MÉTODOS NUMÉRICOS PARA LA COMPUTACIÓN

**Tema 6: Matrices dispersas**

**2020/21**

X de diciembre de 2020

**Grupo 03:** José María Amusquívar Poppe y Prashant Jeswani Tejwani

Universidad de Las Palmas de Gran Canaria

Escuela de Ingeniería en Informática

Índice

[Actividad práctica 1 3](#_Toc58866292)

[Ejercicio 1 3](#_Toc58866293)

[Ejercicio 2 4](#_Toc58866294)

[Ejercicio 3 (optativo) 6](#_Toc58866295)

[Ejercicio 4 (optativo) 8](#_Toc58866296)

[Actividad práctica 2 9](#_Toc58866297)

[Ejercicio 1 9](#_Toc58866298)

[Ejercicio 2 (optativo) 9](#_Toc58866299)

[Actividad práctica 3 9](#_Toc58866300)

[Ejercicio 1 (optativo) 9](#_Toc58866301)

[Ejercicio 2 (optativo) 9](#_Toc58866302)

[Actividad práctica 4 9](#_Toc58866303)

[Ejercicio 1 (optativo) 9](#_Toc58866304)

[Referencias 10](#_Toc58866305)

# Actividad práctica 1

Se realiza la codificación COO y CSR en Matlab.

## Ejercicio 1

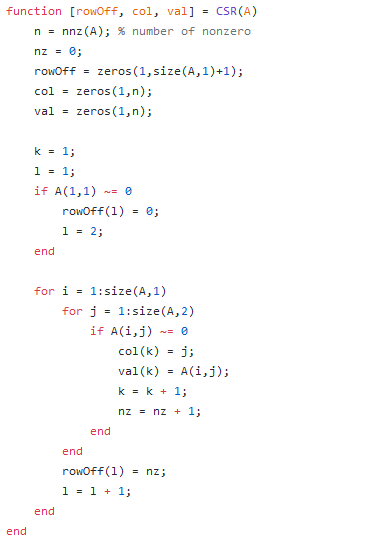
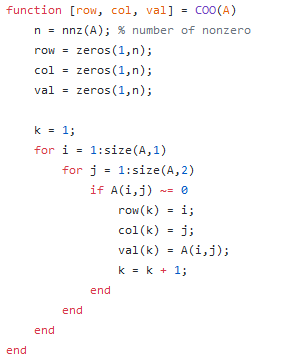
Se implementa dos funciones en Matlab que devuelvan la codificación COO y CSR para una matriz que se le pase como entrada.

Figura 1. Codificación CSR

Figura 2. Codificación COO

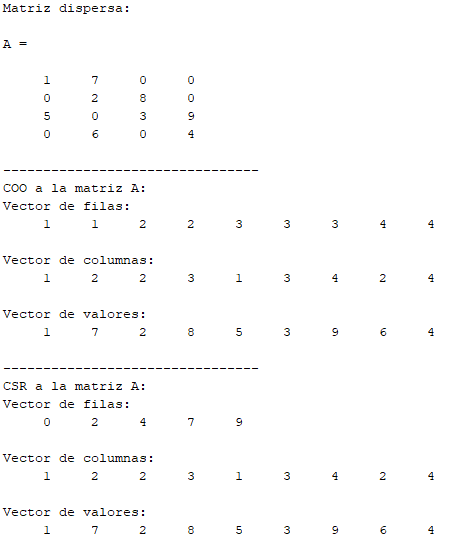
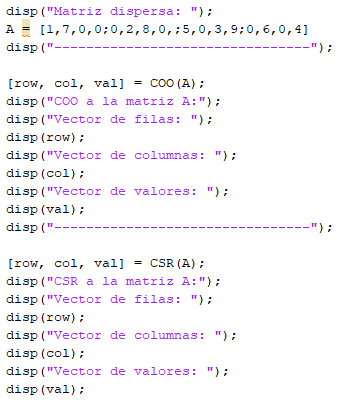


Figura 3. Ejemplo de codificación

Figura 4. Resultado obtenido

## Ejercicio 2

Se programa un generador de matrices aleatorias escasas en Matlab, tomando como entrada la densidad deseada de valores nulos para la matriz y el rango de valores.



