

Edición de Gráficas, Figuras e Imágenes en Gnuplot, Xfig e ImageMagick y su Inclusión en un Documento L^AT_EX

Luis Gerardo de la Fraga

Cinvestav, Departamento de Computación.
Av. IPN 2508. 07360 México, D.F., México.

Correo-e: fraga@cs.cinvestav.mx

<http://cs.cinvestav.mx/~fraga>

2 de julio de 2008

Resumen

En este documento se explicará muy brevemente como realizar gráficas con `gnuplot`, figuras con `Xfig` y edición de imágenes con `ImageMagick`. Todos son herramientas que se utilizan desde la línea de comandos usando scripts. Se pondrá énfasis en como incluir ecuaciones matemáticas.

1. Haciendo gráficas con Gnuplot

Gnuplot es un programa para dibujar funciones y datos, es portable y funciona de forma interactiva en la línea de comandos. El sitio de gnuplot es <http://gnuplot.sourceforge.net>.

Lo que vamos a aprender aquí es:

1. Cómo realizar gráficas de funciones
2. Cómo guardar una gráfica en EPS para incluirla en un documento L^AT_EX
3. Cómo incluir expresiones matemáticas en una gráfica hecha con gnuplot: en L^AT_EX con `psfrag` y exportando la gráfica en `ps/tex`

1.1. Gráficas de funciones

Es muy fácil realizar una gráfica con `gnuplot`. En una terminal damos `gnuplot` y en el prompt interactivo damos `plot sin(x)` y se desplegará (por defecto en una ventana de X Windows) la gráfica de la función $y = \sin(x)$.

Es preferible trabajar con scripts (deja un archivo que se almacena en disco duro), que trabajar en la línea de comandos. Para realizar la gráfica de la función $y = \sin(x)$ en el intervalo $[0, \pi]$, editamos un archivo que llamaremos `sen.gnu`, con el contenido:

```
set xrange [0:3.1416]
plot sin(x)
pause -1
```

Se realizó la gráfica dando `gnuplot sen.gnu`. El `pause -1` sirve para que se despliegue la gráfica hasta que se dé enter en la línea de comandos.

1.2. Gráficas en EPS

El script para generar la gráfica de la parábola $y = x^2$ en el intervalo de $[-10, 10]$ sería el siguiente:

```
set term post eps "Times" 20
set output 'par.eps'
set xrange [-10:10]
set nokey
```

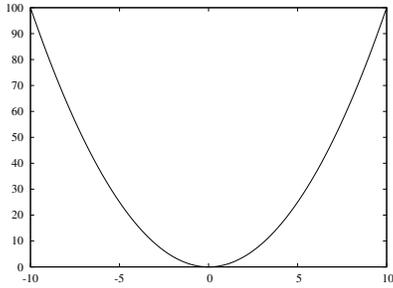


Figura 1: Gráfica de la función $y = x^2$

```
# Y la función:
y(x) = x*x

plot y(x)
```

La primera línea del script habilita la salida a postscript encapsulado, y la segunda línea indica el nombre del archivo de salida. La gráfica de salida se incluye en la fig. 1. Como se vé en el script, no es necesario interrumpir la salida con *pause -1* ya que la salida es el archivo *par.eps*.

Todas la opciones que permite la terminal (como se llama a la salida en *gnuplot*) se obtiene dando *gnuplot* y *help set post*. Todas la posibilidades de una terminal, se logra con el comando *test*. Por ejemplo, para la terminal postscript sus posibilidades se ven en la fig. 2

1.3. Un ejemplo más elaborado

El siguiente script realiza la gráfica una elipse y una parábola. La elipse está rotada 45° y se ponen dos etiquetas para identificar cada curva. La elipse está trazada de forma paramétrica, por lo que debe de usarse el comando *multiplot*. La gráfica de salida se observa en la fig. 3

```
set term post eps "Times"
set output "dosf.eps"
# Habilita poner acentos:
set encoding iso_8859_1
set multiplot
```

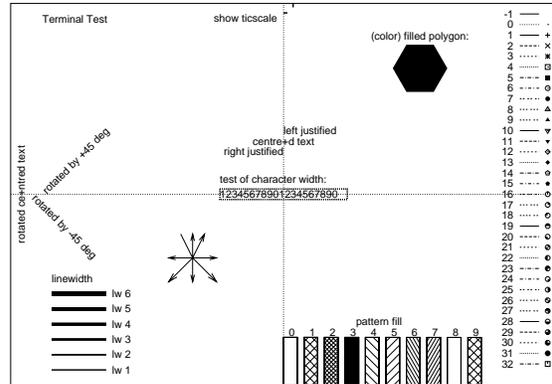


Figura 2: Posibilidades de la terminal postscript

```
set nokey
set size ratio -1 0.6,0.6

set xrange [-10:10]
set yrange [-10:10]
set xlabel "Abscisas"
set ylabel "Ordenadas"

# Graficamos la parábola
f(x) = x*x
set label "Parábola" at -8,8
# set label "a" at -6,7
plot f(x)

# Gráfica de una elipse en
# forma paramétrica
a = 10
b = 5
x(t) = a*cos(t)
y(t) = b*sin(t)
ang=45*3.1416/180
s = sin(ang)
c = cos(ang)
xp(t) = x(t)*c - y(t)*s
yp(t) = x(t)*s + y(t)*c
set parametric
set label "Elipse" at -8,3
plot xp(t), yp(t) w l lt 3
set nomultiplot
```

