ESCUELA
POLITÉCNICA
NACIONAL

Métodos numéricos.

Nombre: Luis Enrique Pérez Señalin.

CONJUNTO DE EJERCICIOS

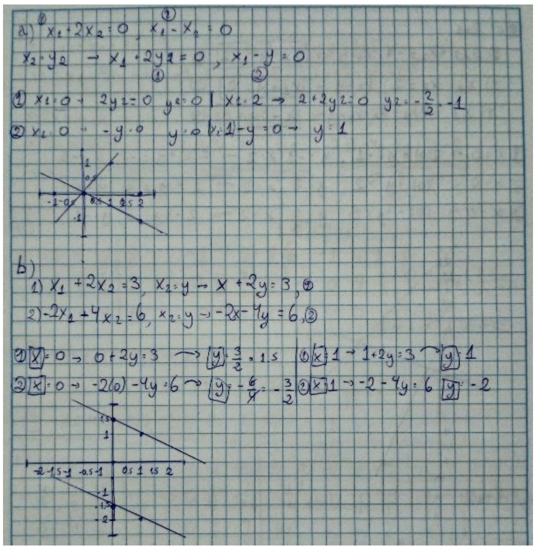
1. Para cada uno de los siguientes sistemas lineales, obtenga, de ser posible, una solución con métodos gráficos. Explique los resultados desde un punto de vista geométrico.

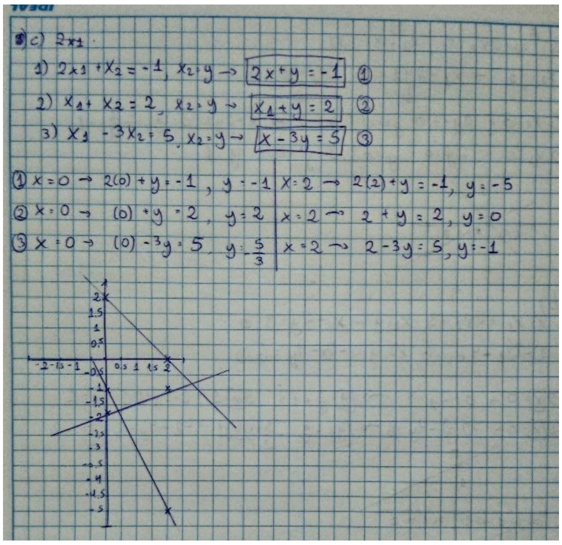
a. $x_1 + 2x_2 = 0,$
$x_1 - x_2 = 0.$

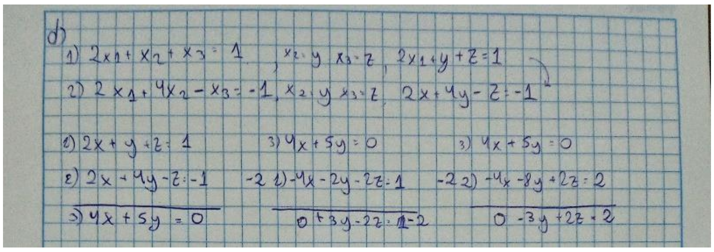
b. $x_1 + 2x_2 = 3,$
$-2x_1 - 4x_2 = 6.$

