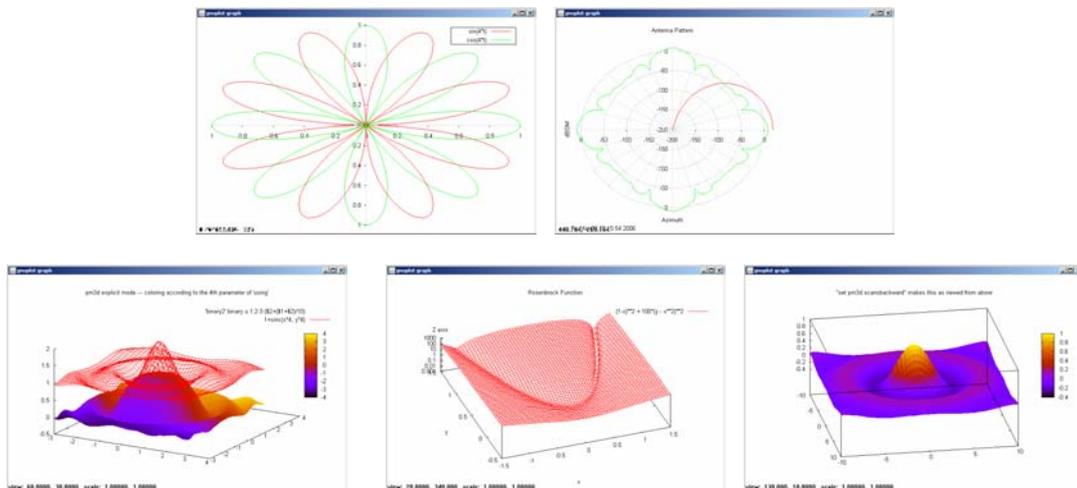


BREVE INTRODUCCIÓN A gnuplot

1. INTRODUCCIÓN

gnuplot es un programa de visualización gráfica de datos científicos. Permite realizar gráficos 2D y 3D de curvas, líneas de nivel y superficies, tanto a partir de funciones como de datos discretos.

gnuplot es software libre, lo cual significa que cualquier persona lo puede utilizar y distribuir gratuitamente. Está disponible para casi todas la plataformas. La primera versión data de 1986. La última versión es la 4.0. Información sobre el programa se puede encontrar en su página web: <http://www.gnuplot.info>



gnuplot funciona mediante comandos, que pueden usarse tanto en modo interactivo como escribiendo **scripts** (secuencia de comandos escritos en un fichero).

En estas notas se expone brevemente cómo utilizar **gnuplot** para dibujar curvas planas. El sistema operativo de referencia es Linux. Algunas distribuciones de Linux (por ejemplo RedHat) traen gnuplot entre los paquetes que se instalan por defecto. La versión dependerá de la versión de RedHat que se instale.

1.1 ¿ Cómo se ejecuta gnuplot ?

- a) Desde un terminal, escribiendo en la línea de comandos

```
>> gnuplot
```

se ejecuta el programa en modo interactivo. Aparecerá, tras unas líneas de información, el prompt

```
gnuplot>
```

en el que se pueden escribir los comandos.

- b) Ejecutando directamente un script: para ello, en la línea de comandos del terminal de Linux se escribe

```
>> gnuplot "fichero_script"
```

siendo "**fichero_script**" un fichero conteniendo comandos de **gnuplot**. Se ejecutarán de forma inmediata las ordenes contenidas en el fichero. Cuando éstas se terminan, se devolverá el control al Sistema Operativo, es decir, se termina la ejecución de gnuplot.

1.2 Algunos comandos utilitarios

COMANDOS UTILITARIOS	
<code>pwd</code>	De " P rint W orking D irectory". Indica cual es el directorio por defecto. Importante porque es en ese directorio donde gnuplot buscará los ficheros cuando se lo indiquemos.
<code>cd "directorio"</code>	De " C hange D irectory". Cambia el directorio por defecto. El path del directorio se puede indicar en forma absoluta o relativa. Ejemplos: <code>gnuplot> cd "/home/roger/graficos"</code> <code>gnuplot> cd "../pruebas"</code> <code>gnuplot> cd "antiguos"</code>
<code>load "fichero"</code>	Si " f ichero" es un script, es decir, contiene comandos gnuplot , esta orden ejecuta dichos comandos. Cuando se termina, se vuelve al modo interactivo. Aunque no es obligatorio, es muy recomendable para el usuario que el fichero lleve una extensión que nos recuerde qué contiene. Se puede usar, por ejemplo, la extensión ***.gnu , para recordarnos que contiene un script de gnuplot .
<code>clear</code>	Borra el terminal gráfico.
<code>exit / quit</code>	En la línea de comandos de gnuplot , terminan la ejecución del programa.