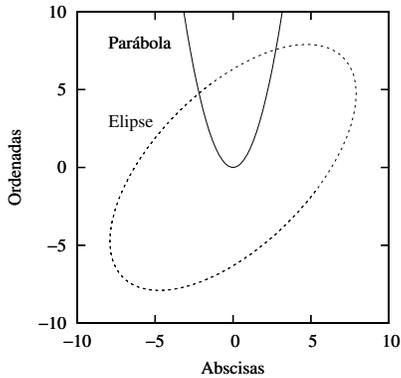


Figura 3: Gráfica de una elipse y una parábola

Una forma rápida de verificar la gráfica es con ghostview, para este caso daríamos `gv dosf.eps`.

En el archivo `dosf.eps` se modificó el `BoundingBox` con las coordenadas medidas en ghostview. Se cambió a `%%BoundingBox: 74 50 240 200`.

1.4. Inclusión de expresiones matemáticas

Existe dos formas de incluir expresiones matemáticas en una gráfica hecha con gnuplot. La primera es usando el paquete `psfrag` y la segunda es exportando la gráfica en `ps/tex`.

Para usar `psfrag` hay que poner una etiqueta única en la gráfica y sustituirla con `psgraf`. Por ejemplo, en la gráfica realizada en la sección 1.3. Descomentamos la línea:

```
# set label "a" at -6,7
```

y al incluir la gráfica realizamos:

```
\begin{figure}
\centering
\psfrag{a} [] [] {$y = x^2$}
\includegraphics[width=0.7\columnwidth]{dosf1.eps}
\caption{Gráfica de una elipse y una parábola}
\label{fig:3}
\end{figure}
```

La gráfica queda como se observa en la fig. 4.

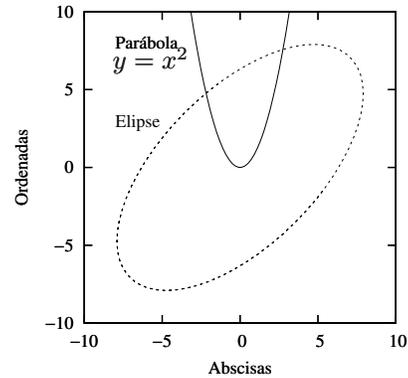


Figura 4: La gráfica de la fig. 3 con la ecuación de la parábola incluida con `psfrag`

Este modo tiene la desventaja que solo funciona para generar un documento postscript, que puede convertirse en PDF con el comando `ps2pdf`.

Voy a mostrar como exportar una figura en `ps/tex` con un ejemplo. El siguiente script genera una circulo e incluye la ecuación del mismo:

La gráfica se debe incluir con una línea como:

```
\scalebox{0.6}{\input{Figs/circulo.tex}}
```

La gráfica generada la podemos ver en la fig. 5

Un comentario: este último modo permite que se usen los tipos de letra de \LaTeX , lo que genera un documento con los requerimientos del Xplore para publicarse en la IEEE.

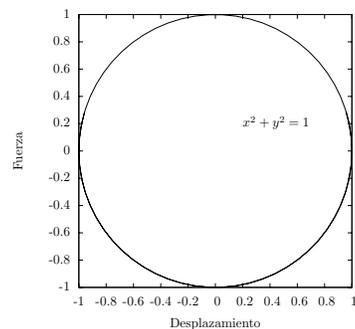


Figura 5: La gráfica de un círculo con su ecuación incluida

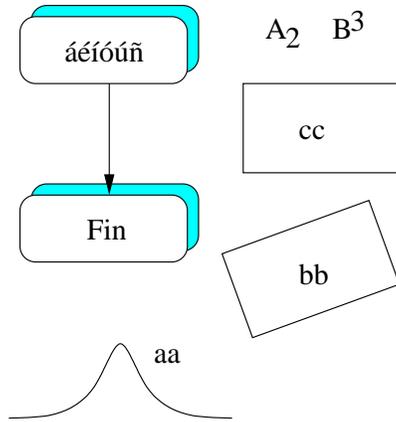


Figura 6: Una figura hecha con Xfig

2. Figuras con Xfig

Xfig es un programa gráfico para la realización de dibujos. Viene incluido en cualquier distribución de GNU/Linux ó puede descargarse desde cualquier máquina espejo del CTAN (por ejemplo, <ftp://ftp.tex.ac.uk/pub/archive/graphics>).

