**Tarea 2.- Resolución de sistemas de ecuaciones lineales por métodos directos e iterativos.**

**Objetivos:**

**Los objetivos que debemos alcanzar en esta práctica son los siguientes:**

* **Conocer un caso de estudio práctico: Reconstrucción de Imágenes de tomografía Computarizada.**
* **Trabajar con las ecuaciones normales para la resolución del caso de estudio.**
* **Utilizar los métodos: Ecuaciones normales y QR.**
* **Utilizar el entorno MATLAB.**

**Introducción.-**

Pretendemos resolver el sistema de ecuaciones

***Ax=b***, (1)

donde *A* es una matriz rectangular *mxn*, *b* un vector *mx1* y *x* un vector de *nx1*. El sistema de ecuaciones es rectangular.

Vamos a suponer que los datos *A* y *b* son obtenidos en una prueba de una tomografía computarizada en un escáner TC con 1025 sensores, y que puede trabajar con las siguientes resoluciones: 32x32, 128x128 y 256x256.

La matriz *A* viene dada por las características del escáner utilizado y es fija. En el vector *b* se almacena el sinograma (Tomografía computarizada realizada al paciente). La solución *x del sistema (1)* almacena la imagen en un vector unidimensional. Utilizando el entorno MATLAB la imagen del TC se reconstruye con la expresión: *IM= reshape(x, t, t), con t=sqrt(n) donde n es la resolución al cuadrado*.

La imagen reconstruida puedes visualizarla con el comando:

figure, imcontrast(imshow(IM)).

En las siguientes imágenes puedes observar que la imagen reconstruida tiene bastante buena calidad, pero que la resolución utilizada pierde mucha información de la imagen.

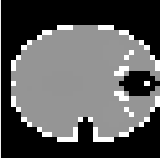
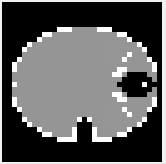
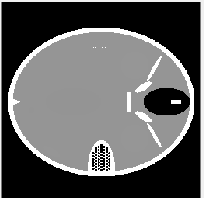
  

Fig. Reconstruida. 32x32 Fig. Referencia 32x32 Fig. Referencia 256x256

La variable *m* denota el número de filas y toma el valor del número de sensores por el número de vistas realizadas.

La variable n, es el número de columnas y coincide con la resolución de la imagen a reconstruir al cuadrado.

En la tarea debemos emplear los datos ubicados en la Carpeta de “Datos TC “del canal PL1\_GA1\_ del equipo de Teams.

Los datos que se proporcionan corresponden al fantoma Forbild ( https://www.researchgate.net/figure/The-FORBILD-phantom\_fig2\_264831864) de diferentes resoluciones. Por ejemplo, el fichero zip IMref\_TC con tienen los ficheros IMref32.mat, …., IMref512x512, que almacenan las imágenes de referencia para las resoluciones de 32x32, 64x64, …., 512x512. En el fichero SistemasResolución32.zip, hay una colección de 8 matrices y sinogramas para la resolución de 32x32 para diferente número de proyecciones. En el fichero Sistema128x30 están los datos para reconstruir una imagen de resolución 128x128 para 30 proyecciones. La carpeta ‘Ejemplo Fantomas de Shepp Logan ‘ contiene los datos de un ejemplo de resolución 60x60 para un TC de 1092 detectores y 30 proyecciones, entre otros ejemplos que se pueden emplear. **Utiliza para la práctica la colección de datos con las dimensiones que la computadora te permita resolver los sistemas de ecuaciones lineales asociados.**

**En los cálculos debes tomar tiempos de ejecución y comparar la calidad** de la imagen obtenida.

Para tomar tiempos de computación en la resolución de los sistemas utiliza los comandos **tic** y **toc.** ( tic, código, toc), si quieres puedes utilizar otros comandos.

Para obtener una **medida de calidad** de la Imagen reconstruida puedes utilizar el **Error relativo**: Er= norm(IMref-IM)/norm(IMref).

