

Ejercicio 1

El número π puede estimarse a partir de la probabilidad que un punto aleatorio en el cuadrado $[-1,1] \times [-1,1]$ caiga dentro de un círculo de radio 1. Si la distribución de puntos es uniforme, esa probabilidad será $\pi r^2 / (4r^2) = \pi/4$.

Estime el número π empleando 10, 100, 10.000 y 100.000 puntos.

Ejercicio 2

Resolver con Numpy los siguientes sistemas de ecuaciones lineales:

$$\begin{aligned} 1x_1 + 2x_2 + 5x_3 + 6x_4 &= 1 \\ 4x_1 - 4x_2 - 6x_3 + 8x_4 &= 6 \\ -12x_1 + x_2 + 3x_3 + 9x_4 &= 7 \\ 18x_1 + 6x_4 &= 2 \end{aligned}$$

$$\begin{aligned} x_1 + 2x_2 + x_3 &= 4 \\ x_2 + x_3 &= 3 \\ x_1 + x_3 &= 5 \end{aligned}$$

$$\begin{aligned} 36x_1 + 51x_2 + 13x_3 &= 3 \\ 52x_1 + 34x_2 + 74x_3 &= 45 \\ 7x_2 + 1.1x_3 &= 33 \end{aligned}$$

$$\begin{aligned} 3x_1 + 9x_2 - 10x_3 &= 24 \\ x_1 - 6x_2 + 4x_3 &= -4 \\ 10x_1 - 2x_2 + 8x_3 &= 20 \end{aligned}$$

Ejercicio 3

Genere una función llamada `Matriz`, que dado un entero n , genere una matriz “ $n \times n$ ” de números aleatorios enteros entre -100 y 100.

Genere una función llamada `Vectorb`, que dada una matriz, devuelva el vector de términos independientes que hace que la solución del problema sea el vector unidad.

Resuelva el sistema para distintos n 's, entre 2 y 1.000 (dependiendo de la capacidad de la máquina) por medio del cálculo de la inversa de la matriz y empleando el método `np.linalg.solve`.

Almacene los tiempos correspondientes a cada forma de resolución en un vector.