Computational Astrophysics - Taller 9

Willinton Caicedo Tez
Angel Daniel Martínez Cifuentes
Computational Astrophysics
Observatorio Astronómico Nacional - Universidad Nacional de Colombia

Resolviendo Grandes Sistemas de Ecuaciones Lineales

El sistema de ecuaciones lineales a resolver es de la forma

$$\mathrm{A}\mathbf{x} = \mathbf{b} \tag{1}$$

donde A es una matrix n×n y **b** un vector de dimensión n.

1. **Determinante**

   Para saber si el sistema tiene solución se halla el determinante, el cual tiene que ser distinto de cero. Para ello, se usa un paquete de Numpy para calcular el determinante. Sin embargo, los archivos LSE4_A.dat y LSE5_A.dat, son matrices 1000×1000 y 2000×2000 respectivamente, por lo que se puede incurrir en un problema de coma flotante (*floating point overflow* o *floating point underflow*). Se usa por lo tanto el paquete de Numpy para calcular el log-determinante (logaritmo natural del determinante). El paquete usado es numpy.linalg.slogdet, y retorna el signo y logaritmo natural (logdet) del valor absoluto del determinante [Numpy]. Si el determinante es 0, entonces el signo será 0 y logdet será $-\infty$. Los cálculos para las 5 matrices se muestran en la Tabla 1.

	Determinante	sign/LogDet
LSE1_A	$1{,}049 \times 10^{-05}$	-1 / -11.46
LSE2_A	$-2432{,}72$	-1 / $+7.796$
LSE3_A	$9{,}441 \times 10^{37}$	-1 / $+87.44$
LSE4_A	–	-1 / 1237.2
LSE5_A	–	$+1$ / 3177.4

Tabla 1: Determinante y numpy.linalg.slogdet. Ningún valor es nulo, así el sistema tiene solución única.

2. **Eliminación Gaussiana y *back-substitution***

   Primero se hace un test con un sistema de ecuaciones conocido:

$$\begin{pmatrix} 5 & 3 & 4 \\ 2 & 1 & 5 \\ 5 & 4 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \\ 2 \end{pmatrix} \tag{2}$$

   El determinante es det(A)= -14 y las soluciones son

$$x_1 = -1{,}2142$$
$$x_2 = +1{,}7857$$
$$x_3 = +0{,}9285$$

   con un tiempo de cómputo promedio de tiempo para 10 veces de $3{,}957 \times 10^{-5}$ s.

   Usando la función numpy.linalg.solve para calcular sistemas de ecuaciones de Numpy, se obtiene un promedio de tiempo de $5{,}173 \times 10^{-5}$ s. Esta función de Python requiere un vector cuadrado y un vector columna en forma de array, y retorna el vector **x** correspondiente a las soluciones del sistema de ecuaciones.

3. **Comparación Eliminación Gaussiana con Numpy.linalg.solve**

   Al correr el programa 10 veces para cada matriz, con cada método, se obtiene la tabla de datos de la Tabla 2.

	Tiempo Gauss [s]	Tiempo Numpy [s]
LSE1_A	0.00023	0.01865
LSE2_A	0.02369	0.00021
LSE3_A	0.13891	0.00046
LSE4_A	3.13449	0.01610
LSE5_A	17.1380	0.23646

Tabla 2: Tiempos para los dos métodos distintos.

Se observa que en ambos métodos el tiempo de cómputo aumenta con la dimensión de la matriz, sin embargo se llega de forma más rápida al resultado (el cual es el mismo), a través de la función de NumPy.

Referencias

Numpy. numpy.linalg.slogdet. https://docs.scipy.org/doc/numpy/reference/generated/numpy.linalg.slogdet.html. Accessed: 2019-05-19.