Otras métricas que debes utilizar para medir la calidad de las imágenes reconstruidas están disponibles en el entorno MATLAB:

- **SSIM** (Índice de Similitud Estructural)

- **PSNR** ( Relación señal-ruido pico)

Nota: con help ssim y help psnr puedes obtener la información de las funciones.

Primero que nada debes realizar una descripción de la computadora que vas a utilizar para realizar la tarea, también informa de la versión de MATLAB que estás utilizando ( utiliza el comando ‘ver’ de MATLAB).

Nota: **: Las implementaciones en MATLAB deben realizarse utilizando los formatos ‘live script’ y comentar los resultados en el mismo fichero.**

**Ejercicios:**

1.- El método de las ecuaciones normales para la resolución de un sistema de ecuaciones lineales

*Ax=b ( 1)*

consiste en resolver el sistema

*A’\*A x= A’\*b*  ( 2) .

a.- Resuelve el sistema de ecuaciones lineales (1) utilizando el comando de MATLAB /.

b.- Resuelve el sistema de ecuaciones lineales (2) utilizando el comando /.

Nota: Si tienes problemas de almacenamiento con el cálculo de A’\*A, intenta resolver el problema sin realizar el cálculo explícito.

c.- Compara los tiempos de ejecución y la calidad de las imágenes obtenidas.

**Puntos: 3**

2.- a.- Implementa el algoritmo de resolución del problema Lineal de Mínimos Cuadrados vía Householder (página 38 del fichero QR\_2024\_25). Comprueba que funciona adecuadamente, para ello utiliza el ejemplo de los apuntes.

b.- Utiliza el método implementado en el apartado (a) para resolver los casos de test utilizados en el ejercicio 1. Compara los costes computacionales y el error en la calidad de los resultados obtenidos. Comenta los resultados obtenidos.

c.- Utiliza el método de descomposición *QR* que contiene el entorno MATLAB. Utiliza el comando help para conocer todas las posibilidades del comando *qr*.

Calcula los tiempos de cómputo en la resolución del sistema de ecuaciones *Ax=b* utilizando el método qr de MATLAB para las posibilidades:

*- [Q,R]= qr(A); x= R\Q’\*b;*

*- [Q,R]= qr(A,0); x= R\Q’\*b;*

*- [c,R] = qr(A,b); x= R\c;*

*- [c,R,P] = qr(A,b) x= P\*( R\c);*

**Nota:** puede que algunas opciones tarden bastante tiempo computacional y otras pueden que te saturen la memoria disponible, depende de la máquina que estés utilizando.

d.- Compara los tiempos de ejecución, la calidad de las imágenes obtenidas y los recursos de memoria de todos los procesos utilizados (ejercicio 1 y 2).

**Nota: puntos 4.**

3.- Utiliza la descomposición *QR* con pivotamiento para resolver el sistema de ecuaciones lineales *Ax=b*, en MATLAB dispones de varias posibilidades pero vas a utilizar las dos siguientes:

- [*Q,R,P]= qr(A,0); x= P\*( R\Q’\*b).*

*- [c,R,P] = qr(A,b) x= P\*( R\c).*

a.- Calcula los tiempos de cómputo en la resolución del sistema de ecuaciones Ax=b utilizando las dos posibilidades.

b.- Compara y analiza los tiempos de cómputo, los requerimientos de memoria y la calidad en la solución de los métodos del apartado (a) con respecto a los del ejercicio (2).

**Nota: puntos 3.**

**4.- Ejercicio Voluntario.**

En la carpeta ‘ QR-Bloques ’ de la UD2 esta explicado el algoritmo a bloques del método QR. Realiza una implementación de dicho algoritmo, valida su funcionamiento y compara tiempos computacionales con respecto al implementado en 2(a) y el método QR implementado en MATLAB. Las hojas proporcionadas se han obtenido del libro “Matrix Computations”. G.Golub & C.Van Loan.

**Importante: Comenta los resultados obtenidos y comprueba que los algoritmos implementados funcionan bajo un ejemplo de dimensión pequeña.**