Figura 4. Ejemplo de matriz dispersa

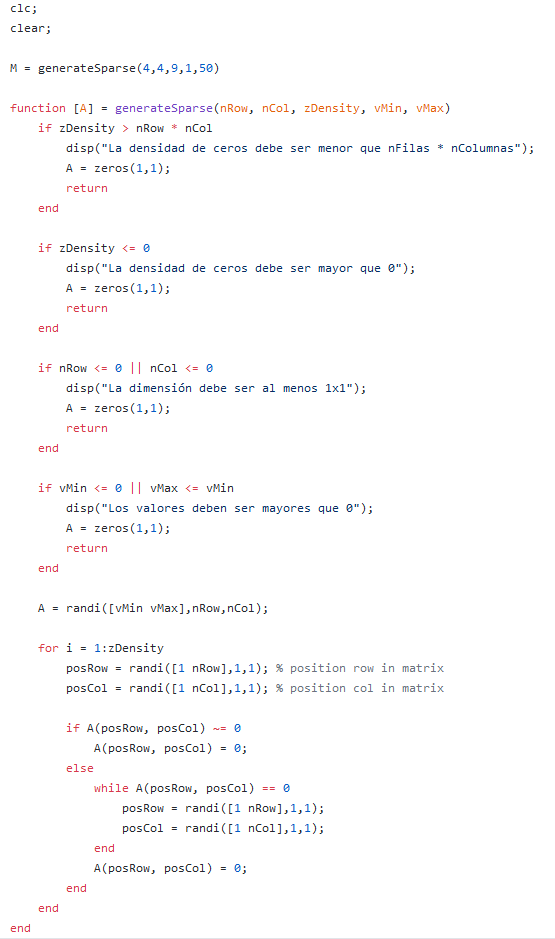
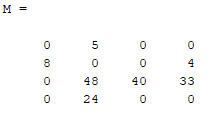


Figura 5. Generador de matrices dispersas

Figura 6. Resultado obtenido

## Ejercicio 3 (optativo)

A continuación, se caracteriza el ahorro promedio que se produce en almacenamiento en función de los diferentes valores de densidad del ejercicio 2.

