Tema 8: Tratamiento de datos. Ficheros

1. Introducción

En el tema anterior hemos visto una forma de entrar datos o imprimir resultados a través de funciones como input o disp. En este tema se estudiarán otras formas de intercambiar datos entre el Matlab y diversos programas o aplicaciones.

Veremos que Matlab dispone de comandos de entrada y salida que actúan sobre ficheros y que permiten abrir y cerrar ficheros e importar y exportar datos.

2. Importar y exportar datos

Antes de estudiar opciones más complejas, es interesante comentar que la opción de copiar y pegar (Copy/Paste) puede ser adecuada en muchos casos. Por ejemplo, copiar elementos de Excel y depositarlos en Matlab entre corchetes funciona con frecuencia.

Nota: Esta opción puede generar problemas por ejemplo, según se hayan introducido los números decimales.

2.1. Utilizando el Current Directory

Si tenemos un fichero .txt, .data,... y lo hemos situado en el directorio actual de trabajo en Matlab, pinchando sobre él con el botón derecho tenemos la opción Import Data (figura 28). Se abre entonces un menú donde se nos ofrecen posibilidades como elegir entre varios separadores de columnas. Si todo está como queremos la pestaña Next crea una variable con el nombre del fichero y el contenido del mismo que ya puede ser usada en la sesión de trabajo (figura 29).

Esta opción de Matlab nos permite trabajar de forma muy simple en caso de ficheros de datos adecuados.

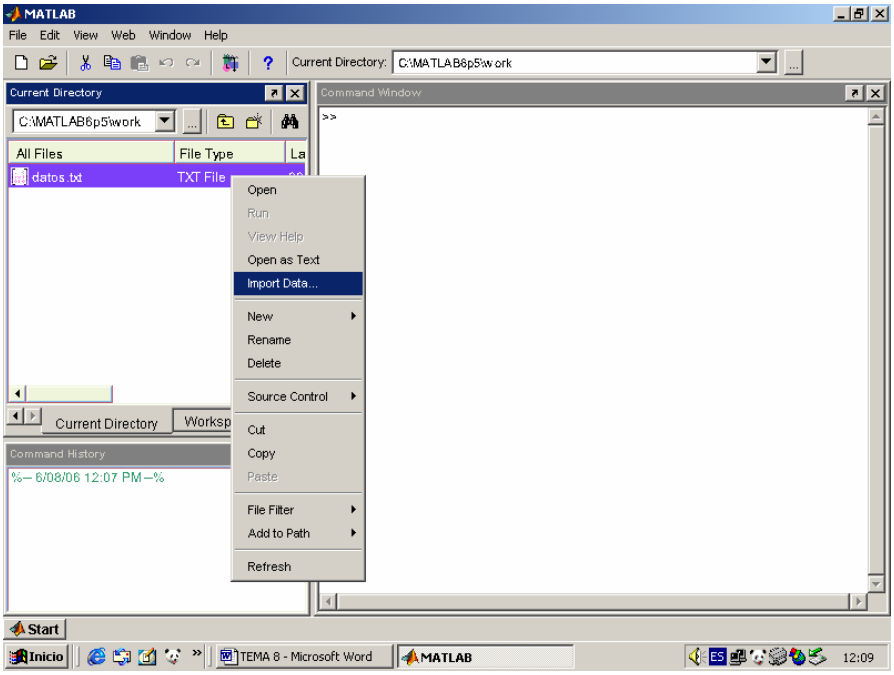

Figura 28

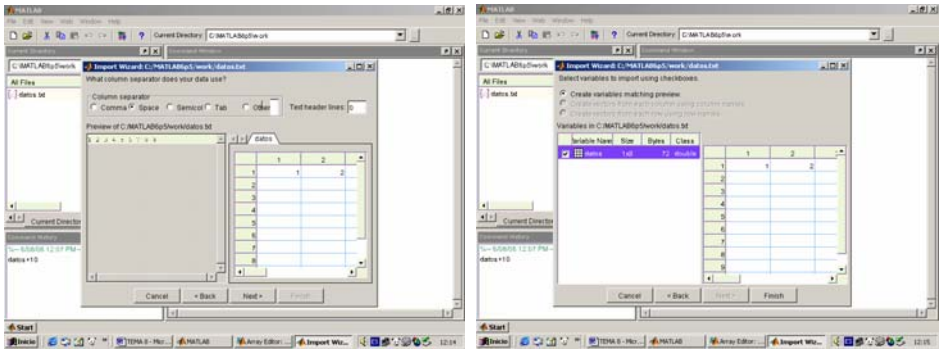

Figura 29

2.2. Comandos

Comandos fopen, fclose y fprintf

Estos comandos sirven para abrir y cerrar ficheros.

Para abrir un fichero, se utiliza el siguiente comando:

variable = fopen('fichero','permiso') donde:

- variable es el nombre de la variable que guarda el identificador del fichero.
- 'fichero' especifica el nombre externo y la dirección del archivo.
- 'permiso' indica el modo de apertura del fichero:

  r abre un fichero existente para lectura.

r+ abre un fichero existente para lectura y escritura.

w crea un fichero nuevo para escritura.

w+ crea un fichero nuevo para lectura y escritura.