La figura 6 fue realizada con Xfig, exportada en EPS e incluida en este archivo. Xfig es muy intuitivo de usar. Una de sus ventajas es que la profundidad de cada primitiva de dibujo se controla con un número de profundidad (por defecto es 50) y puede visualizarse cada capa en dibujos complicados.

Una consideración importante: el dibujo que realizamos debe tener el tamaño lo más cercano que se pueda a como quedará sobre el papel. Para esto recomiendo que se use un acercamiento de 2, de esta forma el área de dibujo será de $10 \times 7 \text{ cm}^2$. Para que cada vez que se ejecute Xfig se seleccione de forma automática el acercamiento de 2 y la rejilla más fina, se edita el archivo `.Xdefaults` que está en el directorio de casa con el contenido:

```
Fig*zoom: 2.0
Fig*startgridmode: 1
```

Para la inclusión de ecuaciones en la figura, es preferible usar los tipos de letra de \LaTeX . Para poder realizarlo hay que

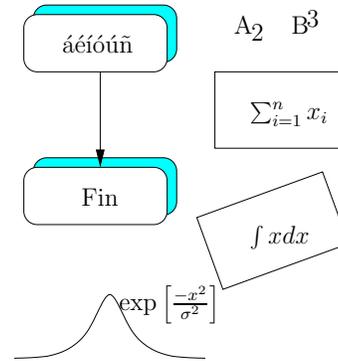


Figura 7: Una figura hecha con Xfig con ecuaciones en \LaTeX

1. Seleccionar el bandera de ‘especial’ para escribir texto
2. Seleccionar los tipos de letra de \LaTeX
3. Exportar en el formato “PS/LaTeX combinado”

EL archivo generado se incluye como se explicó al final de la sección 1.4. Las ecuaciones que se incluyen en la fig. 7 fueron editadas directamente en el fuente del archivo `*.pstex_t` donde estaban las etiquetas de ‘aa’, ‘bb’ y ‘cc’ de la fig. 6. De esta forma también se genera un archivo que cumple con las reglas del Xplore de la IEEE.

3. Edición de texto en imágenes

Lo más común es incorporar una etiqueta como ‘(a)’ ó ‘(b)’ en la imagen (lo que ahorra espacio si se pone esta etiqueta con texto fuera de la imagen). Esto lo podemos realizar con los programas de la suite de ImageMagick (en www.imagemagick.org).

La imagen de la fig. 8 fue creada con el siguiente script:

```
P1="488,107"
P2="487,208"
P3="323,197"
P4="324,97"
```

```
P5="638,449"
P6="298,420"
P7="701,368"
P8="632,647"
```

```
P9="486,309"
P10="323,297"
```

```
convert a.jpg -fill none -stroke black -strokewidth 4 \
  -draw "polyline $P1 $P2 $P3 $P4 $P1" \
  -draw "line $P5 $P6 line $P5 $P7 line $P5 $P8" \
  -strokewidth 1 -fill black -font Times-Bold -pointsize 60 \
  -draw 'text 30,723 "(a)">' \
  -stroke black \
  -draw "circle $P9 486,314" \
  -draw "circle $P10 323,302" b.jpg
```

No es aconsejable convertir la imágenes a EPS debido a que su tamaño se incrementa considerablemente. Las coordenadas fueron medidas con el programa `display`. Para incluir la imagen en el documento \LaTeX se puso en el preámbulo: `\DeclareGraphicsRule{.jpg}{eps}{.bb}{'convert #1` Se generó un archivo `*.bb` donde \LaTeX lee el tamaño de la imagen como si estuviese en el formato EPS con el comando `ebb` del paquete `dvipdfm`. Se editó este archivo cambiando los límites del `BoundingBox` al tamaño de la imagen en pixels. Y para generar el archivo `*.ps` se hizo

```
dvips -R0 -o edicion.ps edicion
```

Es documento en PDF se generó a partir del postscript como:

```
ps2pdf edicion.ps edicion.pdf
```

Además, todas las figuras se pusieron en un directorio `Figs` y se puso en el preámbulo:

```
\graphicspath{{Figs/}}
```

Por último, no olviden checar la sintaxis de las palabras del documento con el programa `aspell`.

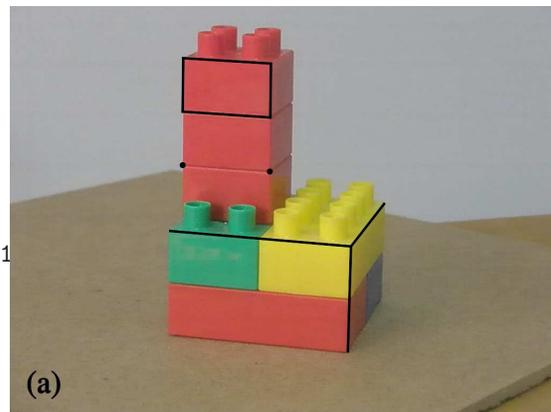


Figura 8: Una imagen con texto y dibujos.