En un **script**, se pueden escribir varias órdenes en una misma línea, separándolas por un punto y coma. Las líneas que comienzan por el carácter **#** son ignoradas. ATENCIÓN: **gnuplot** distingue entre mayúsculas y minúsculas.

1.3 Comandos básicos de dibujo

COMANDOS BÁSICOS DE DIBUJO	
<code>plot</code>	Para dibujar curvas planas y gráficos 2D
<code>splot</code>	Para dibujar superficies
<code>replot</code>	Para hacer modificaciones de un plot o un splot anterior.

1.4 Dibujo de curvas definidas por funciones

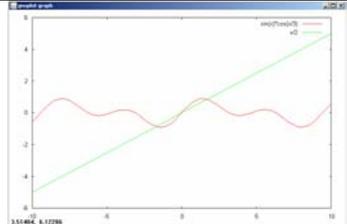
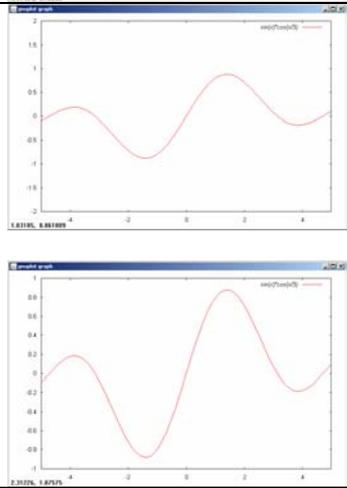
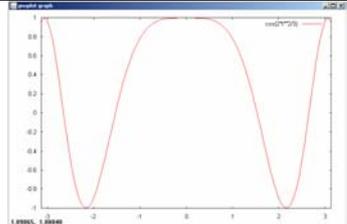
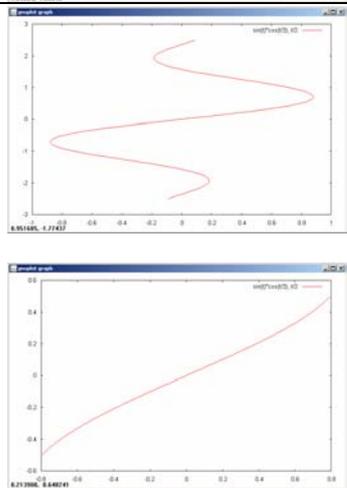
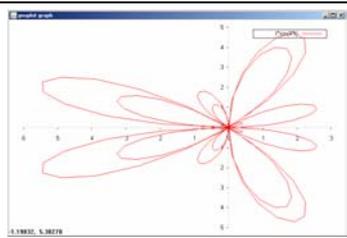
Con el comando **plot** se pueden dibujar curvas definidas por una función. Para ello se utiliza el comando en la forma:

```
gnuplot> plot funcion
```

donde **funcion** es la expresion de una funcion **f(x)**. En las expresiones, las operaciones aritméticas y la inmensa mayoría de las funciones se escriben como en **FORTRAN**. La lista de las funciones matemáticas intrínsecas disponibles en **gnuplot**, así como su descripción se puede obtener tecleando el comando

```
gnuplot> help functions
```



