

## Práctica 4: Empezando a trabajar con ficheros .m

1. Construir una función que calcule el seno de cualquier valor más 5. Utilizarla para calcular  $\sin(45)+5$ ,  $\sin(n)+5$  con  $n$  los 10 primeros números naturales.

Se crea un fichero .m en la forma:

```
function p=f(x);  
p=sin(x)+5;  
% esta función determina el seno de cualquier numero y le suma 5 unidades
```

Una vez guardado, en la línea de comandos ejecutamos:

```
>> f(45)  
ans =  
    5.8509  
  
>> f([1:10])  
ans =  
Columns 1 through 7  
    5.8415    5.9093    5.1411    4.2432    4.0411    4.7206    5.6570  
Columns 8 through 10  
    5.9894    5.4121    4.4560  
  
>> % como vemos la función se aplica tanto en números como en vectores
```

2. Repetir el ejercicio anterior para el cálculo del cubo de los valores.

```
function p=f(x);  
p=x.^3;  
% esta función determina el cubo de cualquier número
```

```
>> f(45)  
ans =  
    91125  
>> f([1:10])  
ans =  
Columns 1 through 6  
     1     8    27    64   125   216  
Columns 7 through 10  
   343   512   729  1000
```

```
>> % Importante señalar en la necesidad de utilizar el punto delante de la operación elevar al cuadrado si queremos aplicar la función con éxito sobre vectores
```

3. Define la función  $g(x) = \frac{x^3 - x}{x^2 + 1}$

a. Calcula  $g(4)$ .

b. Calcula  $g(x)$  siendo  $x = (-2, -1.9, -1.8, \dots, 1.8, 1.9, 2)$

>> % Si definimos la función:

```
function p=g(x);  
p=(x^3-x)/(x^2+1);
```

>> % Apartado a

```
>> g(4)
```

```
ans =
```

```
3.5294
```

>> El apartado b nos da problemas:

```
>> g([-2:0.1:2])
```

```
??? Error using ==> ^
```

```
Matrix must be square.
```

```
Error in ==> C:\MATLAB6p5\work\g.m
```

```
On line 2 ==> p=(x^3-x)/(x^2+1);
```

>>% Cambiamos la definición de g utilizando el punto delante de las operaciones necesarias para que pueda aplicarse sobre vectores:

```
function p=g(x);  
p=(x.^3-x)/(x.^2+1);
```

>> % Apartado b

```
>> g([-2:0.1:2])
```

```
ans =
```

```
Columns 1 through 7
```

```
-1.2000 -1.0757 -0.9509 -0.8260 -0.7011 -0.5769 -0.4541
```

```
Columns 8 through 14
```

```
-0.3335 -0.2164 -0.1045 0 0.0945 0.1756 0.2396
```

```
Columns 15 through 21
```

```
0.2824 0.3000 0.2897 0.2505 0.1846 0.0980 0
```

```
Columns 22 through 28
```

```
-0.0980 -0.1846 -0.2505 -0.2897 -0.3000 -0.2824 -0.2396
```

```
Columns 29 through 35
```

```
-0.1756 -0.0945 0 0.1045 0.2164 0.3335 0.4541
```

```
Columns 36 through 41
```

```
0.5769 0.7011 0.8260 0.9509 1.0757 1.2000
```

```
>>
```

4. Construir una función que calcule el ángulo formado entre dos vectores.

```
function p=ang(u,v);
p=acos(dot(u,v)/(norm(u)*norm(v)));
% Determina en radianes el ángulo formado entre dos vectores
```

5. Define una función que permute las filas i y j de una matriz.

```
function p=permuta(a,i,j);
% extraemos la fila i de la matriz a y la guardamos
e=a(i,:);
% igualemos la fila i a la fila j
a(i,:)=a(j,:);
% igualamos la fila j al valor guardado de la fila i
a(j,:)=e;
p=a;
```

```
>> % Verificamos que funciona con este ejemplo:
```

```
>> a=[1 2 3;0 0 0; 1 1 1]
```

```
a =
```

```
1 2 3
```

```
0 0 0
```

```
1 1 1
```

```
>> permuta(a,1,2)
```

```
ans =
```

```
0 0 0
```

```
1 2 3
```

```
1 1 1
```

```
>>
```

6. Construir una función  $\text{ecu}(a,b,c)$  que determine las raíces de la ecuación  $ax^2+bx+c=0$ .

```
function [p,q]=raiz(a,b,c);
% esta función determina de forma simple las raíces de un polinomio de 2º
% grado  $ax^2+bx+c$ . Mas adelante se mejorará.
d=sqrt(b^2-4*a*c);
p=(-b+d)/2;
q=(-b-d)/2;
```

```
>> % Veamos un ejemplo
```

```
>> [p,q]=raiz(1,-1,0)
```

```
p =
```

```
1
q =
0
>>
```

7. Define como variable global la matriz  $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$ . Define una función que calcule el incremento de todos los elementos de A en un número a elegir.

```
>> % En la línea de comandos de la ventana de trabajo escribimos
> global A
>> A=[1 2 3;4 5 6];
```

Se crea el fichero .m siguiente:

```
function p=f(n);
global A;
p=A+n;
```

```
>> % Ejecutamos en ejemplo:
```

```
>> f(3)
ans =
4 5 6
7 8 9
```

8. Construir una función tal que, dados dos vectores u, v, devuelva un vector con los elementos de u excepto el primero, y los elementos de v excepto el último.

```
function p=prult(u,v);
%creamos el vector formado por los elementos de u excepto el primero
u1=u(2:length(u));
%creamos el vector formado por los elementos de v excepto el último
v1=v(1:length(v)-1);
p=[u1,v1];
```

```
>> % veamos un ejemplo:
```

```
> u=[1 2 3 4 5]; v=[6,7,8];
>> prult(u,v)
ans =
2 3 4 5 6 7
>>
```