Práctica 2: Programación en MATLAB/OCTAVE

Ejercicio 1 (ordenador)

Hacer un programa que nos averigüe el máximo de un vector de números, por ejemplo del vector:

$$v = [2, 7.1, -1, 2.3, 44, 11].$$

Nota: Matlab/Octave trae incorporada la función max( ) que nos busca el máximo. Pero se trata de que practicar la programación.

Ejercicio 2 (ordenador)

Se sabe desde hace más de dos siglos que la suma de los inversos de los cuadrados de los números naturales es:

$$1+\frac{1}{2^2}+\frac{1}{3^2}+\frac{1}{4^2}+\frac{1}{5^2}+\frac{1}{6^2}+\frac{1}{7^2}+\frac{1}{8^2}+\cdots\cdots=\frac{\pi^2}{6}$$

Comprueba la veracidad de esa fórmula, para ello deberás sumar, por ejemplo, los 10000 primeros términos y comprobar que el resultado de la suma está muy próximo a:

$$\frac{\pi^2}{6}=1.64493$$

Ejercicio 3 (ordenador)

La sucesión de Fibonacci: 1, 1, 2, 3, 5, 8, 13, ... es muy conocida, no sólo en el mundo de las Matemáticas, por ejemplo se habla de ella en el libro "El código Da Vinci", en series de televisión como "Numbers", etc. En esta sucesión cada elemento es la suma de los dos anteriores.

Figura: Chimenea de una ciudad finlandesa con la sucesión de Fibonacci.

Hacer un programa que nos pida un número n y nos devuelva el término que ocupa la posición n.

Ejercicio 4 (ordenador)

Observa estas matrices:

$$\begin{pmatrix} 1 & 2 \\ -3 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 0 \\ -3 & 1 & 2 \\ 0 & -3 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 0 & 0 \\ -3 & 1 & 2 & 0 \\ 0 & -3 & 1 & 2 \\ 0 & 0 & -3 & 1 \end{pmatrix}, \dots$$

Hacer un programa que construya una matriz A de dimensiones 20×20 de este tipo, es decir, toda compuesta por ceros, salvo en la diagonal donde va siempre un 1, en la diagonal paralela inmediatamente superior donde va siempre un 2 y en la diagonal inmediatamente inferior donde va siempre un -3.

Después averigua el valor de la suma de todos los elementos de la matriz $A^2 - 6A + 5I$

Ayuda: La potencia de una matriz se calcula con la orden $A\text{^}n$

El producto de un escalar (un número) α por una matriz A se calcula con la orden $\alpha * A$.

La suma de matrices de dos matrices A y B en Matlab se obtiene con la orden $A + B$ y la diferencia con la orden $A - B$.

La función eye(20) permite crear una matriz identidad de orden 20.

Ejercicio 5 (ordenador)

En este ejercicio se trabaja con potencias de matrices.

Presentamos dos códigos:

Código 1:

```
tic
A = rand(1000, 1000);
S=zeros(1000,1000);
for i=1:10
    B=S+A^i;
end
toc
```

La orden tic toc de Matlab sirve para que nos salga por pantalla el tiempo empleado en ejecutarlo.

Código 2:

```
tic
A = rand(1000, 1000);
S=zeros(1000,1000);
potencia=eye(1000);
for i=1:10
    potencia=potencia*A;
    B=S+potencia;
end
toc
```

Prueba estos códigos en Matlab. ¿Qué hacen estos programas? ¿Qué código es más eficiente? Razona las respuestas.

Ejercicio 6 (lápiz y papel; último apartado con ordenador)

Sea la matriz $A=\begin{pmatrix} 0 & \frac{1}{3} & 0 \\ \frac{1}{3} & 0 & 0 \\ 0 & 0 & \frac{1}{3} \end{pmatrix}$. Se desea calcular $S = I + A + A^2 + A^3 + A^4 + A^5 + \cdots$

Para poder hacerlo, sigue los siguientes pasos:

a) Comprueba que $A^2 = \begin{pmatrix} \frac{1}{3^2} & 0 & 0 \\ 0 & \frac{1}{3^2} & 0 \\ 0 & 0 & \frac{1}{3^2} \end{pmatrix} = \frac{1}{3^2}\, I$

Entonces: $A^4 = A^2A^2 = \frac{1}{3^2}\, I \frac{1}{3^2}\, I = \frac{1}{3^4}\, I$; $A^6 = A^4A^2 = \frac{1}{3^4}\, I \frac{1}{3^2}\, I = \frac{1}{3^6}\, I$, etc.

b) Visto lo anterior, será fácil calcular:

$$M = I + A^2 + A^4 + A^6 + \cdots = I + \frac{1}{3^2}\, I + \frac{1}{3^4} I + \frac{1}{3^6}\, I + \frac{1}{3^8}\, I + \cdots$$

$$= \left(1 + \frac{1}{3^2} + \frac{1}{3^4} + \frac{1}{3^6} + \cdots\right) I$$

Calcula la matriz M sabiendo que los infinitos términos de una progresión geométrica (aquella en la que un término se obtiene multiplicando el anterior por una constante llamada razón e indicada por r) se pueden sumar con la fórmula:

$a_1 + a_2 + a_3 + a_4 + \cdots = \frac{a_1}{1-r}$

siempre que la razón sea un número $|r| < 1$.