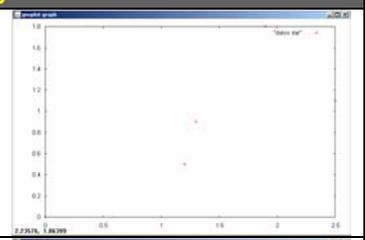
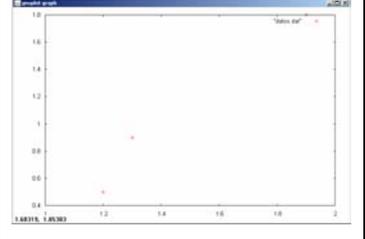
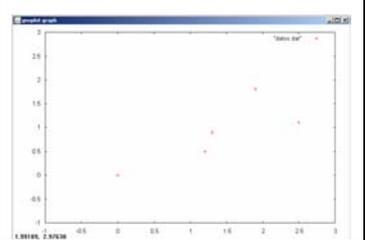
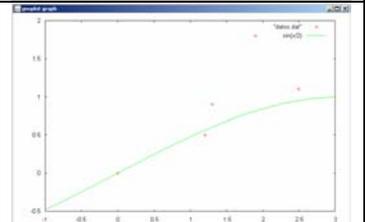
<p>Se pueden dibujar varias funciones juntas</p> <pre>>> plot sin(x)*cos(x/3), x/2</pre>	
<p>Se pueden determinar los rangos, en x y en y, que se quieren visualizar. El primero que se especifica es el x-rango y el segundo el y-rango:</p> <pre>>> plot [-5:5] [-2:2] sin(x)*cos(x/3)</pre> <p>>> plot [-5:5] sin(x)*cos(x/3)</p>	
<p>Se pueden utilizar otras variables independientes, pero hay que indicarlas expresamente en el x-rango:</p> <pre>>> plot [t=-pi:pi] cos(2*t**2/3)</pre>	
<p>Se pueden dibujar funciones definidas por sus ecuaciones paramétricas. La variable independiente por defecto, en este caso, es t</p> <pre>>> set parametric >> plot sin(t)*cos(t/3), t/2</pre> <p>También se puede indicar el rango de variación de la variable independiente:</p> <pre>>> plot [-1:1] sin(t)*cos(t/3), t/2</pre> <p>Y también se puede utilizar otra variable, si se indica expresamente:</p> <pre>>> plot [r=-1 :1] sin(r)*cos(r/3), r/2</pre>	
<p>ATENCIÓN: la orden set parametric afectará a todos los plot siguientes, hasta que se desactive:</p> <pre>>> unset parametric</pre>	
<p>Se pueden dibujar funciones en coordenadas polares. La variable independiente por defecto, en este caso, es también t</p> <pre>>> set polar >> plot [-2*pi:2*pi] t*sin(t) >> unset polar</pre>	

1.5 Dibujo de curvas definidas por puntos

Con el comando **plot** se pueden dibujar curvas definidas por un conjunto discreto de puntos cuyas coordenadas están almacenadas en un fichero.

```
gnuplot> plot "fichero.dat"
```

(la extensión *****.dat** no es obligatoria). El fichero **"fichero.dat"** debe contener, en cada línea, la abscisa y la ordenada de uno de los puntos. La orden anterior dibujará, en un rectángulo que contenga todos los puntos del fichero, un **marcador** en el lugar de cada punto. Si no se ha indicado expresamente, no se unirán los puntos entre sí con ninguna clase de línea.

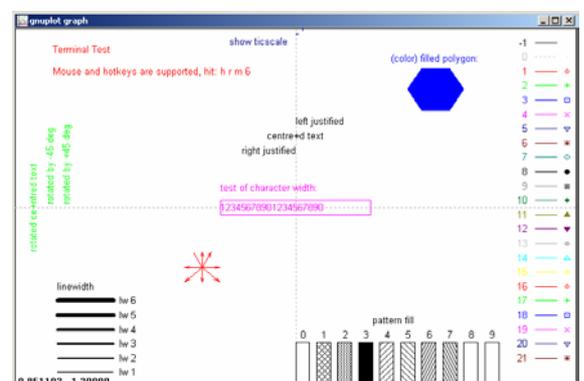
EJEMPLO: GRAFICAS DE CURVAS DEFINIDAS POR PUNTOS	
<pre>>> plot "datos.dat" 0. 0. 1.2 0.5 1.3 0.9 1.9 1.8 2.5 1.1</pre>	
<p>Se pueden determinar los rangos de las variables, igual que para las funciones: el primero es el rango para las abscisas y el segundo el rango para las ordenadas.</p> <pre>>> plot [1:2] "datos.dat"</pre>	
<pre>>> plot [-1:3] [-1:3] "datos.dat"</pre>	
<p>Se pueden combinar gráficas de función y de datos:</p> <pre>>> plot [-1:3] "datos.dat", sin(x/2)</pre>	

1.6 Modificadores del comando plot

Hay una serie de parámetros opcionales que se pueden utilizar para modificar el resultado del comando **plot**. Se exponen a continuación algunos (pocos) de ellos.

Para ver las características por defecto de las gráficas **gnuplot**, se puede usar el comando **test**.

Por ejemplo, se puede ver así el color por defecto asignado a cada curva, según su número de orden: roja la primera, verde la segunda, azul la tercera, ... etc. También se pueden ver los grosores de línea correspondientes a los distintos índices, los marcadores, etc.



