**Introducción a Matlab: Creación/indexado/operaciones con matrices. Gráficos**

En algunos apartados se incluye la respuesta **(en color azul)** a las cuestiones planteadas. Respecto de los ejercicios que están en rojo : las respuestas y el código utilizado hay que entregarlos en la tarea de Moodle "Ejercicios Comp. Matlab 1" antes del 21/9/21,12 horas, en la correspondiente Hoja de Entrega.

**A. Creación de vectores y matrices.** Dar los comandos de Matlab para crear:

1. Un vector fila **x** con los números 1 a 10 y convertirlo en otro vector **y** en precisión simple. Usando whos verificar el tamaño de ambos vectores en memoria:
   x=[1:10], y=single(x), whos x y
2. Un vector columna con los números pares de 2 a 20 x=[2:2:20]'
3. Un vector fila con los números de la tabla del 7.
4. Un vector x con elementos desde 0 a $2\pi$ (2*pi) a saltos de 0.01.
5. Dar un comando para obtener el valor del último elemento de x, sin conocer el número de elementos de x.
6. Dar el comando para obtener el número de elementos de x, guardar el número de filas en la variable n y el nº de columnas en la variable m. length(x), [n,m]=size(x)
7. Un vector columna con los números del 10 al 5 en orden decreciente: x=[10:-1:5]'
8. Una matriz cuya primera fila sean los números pares menores o iguales que 10 y su segunda fila los correspondientes impares. A = [[2:2:10]; [1:2:10]]
9. Un vector con 101 elementos (equiespaciados) desde 0 a $\sqrt{3}$ (raíz cuadrada de 3: sqrt(3)). [0:sqrt(3)/100:sqrt(3)]=linsapce(0, sqrt(3), 101)

**B. Crear vectores y matrices utilizando funciones básicas** (zeros, ones, rand)

Dar los comandos de Matlab para crear:

1. Una matriz de tamaño 3x4 inicializada con ceros. A=zeros(3,4)
2. Una matriz con 3 filas y seis columnas cuyos elementos valen 7.
3. Un vector fila de 5 elementos con números aleatorios entre 0 y 1 con una distribución uniforme (rand): **x=rand(1,5)**
4. Un vector fila de 5 elementos con números aleatorios entre -1 y 1 con una distribución uniforme (función rand) x=2*rand(1,5)-1

**C. Acceder a un dato o conjunto de datos de una matriz o vector**

1. Primero creamos la **matriz** A=[1 2 3 4;5 6 7 8;9 10 11 12]. **Sin preguntar por las dimensiones de A, escribir los comandos para extraer los siguientes elementos** de A que se indican a continuación, almacenándolos en una nueva matriz B:

1. La tercera fila de A (B=A(3,:)) ¿Cuál es la dimensión de B?
2. La primera y la tercera filas de A.
3. La última columna de A.
4. Los elementos de índice impar de la última columna de A: B=A(1:2:end,end)
5. Los elementos 2 y 4 de las filas 1 y última de A. B=A([1 end], [2 4])
6. Escribir una sentencia para extraer de la matriz A la matriz C=[10 11;6 7].
7. Escribir el comando para obtener la matriz C=[12 11;8 7;4 3] a partir de A.
8. Escribir una sentencia para mostrar las dimensiones de A, almacenando el resultado en las variables n y m: [n,m]=size(A)

**2.** Creado el vector x = [ 2 4  7 5 9 3],

a) Extraer los elementos mayores que 4 en un vector que llamaremos y. y=x(x>4)
b) Poner a cero los elementos menores o iguales a 4   x(x<=4)=0 Confirmar qué valores lleva el vector x.
b) Sumar 2 a los elementos mayores que 5 de x.
c) Crear un nuevo vector y que tenga 1's en las posiciones de los elementos de x que sean mayores que la media (función mean) y 0's en las que son menores que la media. m=mean(x); y = x; y(x>=m)=1; y(x<m)=0;
d) Poner a cero los elementos pares (usar función rem) x( rem(x,2)==0) = 0;
e) Extraer los elementos mayores que 3 y , guardar en un nuevo vector y. y=x(x>3)
f) Cambiar el signo de los valores de x verificando que 2<=x(i) < 5:
selected = ( x>=2 & x<5); x(selected)=-x(selected);

