Cálculo Simbólico con MATLAB. Introducción a la Programación.

Bloque V: Introducción a la Programación con Matlab.

Cristina Solares

Universidad de Castilla-La Mancha

11 de septiembre de 2020

- Hemos visto que cuando el problema es sencillo podemos **escribir instrucciones directamente en la línea de comandos** de MATLAB.
- Cuando el número de instrucciones es elevado, se pueden escribir los comandos en un fichero, llamado **archivo de instrucciones**, con la extensión `.m`. Si el fichero se llama `ejemplo.m`, para ejecutar las instrucciones basta con escribir `ejemplo` en la línea de comandos. Las variables de dicho fichero son globales en la sesión actual de MATLAB y se puede hacer referencia a las mismas desde la línea de comandos. En la barra de menús seleccionamos NewScript para crear un nuevo fichero de instrucciones y lo guardamos en nuestro directorio de trabajo. Si queremos modificarlo podemos abrirlo directamente o escribir `edit ejemplo` en la ventana de comandos.
- Otra opción es crear lo que se denomina un **archivo de función**. Es un fichero donde se incluyen las instrucciones de nuestro programa dentro de una función, ahora las variables son locales a dicha función. Para definir la función se utiliza la palabra clave `function`. Dicha función puede tener argumentos y devolver valores, ambos son opcionales. Si la función se llama `ejemplofuncion`, el fichero se llama `ejemplofuncion.m`. Para utilizarlo basta con llamar a la función desde la línea de comandos pasándole los argumentos correspondientes. El fichero debe estar guardado en nuestro directorio de trabajo. Se pueden definir varias funciones dentro del mismo archivo de función, pero esto lo veremos más adelante.

La altura h de las olas, en mar abierto, depende de la velocidad v del viento y de la duración del tiempo t que el viento haya estado soplando a esa velocidad. En la siguiente tabla aparecen valores de la función $h = f(v, t)$ en pies (ft):

Duración (horas)

Velocidad del viento (nudos)

v \ t	**5**	**10**	**15**	**20**	**30**	**40**	**50**
10	2	2	2	2	2	2	2
15	4	4	5	5	5	5	5
20	5	7	8	8	9	9	9
30	9	13	16	17	18	19	19
40	14	21	25	28	31	33	33
50	19	29	36	40	45	48	50
60	24	37	47	54	62	67	69

Crear un programa en MATLAB tal que dada la matriz A con las alturas de ola anteriores devuelva la submatriz con las alturas de ola cuando la velocidad del viento está entre 20 y 40 nudos y el viento sopla durante un tiempo comprendido entre 15 y 40 horas.

Introducción a la Programación. Ejemplo 1.

Creamos un archivo de instrucciones llamado **ejemplo1.m**, en este archivo insertamos el código del programa

ejemplo1.m

```
A=[2 2 2 2 2 2 2;4 4 5 5 5 5 5;5 7 8 8 9 9 9;9 13 16 17 18 19 19;
14 21 25 28 31 33 33;19 29 36 40 45 48 50; 24 37 47 54 62 67 69];
v=A(3:5,3:6)
```

En la ventana de comandos escribimos

```
>> ejemplo1
v =
     8     8     9     9
    16    17    18    19
    25    28    31    33
```

Introducción a la Programación. Ejemplo 2.

Crear un programa en Matlab que, dado un vector con las temperaturas máximas diarias en Ciudad Real, durante el mes de enero de 2015, devuelva un vector con los días del mes donde la temperatura máxima ha estado comprendida entre 12°C y 15°C. Creamos un archivo de funciones llamado **ejemplo2.m**, en este archivo colocamos el código del programa dentro de una función.

ejemplo2.m

```
function diasmes=ejemplo2(vectortemp)
 n=length(vectortemp);
 diasmes=[];
 for i=1:n
    if (12<=vectortemp(i))&&(vectortemp(i)<=15)
        diasmes=[diasmes,i];
    end
 end
end
```

Desde la línea de comandos hacemos

```
>> ejemplo2([9 8 12 15 3 2 18 11 10 9 0 2
3 1 17 18 9 8 12 15 3 2 18 11 10 9 2 3 12 14 15])
ans =
     3     4    19    20    29    30    31
```

Introducción a la Programación. Ejemplo 3.