1.6.1 Modificadores del aspecto de las líneas

Los modificadores siguientes se describen a continuación de la orden **plot** y sirven para elegir el estilo con que se dibujará la curva.

with lines	se unen los puntos con líneas rectas
with points	se marcan sólo los puntos
with linespoints	se marcan los puntos y se unen con líneas rectas
linewidth k	determina el grosor de la línea
linetype n	determina el tipo de línea, lo que a su vez determina el color

EJEMPLOS

```
gnuplot> plot "fichero.dat" with lines
se dibujará la gráfica de datos, uniendo los puntos mediante segmentos rectos

gnuplot> plot sin(x)*cos(x) linewidth 2
se dibujará la gráfica de la función con línea de grosor 2

gnuplot> plot sin(x)*cos(x) linetype 14 linewidth 2
se dibujará la gráfica de la función con línea tipo 14 (color cyan) de grosor 2
```

1.6.1 Modificador index

En un mismo fichero puede haber más de un conjunto de puntos, definiendo más de una curva. Conjuntos de puntos definiendo distintas curvas deben estar separados por 2 LINEAS EN BLANCO. El modificador **index** sirve para indicar qué conjuntos se deben representar. Los conjuntos se consideran numerados, pero ATENCIÓN: el primero es el CERO.

EJEMPLOS

```
gnuplot> plot "fichero.dat" index 0:1
indica que se dibujen los dos primeros conjuntos de datos del fichero "fichero.dat".

gnuplot> plot "fichero.dat" index 3
indica que se dibuje el cuarto conjunto de datos del fichero
```

1.7 El comando set

Las características de la gráfica se pueden modificar también utilizando el comando **set**. De esta forma las modificaciones se aplicarán a todas las gráficas que se realicen posteriormente. Algunos ejemplos:

set style data lines	establece el estilo "lines" para todas las graficas de datos
set style function linespoints	establece "linespoints" para las graficas de funciones
set style line 1 linewidth 2	establece que las líneas de tipo 1 tengan grosor 2
set xrange [xmin:xmax]	establece el rango x para todas las gráficas posteriores
set xrange restore	restaura el rango en x por defecto
set yrange [ymin:ymax]	ídem para y
set yrange restore	
set title "Un titulo"	establece el título para todas las gráficas posteriores
set title	restaura el título a su valor por defecto (vacío)
reset	restaura todas las características gráficas a sus valores por defecto

1.8 Cómo salvar a fichero una gráfica gnuplot

Se puede hacer que una gráfica **gnuplot** se almacene en un fichero, para luego incluirla en un texto o imprimirla. Para ello hay que redirigir la salida del programa hacia el dispositivo o driver adecuado. Explicamos aquí cómo dirigir la salida hacia un fichero **postscript**. La orden

```
gnuplot> set terminal
```

nos mostrará la lista de los terminales disponibles en nuestro sistema. Para más información consultar, en **gnuplot**,

```
gnuplot> help set terminal  
gnuplot> help set output
```

En primer lugar hay que elegir el **terminal postscript** (por defecto el terminal de salida de **gnuplot** es **X11**)

```
gnuplot> set terminal postscript
```

En segundo lugar hay que re-dirigir la salida hacia un fichero:

```
gnuplot> set output "grafica.ps"
```

A continuación se dan las órdenes adecuadas para conseguir la gráfica que nos interese. Ésta no generará una ventana gráfica, sino un fichero de nombre **grafica.ps**.

Como todos los comandos **set**, seguirán estando activos hasta que no se restauren, por ejemplo, con el comando **reset**.

1.9 Un ejemplo de script para dibujar curvas de datos contenidos en un fichero

El siguiente script puede ser usado para dibujar una curvas a partir de las coordenadas de una serie de puntos, contenidos en un fichero de nombre puntos.dat

SCRIPT curvawin.gnu PARA DIBUJAR UNA CURVA DE DATOS EN UNA VENTANA

```
set lmargin 6  
set rmargin 6  
set title "Aquí se puede poner un titulo" tc lt 1  
plot "datos.dat" with lines lt 3 lw 2  
pause mouse  
reset
```

El fichero **datos.dat** tendrá un contenido como, por ejemplo, el siguiente:

```
.000000    .000000  
.300000    .295520  
.600000    .564642  
.900000    .783327  
1.200000    .932039  
1.500000    .997495  
1.800000    .973848  
2.100000    .863209  
2.400000    .675463  
2.700000    .427380
```

Si las órdenes anteriores se almacenan en un fichero de nombre, por ejemplo, **curvawin.gnu**, cuando se disponga de un fichero **datos.dat** conteniendo las coordenadas de los puntos que determinan la curva que se desea dibujar, bastará escribir, en un terminal de Linux

```
>> gnuplot curvawin.gnu
```

para obtener la gráfica en una ventana. La ventana permanecerá activa hasta que se "pinche" en ella con el ratón.

