

# INFORME LABORATORIO 4

Este laboratorio está bien ubicado en el tiempo en el máster, ya que lo hacemos con conocimiento y experiencia en Python básico, de manera que podemos ampliar el conocimiento que tenemos del lenguaje con paquetes y funcionalidades que mejoran el rendimiento en el cálculo científico.

He detectado un detalle en el laboratorio que quizá se pueda mejorar: si se siguen las instrucciones al pie de la letra, se acaba con un fichero denominado `reducooperation.ipynb` sin modificar. Imagino que la intención era que se llamase `reducooperation-alumnox.ipynb` desde el principio, e ir modificándolo. Yo he seguido los pasos literalmente para esta parte.

He añadido también los archivos de output de slurm, aunque las salidas también están registradas en las celdas de las libretas de Jupyter. Respecto al último ejercicio, del método de Montecarlo, he tratado los puntos de forma vectorizada con arrays de NumPy. Los he generado con la función de distribución uniforme que incluye y he aprovechado las propiedades de conversión de tipo booleano a entero para sumar el número de puntos que caen dentro del círculo. Aquí he optado por medir también el tiempo de compilación de njit porque considero que así la comparación es más justa. En este cálculo, más que ejecutar la función varias veces y hacer la media, pienso que es más interesante ejecutarla sólo una vez pero con mayor número  $N$  de puntos. Por eso he incluido el tiempo de compilación. Con la optimización se ha reducido el tiempo en más de 10 veces, aunque no he podido mejorarlo más en el tiempo en que lo he intentado.