

Lo que he aprendido sobre Gnuplot

CARLOS ANDRES RODALLEGA MILLAN

26 de noviembre de 2021

Este documento resumirá todos los conceptos aprendidos en clase sobre el programa de Gnuplot, como se puede apreciar estará dividido en cada sección con las clases en las cuales se aprendió lo que se esta escribiendo.

1. Clase 2-16Nov

El código que se escribió en Atom y se puso en el programa fue el siguiente:

```
#Esto es un comentario en GNUPlot
#Tenemos que decirle que tipo de grafica vamos a realizar (png,jpg, etc), set significa use
set terminal pngcairo #El tipo de gráfico
set output "grafica_16Nov.png" #Nobmre de la gráfica.
#Instrucciones de la gráfica
set xlabel "Eje horizontal"
set ylabel "Eje vertical"
#para pintar se usa el comando plot
plot cos(x)
unset output #Cerrar archivo
```

El resultado de este código fue el siguiente:

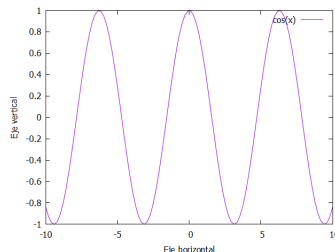


Figura 1: Es un archivo .png que contiene la imagen de la función coseno

1.1. Explicando paso a paso el código anterior

Lo primero debemos entender que para hacer **comentarios** usamos **#**, esta herramienta nos ayuda muchísimo para guiarnos en nuestro código.

Debemos especificarle al programa el tipo de gráfica que vamos a realizar, por eso usamos el siguiente comando, en este comando le decimos a Gnuplot que nos cree un archivo .png con el nombre **"grafica16Nov"**:

```
set terminal pngcairo
set output "grafica_16Nov.png"
```

Ahora le debemos dar las instrucciones de los ejes a nuestro programa, etiquetando los ejes de la siguiente forma:

```
set xlabel "Eje horizontal"
set ylabel "Eje vertical"
```

Después, debemos decirle al programa la orden a realizar, es decir, que haga la gráfica.

```
plot cos(x)
```

Por último, siempre debemos cerrar nuestro programa.

```
unset output
```

2. Clase 3-23Nov

2.0.1. Primera gráfica de esta clase

Tengamos en cuenta que aquí en adelante se ignora los comentarios del texto, estos serán puestos en la explicación de paso a paso del código anterior. Presentamos directamente le archivo que se usó esta clase:

```
set terminal pngcairo font "times,26" enhanced color size 900,600
set output "Grafica26Nov.png"
set xrange[-5:5]
set yrange[0:1.2]
set xlabel "x"
set ylabel "P(X)"
set key top right
P1(x)=exp(-x**2)
P2(x)=exp(-(0.7*x)**2)
plot P1(x) w lp ps 2 pt 7 lc 4 t "P_1(X)", P2(x) w l lw 4 t "P_2(X)"
unset output
```

El resultado del anterior código es el siguiente:

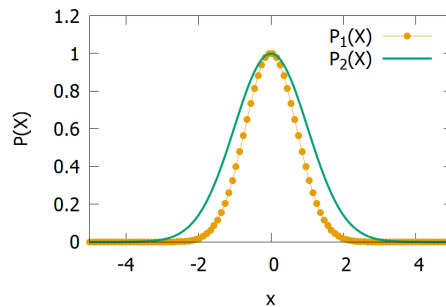


Figura 2: Es un archivo .png que contiene la imagen de ambas funciones Gaussianas

Gnuplot toma x,y,z como los nombres de las variables independientes para cartesianas. (r,t son los nombres de las variables independientes para polares.) Podemos apreciar que en nuestra anterior gráfica tenemos un estilo y color para cada línea.

2.1. Ahora vamos a explicar paso a paso el código anterior.

```
set terminal pngcairo font "times,26" enhanced color size 900,600
set output "Grafica26Nov.png"
```

Como habíamos visto antes, esto nos permite crear y empezar nuestro grafico, con esto, estamos haciendo que lo cree en un archivo .png. Después le estamos especificando el tipo de fuente a usar, el tamaño de fuente y las dimensiones del archivo a realizar.

- Rango

```
set xrange[-5:5]
set yrange[0:1.2]
set xlabel "x"
set ylabel "P(X)"
```

Si no especificamos el rango, este vendrá con valores por defecto de $[-10,10]$ para cada eje coordinado.

- Etiquetas de rangos.

```
set xrange[-5:5]
set yrange[0:1.2]
set xlabel "x"
set ylabel "P(X)"
```

Los anteriores comandos, nos permiten asignarle un nombre a cada eje coordinado.

