286278e-01	3.67765931e-01	5.24783675e-01
5.85072668e-01	5.58666858e-01	5.58924095e-01	7.13922960e-01
1.05820914e+00	1.48050415e+00	1.77246941e+00	1.75971475e+00
1.43599341e+00	1.00412784e+00	7.70555052e-01	9.27549675e-01
1.27204932e+00	9.86754544e-01	5.52038357e-01	2.03044752e-01
8.79114688e-02	2.03353845e-01	4.24010591e-01	6.01156046e-01
6.65925445e-01	6.71084299e-01	7.44715521e-01	9.86031205e-01
1.37339471e+00	1.75157409e+00	1.91784371e+00	1.76074119e+00
1.36111600e+00	9.73941056e-01	8.73910597e-01	1.13685136e+00
1.12592633e+00	7.26996580e-01	2.93834981e-01	4.38600832e-02
6.36001691e-02	2.84508695e-01	5.49728074e-01	7.25634884e-01
7.87072543e-01	8.23173827e-01	9.61298023e-01	1.25982429e+00
1.64354336e+00	1.93199311e+00	1.95251251e+00	1.67001101e+00
1.24268268e+00	9.46897368e-01	9.92474385e-01	1.32964674e+00
9.64614757e-01	5.00166857e-01	1.20805528e-01	-5.89701621e-11
1.52232509e-01	4.51962357e-01	7.29754996e-01	8.83206204e-01
9.33901835e-01	9.96591202e-01	1.18254855e+00	1.50227874e+00
1.83607119e+00	2.00000000e+00	1.87229425e+00	1.49846762e+00
1.09497824e+00	9.26258640e-01	1.11194587e+00	1.48268187e+00
8.07571169e-01	3.33624388e-01	5.38199039e-02	7.67546751e-02
3.43118120e-01	6.85517446e-01	9.42377454e-01	1.05486458e+00
1.08870354e+00	1.17041973e+00	1.38178104e+00	1.68415095e+00
1.92775649e+00	1.94739213e+00	1.68686442e+00	1.26680167e+00
9.35818023e-01	9.14514209e-01	1.21791504e+00	1.57749844e+00
6.73737338e-01	2.47456652e-01	1.00957564e-01	2.64866362e-01
6.13233380e-01	9.57003772e-01	1.16195004e+00	1.21990548e+00
1.23280629e+00	1.32369312e+00	1.53496509e+00	1.78350446e+00

```

1.90754067e+00 1.78051479e+00 1.41858861e+00 1.00295551e+00
7.84399102e-01 9.13080626e-01 1.29760043e+00 1.60266017e+00
5.79255599e-01 2.52056751e-01 2.56533009e-01 5.41646015e-01
9.29998402e-01 1.23367608e+00 1.36198906e+00 1.35842253e+00
1.34882917e+00 1.43792434e+00 1.62362446e+00 1.78835577e+00
1.77786204e+00 1.51949585e+00 1.09982484e+00 7.38752864e-01
6.58984831e-01 9.22130802e-01 1.34139082e+00 1.55513220e+00
5.35521845e-01 3.46869844e-01 5.01781547e-01 8.73709924e-01
1.25520665e+00 1.48216360e+00 1.51836686e+00 1.45370853e+00
1.42277810e+00 1.49933543e+00 1.63706551e+00 1.69811975e+00
1.55436178e+00 1.19581804e+00 7.69020712e-01 5.06060482e-01
5.74702022e-01 9.40573153e-01 1.34400444e+00 1.44064708e+00
5.47811010e-01 5.20460073e-01 8.07122583e-01 1.22100628e+00
1.54963321e+00 1.67249508e+00 1.61222197e+00 1.49427059e+00
1.44573375e+00 1.50051929e+00 1.57366707e+00 1.52368017e+00
1.26399730e+00 8.48521686e-01 4.66076100e-01 3.32944495e-01
5.41716426e-01 9.66183258e-01 1.30512605e+00 1.27301343e+00]
|c^Tx - b^t(lambda)|/|c^Tx| = [6.70911031e-09]
||A^T+s-c|| = 0
||Ax-b|| = 2.204822481797681e-09
E_x = 0 , la condición de no negatividad se cumple
E_s = 5.897016208677996e-11 , la condición de no negatividad se cumple
Sum_i |x_is_i| = 3.962670517409115e-08 , la condición de complementaridad se
cumple
||Ax-b|| = 2.204824055250458e-09

C:\Users\batma\AppData\Local\Temp\ipykernel_13440\2397109568.py:23:
OptimizeWarning: A_eq does not appear to be of full row rank. To improve
performance, check the problem formulation for redundant equality constraints.