c. $2x_1 + x_2 = -1,$
$x_1 + x_2 = 2,$
$x_1 - 3x_2 = 5.$

d. $2x_1 + x_2 + x_3 = 1,$
$2x_1 + 4x_2 - x_3 = -1.$



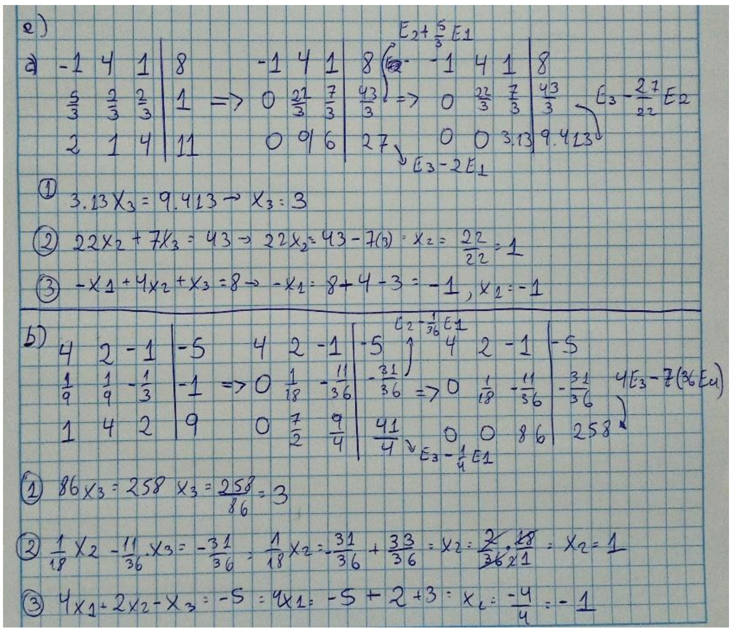




2. Utilice la eliminación gaussiana con sustitución hacia atrás y aritmética de redondeo de dos dígitos para resolver los siguientes sistemas lineales. No reordene las ecuaciones. (La solución exacta para cada sistema es $x_1 = -1$, $x_2 = 2, x_3 = 3$.)

a. $-x_1 + 4x_2 + x_3 = 8,$
$\frac{5}{3}x_1 + \frac{2}{3}x_2 + \frac{2}{3}x_3 = 1,$
$2x_1 + x_2 + 4x_3 = 11.$

b. $4x_1 + 2x_2 - x_3 = -5,$
$\frac{1}{9}x_1 + \frac{1}{9}x_2 - \frac{1}{3}x_3 = -1,$
$x_1 + 4x_2 + 2x_3 = 9,$



3. Utilice el algoritmo de eliminación gaussiana para resolver, de ser posible, los siguientes sistemas lineales, y determine si se necesitan intercambios de fila:

a. $$\begin{aligned} x_1 - x_2 + 3x_3 &= 2, \\ 3x_1 - 3x_2 + x_3 &= -1, \\ x_1 + x_2 &= 3. \end{aligned}$$

b. $$\begin{aligned} 2x_1 - 1.5x_2 + 3x_3 &= 1, \\ -x_1 + 2x_3 &= 3, \\ 4x_1 - 4.5x_2 + 5x_3 &= 1, \end{aligned}$$

c. $$\begin{aligned} 2x_1 &= 3, \\ x_1 + 1.5x_2 &= 4.5, \\ -3x_2 + 0.5x_3 &= -6.6. \\ 2x_1 - 2x_2 + x_3 + x_4 &= 0.8. \end{aligned}$$

d. $$\begin{aligned} x_1 + x_2 + x_4 &= 2, \\ 2x_1 + x_2 - x_3 + x_4 &= 1, \\ 4x_1 - x_2 - 2x_3 + 2x_4 &= 0, \\ 3x_1 - x_2 - x_3 + 2x_4 &= -3. \end{aligned}$$

Respuestas:
A: [1.1875 1.8125 0.875]
B: [-1. -0. 1.]
C: [1.5 2. -1.2 3.]
D: No existe solución única.

Eliminación Gaussiana con sustitución hacia atrás

```
%load_ext autoreload
```

```
%autoreload 2
from src import eliminacion_gaussiana
import numpy as np
```

```
[08-10 12:41:37][INFO] 2024-08-10 12:41:37.927547
[08-10 12:41:38][INFO] 2024-08-10 12:41:38.011599
[08-10 12:41:38][INFO] 2024-08-10 12:41:38.015001
```

```
A = np.array([[1, -1, 3, 2],
     [3, -3, 1, -1],
     [1, 1, 0, 3]])

B = np.array([[ 2, -1.5, 3, 1],
     [-1,    0, 2, 3],
     [ 4, -4.5, 5, 1]])

C = np.array([[2,   0,   0,  0,    3],
     [1, 1.5,   0,  0,  4.5],
     [0,  -3, 0.5,  0, -6.6],
     [2,  -2,   1,  1,  0.8]])

D = np.array(
    [[1,  1,  0, 1,  2],
     [2,  1, -1, 1,  1],
     [4, -1, -2, 2,  0],
     [3, -1, -1, 2, -3]])
```

```
R = eliminacion_gaussiana(A)
print(R)
```

```
[1.1875 1.8125 0.875 ]
```

```
R = eliminacion_gaussiana(B)
print(R)
```

```
[-1. -0.  1.]
```

```
R = eliminacion_gaussiana(C)
print(R)
```

```
[ 1.5  2.  -1.2  3. ]
```

```
R = eliminacion_gaussiana(D)
print(R)
```

```
ValueError: No existe solución única.
```

