

## Tarea 2

### Parte a)

Implemente en Matlab/Octave/SciLab tres variantes de la versión estándar de la multiplicación de matrices.

- i) utilizando la multiplicación de matrices de la herramienta
- ii) multiplicando vectores  $C(i,j) = A(i,:) * B(:,j)$
- iii) multiplicando coeficientes

Evalúe los tiempos de ejecución para las distintas variantes con matrices de 2048x2048, 4096x4096, 8192x8192.

Las tres variantes realizan (aproximadamente) la misma cantidad de operaciones. ¿Por qué cree que son distintos los tiempos entre una variante y otra?

iv) Describa (si no lo hizo antes) las características de la plataforma de ejecución e intente relacionar dichas características con los resultados obtenidos.

v) Considere un equipo (llamado HWALN) con las siguientes características: Intel Xeon Gold 6138 CPU de 20 cores a 2.00GHz, 128 GB de RAM.

¿Es dicha información suficiente para poder estimar el comportamiento de las variantes anteriores?

En caso que no, ¿qué datos faltarían? Busque los valores y, considerando sus resultados explique cómo se deberían comportar sus implementaciones en HWALN.

### Parte b)

i) Evalúe las estrategias de ordenamiento disponibles en Matlab/Octave (En Matlab: help spfun, en Octave ver: <https://octave.org/doc/v4.0.1/Sparse-Functions.html#Sparse-Functions>).

- A partir de la documentación, describir como trabaja Matlab/Octave las matrices dispersas
- Bajar la matriz dispersa bcsstk15 del sitio Suite Sparse y levantarla en Matlab/Octave <https://sparse.tamu.edu/HB/bcsstk15>
- Aplicarle las diferentes estrategias de ordenamiento disponibles y factorizar con chol. Analizar la cantidad de memoria utilizada y patrón de la matriz L utilizando las funciones whos y spy.

Observe la cantidad de elementos distintos de 0 que se obtienen al factorizar una matriz simétrica y definida positiva.

ii) Evalúe la resolución de sistemas lineales con el método del gradiente conjugado (pcg en Matlab/Octave) generando sistemas lineales utilizando las matrices de la parte i.