Calculando M tendremos la suma de las potencias pares de A.

c) Ahora vamos a calcular la suma de las potencias impares. Para ello observa que:

$$A^3 = A^2A = \frac{1}{3^2}\, IA = \frac{1}{3^2}\, A$$

$$A^5 = A^4A = \frac{1}{3^4}\, IA = \frac{1}{3^4}\, A$$

$$\cdots$$

Por tanto, es fácil calcular:

$$H = A + A^3 + A^5 + A^7 + \cdots = A + \frac{1}{3^2}\, A + \frac{1}{3^4}\, A + \cdots = \left(1 + \frac{1}{3^2} + \frac{1}{3^4} + \frac{1}{3^6} + \cdots\right) A$$

Calcula H.

d) Con lo anterior es ahora fácil calcular S:

$$S = I + A + A^2 + A^3 + A^4 + A^5 + \cdots = M + H$$

Comprueba que sale:

$$S = \begin{pmatrix} \frac{9}{8} & \frac{3}{8} & 0 \\ \frac{3}{8} & \frac{9}{8} & 0 \\ 0 & 0 & \frac{3}{2} \end{pmatrix} = \begin{pmatrix} 1.125 & 0.375 & 0 \\ 0.375 & 1.125 & 0 \\ 0 & 0 & 1.5 \end{pmatrix} \qquad (1)$$

e) Vamos a comprobar con un programa en Matlab el resultado obtenido en el apartado anterior. El programa te pedirá (usando la orden input) qué cuántos sumandos se quiere sumar de la expresión anterior (los 100 primeros por ejemplo) y el programa debe devolver el valor de dicha suma. Lógicamente el resultado debe ser parecido al valor límite de S dado en la expresión (1). Hazlo de la forma más eficiente posible (ver ejercicio 5).

Ejercicio 7 (ordenador)

Hacer un programa que genera una lista aleatoria de 1000 números enteros entre 1 y 100. A continuación debe preguntarnos (con la orden input) por un número entre 1 y 100 y el programa debe devolvernos cuántas veces aparece en la lista y en qué posiciones se encuentra (las dos cosas).

Funciones en Matlab

Vamos a ver cómo se trabajan con funciones en Matlab.

Supongamos que queremos escribir una función llamada nombre_funcion. Crearemos un script o programa llamado nombre_funcion.m con la siguiente estructura:

function [salida1, salida2, …]=nombre_funcion(entrada1,entrada2, …)

conjunto de ordenes;

¡Es obligatorio que el nombre del script coincida con el nombre de la función!

Por ejemplo, vamos a hacer una función que reciba tres números de entrada y nos diga cuántos hay positivos. El script se llamará mi_fun.m

```
function [contador]=mi_fun(a,b,c)
contador=0;
if a>0
contador=contador+1;
end
if b>0
contador=contador+1;
end
if c>0
contador=contador+1;
end
```

Una vez que hemos salvado este fichero en el directorio de trabajo, nos podremos ir a la línea de comandos de Matlab/Octave y ejecutar: mi_fun(2, -1, 3), por ejemplo, y el programa nos devuelve 2. Podemos volver a llamarla mi_fun(3,4,5.1) y el programa nos devolverá un 3. Podemos llamar a una función todas las veces que se desee, tanto desde la ventana de comandos como desde dentro de otros programas o scripts.

Nota: Matlab también admite la declaración: `function contador=mi_fun(a,b,c)`

- La función puede tener varias variables de entrada y de distintos tipos (numéricas, de caracteres, vectores o matrices) y puede que devuelva varias variables o ninguna. Por ejemplo, la función siguiente tiene como variable de entrada el radio de una circunferencia y nos devuelve el perímetro de la circunferencia y el área del círculo:

```
function [perimetro,area]=nueva_fun(r)
perimetro=2*pi*r;
area=pi*(r^2);
```

Si después en la línea de comandos ponemos:

>>[a, b]=nueva_fun(5)

Veremos que nos devuelve a=31.41 y que b=78.5

La definición anterior también es válida usando espacios en blanco en lugar de comas. Así, podríamos escribir:

```
function [perímetro area]=nueva_fun(r)
```

y podríamos escribir:

[a b]=nueva_fun(5)

en una llamada a la misma.

También podemos hacer que la función nos devuelva una matriz o un vector. Por ejemplo:

```
function v=nueva_fun(r)
perimetro=2*pi*r;
area=pi*(r^2);
v=[perímetro area];
```

Después podemos escribir:

vector=nueva_fun(3);

y tendremos en vector(1) el perímetro y en vector(2) el área.

