

Herramientas: Formatos gráficos Postscript y SVG. Parte I.

Hay situaciones en las que es necesario realizar dibujos de ciertos resultados matemáticos. En muchos casos es deseable realizar dibujos de forma automática. Es decir, poder escribir un programa que realice el dibujo por nosotros. Existen un par de formatos gráficos que permiten realizar dichos dibujos de forma sencilla y exportar el resultado a otros formatos como el pdf.

El formato Postscript.

Es el formato más veterano y es el usado por muchas impresoras láser. Consiste en un simple fichero de texto (se puede editar con gedit o kwrite) con la extensión ps o eps.

Dibujando líneas con Postscript.

Un ejemplo (más adelante se comenta este código):

```
!PS-Adobe-3.0 EPSF-3.0
%%Creator: Yo
%%Title: Ejemplo1
%%CreationDate: 2008-01-03
%%DocumentData: Clean7Bit
%%Origin: 0 0
%%BoundingBox: 0 0 200 200
%%LanguageLevel: 2
%%Pages: 1
%%Page: 1 1

newpath
100 200 moveto
200 150 lineto
100 0 lineto
100 200 lineto
2 setlinewidth
stroke

%%EOF
```

Si lo salvamos como ejemplo1.eps y lo abrimos con kghostview o gimp tendremos:



El código anterior genera un simple triángulo.

Usando **gimp** se puede exportar a otro formato como el png e insertarlo en un documento de texto.

Vamos a comentar el código:

Esta es la cabecera se pueden indicar algunos datos sobre el documento como el creador, el título o la fecha de creación:

```
!PS-Adobe-3.0 EPSF-3.0
%%Creator: Yo
%%Title: Ejemplo1
%%CreationDate: 2008-01-03
```

%%DocumentData: Clean7Bit

Situación del origen de coordenadas, es recomendable el (0,0):

%%Origin: 0 0

El tamaño de la imagen en puntos, en este caso 200x200:

%%BoundingBox: 0 0 200 200

Se hace el dibujo:

Con **newpath**, en este caso, se indica que se va a crear un conjunto de segmentos uno a continuación de otro, una *polilínea*:

newpath

Posición de la primera coordenada del primer segmento, en este caso (100,200), se especifica con

moveto:

100 200 moveto

Posición del otro extremo del primer segmento (200,150). Se indica con **lineto**:

200 150 lineto

Final del segundo segmento (100,0). A partir de ahora se usa el final de un segmento como principio del siguiente:

100 0 lineto

100 200 lineto

Grosor de la línea:

2 setlinewidth

Se dibuja la línea.

stroke

Se podría insertar otro **newpath** para dibujar otras líneas.

El fin del dibujo se indica con **%%EOF**.

%%EOF

Inserción de texto.

Se pueden insertar textos, por ejemplo, para insertar un texto en el dibujo anterior:

%!PS-Adobe-3.0 EPSF-3.0

%%Creator: Yo

%%Title: Ejemplo1

%%CreationDate: 2008-01-03

%%DocumentData: Clean7Bit

%%Origin: 0 0

%%BoundingBox: 0 0 200 200

%%LanguageLevel: 2

newpath

100 200 moveto

200 150 lineto

100 0 lineto

100 200 lineto

2 setlinewidth

stroke

%Se elije el tipo de letra Times-Roman:

/Times-Roman findfont

%El tamaño del tipo de letra. En este caso 12

12 scalefont

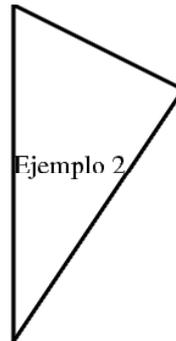
```

%Se aplica Times 12 como tipo de letra a usar
setfont
%Se prepara para insertar el texto
newpath
%El texto se insertará en las coordenadas (100,100)
100 100 moveto
%Se inserta el texto Ejemplo 2
(Ejemplo 2) show

%%EOF

```

Se obtiene:



Un ejemplo de Postscript desde Octave.

Usando Octave, y con lo que se ha visto de Postscript hasta ahora, se pueden realizar dibujos más complicados de forma sencilla. Por ejemplo, con el siguiente código en Octave:

```

function ejemplo3()

    out=fopen("ejemplo3.eps", "w");

    fprintf(out, "%%%!PS-Adobe-3.0 EPSF-3.0\n");
    fprintf(out, "%%%%%Creator: Yo\n");
    fprintf(out, "%%%%%Title: Ejemplo3\n");
    fprintf(out, "%%%%%CreationDate: 2008-01-03\n");
    fprintf(out, "%%%%%DocumentData: Clean7Bi\n");
    fprintf(out, "%%%%%Origin: 0 0\n");
    fprintf(out, "%%%%%BoundingBox: 0 0 200 200\n");
    fprintf(out, "%%%%%LanguageLevel: 2\n");

    fprintf(out, "newpath\n 100 200 moveto\n");
    N=5;
    for x=[1:N]
        fprintf(out, "%f %f lineto\n", 100+100*sin(x*2*pi/N), 100+100*cos(x*2*pi/N) );
    endfor

    fprintf(out, "2 setlinewidth\nstroke\n%%EOF\n");

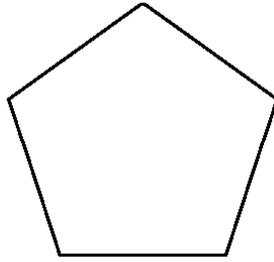
    fclose(out);

    disp("Dibujado\n");

endfunction

```

Se obtiene el dibujo de un pentágono:



Cambiando la línea $N=5$, por $N=6$, $N=7$,... se pueden dibujar hexágonos, heptágonos,...

