Cambio de Sistema de Referencia en Python

Ricardo Juan Cárdenes Pérez

Febrero de 2021

1 Introducción

[Álgebra Lineal]

Supongamos que nos encontramos con dos sistemas de referencia, ambos situados en el plano, que vienen denotados por $\alpha : (O; \vec{e_1}, \vec{e_2})$ y $\beta : (O'; \vec{e_1'}, \vec{e_2'})$, donde sabemos que el punto P(x,y) viene dado en coordenadas del primer sistema. Esto es:

$$x \cdot \vec{e_1} + y \cdot \vec{e_2} \in L(\vec{e_1}, \vec{e_2}) \tag{1}$$

Ahora, supongamos que por algún casual nos interesa obtener las coordenadas del punto P, esta vez en referencia de β. Es fácil ver que, mientras O $\neq O'$, el punto P' (P en refenrecia de β) cambiará completa o parcialmente, sin importar la dependencia o independencia lineal entre sus bases. En este caso, para no complicar más el problema, supondremos que el primer sistema viene dado por las bases canónicas en R^2. Así el punto P cumplirá, para los vectores básicos de α, que:

$$(x, y) = x \cdot \vec{e_1} + y \cdot \vec{e_2} \tag{2}$$

y por tanto su representación sobre los ejes cartesianos sería directa, lo cual es de vital importancia para el problema. Además, como los puntos O y O' están también expresados en base canónica, podríamos representarlos muy fácilmente en el plano, resultando un sistema de coordenadas con tres puntos arbitrarios situados sobre éste.

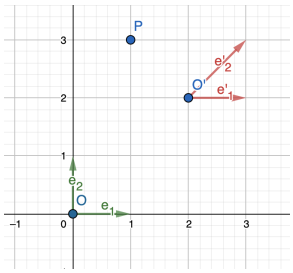

Figure 1: Sistemas de referencia situados sobre el plano

Una vez llegados a este punto, el procedimiento a seguir es muy sencillo. Gracias a la regla de Chasles (deducción a partir de la suma de dos vectores), podemos establecer la siguiente relación vectorial respecto a los puntos P, O, O':

$$\vec{OO'} + \vec{O'P} = \vec{OP} \Rightarrow \tag{3}$$

$$\Rightarrow P' = \vec{O'P} = \vec{OP} - \vec{OO'} \tag{4}$$

Graficamente:

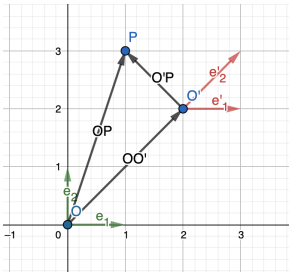

Figure 2: Regla de Chasles

Como $\vec{O'P}$ es el vector que va desde el origen de coordenadas O' (origen de β) hasta el punto P, las coordenadas del punto P' serían las coordenadas de dicho vector, pues es el vetor asociado a este punto. Si la base del nuevo sistema de referencia fuera la canónica, entonces el problema ya estaría resuelto. En caso de que no lo fuera, basta con aplicar al vector resultante asociado a P' un cambio de base, de la canónica a B'=$(\vec{e'_1}, \vec{e'_2})$, mediante una matriz de cambio de base. Para calcularla, sabemos que:

$$\vec{e'_1} = e'_{1x} \cdot (1,0) + e'_{1y} \cdot (0,1) \tag{5}$$

$$\vec{e'_2} = e'_{2x} \cdot (1,0) + e'_{2y} \cdot (0,1) \tag{6}$$

Así, obtenemos que la matriz de cambio de base de B' a la base canónica en el plano es:

$$Q = \begin{pmatrix} e'_{1x} & e'_{2x} \\ e'_{1y} & e'_{2y} \end{pmatrix} \tag{7}$$

Y como para este caso no necesitamos pasar de la base B' a la canónica, sino lo contrario, la matriz cambio de base C de Bc a B' será : $C = Q^{-1}$. Luego:

$$C = Q^{-1} = \begin{pmatrix} e'_{1x} & e'_{2x} \\ e'_{1y} & e'_{2y} \end{pmatrix}^{-1} \Rightarrow \tag{8}$$

$$\Rightarrow C_{Bc \to B'} = \begin{pmatrix} e'_{1x} & e'_{2x} \\ e'_{1y} & e'_{2y} \end{pmatrix}^{-1} \tag{9}$$

