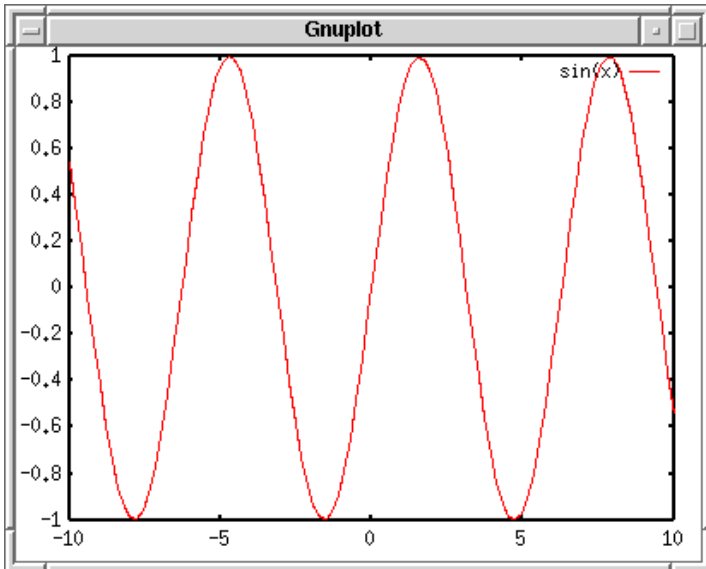


TUTORIAL DE GNUPLOT

Gnuplot es un software libre para crear gráficas de todo tipo. Se muestra a continuación algunas funcionalidades básicas. Se puede utilizar de forma interactiva en la línea de comando o programada mediante scripts.

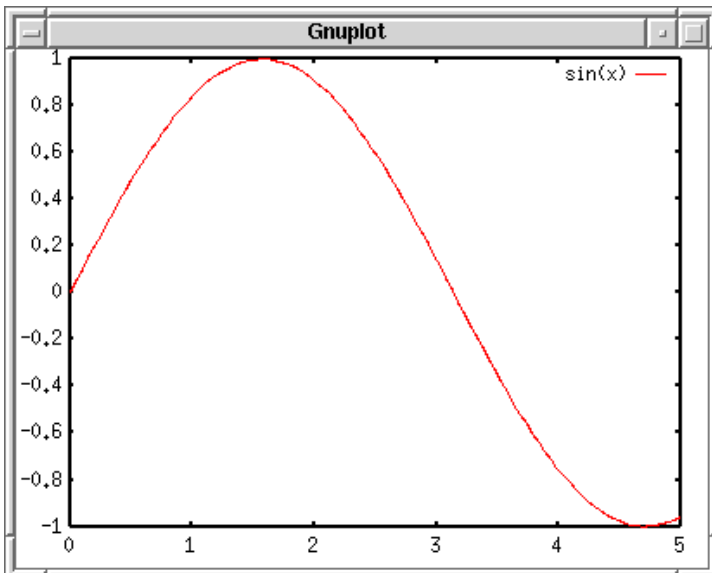
Para ingresar al modo interactivo en la línea de comandos ejecutamos el comando *gnuplot*:

```
gnuplot> plot sin(x)
```



Podemos setear varias cosas como el rango de ploteo:

```
gnuplot> plot [0:5] sin(x)
```



Es posible obtener información de un comando de gnuplot mediante el comando *help*:

```
gnuplot> help set
```

The `set` command can be used to sets *lots* of options. No screen is drawn, however, until a `plot`, `splot`, or `replot` command is given.

The `show` command shows their settings; `show all` shows all the settings.

If a variable contains time/date data, `show` will display it according to the format currently defined by `set timefmt`, even if that was not in effect when the variable was initially defined.

Subtopics available for set:

angles	arrow	autoscale	bar
bmargin	border	boxwidth	clabel
clip	cntrparam	contour	data
dgrid3d	dummy	encoding	format
.....			
zero	zeroaxis	zlabel	zmtics
zrange	ztics		

GRAFICANDO COMANDOS EN 2D

Si tenemos un conjunto de datos (experimentales, salida de un programa, etc.) cuyo formato es de dos columnas de números (X,Y) guardados en un archivo podemos graficarlos:

```
# X Y      Archivo gnuplot.dat
2.1500E-02 1.3060E+00
2.3900E-02 1.2220E+00
2.6800E-02 1.3430E+00
2.9700E-02 1.2580E+00
3.2500E-02 1.2430E+00
```

```
> plot "gnuplot.dat" using 1:2
```

USANDO GNUPLLOT DESDE UN SCRIPT

Es posible automatizar el uso de los comandos de gnuplot mediante un script, por ejemplo:

```
#!/usr/local/bin/gnuplot -persist

# Si se quiere que la salida sea a un archivo postscript:
# set output 'output.ps'
plot "output.dat" using 1:2
# EOF
```

Luego se ejecuta este script llamado `gnuplot.plt` como cualquier script de shell:

```
./gnuplot.plt
```