    status = linprog(c, A_eq=AAA, b_eq=b)
C:\Users\batma\AppData\Local\Temp\ipykernel_13440\2397109568.py:23:
OptimizeWarning: Solving system with option 'cholesky':True failed. It is normal
for this to happen occasionally, especially as the solution is approached.
However, if you see this frequently, consider setting option 'cholesky' to
False.
    status = linprog(c, A_eq=AAA, b_eq=b)
C:\Users\batma\AppData\Local\Temp\ipykernel_13440\2397109568.py:23:
OptimizeWarning: Solving system with option 'sym_pos':True failed. It is normal
for this to happen occasionally, especially as the solution is approached.
However, if you see this frequently, consider setting option 'sym_pos' to False.
    status = linprog(c, A_eq=AAA, b_eq=b)
c:\Program Files\Python310\lib\site-packages\scipy\optimize\_linprog_ip.py:117:
LinAlgWarning: Ill-conditioned matrix (rcond=1.45476e-18): result may not be
accurate.
    return sp.linalg.solve(M, r, sym_pos=sym_pos)
c:\Program Files\Python310\lib\site-packages\scipy\optimize\_linprog_ip.py:117:
LinAlgWarning: Ill-conditioned matrix (rcond=5.43869e-19): result may not be
accurate.

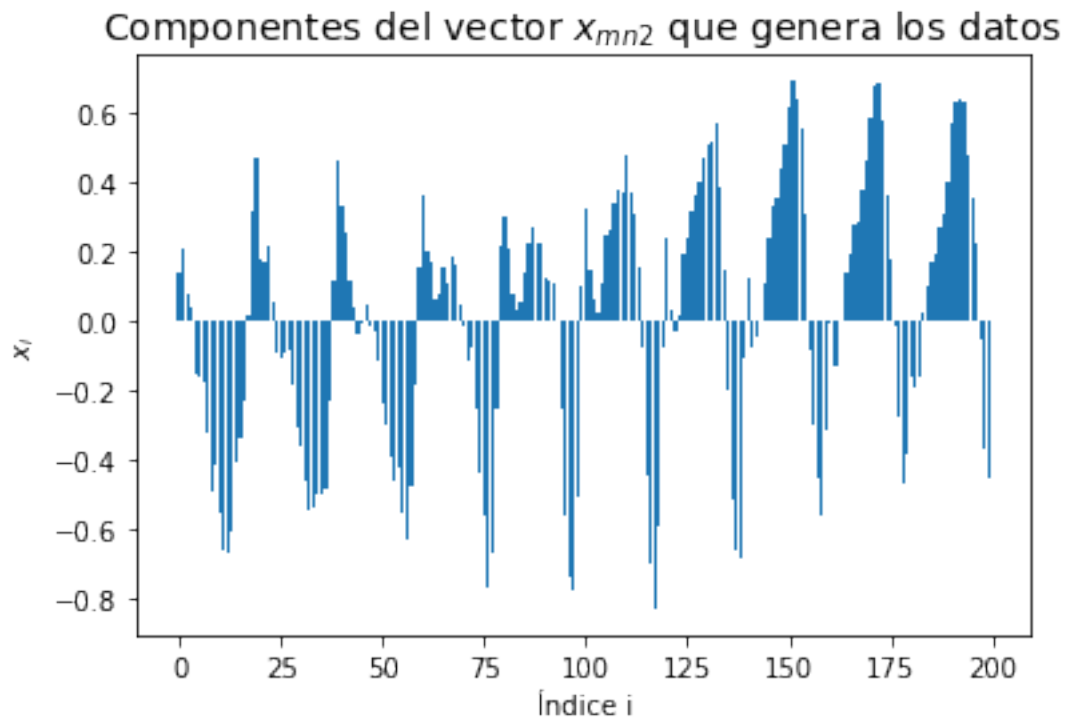
```

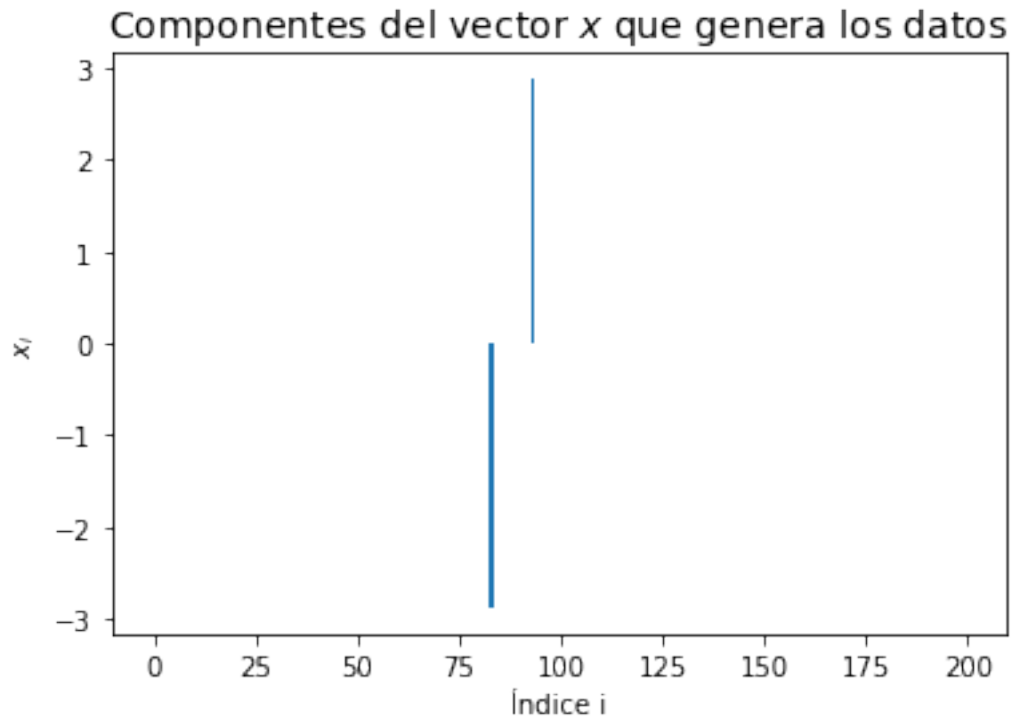
```

    return sp.linalg.solve(M, r, sym_pos=sym_pos)
