El programa DpGraph

www.davidparker.com

El programa DpGraph (Dynamic Photorealistic Graphing, versión 2001.11.09) permite crear bellas superficies 3D, a todo color y con calidad fotográfica, para después mostrarlas en pantalla, publicarlas y manipularlas en Internet. Ha sido programado en lenguaje ensamblador para conseguir la máxima velocidad en la construcción de las gráficas.
Requisitos: Cualquier PC compatible bajo Windows 95, Windows 98, Windows NT 4.0, o posterior. Necesita menos de 450K de espacio en su disco duro.
Instalación: Con el programa de instalación `InstallD.exe` se instala DpGraph y todos los ficheros de ejemplos. Para actualizar el programa, después se debe ejecutar `UpdateDP.exe`.
Deshacer la instalación: Se puede proceder de dos maneras:
1: Haga "click" en `Inicio-Configuración-Panel de control-Agregar/Quitar programas` después busque DpGraph en la ventana, selecciónelo, y haga "click" en el botón `Agregar/Quitar`.
2: Haga "click" en `Inicio-Programas-DpGraph-Unistall`.
Aviso legal: DpGraph no es "shareware"; las copias de usuarios sin licencia son ilegales. DpGraph es una marca registrada por David Parker.

Formato de los ficheros dpg

Es muy ilustrativo hacer "clic" en Open para abrir algunos de los ficheros .dpg que se distribuyen con el programa y luego hacer "clic" en "Edit" para examinar su contenido. DpGraph también permite leer y crear, al grabarlos, ficheros .dpg en binario. DpGraph Viewer, cuya distribución es gratuita, sólo puede leer ficheros ***.dpg de este tipo. Estos ficheros, que están comprimidos y contienen información para la detección de errores, son apropiados para facilitar su transmisión.

Los ficheros de texto ***.dpg utilizados por DpGraph son ficheros estándar ASCII que pueden leer y editar con cualquier editor de texto como por ejemplo Notepad.

- Cada línea del fichero es un comando para DpGraph, excepto las que empiezan con el carácter ' ; ' que son comentarios.
- Entre las líneas de comando se pueden incluir comentarios colocando delante el carácter ' ; ' .
- Cada comando debe de estar en una línea separada.
- En la última línea debe de aparecer el comando GRAPH3D(...).

El siguiente es un ejemplo muy simple de un fichero ***.dpg que sirve para obtener la gráfica del trozo de esfera que queda dentro de una caja. En él se introducen y comentan los comandos imprescindibles para este fín

*Ejemplo*1.*dpg*

```
; Para hacer comentarios como éste, dentro de un fichero ***.dpg,
; se usa el carácter punto y coma.
; Cualquier cosa en una línea, después de ; se ignora
; EN PRIMER LUGAR DEFINIMOS LA CAJA DE VISIÓN:
; Sólo veremos parte de la gráfica que queda dentro de la caja
; Definimos el mínimo y el máximo valor de las variables x, y, z
; Los valores predefinidos por defecto, son -3 y 3, respectivamente.
graph3d.minimumx :=-2 ;Mínimo valor de la variable x
graph3d.maximumx := 2 ;Máximo valor de la variable x
graph3d.minimumy :=-2 ;Mínimo valor de la variable y
graph3d.maximumy := 2 ;Máximo valor de la variable y
graph3d.minimumz :=-2 ;Mínimo valor de la variable z
graph3d.maximumz :=1.5 ;Máximo valor de la variable z
; FINALMENTE, el comando que genera el gráfico (de la esfera),
; aparece en la última línea
GRAPH3D(x^2 + y^2 + z^2=4)
```

Nota: Las versiones anteriores de DpGraph no permitían partir en varias líneas el comando que genera el gráfico. Por ello, cuando se deseaban visualizar varias superficies simultáneamente a veces era preciso escribir una última línea de comando demasiado larga. La versión actual de DpGraph permite descomponer en varias líneas el último comando `graph3D(....)`. Así por ejemplo, en el fichero `ejemplo1.dpg` en vez de escribir:

```
GRAPH3D((x^2+y^2+z^2=1, x^2+y^2+z^2=2, x^2+y^2+z^2=3,x^2+y^2+z^2=4))
```

con la última versión se puede escribir:

```
GRAPH3D((x^2+y^2+z^2=1, x^2+y^2+z^2=2,
            x^2+y^2+z^2=3,x^2+y^2+z^2=4))
```

Gráficas de superficies en forma implícita

Para conseguir la gráfica de una superficie dada en forma implícita, $f(x, y, z) = c$, como es el caso de la esfera $x^2 + y^2 + z^2 = 4$ basta escribir en la última línea

```
GRAPH3D(x^2 + y^2 + z^2 = 4)
```