Destacar que este comando lo que hace es poner en contacto el programa con un fichero, no lo visualiza en pantalla. Para trabajar con ficheros lo primero será abrirlo y lo último cerrarlo.

Para cerrar un fichero se utiliza la función: fclose(fid) que cierra el fichero de identificador fid y devuelve 1 si el cierre es correcto y 0 si es incorrecto.

Para escribir en un fichero se utiliza el comando: fprintf(fid,'format',A,….) que escribe los elementos especificados en A (que en general es una matriz) en el fichero de identificador fid (previamente abierto) con el formato especificado en 'format'. Así, la función fprintf dirige su salida a un fichero indicado por el indicador.

Formato de datos:

%d Enteros.

%f Reales con punto fijo.

%e Reales con formato exponencial.

%g Utiliza uno de los formatos anteriores; el que dé la mayor precisión en el menor espacio.

Si se utiliza el comando fprintf('format',A,….) la escritura de datos se realiza en la pantalla.

Ejemplo:

Creamos un fichero ASCII de nombre "resultados" que contiene los valores de la función exponencial para valores de la variable entre 0 y 1 separados una décima y lo representamos en la pantalla.

```
>>x=0:.1:1
>>y=[x;exp(x)];
>>fid=fopen('resultado.txt','w');
>> fprintf(fid,'%6.2f    %12.8f   \n', y);
```

Este programa escribiría en el fichero resultado.txt los siguientes valores:

```
0.00    1.00000000
0.10    1.10517092
0.20    1.22140276
0.30    1.34985881
```

.....

1.00 2.71828183

Comando fscanf

La lectura de datos a partir de un fichero ASCII se realiza mediante los comandos:

[A,cont]=fscanf(fid,'formato') que lee datos con el formato especificado del fichero abierto con el identificador fid, en un vector columna de nombre A. cont es el número de datos leídos.

[A,cont]=fscanf(fid,'formato',size) que lee datos con el formato especificado del fichero abierto con el identificador fid y los escribe en la matriz A de tamaño size. cont es el número de datos leídos.

Ejemplo:

Se supone que en la carpeta de trabajo de MATLAB: 'work', se encuentra un archivo de nombre datos.txt, cuyo contenido es:

1 2 3 4 5

6 7 8 9 10

>>fid=fopen('datos.txt','r')

>>[A,cont]=fscanf(fid,'%d')

La salida es el vector columna de contenido:1 2 3 4 5 6 7 8 9 10 y cont=10.

Otra posibilidad:

>>[A,cont]=fscanf(fid,'%d',[2,5])

A=

1 3 5 7 9

2 4 6 8 10

cont=10

Nota: Si realizamos estas operaciones seguidas tendremos problemas en la segunda debido a que el fichero ha sido anteriormente leído y es necesario rebobinarlo si se quiere leer de nuevo. Entonces se utiliza

>>frewind(fid)

Otras posibilidades:

>>[A,cont]=fscanf(fid,'%d',[3,3])

A=

```
    1 4 7
    2 5 8
    3 6 9
cont=9

>>[A,cont]=fscanf(fid,'%d',[4,4])
A=
     1 5 9
     2 6 10
     3 7 0
     4 8 0
cont=10

>>[A,cont]=fscanf(fid,'%d',[4,inf])
```

>> %se está fijando como número de columnas de la matriz el valor mínimo que permita la lectura de todos los datos del fichero. La salida es la misma que en el caso anterior.

3. Trabajando con Excel

Matlab permite una muy buena conexión con este programa. Ya hemos comentado que si los datos son adecuados, la opción copiar en Excel y pegar en Matlab entre corchetes y dando un nombre a la variable creada funciona directamente. Esto nos obliga a que los números estén definidos en Excel como en Matlab, por ejemplo, los decimales con puntos y no con comas.

En general, por ejemplo si en el fichero existen cabeceras, lo apropiado es importar el fichero según lo explicado en el punto 2.1. Se crean entonces variables del tipo: data (con los valores numéricos), colheadest (con las cabeceras) y textdata (con el texto de las cabeceras). No existe así ninguna incompatibilidad de formatos.

Si se dispone de la toolbox "exlink" la conexión con este programa es total y directa:

Elegir en Excel del menú Herramientas/Complementos/examinar: (Matlab-toolbox-excelink). Aparece entonces en excel una barra de herramientas con: put matriz, get matriz y evaluate.

- Excel-Matlab: Se crean unos valores en Excel y se selecciona put matriz. Se abre Matlab y pregunta por el nombre para esa matriz. Se crea así una nueva variable en el Workspace sin ningún problema de compatibilidad.

- Matlab.Excel: Se utiliza get matriz. Esto abre el Workspace de Matlab y desde allí se exporta la matriz.

Práctica 8: Tratamiento de datos. Ficheros

1. Crear un fichero de texto con el block de notas con los datos: 1,2,3,4,5,6,7,8,9,10. Importarlo desde Matlab y crear un vector con su contenido.

2. Sea el fichero *datos.txt* que contiene la siguiente información:

7 5 2 1 9 3 4 5 7 8 4

Almacenar estos datos en una matriz A de 4 filas y 3 columnas.

3. Crear un fichero de Excel con las cantidades compradas de varios productos y su importe por unidad. Cargarlo en Matlab y calcular allí el coste total de la compra.