2 MÍNIMOS CUADRADOS

2.1 LA FACTORIZACIÓN Q R

El objetivo del primer apartado de la práctica es observar el comportamiento de las tres factorizaciones Q R de una matriz A que describimos en teoría.

Para ello, en primer lugar, factorizamos la matriz de Hilbert en distintas dimensiones y las vamos a comparar presentando el error en la ortogonalidad del factor Q mediante $\|Q^t\,Q - I\|_F$ frente al número de condición de la matriz H_n para los distintos valores de n, cuando Q se calcula con los tres procedimientos considerados. Completaremos la tabla

		$\|Q^t\,Q - I\|_F$		
n	$\mu(H_n)$	Reflectores	Gram-Schmidt	Gram-Schmidt modificado
2	19'2815	$0'0256 \cdot 10^{-14}$	0	0
10	$1'6075\ 10^{13}$	$0'1138\ 10^{-14}$	6'2869	0'0002
15	$2'4960 \cdot 10^{17}$	$0'1912\ 10^{-14}$	9'7815	1'7090

Y realizaremos una gráfica, en escala logarítmica, de los tres errores, en la que en el eje de abcisas se encontrará el número de condición y en el eje de ordenadas el error.

En segundo lugar compararemos el coste operativo, mediante la factorización de matrices aleatorias de tamaño $n \times n$ para $n = 16, 32, \ldots, 512$. Completaremos la tabla

n	Reflectores	Gram-Schmidt	Gram-Schmidt modificado
16	0'0009	0'0001	0'0001
64	0'0142	0'0011	0'0013
512	1'1328	0'1350	0'4260

Y realizaremos, también con los tres métodos, una segunda gráfica, en escala logarítmica, en la que en el eje de abcisas se encontrará la dimensión de la matriz y en el eje de ordenadas el tiempo de CPU requerido.

2.1.1 Desarrollo

Refectores de Householder

Crear una función de MATLAB [u.norma]=house(x) que, dado un vector $\mathbf{x} \in \mathbb{R}^k$, calcula el vector $\mathbf{u} \in \mathbb{R}^k$ tal que $P(\mathbf{u})\,\mathbf{x}$ tiene nulas sus últimas $k-1$ componentes. Con parámetro de entrada

- x, que contiene el vector que queremos sustituir,

y parámetros de salida

- u, que contiene el vector del reflector.
- norma, norma del vector reflector.

Factorización con reflectores de Householder

Crear una función de MATLAB,

function [Q,R] = factorizacion(A);

que devuelva la factorización Q R de una matriz A cuadrada e invertible A de tamaño $n \times n$ mediante el uso de los reflectores de Householder y del modo más eficiente posible, la factorización Q R de A. Con los siguientes parámetros de entrada

- A, que contiene la matriz a factorizar,

y parámetros de salida

- Q, deberá contener la matriz ortogonal de la factorización Q R de A,
- R, deberá contener en su parte triangular superior el factor R correspondiente a la factorización Q R de A y en los elementos que están por debajo de la diagonal principal la información necesaria para recuperar los reflectores de Householder que se han utilizado en dicha factorización. Se sugiere utilizar la función house.m.

Factorización con el método de Gram-Schmidt

Crear una función de MATLAB,

function [Q,R] = GramSchmidt(A);

que devuelva la factorización Q R de una matriz A cuadrada e invertible A de tamaño $n \times n$, mediante el uso del método de Gram-Schmidt del modo más eficiente posible, la factorización Q R de A. Con los siguientes parámetros de entrada

- A, que contiene la matriz a factorizar,

y parámetros de salida

- Q, deberá contener la matriz ortogonal de la factorización Q R de A,
- R, deberá contener en su parte triangular superior el factor R correspondiente a la factorización Q R de A.

Factorización con el método de Gram-Schmidt Modificado

Crear una función de MATLAB,

function [Q,R] = GramSchmidtMod(A);

que devuelva la factorización Q R de una matriz A cuadrada e invertible A de tamaño $n \times n$, mediante el uso del método modificado de Gram-Schmidt del modo más eficiente posible, la factorización Q R de A. Con los siguientes parámetros de entrada

- A, que contiene la matriz a factorizar,

y parámetros de salida

- Q, deberá contener la matriz ortogonal de la factorización Q R de A,
- R, deberá contener en su parte triangular superior el factor R correspondiente a la factorización Q R de A.

Script de ejecución

Crear tantos programas principales (scripts) de MATLAB como consideres oportuno para ejecutar el programa.