Se puede conseguir lo mismo usando coordenadas esféricas, escribiendo

```
GRAPH3D(RHO = 2)
```

o con coordenadas cilíndricas, escribiendo

```
GRAPH3D(r = 2*sin(phi))
```

Para representar varias superficies, basta ponerlas en una lista como ya hemos visto en `Ejemplo1.jpg`

Las variables para escribir las ecuaciones de las superficies son

`x, y, z`	coordenadas cartesianas.
`r, theta`	r es el radio polar $= x^2 + y^2$; theta es el ángulo polar $0 \leq \theta < 2\pi$
`rho, theta, phi`	rho es el radio esférico $= \sqrt{x^2 + y^2 + z^2}$;
	theta es el ángulo polar $0 \leq \theta < 2\pi$
	phi es el ángulo esférico $\phi = \pi/2 -$ latitud; $0 < \phi < \pi$.
`time`	un parámetro para generar superficies que cambian con el tiempo.

Se puede dar movimiento a las superficies usando el tiempo `time` como una variable: Un hiperboloide que pasa de una a dos hojas se obtiene con el comando

```
GRAPH3D( X^2 + Y^2 + SIN(time) = Z^2 )
```

En Ejemplo2.dpg se puede ver como se usa la variable `time` para obtener un plano que se mueve tangente a un cilindro, y una esfera cuyo radio cambia con el tiempo. Se utiliza la linea de comando

```
GRAPH3D((sin(time)*x+cos(time)*y=2, r=2, rho=1+|sin(time)|))
```

Nota. Con la experiencia puede encontrar útil el siguiente procedimiento para representar nuevas superficies: Abra el fichero `Defaults.dpg` que contiene todos los comandos usuales, vaya a `Edit` modifique la ecuación predeterminada y los valores asignados y haga "clik" en `Execute` Como todas las opciones están presentes en `Defaults.dpg` pueden ser modificadas con un mínimo tecleo. Por último, vaya a `Save` para grabar el fichero modificado con un nombre diferente.

Recintos definidos por desigualdades

DpGraph puede representar desigualdades 3D, como `GRAPH3D(Z*X >Y)`. Al representar desigualdades como ésta recuerde que el color "azul" señala la región donde la desigualdad es cierta y el color "rojo" señala la región donde la desigualdad es falsa. Si la desigualdad ocurre en la superficie, como en `GRAPH3D(Z*X >=Y)` entonces DpGraph utiliza azul intenso para el lado de la superficie donde la desigualdad es cierta y color magenta para el otro lado. Si la desigualdad no ocurre sobre la superficie, como en `GRAPH3D(Z*X >Y)`, entonces DpGraph utiliza un rojo intenso en el lado de la superficie donde la desigualdad estricta es cierta y color lila para el otro lado. DpGraph también puede mostrar la intersección de regiones definidas por desigualdades como se puede ver el ejemplo Ejemplo3.dpg donde interviene el comando `GRAPH3D( rho <= 3 & z <1.5 )`

Superficies parametrizadas

DpGraph dibuja también superficies dadas en forma paramétrica. Los parámetros para las superficies son U, V. Para definir un dominio rectangular donde varían estos parámetros se usan los comandos

```
graph3d.minimumu :=-3
graph3d.maximumu := 3
graph3d.minimumu :=-3
graph3d.maximumu := 3
graph3d.stepsu := 40
graph3d.stepsv := 40
```

donde los dos últimos comandos sirven para definir el número intervalos en que se descompone el correspondiente intervalo de variación del parámetro.

Para dibujar superficies dadas en forma paramétrica se puede usar uno de los tres comandos `RECTANGULAR(...), CYLINDRICAL(...), SPHERICAL(...)`
Los parámetros para las superficies siempre son son U, V.

El ejemplo Ejemplo4.dpg sirve para obtener la gráfica de un trozo de esfera descrito en forma paramétrica, usando como parámetros la longitud V y la latitud U. En él intervienen los siguientes comandos para definir el dominio rectangular donde varían los parámetros U, V, y los que sirven para fijar la resolución (número de intervalos en que se descompone el intervalo de variación de cada parámetro)

```
; Se define el mínimo y el máximo valor de los parámetros u, v
; Los valores predefinidos por defecto son -3 y 3, respectivamente.
graph3d.minimumu := 0
graph3d.maximumu := pi/4
graph3d.minimumv := -pi/2
graph3d.maximumv := pi/2
; Se define la resolución para cada parámetro
; (el valor por defecto para cada parámetro es 40)
graph3d.stepsu := 8
graph3d.stepsv := 30
GRAPH3D(RECTANGULAR(2*cos(u)*cos(v), 2*cos(u)*sin(v), 2*sin(u)))
```