**3.** Sea el vector notas = 6+1.2*randn(1,70) que contienen las notas de 70 alumnos de la asignatura de Algorítmica Numérica:

a) Calcular la nota media (comando mean) y guardar en med el valor y la nota máxima (comando max) de los alumnos y guardar en mx.
b) Calcular la nota media de los alumnos aprobados (>=5)
c) Poner un 5 a aquellos alumnos con nota entre 4.5 y 5. En MATLAB se pueden combinar condiciones usando el operador lógico & (AND). Ver ejercicio anterior.

## D. Operaciones básicas con vectores y matrices.

**1.** Crear el vector **x**=[0:3:30] y a partir de él dar los comandos de Matlab para:

a) Sumar 3 a todos los elementos de x: x=x+3
b) Sumar 1 a los elementos con índice par: x(2:2:end)=x(2:2:end)+1
c) Poner a cero los elementos de x con índice impar.
d) Crear un vector d con la diferencia entre elementos consecutivos de x.
Esto es d(1) = x(2)-x(1), d(2)=x(3)-x(2), etc.
e) Multiplicar todos sus elementos por 5:
f) Cambiar el signo a los elementos de x con índice impar.

**2.** Dado el vector x=[0:pi/2:2*pi], calcular un vector que llamaremos y con los valores de la función seno (sin) aplicada a los elementos de x. Repetir con la función exp().

**3.** Crear los vectores fila x=[1 2 3 4]; y=[5 6 7 8];

a) Formar un vector fila de tamaño 8 concatenando x e y: A=[x, y] o A=[x y]
b) Formar una matriz A de tamaño 2x4 cuyas filas sean los vectores x e y: A=[x; y]
c) Formar una matriz 4x2 cuyas columnas sean los vectores x e y

**4.** Crear una matriz A de 5 x 7 con números aleatorios usando la función randn(5,7). Calcular con un solo comando MATLAB un vector con:

a) La media (mean) de cada columna de A y guardarlo en y (mean(A)) (Fijarse que y es un vector 1x7)
b) La media de cada FILA de A. mean(A') (Vector de dimensión 1x5)
c) La media de TODOS los elementos de A. mean(A(:)) (dimensión 1x1)
d) El elemento máximo de cada columna de A: max(A) (Vector 1x7)
e) El elemento máximo de A: max(A(:)) (vector 1x1)
f) La máxima dispersión (max –min) de cada columna de A dis=max(A)-min(A) ¿Qué dimensión tiene el vector dis?

## E. Operaciones de álgebra lineal con vectores y matrices

Crear las matrices A=[1 2; -1 1], B = [-1 0; 0 1], y los vectores x=[2 3], **y =[-1 1]**

**1.** Multiplicar las matrices A y B siguiendo las reglas del álgebra lineal: A*B
**2.** Multiplicar la matriz A por el vector x: A*x'
**3.** Calcular el producto escalar de los vectores x e y: p=x*y'
**4.** Comprobad que la multiplicación matricial no es conmutativa ( A*B =/ B*A).
**5.** Ejecutad C=A*A*A*A y comprobar que es equivalente a C=A^4 y no a C=A.^4

## F. Operaciones punto a punto con vectores

**1.** Crear los siguientes vectores x=[1 2 3 4]; y=[5 6 7 8];

a) Multiplicar cada elemento de x por el correspondiente de y: z=x.*y
b) Dividir cada elemento de x por el correspondiente de y.
c) Elevar al cuadrado cada elemento de x: z=x.^2
d) Calcular un vector z conteniendo el recíproco (1/x) de cada elemento de x.
e) Elevar los elementos de x a la potencia indicada por el correspondiente elemento de y. z=x.^y
f) A partir del vector x, dar un comando para obtener el vector [1 4 27 256]