Finalmente, si partimos de la ecuación (4), esta vez conociendo (9) y suponiendo P' = $\begin{pmatrix} x' \\ y' \end{pmatrix}$, deducimos que:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = Q^{-1} \cdot (\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} O'_x \\ O'_y \end{pmatrix}) \Rightarrow \tag{10}$$

$$\Rightarrow \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} e'_{1x} & e'_{2x} \\ e'_{1y} & e'_{2y} \end{pmatrix}^{-1} \cdot (\begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} O'_x \\ O'_y \end{pmatrix}) \Rightarrow \tag{11}$$

$$\Rightarrow \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} e'_{1x} & e'_{2x} \\ e'_{1y} & e'_{2y} \end{pmatrix}^{-1} \cdot \begin{pmatrix} x - O'_x \\ y - O'_y \end{pmatrix} \tag{12}$$

En caso de que B'=$(\vec{e'_1}, \vec{e'_2})$ = B_c, sabemos que Q = $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, por lo que C = Q^{-1} = $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ y, finalmente:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x - O'_x \\ y - O'_y \end{pmatrix} \Rightarrow \tag{13}$$

$$\Rightarrow \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x - O'_x \\ y - O'_y \end{pmatrix} \Rightarrow \tag{14}$$

$$\begin{cases} x' = x - O'_x \\ y' = y - O'_y \end{cases} \tag{15}$$

2 Modelo matemático en Python

Generar un algoritmo que realice el cambio entre dos sistemas de referencia en Python, implementando el desarrollo realizado en las páginas anteriores, es relativamente sencillo, pues para el calculo matricial podemos servirnos de la librería numpy (de código abierto). Para ello, basta con importar este módulo al principio del programa, el cual suele ser abreviado por las letras np.

```
import numpy as np
```

Una vez hecho esto, los siguientes comandos pueden ser de gran utilidad:

array([x,y,...]) - Genera una matriz con las listas x, y, etc, como filas.
linalg.inv(X) - Genera la matriz inversa de X.
matmul(X,Y) - Genera la matriz resultante de la operación X·Y

Para llamar a estos métodos, basta con coger el nombre con el que hemos importado la librería, normalmente np, y añadirle el método al final con un punto entre medias que los separa.

ej: x = np.array([[1,2,3],[4,5,6],[7,8,9]])

Resultado: x = $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$

Sabiendo esto, el problema se resulve prácticamente solo. El algoritmo consistirá en una función que reciba 6 parámetros: origen de α (O), origen de β (O'), $\vec{e_1'}$, $\vec{e_2'}$, P_x, P_y; donde B' = $(\vec{e_1'}, \vec{e_2'})$ es la base del sistema de referencia final y P(P_x, P_y) es un punto en coordenadas del primer sistema. Una vez llamada la función con sus parámetros correspondientes, basta con que ésta aplique los datos a la siguiente ecuación y devuelva el resultado.

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} e'_{1x} & e'_{2x} \\ e'_{1y} & e'_{2y} \end{pmatrix}^{-1} \cdot \begin{pmatrix} x - O'_x \\ y - O'_y \end{pmatrix} \tag{16}$$

(*) Ejemplo de código en la siguiente página

```python
# Ricardo Juan Cárdenes Pérez
# 7/2/2021

import numpy as np
import matplotlib.pyplot as plt

def ref_sis_2d(o,op,e1,e2,p):

    # Operaciones matriciales
    q = np.array([e1,e2]).T

    try:
        c = np.linalg.inv(q)

    except Exception as err:
        print(err.args[0])
        return 'El sistema de referencia 2 no es bidimensional'

    pp = np.matmul(c, p - op)

    # Devolvemos el resultado
    return pp


if __name__ == '__main__':

    # Declaramos sistemas de referencia
    comp = {'O':'', 'O\'': '', 'e1': '', 'e2': '', 'P': ''}
    for i in comp.keys():
        n1 = int(input(f'Introduce la coordenada en X de {i}: '))
        n2 = int(input(f'Introduce la coordenada en Y de {i}: '))
        comp[i] = np.array([[n1],[n2]])

    pp = ref_sis_2d(comp['O'],comp['O\''],comp['e1'],comp['e2'],comp['P'])
    print(pp)

```

Figure 3: Ejemplo de código

Como podemos observar, si introducimos los vectores $\vec{e_1'}$ *y* $\vec{e_2'}$ **iguales o proporcionales** se produce un error ('Singular matrix'). Esto es porque

$$\vec{e_1'} = \vec{e_2'} \Rightarrow r(\vec{e_1'}, \vec{e_2'}) = 1, \text{ y como } \mathrm{r}(\vec{e_1'}, \vec{e_2'}) \neq 2 \Rightarrow |\vec{e_1'}, \vec{e_2'}| = 0, \quad \nexists \begin{pmatrix} e'_{1x} & e'_{2x} \\ e'_{1y} & e'_{2y} \end{pmatrix}^{-1}.$$