Al final se pueden encontrar ejemplos más complicados usando Octave.

Usando gimp o ps2pdf podemos exportar el dibujo a otros formatos como pdf o jpeg.

Dibujo de arcos y círculos.

Para dibujar un arco se usaría la orden `arc` con la siguiente sintaxis:

`x y radio ángulo_inicio ángulo_final arc`

Por ejemplo:

```
%!PS-Adobe-3.0 EPSF-3.0
%%Creator: Yo
%%Title: Ejemplo4
%%CreationDate: 2008-01-03
%%DocumentData: Clean7Bit
%%Origin: 0 0
%%BoundingBox: 0 0 200 200
%%LanguageLevel: 2

%Se indica que se va a dibujar un nuevo camino
newpath
%Se dibuja un arco con centro (100,100) y radio 50. Comienza en 0° y acaba en 270°
100 100 50 0 270 arc
%Se selecciona un espesor de 2 para el arco
2 setlinewidth
%Se dibuja
stroke
%%EOF
```

Se obtiene la siguiente figura:



Figuras rellenas.

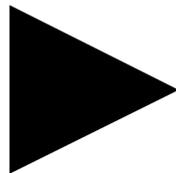
Se usará fill para rellenar las figuras. Por ejemplo:

```
%!PS-Adobe-3.0 EPSF-3.0
%%Creator: Yo
%%Title: Ejemplo5
%%CreationDate: 2008-01-03
%%DocumentData: Clean7Bit
%%Origin: 0 0
%%BoundingBox: 0 0 200 200
%%LanguageLevel: 2
```

```
newpath
5 195 moveto
195 100 lineto
5 5 lineto
fill
stroke
```

```
%%EOF
```

Dibujará:



Grises y colores.

En el ejemplo anterior se ha dibujado un triángulo negro. Se puede dibujar en gris usando setgray. Su sintaxis es:

intensidad **setgray**

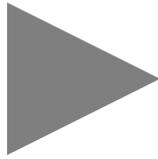
La intensidad es un número de 0 a 1. Así por ejemplo:

```
%!PS-Adobe-3.0 EPSF-3.0
%%Creator: Yo
%%Title: Ejemplo6
%%CreationDate: 2008-01-03
%%DocumentData: Clean7Bit
%%Origin: 0 0
%%BoundingBox: 0 0 200 200
%%LanguageLevel: 2
```

```
newpath
5 195 moveto
195 100 lineto
5 5 lineto
0.5 setgray
fill
stroke
```

```
%%EOF
```

Dibujará:



Una vez establecido un nivel de gris el resto de figuras se dibujan con dicho nivel de gris, por lo que es buena idea reestablecer el nivel de gris antes de seguir dibujando. Por ejemplo:

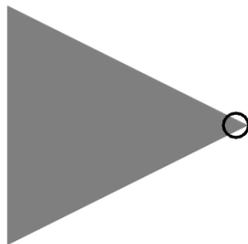
```
%!PS-Adobe-3.0 EPSF-3.0  
%%Creator: Yo  
%%Title: Ejemplo7  
%%CreationDate: 2008-01-03  
%%DocumentData: Clean7Bit  
%%Origin: 0 0  
%%BoundingBox: 0 0 200 200  
%%LanguageLevel: 2
```

```
newpath  
5 195 moveto  
195 100 lineto  
5 5 lineto  
0.5 setgray  
fill  
stroke
```

```
newpath  
0 setgray  
185 100 10 0 360 arc  
2 setlinewidth  
stroke
```

```
%%EOF
```

Dibujará:



Los colores se manejan de forma parecida usando `setrgbcolor`, que admite 3 parámetros, entre 0 y 1, indicando el nivel de rojo, verde y azul a usar:

rojo verde azul **setrgbcolor**

Por ejemplo:

```
%!PS-Adobe-3.0 EPSF-3.0  
%%Creator: Yo
```

```
%%Title: Ejemplo8
%%CreationDate: 2008-01-03
%%DocumentData: Clean7Bit
%%Origin: 0 0
%%BoundingBox: 0 0 200 200
%%LanguageLevel: 2
```

```
newpath
1 0 0 setrgbcolor
25 100 10 0 360 arc
fill
stroke
```

```
newpath
0 1 0 setrgbcolor
50 100 10 0 360 arc
fill
stroke
```

```
newpath
0 0 1 setrgbcolor
75 100 10 0 360 arc
fill
stroke
```

```
newpath
0.25 0.5 0.7 setrgbcolor
125 100 10 0 360 arc
fill
stroke
```

```
%%EOF
```

