

INTRODUCCIÓN A GNUPLOT

Juan José Gómez Cadenas
Departamento de Física Atómica y Nuclear

1 Introducción a gnuplot

gnuplot es un programa interactivo, que permite representar gráficamente todo tipo de funciones. Es muy fácil de utilizar (de hecho basta con conocer dos comandos: **plot** y **splot**) y sin embargo es muy potente. Puede producir una gran variedad de gráficos, con una amplia gama de opciones para personalizarlos y puede enviarlos a todo tipo de terminales (impresoras, terminales, ficheros postscript, etc).

Este manual describe los elementos básicos, de **gnuplot**.

2 gnuplot para impacientes

1. *Para llamar a gnuplot:* **gnuplot (return)**
2. *Para salir de gnuplot:* **quit (return)**
3. *Ayuda dentro de gnuplot:*
 - (a) Tecleamos el comando **gnuplot (return)**. Entramos en gnuplot interactivo. Aparece el símbolo **gnuplot>**.
 - (b) Para ayuda introductoria: **help plot**
 - (c) Para ayuda sobre un tópico específico: **help (tópico)**
 - (d) Listado de todos los tópicos: **help**
4. *Manejo de Ficheros*
 - (a) Dibujar el contenido de un fichero: **plot 'fichero'**
 - (b) Cargar un fichero de macros: **load 'fichero'**
 - (c) Salvar lista de comandos en un fichero : **save 'fichero'**
5. *Comandos PLOT y SPLIT*

- (a) **plot** y **splot** son los comandos básicos para dibujar. **plot** se usa para dibujar funciones uni y bidimensionales; **splot** se utiliza para dibujar superficies tridimensionales. Ambos comandos permiten dibujar el contenido de un fichero de datos.
- (b) Sintaxis:
plot rangos función título
splot rangos función título
 Donde **función** puede ser:
 - i. Una expresión matemática.
 - ii. El nombre de un fichero de datos entre comillas.
 - iii. Una pareja (en el caso de **plot**) o un triplete (en el caso de **splot** de expresiones matemáticas, en el caso de funciones paramétricas. El usuario puede definir funciones y variables. Ver ejemplos más abajo.

3 Ejemplos Básicos

Para empezar a familiarizarnos con **gnuplot** podemos ensayar los siguientes comandos:

1. **plot cos(x)**
2. **plot [-pi:pi] sin(x**2)**
3. **splot [-3:3] [-3:3] x**2*y**

El primer comando produce un gráfico de $\cos x$, utilizando para el rango de x, y los valores por defecto de **gnuplot**. El segundo comando produce un gráfico de la función $\sin x^2$ entre $-\pi$ y π . El tercer comando produce un gráfico de una superficie tridimensional de la función $f(x, y) = x^2 y$ entre -3 y 3 para ambos rangos de x e y . Para ello, hemos utilizado las funciones *intrínsecas* de **gnuplot**, *sin()*, *cos()* y *exp()*, al igual que la constante **pi**, también definida por **gnuplot**. El programa tiene un gran número de funciones y constantes definidas, de las cuales se listan algunas en la tabla 6. El resultado de los comandos anteriores se muestra en la figura 1.

El comando **set** se usa para controlar muchas opciones disponibles en **gnuplot**. Por ejemplo, el comando **set terminal** define el tipo de terminal (de hecho, de dispositivo gráfico) que va a utilizarse. En LINUX, el defecto