**2.** Crear un vector x con las fracciones 1/1, 1/2 , 1/3, 1/4, … 1/10. x=1./[1:10]

**3.** Generar un vector x con los primeros 11 números de la serie: 0, 1/2, 2/3, 3/4, … ( 0/1, 1/2, 2/3, 3/4, …).
n=0:10; x=n./(n+1)

**4.** Crear el vector x con los términos de la sucesión $(-1)^n$ con n= 0, …, 10 x=(-1).^[0:10]

**5.** 1. Crear un vector c con los cuadrados de los números enteros del 0 al 10. c=[0:10].^2
2. Con un solo comando, crear el vector dif conteniendo la diferencia entre cuadrados consecutivos del apartado anterior (1^2-0^2, 2^2-1^2, 3^2-2^2,…, 10^2-9^2). dif=c(2:end)-c(1:end-1)

**6.** 1. Crear un vector llamado x con los números de la sucesión $\dfrac{(-1)^n}{2n+1}$ con n = 0, …, 100.
n=0:100; x=((-1).^n)./(2*n+1)
2. Sumar los números de la sucesión del apartado anterior y llamar vap. vap=sum(x)

3. La siguiente serie (fórmula de Leibniz) proporciona valores aproximados de $\pi/4$:

$$\sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} = \frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \ldots = \frac{\pi}{4}.$$

Por lo que el valor obtenido en el apartado 2 es un valor aproximado de pi/4. Repetir los pasos anteriores con 10000 términos (llamaremos vap2 al resultado).
¿Cuál de los dos resultados aproximados está más próximo al valor de pi/4? Para ello calcularemos los errores error1=abs(vap-(pi/4)) y error2=abs(vap2-(pi/4)).

**7.** a) Construir el vector n con los números enteros de 0 a 10.
b) Construir el vector (que llamaremos a) de las potencias desde 0 a 10 de 2, esto es $2^0$, $2^1$, $2^2$,..., $2^{10}$
c) Construir el vector s con los elementos inversos del vector a.
d) Calcular un valor aproximado de 2 (que llamaremos vaprox1) utilizando la relación matemática

$$1+\frac{1}{2}+\frac{1}{2^2}+\frac{1}{2^3}+\frac{1}{2^4}+\ldots=2$$

y considerando los 11 primeros sumandos de la serie anterior. Calcular qué error (error1) produce dicha aproximación (error1=abs(vaprox1-2)).
d) Repetir el paso anterior considerando los primeros 10000 términos, llamaremos aprox2 al resultado error2 al error. ¿Cuál de las dos aproximaciones es mejor, proporciona un resultado más próximo al valor 2?

## G. Gráficas en MATLAB

**1.** Vamos a pintar las gráficas de las funciones $f_1(x)=\sin(3x+x^2)$ y $f_2(x)=\sqrt{\left|\frac{\sin(1+x)}{1+x}\right|}$ en el intervalo [0,pi]. Para ello seguiremos los siguientes pasos:

1.1. Crear un conjunto de puntos, que llamaremos x, con "bastante resolución" en el intervalo dado, por ejemplo x=[0:0.01:pi];

1.2. Evaluamos las funciones en los puntos del vector x y guardamos los resultados en los vectores **f1** y **f2** respectivamente:
f1=sin(3*x+x.^2);
f2=…….;

1.3. Pintamos las gráficas la primera en azul y la segunda en rojo:
plot(x,f1,'b',x,f2,'r');

*Nota: El comando plot admite el formato* ***plot(x,f1,'c')*** *donde 'c' es una letra que representa el color deseado ('r', rojo, 'b', azul, 'g', verde, 'k', negro, 'y', amarillo, etc.).*