c:\Program Files\Python310\lib\site-packages\scipy\optimize\_linprog_ip.py:117:
LinAlgWarning: Ill-conditioned matrix (rcond=3.10799e-21): result may not be
accurate.
    return sp.linalg.solve(M, r, sym_pos=sym_pos)
c:\Program Files\Python310\lib\site-packages\scipy\optimize\_linprog_ip.py:117:
LinAlgWarning: Ill-conditioned matrix (rcond=7.97324e-20): result may not be
accurate.
    return sp.linalg.solve(M, r, sym_pos=sym_pos)
c:\Program Files\Python310\lib\site-packages\scipy\optimize\_linprog_ip.py:117:
LinAlgWarning: Ill-conditioned matrix (rcond=1.76113e-18): result may not be
accurate.
    return sp.linalg.solve(M, r, sym_pos=sym_pos)
c:\Program Files\Python310\lib\site-packages\scipy\optimize\_linprog_ip.py:117:
LinAlgWarning: Ill-conditioned matrix (rcond=1.58417e-18): result may not be
accurate.
    return sp.linalg.solve(M, r, sym_pos=sym_pos)

```

[ ]: Text(0.5, 1.0, 'Componentes del vector  $x_{mn2}$  que genera los datos')





Los picos de las gráficas  $x_{real}$  y los de  $x_{mn2}$  parecen coincidir, así como donde se concentran los datos de  $x$  coinciden con los de  $x_{mn2}$ . Ahora podemos ver que las gráficas de  $x$  y  $x_{real}$  se parecen aún mas, ambas tiene picos más “agresivos” pero puesto que  $x$  se cálculo minimizando la norma 1 esto máxima los 0 por lo que perdimos algunos picos.