4. Use el algoritmo de eliminación gaussiana y la aritmética computacional de precisión de 32 bits para resolver los siguientes sistemas lineales.

a. $\frac{1}{4}x_1 + \frac{1}{5}x_2 + \frac{1}{6}x_3 = 9,$
$\frac{1}{3}x_1 + \frac{1}{4}x_2 + \frac{1}{5}x_3 = 8,$
$\frac{1}{2}x_1 + x_2 + 2x_3 = 8.$

b. $3.333x_1 + 15920x_2 - 10.333x_3 = 15913,$
$2.222x_1 + 16.71x_2 + 9.612x_3 = 28.544,$
$1.5611x_1 + 5.1791x_2 + 1.6852x_3 = 8.4254.$

c. $x_1 + \frac{1}{2}x_2 + \frac{1}{3}x_3 + \frac{1}{4}x_4 = \frac{1}{6},$
$\frac{1}{2}x_1 + \frac{1}{3}x_2 + \frac{1}{4}x_3 + \frac{1}{5}x_4 = \frac{1}{7},$
$\frac{1}{3}x_1 + \frac{1}{4}x_2 + \frac{1}{5}x_3 + \frac{1}{6}x_4 = \frac{1}{8},$
$\frac{1}{4}x_1 + \frac{1}{5}x_2 + \frac{1}{6}x_3 + \frac{1}{7}x_4 = \frac{1}{9}.$

d. $2x_1 + x_2 - x_3 + x_4 - 3x_5 = 7,$
$x_1 + 2x_3 - x_4 + x_5 = 2,$
$-2x_2 - x_3 + x_4 - x_5 = -5,$
$3x_1 + x_2 - 4x_3 + 5x_5 = 6,$
$x_1 - x_2 - x_3 - x_4 + x_5 = -3.$

Respuestas:
A: [-227.07666, 476.92264, -177.69217]
B: [0.9999997, 1.0000001, 0.9999998]
C: [-0.03174075, 0.5951853, -2.3808312, 2.7777011]
D: [1.8830411, 2.807017 , 0.73099416, 1.4385967 0.09356727]

Eliminación Gaussiana con sustitución hacia atrás

```
%load_ext autoreload
```

```
%autoreload 2
from ejercicio4 import eliminacion_gaussiana_32bits
import numpy as np
```

```
A = np.array(
    [[1/4, 1/5, 1/6, 9],
     [1/3, 1/4, 1/5, 8],
     [1/2,   1,   2, 8]])

B = np.array(
    [[  3.333,  15920, -10.333,  15913],
     [  2.222,  16.71,   9.612, 28.544],
     [ 1.5611, 5.1791,  1.6852, 8.4254]])

C = np.array(
    [[  1, 1/2, 1/3, 1/4, 1/6],
     [1/2, 1/3, 1/4, 1/5, 1/7],
     [1/3, 1/4, 1/5, 1/6, 1/8],
     [1/4, 1/5, 1/6, 1/7, 1/9]])

D = np.array(
    [[2,  1, -1,  1, -3,  7],
     [1,  0,  2, -1,  1,  2],
     [0, -2, -1,  1, -1, -5],
     [3,  1, -4,  0,  5,  6],
     [1, -1, -1, -1,  1, -3]])
```

```
R_A = eliminacion_gaussiana_32bits(A)
R_B = eliminacion_gaussiana_32bits(B)
R_C = eliminacion_gaussiana_32bits(C)
R_D = eliminacion_gaussiana_32bits(D)
```

```
print(f'Respuesta de A: {R_A}')
print(f'Respuesta de B: {R_B}')
print(f'Respuesta de C: {R_C}')
print(f'Respuesta de D: {R_D}')
```

```
Respuesta de A: [-227.07666  476.92264 -177.69217]
Respuesta de B: [0.9999997 1.0000001 0.9999998]
Respuesta de C: [-0.03174075  0.5951853  -2.3808312   2.7777011 ]
Respuesta de D: [1.8830411  2.807017   0.73099416 1.4385967  0.09356727]
```