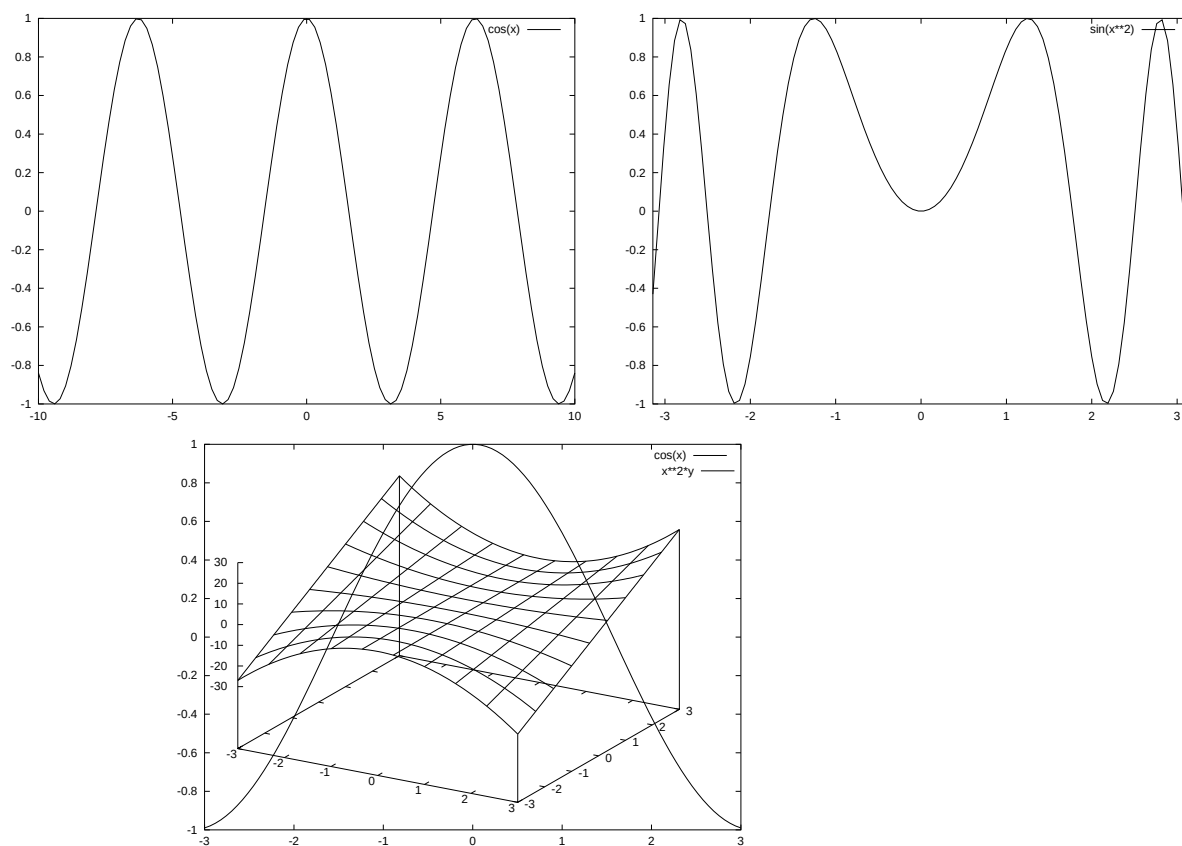


Figure 1: Ejemplo 1.

es un terminal del tipo X11, pero existe una gran variedad de dispositivos gráficos. El comando **set terminal**, ejecutado desde el programa, facilita una lista completa de dichos dispositivos.

Otro ejemplo de uso del comando **set** son los comandos **set xlabel**, **set ylabel**, **set xlabel** y **set title**. Por ejemplo:

- **set xlabel 'x'** añade el título “x” al eje X.
- **set ylabel 'y'** añade dicho título al eje Y.
- **set title 'Título'** etc.

4 Constantes y Funciones Definidas por el Usuario

Las expresiones aritméticas y lógicas utilizadas por **gnuplot** son similares a las de un lenguaje de alto nivel, como Fortran o C. Por ejemplo ****** para la exponencial, **&&** para el *AND* lógico, **||** para el *OR*, etc.

En **gnuplot** pueden definirse constantes. Por ejemplo

- **c = 2.9979e+8** asigna a la constante **c** la velocidad de la luz en ms^{-1} .
- **print c** imprime el valor que hemos asignado a **c**.

Ahora supongamos que queremos dibujar la función gaussiana:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \quad (1)$$

Primero definimos las constantes μ y σ (el valor medio y la varianza de la función gaussiana). Por ejemplo:

- **mu = 10.98765**
- **sigma = 6.43321**

A continuación debemos teclear:

```
plot 1/(sqrt(2*pi)*sigma)*exp(-(x-mu)**2/(s*sigma**2))
```

Es evidente que teclear una larga función como la anterior resulta tedioso, sobre todo si hay que repetirla muchas veces. **gnuplot**, sin embargo, permite definir una expresión con esta:

```
ga(x,mu,sigma)= 1/(sqrt(2*pi)*sigma)*exp(-(x-mu)**2/(2*sigma**2))
```

A continuación uno puede utilizar la expresión en conjunción con el comando **plot**. Por ejemplo:

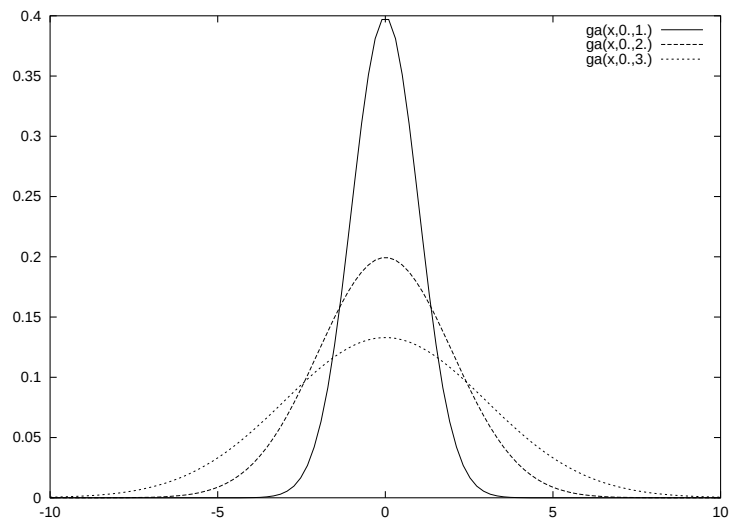


Figure 2: Ejemplo 2.

plot ga(x,0.,1.),ga(x,0.,2.),ga(x,0.,3.)

Que nos dibuja, en el mismo gráfico tres gaussianas, centradas en cero, de anchuras 1,2 y 3, como se muestra en la figura 2

Es importante recordar que **gnuplot** trata a los números sin punto decimal como enteros a los que aplica la aritmética propia de números enteros en lenguajes como C o Fortran (por ejemplo, $1./4. = 0.25$ pero $1/4 = 0$).

gnuplot permite definir funciones muy sofisticadas. Por ejemplo, la función:

$$f(x) = \begin{cases} x(1-x) & x < -1 \\ x-1 & -1 \leq x \leq 4 \\ \sqrt{x} & x > 4. \end{cases} \quad (2)$$

Puede definirse así:

- **f1(x)=(x;-1) ? x*(1-x) : x-1**
- **f2(x)=(x;4) ? f1(x) : sqrt(x)**

La definición anterior está basada en los operadores ternarios de C, que son de la forma

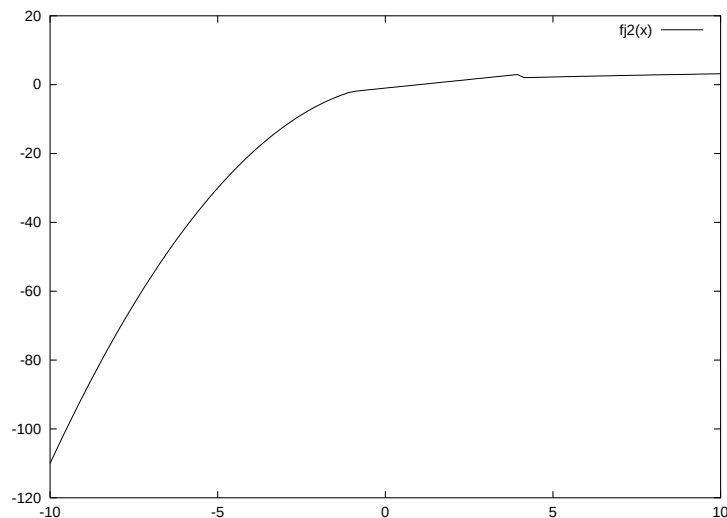


Figure 3: Ejemplo 3.

(condición) ? expresión 1 : expresión 2.

Cuyo significado es: “Si *(condición)* es cierta, evalúa *expresión 1* en otro caso evalúa *expresión 2*”.

A continuación, el comando:

plot[-10:10] f2(x)

Nos dibuja, la función, empalmando tan suavemente como puede en los puntos en los que la función cambia como se muestra en la figura 3

5 Scripts

gnuplot permite colocar un conjunto de comandos que se quiera utilizar muchas veces en un fichero de comandos o *script file*. Ese fichero de comandos puede cargarse, en modo interactivo, con el comando **load**. Alternativamente, se puede ejecutar el script en *modo batch*. Por ejemplo, para ejecutar el script *miscrypt.gp* en modo batch, tecleamos el comando:

gnuplot miscrypt.gp

Este comando llama a **gnuplot**. Cuando terminar de ejecutar, sale y devuelve control. Este es un truco útil, por ejemplo para mandar gráficos a un fichero gráfico en lugar de visualizarlos en la pantalla.

Algunos caracteres especiales en el lenguaje de script de **gnuplot** son el **#**, que se utiliza como signo de comentario, y el signo

que es el signo de continuación. Múltiples comandos pueden situarse en una misma línea a condición de separarlos por un punto y coma ;.

El comando *save* permite guardar en un fichero las variables, funciones y valores especiales que se hayan definido durante la sesión. Por ejemplo el comando:

save 'miscosas.gp'

Salva en el fichero *miscosas.gp* el contenido de las variables, funciones y valores especiales definidos en la sesión.

