

EJERCICIOS DE MATLAB.

OPERACIONES MATRICIALES

- 1) Crear una matriz de números reales y tamaño 2×4:

$$M = \begin{pmatrix} 0 & 2 & 4 & 6 \\ 0 & 3 & 5 & 7 \end{pmatrix}$$

Guardar la variable M en el fichero *matriz.mat*. A continuación, limpiar el espacio de trabajo y volver a cargar el fichero, comprobando que la variable está en memoria. Por último, seleccionar la segunda fila, la cuarta columna, y las submatriz formada por las dos primeras filas y las dos primeras columnas.

- 2) Definir las siguientes matrices en MATLAB:

$$A = \begin{pmatrix} 1 & 3 & 5 & 8 \\ 2 & 6 & 5 & 3 \\ 4 & 1 & 9 & 7 \\ 1 & 8 & 0 & 2 \end{pmatrix}; B = \begin{pmatrix} 1 & 9 & 5 & 8 \\ 12 & 5 & 5 & 9 \\ 4 & 2 & 9 & 74 \\ 0 & 6 & 0 & 3 \end{pmatrix}; C = \begin{pmatrix} 1 & 9 \\ 10 & 2 \end{pmatrix}$$

- Realizar los siguientes cálculos básicos con estas matrices:
3·A, A-7, A·B^T, A⁻¹, B⁻¹
- Realizar ahora los siguientes cálculos, siendo D la submatriz de A formada por las 1ª y 3ª filas y columnas, y E la submatriz de B formada por las 2ª y 4ª filas y columnas:
D·E^T, D·C, C·E
- Resolver las siguientes ecuaciones:
A·x=B, D·x=C
- Siendo F la submatriz de A formada por las filas 2, 3 y 4, y G la submatriz de B formada por las columnas 1, 2 y 4, calcular:
F·G

- 3) Repetir el ejercicio 1) con la siguiente matriz de números complejos:

$$N = \begin{pmatrix} j & 1+j & -j & 5 \\ -j & 4+2j & 3 & 9-3j \end{pmatrix}$$

- 4) Este ejercicio nos servirá para varias cosas. Por un lado, para ver la conveniencia de trabajar con matrices en lugar de con escalares y bucles, y por otro lado para ver un primer ejemplo de cómo desarrollar funciones propias en Matlab (*help function*). Los elementos básicos de Matlab son matrices, los escalares son matrices de 1×1 elemento. En general, es mejor trabajar con matrices que operar con escalares y bucles, y en caso que no haya más remedio que utilizar bucles es mejor declarar las variables, aunque no es algo necesario en Matlab. Para ello, implementar una función que calcule los cuadrados de los N=1000 y 10000 primeros números naturales de 3 formas distintas:

- Utilizando un bucle *for* sin declarar las variables.
- Utilizando un bucle *for* declarando previamente la variable.
- Calculando el cuadrado del vector correspondiente (operación componente a componente).

Calcular, con la ayuda de la instrucción de Matlab *cpitime* el tiempo que se tarda en realizar el cálculo con cada uno de los tres métodos. La función a implementar debe tener como variable de entrada el número N, y como salida el tiempo que se tarda con cada uno de los tres métodos (t1, t2, t3).

- 5) Obtener utilizando la función *linspace*, un vector de 20 elementos que recorra el intervalo $[0, \pi]$. Generar utilizando la sentencia *valorinicial:paso:valorfinal*, un vector de referencia de 1000 puntos que recorra el intervalo $[0, 10]$.
- 6) Generar un vector que sea una secuencia aleatoria uniforme (*rand*) de amplitud unidad con 250 puntos. Encontrar:
 - Aquellas componentes del vector cuyo valor sea mayor que 0.9.
 - Aquellas componentes del vector cuyo valor sea menor o igual que 0.15.
- 7) Generar un vector de 250 puntos que sea una secuencia aleatoria normal (*randn*) con media cero y desviación estándar igual a 3. Encontrar:
 - Aquellas componentes cuyo valor esté situado entre 2 y 3.
 - Aquellas componentes cuyo valor esté situado entre -1 y -2.
- 8) Existe un procedimiento para determinar los número primos aplicando una especie de "criba"; de hecho este procedimiento se conoce como "*criba de Eratóstenes*". Este procedimiento consiste en la eliminación sucesiva de todos los múltiplos de un determinado número, a partir de 2, considerando dicho número inicial como primo si no ha sido eliminado. Así el procedimiento empieza por 1 que se considera primo, seguidamente se pasa al 2 (primo); eliminaríamos a continuación todos los múltiplos de 2; quedarían así eliminados 4,6,8,10,12, El siguiente número sin eliminar es el 3, primo, seguidamente eliminaríamos sus múltiplos, 6 (ya eliminado),9,.....El siguiente número es el 5 (el 4 ha sido eliminado) que es primo, a continuación se eliminan sus múltiplos.....Al final sólo quedan sin eliminar los números primos. Determinar un programa en MATLAB que determine los 100 primeros números primos utilizando este procedimiento.
- 9) Escribir un programa que pida un número del teclado. Mientras este número sea distinto de la unidad hará lo siguiente: si el número es par lo dividirá por 2, y si es impar lo multiplicará por 3 y le sumará 1. El programa debe visualizar por pantalla la secuencia generada. Por ejemplo, si el número introducido es 9, la secuencia será: 28 14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1.