En general, el primer sistema de ecuaciones lineales que uno estudia es un sistema de 2×2, es decir 2 ecuaciones con 2 incógnitas, por ejemplo:

$$\begin{aligned} y &= a_1 x + b_1 \\ y &= a_2 x + b_2 \end{aligned}$$

donde y, x, a_1, a_2, b_1, y b_2 pertenecen a R, es decir a los números reales. En particular sabemos que en este caso si,

$$\det \begin{pmatrix} 1 & -a_1 \\ 1 & -a_2 \end{pmatrix} = a_1 - a_2 \neq 0$$

existe una única solución para x y y que satisface ambas ecuaciones. Una forma de resolver el sistema de ecuaciones lineales anterior es simplemente igualando la variables y, es decir:

$$a_1 x + b_1 = a_2 x + b_2$$

de la cual podemos despejar x de la siguiente forma:

$$x = \frac{b_2 - b_1}{a_2 - a_1}$$

Ahora, ¿qué ocurriría si reemplazamos los coeficientes e incógnitas por matrices de dimensión 2×2? Esto significa que las ecuaciones se transforman a la siguiente forma:

$$\begin{aligned} Y &= A_1 X + B_1 \\ Y &= X A_2 + B_2 \end{aligned}$$

donde Y, X, A_1, A_2, B_1 y B_2 pertenecen a $R^{2\times 2}$. Note que el orden de la multiplicación de X por las matrices A_1 y A_2 no es un error, así se propone la generalización. Considere la siguiente definición para X, Y, A_k y B_k para $k \in \{1, 2\}$:

$$X = \begin{pmatrix} x_1 & x_3 \\ x_2 & x_4 \end{pmatrix}, Y = \begin{pmatrix} y_1 & y_3 \\ y_2 & y_4 \end{pmatrix}, A_k = \begin{pmatrix} a_{k,1} & a_{i,3} \\ a_{k,2} & a_{k,4} \end{pmatrix}, B_k = \begin{pmatrix} b_{k,1} & b_{k,3} \\ b_{k,2} & b_{k,4} \end{pmatrix}$$

Al aplicar el mismo procedimiento anterior para el caso con coeficientes reales, uno no encuentra directamente un sistema de ecuaciones lineales en su forma tradicional. Sin embargo sí se puede re-escribir como un sistema de ecuaciones lineales de dimensión 4×4, solo considerando como incógnita la matriz X. Para el manejo del problema considere el siguiente orden de los coeficientes. Poner especial atención a los coeficientes del lado derecho ya que definen el orden de las filas de la matriz C y el orden de los x_i ya que define el orden de las columnas de la matriz C,

$$\underbrace{\begin{pmatrix} c_{1,1} & c_{1,2} & c_{1,3} & c_{1,4} \\ c_{2,1} & c_{2,2} & c_{2,3} & c_{2,4} \\ c_{3,1} & c_{3,2} & c_{3,3} & c_{3,4} \\ c_{4,1} & c_{4,2} & c_{4,3} & c_{4,4} \end{pmatrix}}_{C} \begin{pmatrix} x_1 \\ x_3 \\ x_2 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_{2,1} - b_{1,1} \\ b_{2,2} - b_{1,2} \\ b_{2,3} - b_{1,3} \\ b_{2,4} - b_{1,4} \end{pmatrix}$$

Notar que los coeficientes $c_{i,j}$, para $i, j \in 1, 2, 3, 4$, son desconocidos y los debe obtener. Para el desarrollo de las siguientes preguntas considere las siguientes matrices:

$$A_1 = \begin{pmatrix} 0.5881308011 & 0.8977137279 \\ 0.8915307295 & 0.8158374773 \end{pmatrix}, A_2 = \begin{pmatrix} 0.0358895856 & 0.6917575818 \\ 0.378680942 & 0.5185109454 \end{pmatrix}, B_1 = \begin{pmatrix} 0.6579514656 & 0.1938502179 \\ 0.2723164021 & 0.7186059336 \end{pmatrix}, B_2 = \begin{pmatrix} 0.7830036095 & 0.8503276398 \\ 0.775244894 & 0.0366643064 \end{pmatrix}$$

Versión Texto Plano:

```
A1 = np.array([[0.5881308011,0.8977137279],[0.8915307295,0.8158374773]])
A2 = np.array([[0.0358895856,0.6917575818],[0.378680942,0.5185109454]])
B1 = np.array([[0.6579514656,0.1938502179],[0.2723164021,0.7186059336]])
B2 = np.array([[0.7830036095,0.8503276398],[0.775244894,0.0366643064]])
```

1.- [5 puntos] (Verdadero o Falso) En el caso original, con coeficientes reales, se indicó que si $a_1 = a_2$ no tiene solución única el problema. ¿Es posible que exista solución única en el problema modificado si $A_1 = A_2$? (**Ingrese un 0 si es Falso o un 1 si es Verdadero**).