- Forma de la línea de nuestra gráfica y graficación.

```
plot P1(x) w lp ps 2 pt 7 lc 4 t "P_1(X)", P2(x) w l lw 4 t "P_2(X)"
```

En esta línea de códigos le estamos diciendo a Gnuplot que haga las gráficas que están separadas por comas (,) y que las haga de con línea y puntos (**lp**), después especificamos el tamaño de cada punto ("**pt7**"), el color "line color"(**lc 4**) y por último le decimos que le asigne un nombre que se ubica arriba a la derecha **t "P(x₁)"**.

lw ancho de la línea.

1 morado, 2 verde, 3 azul clarito, 4 naranja.

Por último, como hemos venido haciendo, cerramos nuestra gráfica con :

```
unset output
```

2.2. Segunda gráfica de esta clase "Panal de huevos"

El archivo usado fue:

```
#Grafica en 3d
#Todas las instrucciones que queremos darle a GNUPLOT, las colocaremos siempre en un archivo aparte.
#Eso es un script.
#Enhanced, es para usar subindices o superindices, símbolos.
#Antes de comenzar, colocar/limpiar nuestro espacio. Reset elimina la información en el buffer.
#-----Empieza la línea de código-----
reset

set terminal pngcairo font "Times,26" enhanced color size 900,600

set output "Grafica26_2Nov_3D.png"
set pm3d
set view map
#forma de vista de mapa

#Panal de huevos.
set xlabel "eje x"
set ylabel "eje y"
set xrange[-10:10]
set yrange[-10:10]
```

```

f(x,y)=cos(x)*sin(y)
set isosample 500
set ztics 0.5
set cbtics 0.5
#Espacio de las letras de los números del eje z
#Lo anterior organiza la vista.
#isosample es una grilla, cantidad de línea, 50 líneas por variable inde.
unset surface
splot f(x,y) t" "

#Pareja pm3d, para pintarlo, aparecen colores y la escala.
#Unset surface, para quitar la malla
#x al cuadrado
unset output

```

El grafico realizado fue el siguiente:

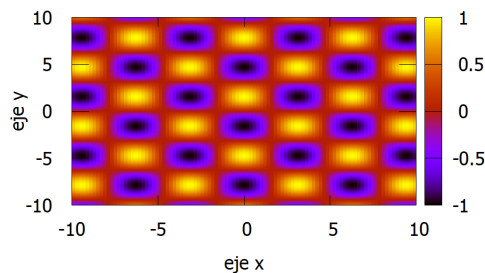


Figura 3: Esta fue la figura realizada en 3d con el anterior código, vista desde arriba.

Texto por si acaso

2.3. Ahora vamos a explicar paso a paso el código anterior.

3. Clase Revelo

El siguiente archivo es interesante porque se puede crear diferentes gráficas en una misma gráfica. Presentamos directamente le archivo que Revelo nos pasó para esta clase:

```

set terminal pngcairo transparent enhanced font "Times-New-Roman,12"
set output "grafica1_clase4.png"

set xrange[0:2*pi]
set yrange[-1.1:1.1]
set xlabel "Angulo ({/Symbol Q})"
set ylabel "Eje vertical"
set key bottom left
plot sin(x)

unset output
reset

#grafica 3d
set terminal pngcairo transparent enhanced font "Times-New-Roman,14"
set output "grafica3D_clase4.png"

```

set multiplot	set isosamples 150	
	set xrange[-0.5*pi:0.5*pi]	reset
set origin 0,0	set yrange[0:4*pi]	
set size 0.5,0.5	splot sin(x)*cos(y)	set origin 0.5,0.5
set pm3d map		set size 0.5,0.5
set hidden3d	reset	set pm3d at bs
set isosamples 150		unset surface
set xrange[0:2*pi]	set origin 0,0.5	set isosamples 150
set yrange[-2*pi:2*pi]	set size 0.5,0.5	set xrange[0:2*pi]
set zrange[-1:1]	set pm3d	set yrange[-2*pi:2*pi]
set cbrange[-1:1]	unset surface	set ztics 0.5
splot sin(x)*cos(y)	set isosamples 150	set cbticks 0.5
	set xrange[0:2*pi]	unset key
reset	set yrange[-2*pi:2*pi]	splot sin(x)*cos(y)
	set ztics 0.5	
set origin 0.5,0	set cbticks 0.5	unset multiplot
set size 0.5,0.5	unset key	
set hidden3d	splot sin(x)*cos(y)	unser output

2

3.1. Ahora vamos a explicar paso a paso el código anterior.

para windows: powershell en la powershell wgnuplot Clase4gnuplot.plt

4. Libro que dejó la profesora