En el **script** anterior, las órdenes **set lmargin** y **set rmargin** fijan la anchura de los márgenes izquierdo y derecho. De forma análoga existen las órdenes **set tmargin** y **set bmargin** para fijar la anchura del margen superior (top) e inferior (bottom).

La orden **set title** establece el título "**Aquí se puede poner un titulo**" para la gráfica.

La orden **pause mouse** detiene la ejecución del **script** hasta que pulsemos el botón izquierdo del ratón. Si no se incluyera, no daría tiempo de ver la gráfica, ya que la ventana sería eliminada al finalizar la ejecución de gnuplot.

La orden final **reset** restaura todas las características cambiadas con **set** a sus valores por defecto

1.10 Procedimiento para ordenar la realización de una gráfica gnuplot desde dentro de un programa FORTRAN

Supongamos que se desea dibujar la gráfica de una curva a partir de una serie de puntos que se calculan en un programa FORTRAN. Se puede hacer esto desde el mismo programa, siguiendo los pasos siguientes:

- Escribir el script **curvawin.gnu** descrito en el apartado anterior y ponerlo en el mismo directorio en el que se vaya a ejecutar el programa FORTRAN.
- Escribir también el script **curvaps.gnu** que se describe a continuación y ponerlo en el mismo directorio. Este **script** es una variante de **curvawin.gnu** para generar la gráfica en formato **postscript**.

```
SCRIPT curvaps.gnu PARA DIBUJAR UNA CURVA DE DATOS EN FORMATO
postscript

set terminal postscript
set output "datos.ps"
set lmargin 6
set rmargin 6
set title " Aquí se puede poner un titulo " tc lt 1
plot "datos.dat" with lines lt 3 lw 2
pause mouse
reset
```

- En el programa FORTRAN:
 - Crear el fichero de datos **datos.dat** con el formato indicado más arriba. Para ello las operaciones a realizar son las siguientes:
 - Abrir el fichero
open(66, file='datos.dat')
 - Escribir cada línea
write(66,*) x,y
 - Cerrar el fichero
close(66)
 - Incluir la siguiente instrucción (LINUX) si se desea la gráfica en una ventana
call system('gnuplot curvawin.gnu')
 - Incluir la siguiente instrucción (LINUX) si se desea la gráfica en un fichero **postscript**
call system('gnuplot curvaps.gnu')

ATENCIÓN: La orden OPEN anterior creará un fichero nuevo de nombre **datos.dat**.

Si ya existe en el directorio por defecto un fichero con ese nombre, la orden anterior lo "machacará".

EJEMPLO:
PROGRAMA FORTRAN PARA DIBUJAR LA FUNCION
 $X \cdot \exp(-X) \cdot \sin(3 \cdot X)$ EN EL INTERVALO [0,10]
(S.O. LINUX)

```

program dibujo
n=50
a=0.
b=10.
h=(b-a)/n
nfich=77
x=a
open(nfich,file='datos.dat')
do i=1,n+1
  x=a+(i-1)*h
  y=x*exp(-x)*sin(3.*x)
  write(nfich,*) x,y
enddo
close(nfich)
call system('gnuplot curvawin.gnu')
stop
end program dibujo

```

1.11 Subrutina para ordenar la realización de una gráfica gnuplot desde dentro de un programa FORTRAN

La siguiente subrutina, en función del parámetro **iout**, y utilizando los vectores de entrada **x** e **y**, lleva a cabo las acciones anteriores.

SUBROUTINA PARA CREAR EL FICHERO DE DATOS Y EJECUTAR gnuplot

```

subroutine plotdat(x,y,n,iout)
!-----
! Creacion del fichero de datos para dibujar una curva con gnuplot
! y lanzamiento de gnuplot para generar la gráfica
! ATENCION: adaptada a LINUX
! En otro sistema habría que modificar la orden call system
!-----
! Argumentos de entrada:
! x (vector real*4) abscisas de los puntos que definen la gráfica
! y (vector real*4) ordenadas de los puntos
! n longitud de los vectores x e y
! iout parámetro:
! si iout=0 solamente se crea el fichero de datos
! si iout=1 se genera, además, la gráfica en una ventana
! si iout=2 se genera la gráfica en formato postscript
!-----
real*4 x(*),y(*)
!
open(77,file='datos.dat',form='formatted')
write(77,*) (x(i),y(i),i=1,n)
close(77)
!
if (iout.eq.1) then
  call system('gnuplot curvawin.gnu')
else if (iout.eq.2) then
  call system('gnuplot curvaps.gnu')
endif
!
return
end

```