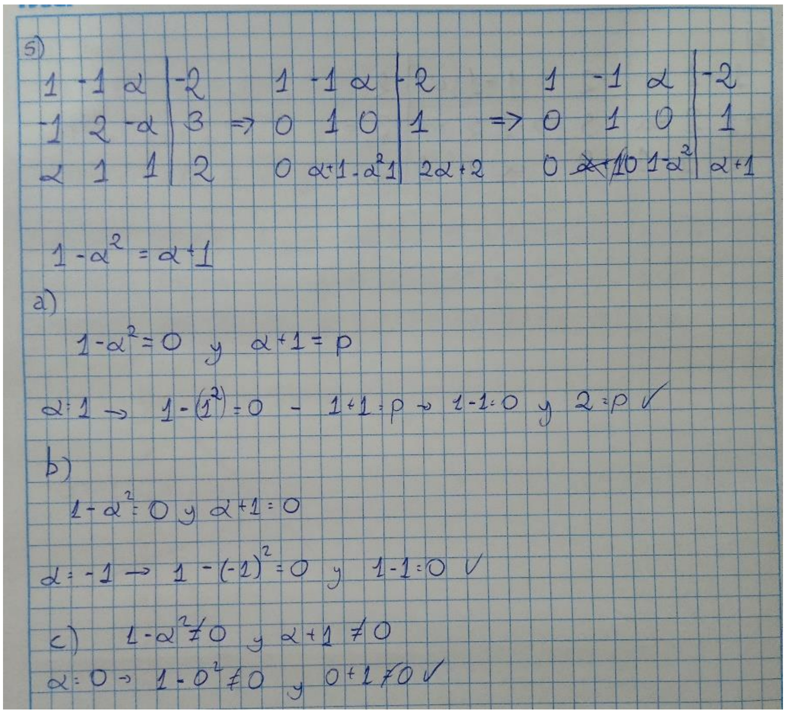
1. Dado el sistema lineal:

$$\begin{aligned} x_1 - \ x_2 + \alpha x_3 &= -2, \\ -x_1 + 2x_2 - \alpha x_3 &= 3, \\ \alpha x_1 + \ x_2 + \ \ x_3 &= 2. \end{aligned}$$

a. Encuentre el valor(es) de α para los que el sistema no tiene soluciones.
b. Encuentre el valor(es) de α para los que el sistema tiene un número infinito de soluciones.
c. Suponga que existe una única solución para una a determinada, encuentre la solución.

Respuesta:

a) $\alpha = 1$
b) $\alpha = -1$
c) $\alpha = 0$



5)

$$\left[\begin{array}{ccc|c} 1 & -1 & \alpha & -2 \\ -1 & 2 & -\alpha & 3 \\ \alpha & 1 & 1 & 2 \end{array}\right] \Rightarrow \left[\begin{array}{ccc|c} 1 & -1 & \alpha & -2 \\ 0 & 1 & 0 & 1 \\ 0 & \alpha+1 & 1-\alpha^2 1 & 2\alpha+2 \end{array}\right] \Rightarrow \left[\begin{array}{ccc|c} 1 & -1 & \alpha & -2 \\ 0 & 1 & 0 & 1 \\ 0 & \cancel{\alpha+1} 0 & 1-\alpha^2 & \alpha+1 \end{array}\right]$$

$1-\alpha^2 = \alpha+1$

a)

$1-\alpha^2 = 0$ y $\alpha+1 = p$

$\alpha = 1 \to 1-(1^2) = 0 \ - \ 1+1 = p \to 1-1=0$ y $2 = p$ ✓

b)

