

SESIÓN 5. PARTE 2.

Números aleatorios

- Calcular números aleatorios entre dos valores “a” y “b”.
<http://matlab.digym.upm.es/calcular-numeros-aleatorios-entre-dos-valores-a-y-b/>
- Ejemplos con los comandos rand y floor. <http://matlab.digym.upm.es/asignar-valores-iniciales-con-floor-y-rand-ejemplos-individuales-y-mezcladas/>

Funciones

- Concepto. <http://matlab.digym.upm.es/concepto-de-funcion-en-matlab/>
- Ejemplo con dos variables de salida. <http://matlab.digym.upm.es/funcion-en-matlab-ejemplo-con-varias-variables-de-salida/>

Ejercicio propuesto 5.3 (no entregable)

Enunciado:

1. Hacer un subprograma function Matriz_xxx (donde xxx con las iniciales del nombre y apellidos) cuyo argumento de entrada sea un vector “a” cEl argumento de salida será una matriz M calculada a partir de la formula siguiente:

$$M(i, j) = f1(i + j) + \sum_{k=1}^n a(k) * f2(i)$$

Donde: $1 \leq i, j \leq n$, siendo n la longitud del vector a. f1 y f2 son dos funciones handle: $f1(x) = x^2$ y $f2(x) = x + 1$

2. Hacer un programa que haga lo siguiente.

- a. Pedir un vector de n elementos y cada elemento tendrá un valor aleatorio entre los valores 1 y 3.
- b. Llamar a la función Matriz_xxx con el valor de entrada del vector “a” calculado anteriormente para obtener la matriz M1
- c. Calcular una matriz M2, que se calculará de la forma siguiente $M2(i, j) = M1(i, j) - 1$
- d. Visualizar la matriz M2.

Solución: (aportada durante la videoconferencia del día 25 de marzo)

- **Subprograma**
https://drive.google.com/open?id=1IZ7gTW30dlQmVe_PsaHue1CISaY2ITa0
- **Programa principal** <https://drive.google.com/open?id=1CanYPcUB0F2aR-ZcT6MNIkwTwyL4O1Xh>

Ejercicio propuesto después de la sesión: Sumatorio de funciones

Calcular:

$$M(i, j) = \sum_{k=1}^n (a(i + j) * f_k(X(k)));$$

Siendo:

f1=@(x) x+1

f2=@(x) x+2

f3=@(x) x+3

1<=i,j<=n

n=longitud del vector a=[3,4,5,6,7,8]

Solución Versión 1. Trasladar resultados funciones a vectores de forma previa al sumatorio.

<https://youtu.be/Uz6MuZX2syQ>

Solución Versión 2. Con variables indexadas (es más intuitiva que la anterior)

```
%con indexación de celdas {}
clc
clear all
f{1}=@(x) x+1 %llaves en la función
f{2}=@(x) x+2
f{3}=@(x) x+3
X=[1,2,3]
a=[3,4,5,6,7,8]
n=length(X)
for i=1:1:n
    for j=1:1:n
        su=0;
        for k=1:1:n
            su=su+(a(i+j)*f{k}(X(k))); %llaves en la función
        end
        M(i,j)=su;
    end
end
end
M
```

Dudas:

- **Dibujar nube de puntos vs recta:** la nube de puntos se puede dibujar con el vector x obtenido de forma aleatoria. Pero para dibujar en el resto de funciones el vector x debe estar ordenado, el comando, por ejemplo X=sort(x) guardaría en X el vector x pero ordenado de menor a mayor.
- **Dibujar una función punto a punto vs uniendo los puntos.** plot(x,y,'g') dibujaría puntos verdes y plot(x,y,'g') dibuja líneas. Observar que si pongo un punto delante de la g dibuja los puntos y en el otro caso no.