

Introducción al cálculo científico con Octave

(Segunda edición)

Sesión 10: relaciones con MATLAB y LaTeX

1 Incluir códigos octave en LaTeX

Os dejamos aquí un ejemplo (ver también sesión 1) de cómo incluir código *octave* (y otros) en LaTeX, para hacer apuntes, artículos,... Recordamos que es preciso tener el paquete `mcode`, y que hemos de cargarlo con un `\usepackage`

El siguiente código LaTeX (nótese que al cargar el paquete con `\usepackage`, hemos incluido las opciones *framed* y *numbered*, para que coloque un cuadro y numere)...

```
\usepackage[framed,numbered]{mcode}
```

```
\begin{lstlisting}
function y = nif (dni)
%
% Esta funcion resuelve la letra correspondiente al dni introducido
%
letras = 'TRWAGMYFPDXBNJZSQVHLCKE';
n=rem(dni,23)+1;
y=letras(n);
endfunction
\end{lstlisting}
```

...produce la siguiente salida...

```
1 function y = nif (dni)
2 %
3 % Esta funcion resuelve la letra correspondiente al dni introducido
4 %
5 letras = 'TRWAGMYFPDXBNJZSQVHLCKE';
6 n=rem(dni,23)+1;
7 y=letras(n);
8 endfunction
```

Otras formas son las siguientes:

- Para un objeto en mitad de una línea: `\mcode{fragmento de código}`,
- Para incluir un fichero externo (con el mismo formato que `\begin{lstlisting}`) usaremos `\lstinputlisting{fichero.m}`,

2 Datos exportados con octave y gráfica con tikz

Si queremos hacer una gráfica en TeX con datos provenientes de una simulación numérica (o cualquier cálculo) realizado con octave, podemos guardar los datos con octave (ver sesión 1 de este curso y sección 2.1) y después usar el paquete *tikz*. Aquí os dejamos un ejemplo.

```
\usepackage{tikz}

\begin{figure}[ht]
\begin{center}
\begin{tikzpicture}[>=latex, scale=1.0,xscale=1,yscale=2]
\coordinate [label=above:{$t$}] (t) at (-6,1.65);
\coordinate [label=right:{$x$}] (y) at (2,0);
\coordinate [label=below:{$cero$}] (punto) at (1.03,-0.1);
\draw[-] (punto) -- (1.03,0.1);
\draw[->] (-6,0) -- (t) ;
\draw[->] (-6,0) -- (y) ;
\draw[solid,line width = 1.0pt] plot file{./datos.mat};
\end{tikzpicture}
\end{center}
\caption{El fichero datos.mat contiene la curva en param\'etricas.}
\label{figura}
\end{figure}
```

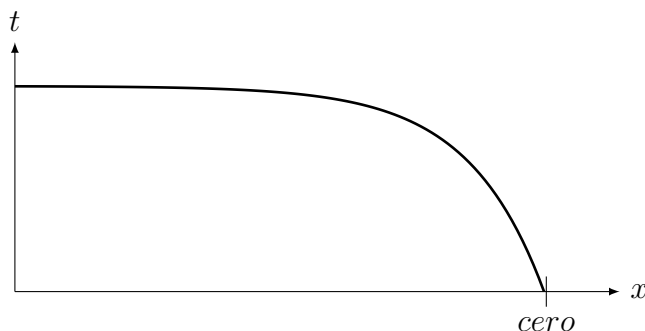


Figure 1: El fichero datos.mat (llamado en la línea indentada) contiene (véase sección 2.1) los puntos de la curva dibujada en paramétricas.