También puede que una función no devuelva nada o que no tenga ninguna variable de entrada o ambas cosas. Por ejemplo:

```
function []=saludo()

for i=1:5
fprintf('Hola ¿qué tal estás?\n');
end
```

Después se puede llamar a la función con `saludo( )` o simplemente con `saludo` (sin paréntesis).

También admite en este caso la declaración: `function saludo()`

Ejercicio 8 (ordenador)

Consideremos la sucesión: $1, 2x\ 3x^2, 4x^3, 5x^4, \ldots$

Hacer una función de dos variables llamada misuma( ), cuyas variables de entrada sean x y n y la función debe devolver el valor de la suma: $1 + 2x + 3x^2, +4x^3 + \cdots + nx^{n-1}$

Hacer un programa principal donde se le pida al usuario un valor de la x y un valor de la n y devuelva la suma anterior usando la función misuma( ).

(Solución: Por ejemplo misuma(0.5,100) debe devolver 4)

Ejercicio 9 (ordenador)

Un número entero entre 0 y 31 se escribe con 5 bits:

$$0 \rightarrow 00000$$
$$1 \rightarrow 00001$$
$$2 \rightarrow 00010$$
$$3 \rightarrow 00011$$
$$\ldots$$
$$30 \rightarrow 11110$$
$$31 \rightarrow 11111$$

Hacer un programa que nos pida un número entero en ese rango (con la orden input) y nos calcule ese número escrito en binario con 5 bits. Crea una función para ello.

Ejercicio 10 (lápiz y papel y ordenador)

En este ejercicio vamos a ver cómo extraer un mensaje oculto en una imagen. Esto se conoce como esteganografía. La imagen se llama "estegolsb1.png" y debes bajártela de Moodle.

Vamos a usar a continuación un alfabeto compuesto de 32 caracteres que incluye las 27 letras del abecedario además del punto, la coma, los símbolos de interrogación y el espacio en blanco.

Cada carácter lo vamos a hacer corresponder con un número:

-la letra A se corresponde con un 0

-la letra B se corresponde con un 1

-la letra C se corresponde con un 2

-etc.

Hemos visto en el ejercicio anterior que un número entre 0 y 31 se convierte en una secuencia de 5 bits. De esta forma, se obtendría la siguiente tabla:

A	B	C	D	E	F	G	H	I	J	K
0	1	2	3	4	5	6	7	8	9	10
00000	00001	00010	00011	00100	00101	00110	00111	01000	01001	01010

L	M	N	Ñ	O	P	Q	R	S	T	U
11	12	13	14	15	16	17	18	19	20	21
01011	01100	01101	01110	01111	10000	10001	10010	10011	10100	10101

V	W	X	Y	Z	.	,	¿	?	
22	23	24	25	26	27	28	29	30	31
10110	10111	11000	11001	11010	11011	11100	11101	11110	11111

Empezamos con el proceso de extracción del mensaje:

1) Leemos la imagen estegolsb1.png.

Eso se hace con la orden:

```
A=imread('estegolsb1.png');
```

La imagen debe estar en el directorio de trabajo de Matlab.

Esta orden está dentro del toolbox (librería de Matlab) llamada Image Processsing. Esta librería está instalada en la versión de Matlab con la que trabajamos en clase.

Después mostramos la imagen en Matlab con la orden:

```
imshow(A)
```

2) A continuación leemos los 15 primeros niveles de gris de los píxeles de la imagen moviéndonos de izquierda a derecha empezando en la esquina superior izquierda. Eso se consigue en Matlab con la orden

```
B=A(1, 1:15)
```

Nos sale la lista $B = [222\ \ 225\ \ 224\ \ 225\ \ 227\ \ 226\ \ 226\ \ 226\ \ 228\ \ 228\ \ 229\ \ 225\ \ 221\ \ 219\ \ 215]$

En general, si B es un vector, la orden $B(i{:}\, i+n)$ obtiene el vector $[B(i), B(i+1), \ldots, B(i+n)]$

3) Creamos una nueva lista llamada *bits*:

- si el nivel de gris es par ponemos un 0
- si el nivel de gris es impar ponemos un 1

De manera que *bits* será un vector de 15 componentes de la forma:

$$bits = [0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]$$

4) Agrupa de 5 en 5 los elementos de la lista *bits* y convierte cada vector de 5 bits en un número. Obtendrás una lista con 3 números: $[11\ \ 0\ \ \ 31]$

5) Convierte cada número en carácter de acuerdo al alfabeto empleado: el 11 es la letra L, el 0 es la letra A, el 31 es el espacio en blanco.

De esta forma hemos sacado los tres primeros caracteres ocultos en la imagen.

Debes razonar de idéntica forma y obtener el 4º y el 5º caracteres ocultos.

(Solución: LA CR)

Nota: Si te ha interesado el tema de esteganografía puedes hacer la tarea opcional 1.