Si queremos salvar tan sólo las funciones que hemos definido:

save function 'miscosas.gp'

Igualmente para las variables (*save variable*) y los valores especiales *save set*).

Los ficheros *ej1.gp*, *ej2.gp*, *ej3.gp* y *ej4.gp* que se han utilizado para generar los dibujos de este manual son ejemplos sencillos de uso del script.

6 Gráficos a la Carta

- **set logscale x**. Muestra el eje X en escala logarítmica.
- **set logscale y**. Muestra el eje Y en escala
- **set nologscale**. Vuelve a la escala lineal.
- **set zeroaxis**. Dibuja los ejes.
- **set nozeroaxis**. Elimina los ejes.
- **set xzeroaxis,yzeroaxis,noxzeroaxis,noyzeroaxis**. Similar, para cada eje por separado.
- **set key x,y** . Especifica donde poner la leyenda del gráfico (en el punto x,y del gráfico).
- **set label x,y** . Especifica donde poner texto adicional en el gráfico:
Por ejemplo:

set label 1 'Gráfico' identifica el texto “Gráfico” con “label 1”, que ahora podemos situar en los puntos, digamos, 1.5 y 4 del texto, con el comando:

set label 1 1.5,4

El ejemplo en la figura 4 es la función normal bivariada (es decir una función de gauss, con media cero y varianza uno, en dos dimensiones).

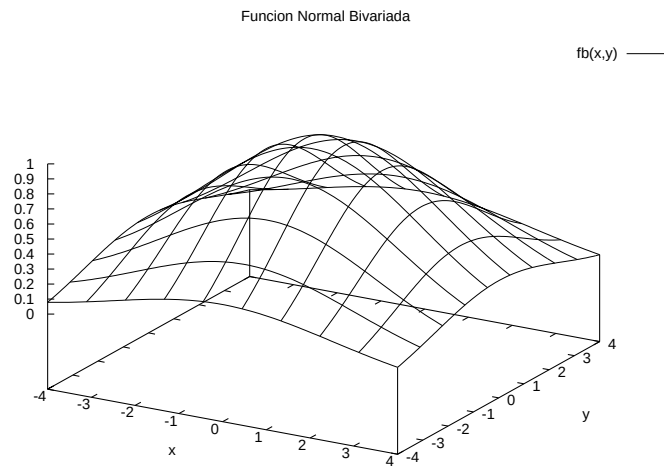


Figure 4: Ejemplo 4.

Consultar el script *ej4.gp*, donde se ha definido la función y se han modificado varios parámetros.

La Tabla 6 da una lista de algunas de las funciones más importantes definidas en **gnuplot** .

Función	Argumento	Devuelve
<i>abs(x)</i>	Cualquiera	Valor absoluto de x
<i>abs(x)</i>	Complejo	Longitud de x, $x = \sqrt{real(x)^2 + im(x)^2}$
<i>acos(x)</i>	Cualquiera	Arco coseno de x en radianes
<i>arg(x)</i>	Complejo	La fase de x en radianes
<i>asin(x)</i>	Cualquiera	Arco seno de x en radianes
<i>atan(x)</i>	Cualquiera	Arco tangente de x en radianes
<i>atan(x)</i>	Cualquiera	Arco tangente de x en radianes
<i>besj0(x)</i>	Radianes	Función de Bessel $j_0(x)$
<i>besj1(x)</i>	Radianes	Función de Bessel $j_1(x)$
<i>cos(x)</i>	Radianes	coseno de x
<i>cosh(x)</i>	Radianes	coseno hiperbólico de x
<i>erf(x)</i>	Cualquiera	Función error real $Erf(x)$
<i>exp(x)</i>	Cualquiera	Función exponencial e^x
<i>imag(x)</i>	Complejo	Parte imaginaria de x como un real
<i>int(x)</i>	Real	Parte entera de x como un real
<i>log(x)</i>	Cualquiera	Logaritmo Neperiano de x
<i>log10(x)</i>	Cualquiera	Logaritmo en base 10 de x
<i>rand(x)</i>	Cualquiera	Generador de números aleatorios
<i>real(x)</i>	Cualquiera	La parte real de x
<i>sgn(x)</i>	Cualquiera	1 si $x > 0$, -1 si $x < 0$
<i>sin(x)</i>	Radianes	seno de x
<i>sinh(x)</i>	Radianes	seno hiperbólico de x
<i>sqrt(x)</i>	Cualquiera	$\sqrt{x}$
<i>tan(x)</i>	Radianes	tangente de x
<i>tanh(x)</i>	Radianes	tangente hiperbólica de x

Table 1: Algunas funciones intrínsecas de **gnuplot** .