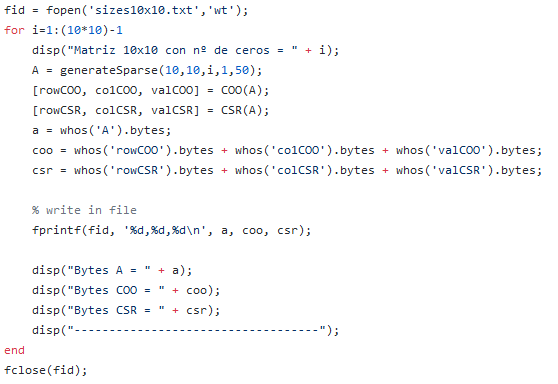


Figura 7. Cálculo de los bytes para distintas densidades según la codificación

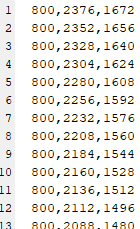


Figura 8. Formato del fichero de salida

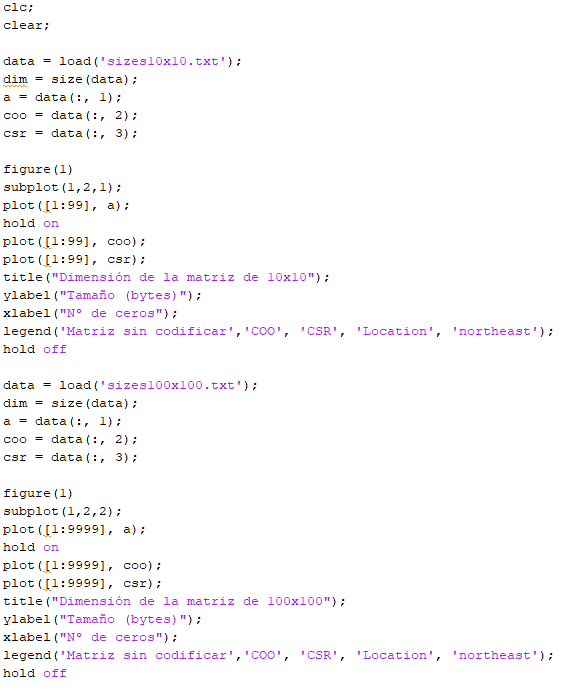
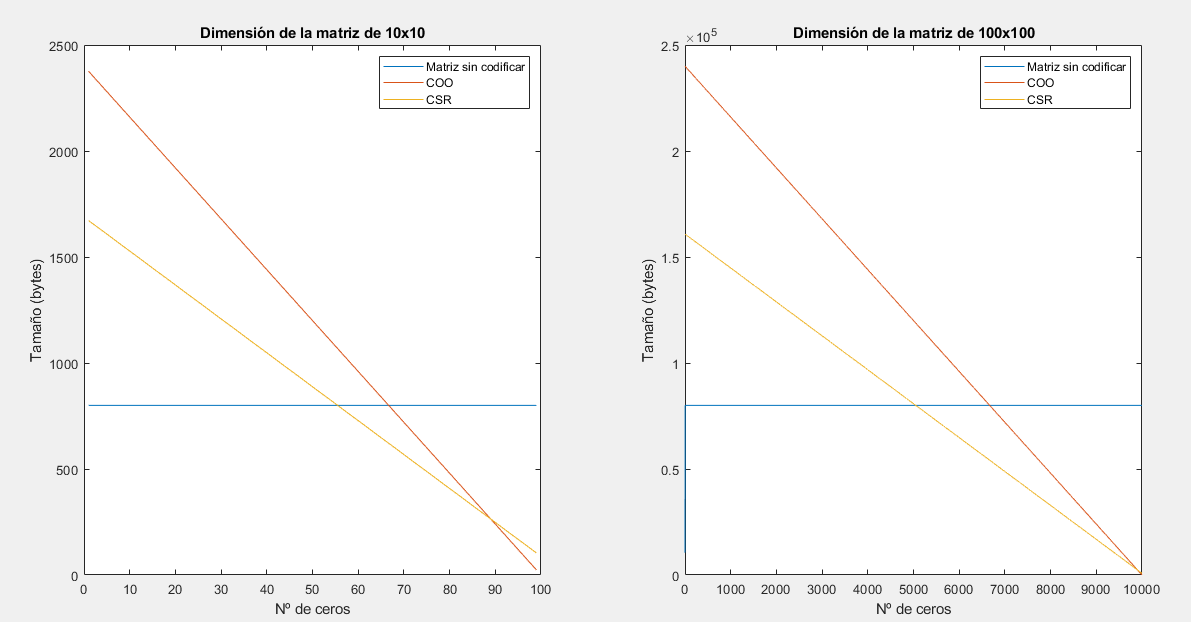
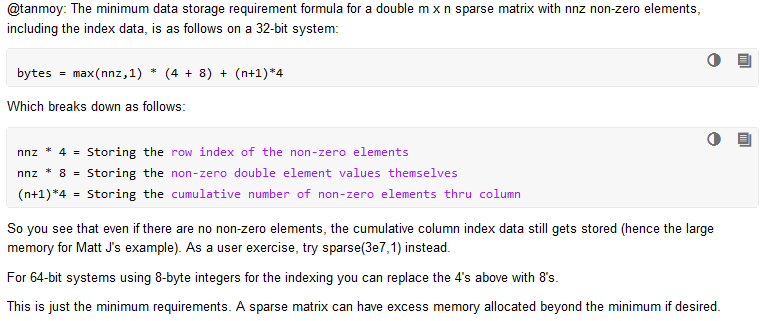