A continuación vamos a definir la función $f(x, y) = sin(x) + cos(y)$ en un archivo de función llamado `func1.m`. Aplicaremos dicha función en los puntos (π, π) y (a, b). Por último vamos a dibujar la función utilizando el comando `fsurf`.

func1.m

```
function a=func1(x,y)
%Ejemplo f(x,y)=sin(x)+cos(y)
a=sin(x)+cos(y);
end
```

Desde la línea de comandos hacemos $f(\pi, \pi)$ y $f(a, b)$

```
>> func1(pi,pi)
ans =
   -1.0000
>> syms a b; func1(a,b)
ans =
cos(b) + sin(a)
```

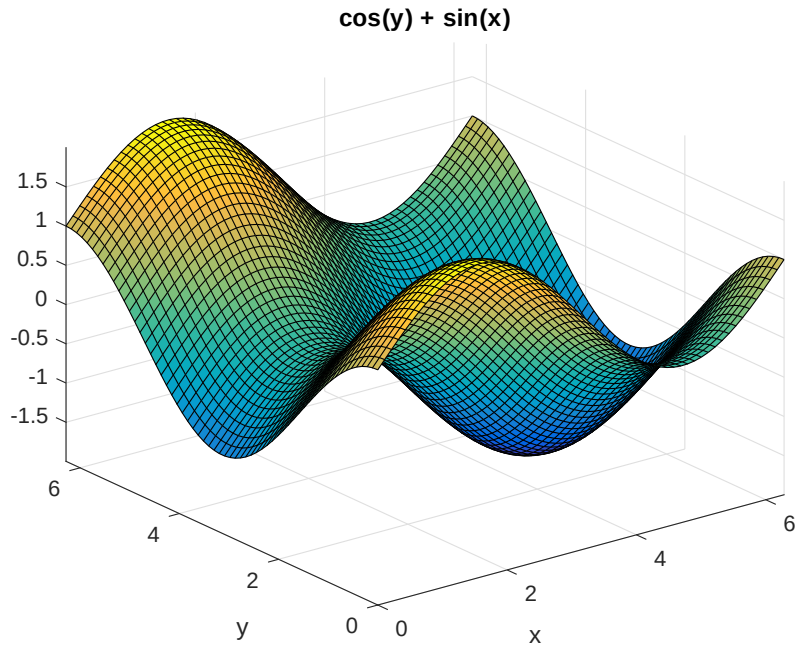
Los comentarios que hemos escrito aparecerán en la pantalla si hacemos

```
>> help func1
Ejemplo f(x,y)=sin(x)+cos(y)
```

Introducción a la Programación. Ejemplo 3.

Para dibujar la función hacemos

```
>> syms x y; fsurf(func1(x,y),[0,2*pi,0,2*pi])
```



Introducción a la Programación. Ejemplo 4.

A continuación creamos un programa que calcula el área de un círculo

calcarea.m

```
function area=calcarea(r)
%El siguiente programa calcula el area de un circulo con radio r
area=pi*r^2;
end
```

Si ejecutamos el programa en la ventana de comandos

```
>> calcarea(2)
ans =
   12.5664
>> help calcarea
El siguiente programa calcula el area de un circulo con radio r
```

Se puede observar que el comando `help` devuelve el comentario.

Introducción a la Programación. Ejemplo 4.

Se pueden **introducir datos** a través del teclado utilizando el comando `input`.

calcarea.m

```
function area=calcarea()
%El siguiente programa calcula el area de un circulo de radio r

%En primer lugar pedimos al usuario que introduzca el radio r
r=input('Introduce un numero entero positivo (radio):');
%El area del circulo se calcula a partir de su radio
area=pi*r^2;
end
```

Si queremos **mostrar la salida** de nuestro programa podemos utilizar el comando `disp`

calcarea.m

```
function calcarea()
%El siguiente programa calcula el area de un circulo con radio r

%En primer pedimos al usuario que introduzca el radio r
r=input('Introduce un numero entero positivo (radio):');
%El area del circulo se calcula a partir de su radio
area=pi*r^2;
disp(area);
end
```

Introducción a la Programación. Ejemplo 5.

Se pueden definir funciones anidadas, es decir, funciones dentro del código de otras funciones.

ecuadrado.m

```
function z=ecuadrado(x)
function y=cuadrado(x)
y=x^2;
end
z=exp(cuadrado(x));
end
```

En la línea de comandos

```
>> ecuadrado(2)
ans =
   54.5982
```

Introducción a la Programación. Ejemplo 5.

Se pueden definir varias funciones dentro de un mismo fichero, una función se llamará igual que el fichero con la extensión `.m`.

ecuadrado.m

```
function z=ecuadrado(x)
z=exp(cuadrado(x));
end

function y=cuadrado(x)
y=x^2;
end
```

En la línea de comandos

```
>> ecuadrado(2)
ans =
   54.5982
```

Introducción a la Programación. Ejemplo 6.

Se pueden definir funciones con dos o más salidas. La salida se muestra entre corchetes y separadas por comas. En el siguiente ejemplo, la función devuelve la medida y desviación típica de un conjunto de datos

ecuadrado.m

```
function [m,s] = estdescriptiva(x)
n = length(x);
m = sum(x)/n;
s = sqrt(sum((x-m).^2/n));
end
```

En la línea de comandos

```
>> %Medidas velocidad en km/s
>> datos=[883 816 778 796 682 711 611 599 1051 781 578 796 774 820 772];
>> [media,desvtipica]=estdescriptiva(datos)
media =
  763.2000

desvtipica =
  115.4182
```