1.4. Lo anterior también se puede hacer pintando primero una gráfica, usando el comando hold on, añadiendo la segunda gráfica y posteriormente desactivando el cargar más gráficas en la figura con el comando hold off**:** plot(x,f1,'b'); hold on; plot(x,f2,'r'); hold off

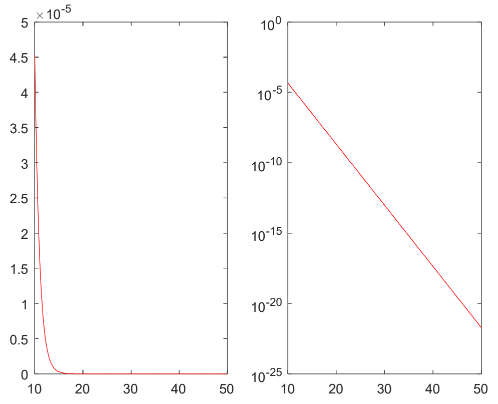
*Nota: El comando hold on permite superponer las gráficas de varios plots en los mismos ejes. Si no se usa, el último plot borra los anteriores (hold off vuelve al modo habitual).*

**2.** Vamos a pintar la gráfica de la función f(x)=exp(-x) en el intervalo [10,50]. Para ello vamos a crear una ventana dividida en dos columnas (subplot (1,2, )):
- En la primera columna (subplot (1,2,1)), incluir la gráfica usando el comando plot de la función $f(x)=e^{-x}$ en el intervalo [10,50], evaluando la función en un soporte x de 10 a 50 a saltos de 0.1. Se observa que los valores de la función se aproximan rápidamente a cero y <u>no se aprecia bien su comportamiento</u> en todo el intervalo ¿Los valores de la función en x=20 y x=40 son del mismo orden? Por ejejmplo ¿se puede ver de qué orden es el valor de la función en x=30? ¿10^-1, 10^-2,….?
- Para solucionar este problema (apreciar mejor los valores de la función próximos a cero) usaremos una <u>escala logarítmica</u> en el eje y (comando semilogy). Para ello, en la segunda columna del objeto gráfico (subplot (1,2,2)): incluir la gráfica de la función usando el comando semilogy(x,f) (<u>escala logarítmica</u> en eje <u>y</u>). Notad que a la vista de esta gráfica, ahora si podríamos responder a la pregunta del punto anterior.

***Notas:*** *1. La función e^x en Matlab se calcula con el comando exp(x).*
*2. Los comandos semilogx/semilogy se usan de forma similar al comando plot pero la gráfica la pintan en <u>escala logarítmica</u> (base 10) en el eje de las x's o y's, respectivamente. El comando loglog usa una escala logarítmica en ambos ejes.*

```
x=10:0.1:50;
f=exp(-x);
subplot(1,2,1), plot(x,y,'r');
```

```
subplot(1,2,2), semilogy(x,y,'r');
```




**3.** Vamos a calcular algunos valores aproximados del nº e considerando los primeros mil valores de la sucesión $\lim_{n\to\infty} (1+\frac{1}{n})^n = e$ . Para ello:

1. Creamos el vector n de 1 a 1000.
2. Creamos el vector vap de valores aproximados con los términos de la sucesión dada para n de 1 a 1000 $((1+\frac{1}{1})^1, (1+\frac{1}{2})^2, \ldots\ldots, (1+\frac{1}{1000})^{1000})$ ¿Qué valor toma el último término de la sucesión?
3. Creamos el vector, que llamaremos error, que contiene los errores (en valor absoluto) que se cometen al considerar los valores aproximados anteriores respecto del nº e (exp(1) en Matlab) restando el valor aproximado y el valor exacto: error=abs(vap-exp(1)). Pintamos la gráfica del error respecto de n con asteriscos rojos (plot(n,error,'*r') ¿El error es cada vez menor?

Prueba ahora a pintar la gráfica anterior en escala semilogarítmica en el eje y.