2.2 COMPRESIÓN DE IMÁGENES EN RECUERDO DE JHON FORBES NASH JR.

2.2.1 Planteamiento

Uno de los tópicos de algebra computacional más empleados en términos del número de veces que aparece en diferentes aplicaciones es la descomposición en valores singulares (*singular value descomposition*, SVD). En este problema presentamos su aplicación en el procesamiento de imágenes. Tiene otras y es frecuentemente su uso en estadística.

2.2.2 Relación con la comprensión de imágenes

Si $A \in \mathbb{R}^{m\times n}$, con una SVD dada por $A = U\,\Sigma\,V^t$ definida como en las clases de teoría. Para un entero positivo $k \leqslant \min(m,n)$, definimos $A_{(k)} \in \mathbb{R}^{m\times n}$ como la matriz A obtenida de sus k autovalores singulares mayores: $A_{(k)} = U_{(k)}\,\Sigma_{(k)}\,V^t_{(k)}$, donde $U_{(k)}$ son las primeras k columnas de U, $V_{(k)}$ son las primeras k columnas de V, y $\Sigma_{(k)}$ es el bloque de las primeras k filas y columnas de Σ.

Es conocido que $A_{(k)}$ minimiza el problema

$$\min_{B\in\mathbb{R}^{m\times n},\,\text{rango}(B)\leqslant k} (\|A-B\|_F).$$

Es decir, entre las matrices de rango menor que k, es la que mejor se aproxima a A.

Su aplicación a un problema de compresión de imágenes consistirá en obtener una matriz A que describa una imagen y reducir el tamaño que necesitamos para obtenerla con suficiente calidad.

Para ello procesaremos el fichero nash.jpg (o cualquier otro fichero imagen en blanco y negro del que dispongáis) mediante el siguiente procedimiento

- A=imread ('nash.jpg'); ← ¡OJO! He hecho el ejercicio con la imagen saturn.png
- A=im2double(A);
- A=rgb2gray(A);

De este modo obtenemos una matriz 1500×981 donde cada entrada es un pixel de la foto con un número entre 0 y 1, es decir 1.4715 Megapixels.

Utiliza la función svd de MATLAB para calcular la descomposición en valores singulares de A. A partir de ella, construye $A_{(k)}$ y $\|A - A_{(k)}\|_F$ para completar la tabla

n	25	50	$\cdots$	100	$\cdots$
$\|A - A_{(k)}\|_F$	33'9296	20'5382	$\cdots$	9'8823	$\cdots$

La imagen que consideres óptima se mostrará en una figura en comparación con la original, etiquétalas para distinguirlas.

Presenta además una gráfica en las que se muestre la evolución de $\|A - A_{(k)}\|_F$ (en el eje de ordenadas) frente al coste de almacenamiento (en el eje de abcisas).

2.2.3 Ejercicios teóricos

1. Demuestra que los autovalores de $A^t A$ son siempre no negativos, donde $A \in \mathbb{R}^{m \times n}$.
2. Demuestra que si u_i y u_j son autovalores de $A^t A$ asociados a distintos autovalores entonces son ortogonales.
3. Justifica porqué la diferencia de tamaño empleado de A frente a $A_{(k)}$ es $mn - (n + m + 1)k$.

2.3 FUNCIONES AUXILIARES EN MATLAB

Las siguientes funciones son útiles, además de las ya conocidas. Consulta la información en la ayuda del programa.

- rand, nos permite generar matrices aleatorias, con distribución uniforme.
- ncond, calcula el número de condición de una matriz, para cada una de las normas conocidas.
- hilb, obtiene la matriz de Hilbert.
- loglog, equivalente a plot pero con escala logarítmica.
- subplot, divide la pantalla en varias ventanas gráficas.

2.4 ENTREGA

Se entregará

- Un informe explicativo de los resultados obtenidos, que incluya los resultados y las gráficas solicitada, conclusiones y justificación de las soluciones empleadas y la documentación de los programas desarrollados. Además debes incluir la respuesta a los ejercicios teóricos que se plantean en el segundo apartado de la práctica.
- Los ficheros gráficos que se consideren oportunos en formato eps o png.
- Los ficheros de MATLAB con los programas necesarios para la ejecución de la práctica y la obtención de los resultados presentados en el informe, que deben de estar acompañados de los correspondientes comentarios

2.5 EVALUACIÓN

El primer ejercicio tendrá una valoración de 7 puntos, el segundo de 3 puntos.

Para el primer ejercicio se proporcionarán los ficheros codificados de los distintos programas con el objetivo de concluir la realización de la práctica en su fase elemental. El uso de estos ficheros reducirá la calificación a un 60%. El reparto de calificación entre: programación, resultados y conclusiones será de 40%, 30% y 30%, respectivamente.