2.1 El fichero datos.mat

En la sesión 2, ya aprendimos a exportar los valores de una variable mediante el comando *save nombrefichero variable*. Así hemos creado el fichero *datos.mat*, cuyo contenido es texto puro y puede ser editado posteriormente a mano en caso necesario. Por completitud, escribimos aquí su contenido

```

# Created by Octave 3.4.0, Tue Feb 4 10:04:38 2014 CET <jjmnieto@ugr.es>
# name: yz
# type: matrix
# rows: 100
# columns: 2
-6 1.357901538141189
-5.929292929292929 1.35781073303541
-5.858585858585859 1.357713274931147
-5.787878787878788 1.357608676384569
-5.717171717171717 1.357496414238397
-5.646464646464646 1.357375927005301
-5.575757575757576 1.357246612059577
-5.505050505050505 1.35710782262307
-5.434343434343434 1.356958864530267
-5.363636363636363 1.356798992756379
-5.292929292929293 1.35662740769104
-5.222222222222222 1.356443251139003
-5.151515151515151 1.356245602027811
-5.080808080808081 1.356033471800985
-5.010101010101010 1.355805799473688
-4.939393939393939 1.355561446326132
-4.868686868686869 1.355299190208186
-4.797979797979798 1.355017719426704
-4.727272727272728 1.354715626184996
-4.656565656565657 1.354391399541631
-4.585858585858586 1.354043417853349
-4.515151515151516 1.353669940664297
-4.444444444444445 1.353269100001012
-4.373737373737374 1.352838891029614
-4.303030303030303 1.352377162028482
-4.232323232323233 1.351881603626265
-4.161616161616162 1.351349737251385
-4.090909090909091 1.350778902735287
-4.020202020202021 1.350166245007411
-3.949494949494949 1.349508699815341
-3.878787878787879 1.348802978398732
-3.808080808080808 1.348045551040327
-3.737373737373737 1.347232629411824
-3.666666666666667 1.346360147626269
-3.595959595959596 1.345423741902231
-3.525252525252526 1.344418728738027
-3.454545454545455 1.343340081486846
-3.383838383838384 1.342182405215609
-3.313131313131313 1.340939909721815
-3.242424242424243 1.339606380573421

```

-3.171717171717172 1.338175148026906
-3.101010101010101 1.336639053668062
-3.03030303030303 1.334990414608658
-2.95959595959596 1.333220985059917
-2.888888888888889 1.331321915090601
-2.818181818181818 1.32928370636343
-2.747474747474747 1.327096164628454
-2.676767676767677 1.324748348735771
-2.606060606060606 1.322228515912558
-2.535353535353535 1.319524063030744
-2.464646464646465 1.316621463571534
-2.393939393939394 1.313506199971544
-2.323232323232324 1.310162691012142
-2.252525252525253 1.306574213888846
-2.181818181818182 1.302722820570994
-2.111111111111111 1.298589248033358
-2.040404040404041 1.294152821910711
-1.96969696969697 1.289391353093486
-1.898989898989899 1.284281026747311
-1.828282828282829 1.278796283201386
-1.757575757575758 1.272909690109912
-1.686868686868687 1.266591805247212
-1.616161616161616 1.259811029250287
-1.545454545454546 1.252533447572292
-1.474747474747475 1.244722660856447
-1.404040404040404 1.236339602881999
-1.333333333333334 1.227342345171659
-1.262626262626263 1.21768588728326
-1.191919191919192 1.207321931736752
-1.121212121212121 1.196198642450831
-1.050505050505051 1.184260385480982
-0.9797979797979801 1.171447450762245
-0.9090909090909092 1.157695753464956
-0.8383838383838382 1.142936513469787
-0.7676767676767682 1.127095911358952
-0.6969696969696972 1.110094719202992
-0.6262626262626263 1.091847904296496
-0.5555555555555554 1.072264203860806
-0.4848484848484853 1.051245668586558
-0.4141414141414144 1.028687172733053
-0.3434343434343434 1.004475888334178
-0.2727272727272734 0.9784907208810859
-0.2020202020202024 0.950601703659157
-0.1313131313131315 0.9206693477099757
-0.06060606060606055 0.8885439441671062

```

0.0101010101010095 0.8540648154762551
0.08080808080808044 0.8170595117547343
0.1515151515151514 0.7773429482707694
0.2222222222222223 0.7347164797286814
0.2929292929292924 0.6889669067299201
0.3636363636363633 0.6398654094406838
0.4343434343434343 0.5871664031327973
0.5050505050505052 0.5306063098737376
0.5757575757575752 0.4699022402223381
0.6464646464646462 0.4047505783365648
0.7171717171717171 0.3348254634166923
0.7878787878787872 0.2597771598886893
0.8585858585858581 0.1792303081761764
0.9292929292929291 0.09278204731206974
1 0

```

3 Variables globales

Al igual que en *MatLAB*, en *Octave* es posible usar *variables globales* de forma que puedan ser usadas por todas las funciones llamadas en una misma sesión y que no tengan que ser recalculadas cada vez que sea necesario utilizarlas, con el ahorro computacional que ello supone.