Figura 10. Gráficas obtenidas para matrices de tamaño 10x10 y 100x100

Figura 9. Fichero para obtener las gráficas



## Ejercicio 4 (optativo)

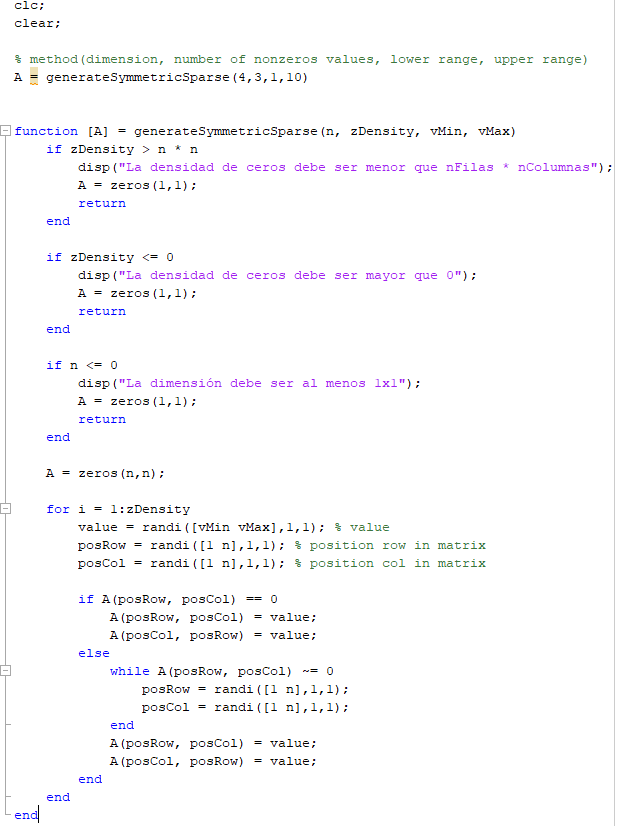
Se implementa un generador de matrices aleatorias escasas simétricas.

Figura 11. Generador de matrices dispersas simétricas

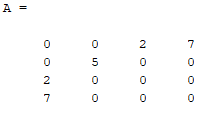


Figura 12. Resultado obtenido

# Actividad práctica 2

Se realiza la codificación dispersa con MKL en C++.

## Ejercicio 1

Se toma un ejemplo de matriz utilizada en la práctica 1 y se reproducen los resultados con las funciones MKL.

## Ejercicio 2 (optativo)

A continuación, se prueban conversiones a otros formatos.

# Actividad práctica 3

## Ejercicio 1 (optativo)

Se realiza una operación de multiplicación de matrices escasas grandes utilizando BLAS, repitiendo y utilizando *SparseBLAS*. Se comparar el tiempo invertido.

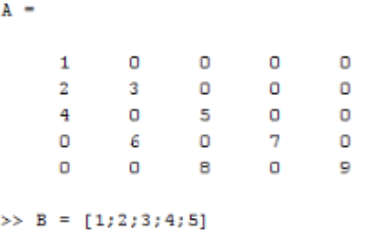
## Ejercicio 2 (optativo)

Se estima la ganancia en GFLOPS resultante.

# Actividad práctica 4

## Ejercicio 1 (optativo)

Se resuelve el siguiente sistema Ax = b aplicando la metodología DSS:



# Referencias

*ULPGC*. (s.f.). Obtenido de https://ncvt-aep.ulpgc.es/cv/ulpgctp21/pluginfile.php/412003/mod\_resource/content/9/6%20Matrices%20dispersas.pdf