0

2.- [5 puntos] (Verdadero o Falso) Si la matriz A_1 fuera la matriz nula, sigue siendo no singular la matriz C. (**Ingrese un 0 si es Falso o un 1 si es Verdadero**).

1

3.- [10 puntos] (Verdadero o Falso) Es posible obtener la factorización LU de C. (**Ingrese un 0 si es Falso o un 1 si es Verdadero**).

1

✔

4.- [15 puntos] Indique el valor obtenido de $c_{3,3}$ y $c_{4,4}$.

$c_{3,3}$ = 0 , $c_{4,4}$ = 0.29

☑

5.- Proponga e implemente en su jupyter notebook una función que obtenga X basado en la factorización PALU. La función debe tener la siguiente firma:

```
solverInterseccionMatricial(A_1, A_2, B_1, B_2):
    # your code
    return X
```

NOTA: En esta pregunta no hay campo para entregar respuesta. Solo se debe incluir la definición de la función en el jupyter notebook con la firma indicada.

En donde A_1, A_2, B_1 y B_2 son ndarrays de 2×2. El retorno de la función debe ser un ndarray de 2×2 que resuelva el sistema de ecuaciones.

6.- [15 puntos] Dado los valores numéricos indicados anteriormente para las matrices A_1, A_2, B_1, y B_2, obtenga la solución numérica para X usando su implementación con PALU.

$x_1 =$ -4.0 , $x_2 =$ 4.014 , $x_3 =$ 3.23 , $x_4 =$ -2.6

✔

7.- [10 puntos] Obtenga los coeficientes en la diagonal de la matriz U de la factorización PALU, es decir $u_{1,1}$, $u_{2,2}$, $u_{3,3}$ y $u_{4,4}$.

$u_{1,1} =$ 0.89 , $u_{2,2} =$ 0.89 , $u_{3,3} =$ 0.66 , $u_{4,4} =$ 0.25

✖

Uno de los primeros algoritmos iterativos que se construyeron en el curso fue el algoritmo para sacar la raíz cuadrada de un número real, sin embargo es mucho más interesante trabajar con números complejos. Sabemos que todo número complejo es posible descomponerlo en su parte real e imaginaria, es decir $z = x + iy$, donde $i^2 = -1$. Lo interesante es que en la práctica uno puede manipular número complejos solo con álgebra real trabajando con cada componente de forma independiente.

1.-Proponga un algoritmo que obtenga la raíz **cúbica** de un número complejo $w_0 = a_0 + \mathrm{i}\, b_0$ solo considerando álgebra real, particularmente **double precision**, y que solo puede aplicar las 4 operaciones elementales, es decir: $+, -, *,$ y $\div$. Su algoritmo debe recibir como entrada la parte real y imaginaria, es decir a_0 y b_0, y retornar la parte real y imaginaria de la raíz cúbica, es decir debe retornar x e y de $z = x + \mathrm{i}\, y$, tal que se cumpla que $z^3 = w_0$.

Considerar que el error absoluto debe ser menor o igual a $1e-5$. Su función debe tener la siguiente firma:

```
solverCubicComplexRoot(a0,b0):
    # your code
    return x, y
```

NOTA: En esta pregunta no hay campo para entregar respuesta. Solo se debe incluir la definición de la función en el jupyter notebook con la firma indicada.

2.- [20 puntos] Implemente su algoritmo (documentando bien su código) y obtenga la raíz cúbica de $w_0 = 9.2 - 2.1\,\mathrm{i}$, considerando el siguiente "dato inicial" para su algoritmo iterativo:

$$z_0 = -1 - 1.5\,\mathrm{i}$$

Parte Real = -1.19(

Parte Imaginaria = -1.74

✔

3.- [30 puntos] Considerando el "dato inicial" de la pregunta anterior, entregue el determinante de la matriz Jacobiana para primera y segunda iteración.

Determinante en primera iteración = 199.3

Determinante en segunda Iteración = 179.9

✔