$1-\alpha^2 = 0$ y $\alpha+1 = 0$

$\alpha = -1 \to 1-(-1)^2 = 0$ y $1-1=0$ ✓

c) $1-\alpha^2 \neq 0$ y $\alpha+1 \neq 0$

$\alpha = 0 \to 1-0^2 \neq 0$ y $0+1 \neq 0$ ✓

EJERCICIOS APLICADOS

2. Suponga que en un sistema biológico existen n especies de animales y m fuentes de alimento. Si x_j representa la población de las j-ésimas especies, para cada $j = 1, \cdots, n$; b_i; representa el suministro diario disponible del i-ésimo alimento y a_{ij} representa la cantidad del i-ésimo alimento.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2, \\ \vdots \qquad \vdots \qquad\quad \vdots \quad &\quad \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m, \end{aligned}$$

representa un equilibrio donde existe un suministro diario de alimento para cumplir con precisión con el promedio diario de consumo de cada especie.

a. Si

$$A = [a_{ij}] = \begin{matrix} 1 & 2 & 0 & 3 \\ 1 & 0 & 2 & 2 \\ 0 & 0 & 1 & 1 \end{matrix}$$

$\mathbf{x} = (x_j) = [1000, 500, 350, 400]$, y $\mathbf{b} = (b_i) = [3500, 2700, 900]$. ¿Existe suficiente alimento para satisfacer el consumo promedio diario?

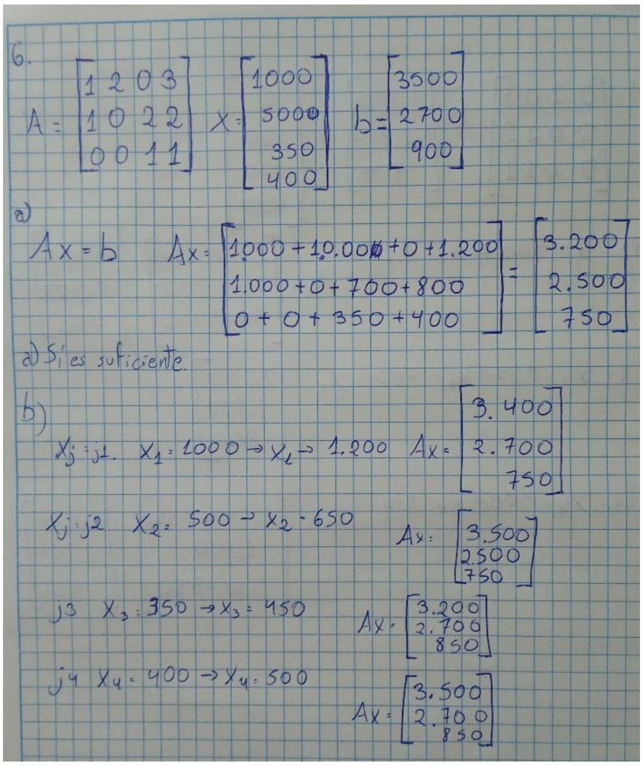
b. ¿Cuál es el número máximo de animales de cada especie que se podría agregar de forma individual al sistema con el suministro de alimento que cumpla con el consumo?

c. Si la especie 1 se extingue, ¿qué cantidad de incremento individual de las especies restantes se podría soportar?

d. Si la especie 2 se extingue, ¿qué cantidad de incremento individual de las especies restantes se podría soportar?

Respuesta:

a) Sí hay suficiente comida para las especies.



6.

$$A = \begin{bmatrix} 1 & 2 & 0 & 3 \\ 1 & 0 & 2 & 2 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad x = \begin{bmatrix} 1000 \\ 5000 \\ 350 \\ 400 \end{bmatrix} \quad b = \begin{bmatrix} 3500 \\ 2700 \\ 900 \end{bmatrix}$$

a)

$$Ax = b \qquad Ax = \begin{bmatrix} 1000 + 10.000 + 0 + 1.200 \\ 1.000 + 0 + 700 + 800 \\ 0 + 0 + 350 + 400 \end{bmatrix} = \begin{bmatrix} 3.200 \\ 2.500 \\ 750 \end{bmatrix}$$

a) Sí es suficiente

b)