Dibujará:



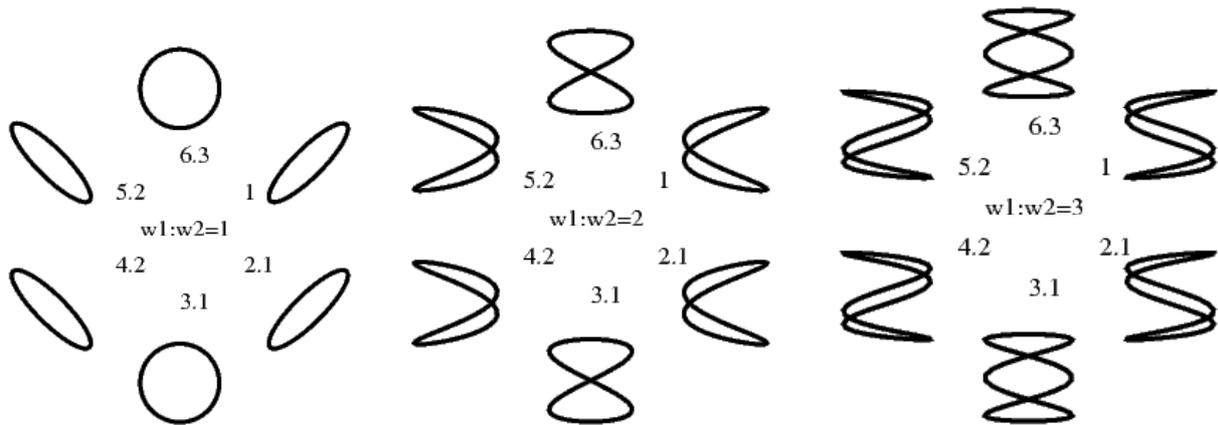
Más información.

Se pueden encontrar unos sencillos apuntes sobre el formato Postscript en:

<http://astronomy.swin.edu.au/%7Epbourke/dataformats/>

Otros ejemplos usando Octave y Postscript: Las figuras de Lissajous.

Con el programa que se mostrará a continuación, diseñado en Octave, se pueden dibujar las figuras de Lissajous a distintos cocientes de las frecuencias y a diferentes fases. Así se obtienen diferentes dibujos cambiando las relaciones entre las frecuencias. Así para relaciones entre las frecuencias $w1:w2$ siendo 1, 2 ó 3 se obtiene:



El programa en Octave sería:

```
function ejemplo9()
```

```
    out=fopen("ejemplo9.eps", "w");
```

```
    fprintf(out, "%%!PS-Adobe-3.0 EPSF-3.0\n");
    fprintf(out, "%%%Creator: Yo\n");
    fprintf(out, "%%%Title: Ejemplo3\n");
    fprintf(out, "%%%CreationDate: 2008-01-03\n");
    fprintf(out, "%%%DocumentData: Clean7Bit\n");
    fprintf(out, "%%%Origin: 0 0\n");
    fprintf(out, "%%%BoundingBox: 0 0 200 200\n");
    fprintf(out, "%%%LanguageLevel: 2\n");
```

```
    N=50;
```

```
    frecuencia=3;
```

```
    N_fases=6;
```

```
    radio=20;
```

```
    radio1=75;
```

```
    for y=[1:N_fases]
```

```
        fase=2*pi/N_fases*y;
```

```
        x=0;
```

```
        fprintf(out, "newpath\n");
```

```
        fprintf(out, "%f %f moveto\n",
```

```
100+radio1*sin(y*2*pi/N_fases)+radio*sin(frecuencia*x*2*pi/N+fase),
```

```
100+radio1*cos(y*2*pi/N_fases)+radio*cos(x*2*pi/N) );
```

```
        for x=[1:N]
```

```
            fprintf(out, "%f %f lineto\n",
```

```
100+radio1*sin(y*2*pi/N_fases)+radio*sin(frecuencia*x*2*pi/N+fase),
```

```
100+radio1*cos(y*2*pi/N_fases)+radio*cos(x*2*pi/N) );
```

```
        endfor
```

```
        fprintf(out, "2 setlinewidth\nstroke\n");
```

```
    fprintf(out, "/Times-Roman findfont\n12 scalefont\nsetfont\n");
```

```
    fprintf(out, "newpath\n");
```

```
    fprintf(out, "%f %f moveto\n", 100+radio1/2*sin(y*2*pi/N_fases),
```

```

100+radio1/2*cos(y*2*pi/N_fases) );
    fprintf(out, "(%1.2g) show\n", fase);

endfor

fprintf(out, "/Times-Roman findfont\n12 scalefont\nsetfont\n");
fprintf(out, "newpath\n");
fprintf(out, "%f %f moveto\n", 80, 100);
fprintf(out, "(w1:w2=%1.2g) show\n", frecuencia);

fprintf(out, "%%EOF\n");

fclose(out);

disp("Dibujado\n");

endfunction

```

P.L. Lucas