La sintaxis básica es:

```

1 global variable1 variable2 etc
2 variable1 = valor1; variable2 = valor2; etc

```

Y no olvidar que una función que use alguna de estas variables declaradas globales, debe contener esta indicación en su cuerpo, de nuevo mediante el comando *global*:

```

1 function y = nombrefuncion (x)
2 global variable1
3 y = variable1.*x;
4 end

```

Os dejamos aquí un ejemplo completo extraído de la sesión 8, de ecuaciones diferenciales, en la que surgió la cuestión. En este ejemplo, las variables declaradas globales serán *a*, *b*, *alpha*, *beta*, *nu*, *k*, *m*, *h*, *A*, *rhs* correspondientes a los datos de un cierto problema de valores en la frontera (véase sesión 8 para más detalles) y a su discretización espacial, que genera una ODE en la variable temporal de la forma:

$$u'(t) = \text{lineas}(u, t) := A * u(t_i) + \text{rhs}, \quad (\text{método de líneas}).$$

La idea es que definamos la función “lineas” anterior, de modo que cada vez que el comando *lsode* la llame para resolver numéricamente la OD, ésta no tenga que recalcular (perdiendo tiempo) las variables globales que la componen. En este ejemplo

sencillo, mediante los comandos *tic* y *toc* y con la ya vieja versión 3.4.0 de Octave se redujo el tiempo de cómputo más del 50%.

```

1 % Datos de la ecuacion e intervalo espacial
2 global a b alpha beta nu k m h A rhs
3 a = 0; b = 2; alpha = 1; beta = 0; nu = 0.1; k = 0.5;
4 % datos de la discretizacin
5 m = 100; h = (b-a)/(m+1);
6 % Matriz A y termino independiente
7 e = ones(m,1); A = spdiags([e -2*e e],[-1 0 1],m,m); A= nu.*A./h.^2;
8 I = speye(m); A= A-k*I;
9 rhs= zeros(m,1); rhs(1) = nu*alpha/h^2; hs(m) = nu*beta/h^2;

```

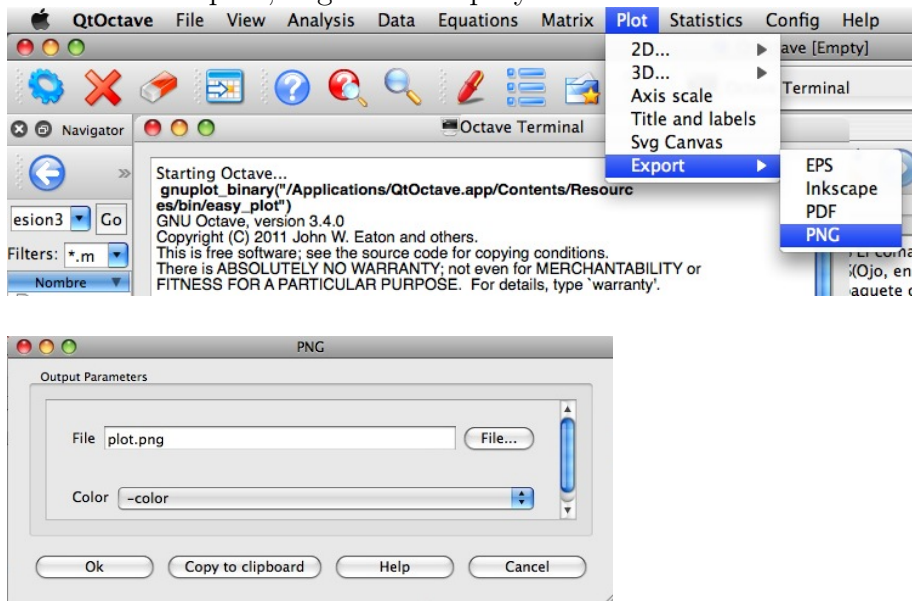
```

1 function ddot = lineas(d,t)
2 global a b alpha beta nu k m h A rhs
3 ddot = A*d+rhs;
4 endfunction

```

4 Generación directa de gráficas e (indirecta) de películas

Desde QtOctave, una vez creado un gráfico, podemos guardarlo directamente usando el menú Plot->Export, eligiendo el tipo y dando nombre al fichero salida:



Directamente desde la línea de comandos también es posible con el comando `print` (que es lo que hace el menú Plot, pero de manera interna).

```

1 > print ("plot.png", "-color", "-FHelvetica:12", "-dpng")

```

Un tratamiento posterior **ajeno a octave**, permite unir varias gráficas a modo de película. Hemos dejado un ejemplo en la sesión 8 usando ffmpeg.