$X_j: j1.\ X_1 = 1000 \rightarrow X_1 \rightarrow 1.200 \quad Ax = \begin{bmatrix} 3.400 \\ 2.700 \\ 750 \end{bmatrix}$

$X_j: j2\ X_2 = 500 \rightarrow X_2 = 650 \quad Ax = \begin{bmatrix} 3.500 \\ 2500 \\ 750 \end{bmatrix}$

$j3\ X_3 = 350 \rightarrow X_3 = 450 \quad Ax = \begin{bmatrix} 3.200 \\ 2.700 \\ 850 \end{bmatrix}$

$j4\ X_4 = 400 \rightarrow X_4 = 500 \quad Ax = \begin{bmatrix} 3.500 \\ 2.700 \\ 850 \end{bmatrix}$

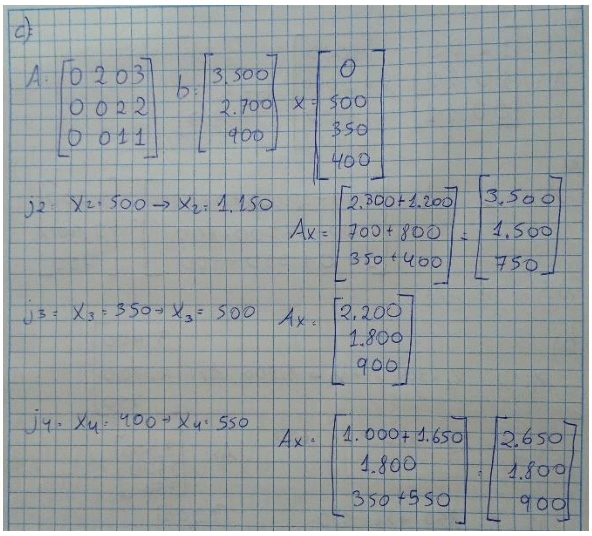
b) Los valores pueden ser:

$x_1 = \ 1.000 \rightarrow 1.200$

$x_2 = \quad 500 \rightarrow \quad 650$

$x_3 = \quad 350 \rightarrow \quad 450$

$x_4 = \quad 400 \rightarrow \quad 500$



c)

$$A = \begin{bmatrix} 0 & 2 & 0 & 3 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 3.500 \\ 2.700 \\ 400 \end{bmatrix} \quad x = \begin{bmatrix} 0 \\ 500 \\ 350 \\ 400 \end{bmatrix}$$

$j2:\ X_2 = 500 \rightarrow X_2 = 1.150 \quad Ax = \begin{bmatrix} 2.300 + 1.200 \\ 700 + 800 \\ 350 + 400 \end{bmatrix} = \begin{bmatrix} 3.500 \\ 1.500 \\ 750 \end{bmatrix}$

$j3:\ X_3 = 350 \rightarrow X_3 = 500 \quad Ax = \begin{bmatrix} 2.200 \\ 1.800 \\ 900 \end{bmatrix}$

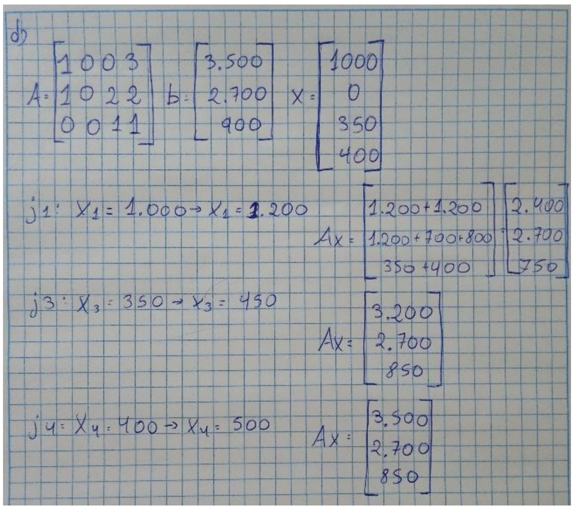
$j4:\ X_4 = 400 \rightarrow X_4 = 550 \quad Ax = \begin{bmatrix} 1.000 + 1.650 \\ 1.800 \\ 350 + 550 \end{bmatrix} = \begin{bmatrix} 2.650 \\ 1.800 \\ 900 \end{bmatrix}$

c) Los valores pueden ser:

$x_2 = \quad 500 \rightarrow \quad 1.150$

$x_3 = \quad 350 \rightarrow \quad\ \ 500$

$x_4 = \quad 400 \rightarrow \quad\ \ 550$



d)

$$A = \begin{bmatrix} 1 & 0 & 0 & 3 \\ 1 & 0 & 2 & 2 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 3.500 \\ 2.700 \\ 900 \end{bmatrix} \quad x = \begin{bmatrix} 1000 \\ 0 \\ 350 \\ 400 \end{bmatrix}$$

$j_1: x_1 = 1.000 \rightarrow x_1 = 1.200$ $\quad Ax = \begin{bmatrix} 1.200+1.200 \\ 1.200+700+800 \\ 350+400 \end{bmatrix} \begin{bmatrix} 2.400 \\ 2.700 \\ 750 \end{bmatrix}$

$j_3: x_3 = 350 \rightarrow x_3 = 450$ $\quad Ax = \begin{bmatrix} 3.200 \\ 2.700 \\ 850 \end{bmatrix}$

$j_4: x_4 = 400 \rightarrow x_4 = 500$ $\quad Ax = \begin{bmatrix} 3.500 \\ 2.700 \\ 850 \end{bmatrix}$

d) Los valores pueden ser:

$x_1 = 1.000 \rightarrow 1.200$

$x_3 = 350 \rightarrow 450$

$x_4 = 400 \rightarrow 500$

EJERCICIOS TEÓRICOS

3. Repita el ejercicio 4 con el método Gauss-Jorda

Respuestas:

A: [-227.07669067, 476.92263794, -177.69216919]

B: [0.9999997, 1.00000012, 0.99999982]

C: [-0.03174108, 0.5951857 , -2.38083124, 2.77770114]

D: [1.88304102, 2.80701733, 0.73099416, 1.43859673, 0.09356727]

Eliminación Gaussiana con sustitución hacia atrás

```
%load_ext autoreload
```

```
%autoreload 2
from ejercicio7 import gauss_jordan_32bits
import numpy as np
```

```
A = np.array(
    [[1/4, 1/5, 1/6, 9],
     [1/3, 1/4, 1/5, 8],
     [1/2,   1,   2, 8]])

B = np.array(
    [[  3.333,  15920, -10.333,  15913],
     [  2.222,  16.71,    9.612, 28.544],
     [ 1.5611, 5.1791,   1.6852, 8.4254]])

C = np.array(
    [[  1, 1/2, 1/3, 1/4, 1/6],
     [1/2, 1/3, 1/4, 1/5, 1/7],
     [1/3, 1/4, 1/5, 1/6, 1/8],
     [1/4, 1/5, 1/6, 1/7, 1/9]])

D = np.array(
    [[2,  1, -1,  1, -3,  7],
     [1,  0,  2, -1,  1,  2],
     [0, -2, -1,  1, -1, -5],
     [3,  1, -4,  0,  5,  6],
     [1, -1, -1, -1,  1, -3]])
```

```
R_A = gauss_jordan_32bits(A)[0]
R_B = gauss_jordan_32bits(B)[0]
R_C = gauss_jordan_32bits(C)[0]
R_D = gauss_jordan_32bits(D)[0]
```

```
print(f'Respuesta de A: {R_A}')
print(f'Respuesta de B: {R_B}')
print(f'Respuesta de C: {R_C}')
print(f'Respuesta de D: {R_D}')
```

```
Respuesta de A: [-227.07669067  476.92263794 -177.69216919]
Respuesta de B: [0.9999997  1.00000012 0.99999982]
Respuesta de C: [-0.03174108  0.5951857  -2.38083124  2.77770114]
Respuesta de D: [1.88304102 2.80701733 0.73099416 1.43859673 0.09356727]
```