Los ejemplos Ejemplo4A.dpg Ejemplo4B.dpg muestran como se obtiene el mismo resultado con coordenadas cilíndricas o esféricas sustituyendo la última línea por una de las siguientes

```
GRAPH3D((CYLINDRICAL(2*cos(u),v,2*sin(u)))
GRAPH3D((SPHERICAL(2,v,pi/2-u)))
```

Curvas

Se pueden representar curvas en el espacio dadas en forma paramétrica, usando uno sólo de los parámetros u, v. Por ejemplo, la hélice de ecuaciones paramétricas

$$x = 2\cos(v),\ y = 2\,\mathrm{sen}(v),\ z = v,\ \text{con}\ -4\pi \leq v \leq 4\pi$$

se puede visualizar en Ejemplo5.dpg donde intervienen los comandos

```
graph3d.minimumv :=-4*pi
graph3d.maximumv := 4*pi
graph3d.minimumz :=-7
graph3d.maximumz := 7
graph3d.stepsv:= 60
GRAPH3D(RECTANGULAR(2*cos(v), 2*sin(v), v/2 ))
```

Usando coordenadas cilíndricas, la última línea se puede sustituir por

```
GRAPH3D(CYLINDRICAL(2, v, v/2))
```

Las curvas planas paramétrica son caso particular de las curvas paramétricas en el espacio, con ecuaciones de la forma $x = f(v),\ y = g(v),\ z = 0$. Se recomienda observarlas en una caja tridimensional de pequeña altura, vista desde arriba, sin perspectiva. Un ejemplo de referencia para este asunto es Ejemplo6.dpg) con el que se dibuja la elipse $x = 3\cos(v)$; $y = 2\,\mathrm{sen}(v)$, $0 \leq v \leq 2\pi$. En este ejemplo se utilizan los siguientes comandos

```
graph3d.box :=false ;Para no dibujar la caja
graph3d.view := top ;Para mirar desde arriba
graph3d.perspective := false ;Sin perspectiva
;Definimos una caja rectangular de pequeña altura:
graph3d.minimumx :=-3
graph3d.maximumx := 3
graph3d.minimumy :=-2
graph3d.maximumy := 2
graph3d.minimumz :=-0.1
graph3d.maximumz := 0.1
; El parámetro para la elipse es el ángulo v.
; Definimos el número de pasos del parámetro v su intervalo de variación.
graph3d.stepsv := 80
graph3d.minimumv := 0
graph3d.maximumv := 2*pi
; Última línea de comando con la ecuación paramétrica de la elipse,
GRAPH3D(RECTANGULAR(3*cos(v), 2*sin(v), 0))
```

También es posible representar gráficas de curvas en el plano dadas en forma implícita, F(x; y) = 0. Basta considerar las superficies 3D dadas por la misma ecuación (cilindros cuya generatriz es el eje z) y observarlas en una caja tridimensional con

un rango muy pequeño para la variable z, y elegir la vista desde arriba sin perspectiva. El ejemplo, Ejemplo7.dpg), que también sirve para dibujar una elipse, se puede tomar como referencia para obtener gráficas de curvas planas dadas en forma implícita

```
graph3d.box :=false ;Para no dibujar la caja
graph3d.view := top ;Para mirar desde arriba
graph3d.perspective := false ;Sin perspectiva
;Definimos una caja rectangular de pequeña altura:
graph3d.minimumx :=-4
graph3d.maximumx := 4
graph3d.minimumy :=-3
graph3d.maximumy := 3
graph3d.minimumz :=-0.1
graph3d.maximumz := 0.1
;Ultima línea de comando con la ecuación de la elipse y los ejes
GRAPH3D(( x^2/9+ y^2/4=1, x=0, y=0 ))
```

Campos de vectores

Se puede utilizar el comando `vector` para visualizar campos de vectores. Un campo radial de vectores normales a esferas centradas en el origen se obtiene con

```
GRAPH3D( vector(x/rho/3, y/rho/3, z/rho/3) )
```

Para dibujar la punta derecha del vector se debe incluir el comando

```
graph3d.vectorarrowhead:= true
```

Para visualizar sólo los vectores seleccionados de un campo vectorial se usa la función

```
near(x,y,z)
```

que vale 1 en el punto de la malla que está más cerca de `(x,y,z)` y en los otros puntos no está definida. Para seleccionar el conjunto de vectores que se desean dibujar basta multiplicar `near(x,y,z)` por los elementos del vector. Por ejemplo, para dibujar `vector(2,3,1)` con su base cerca de `(0,0,0)` basta escribir

```
GRAPH3D( vector(near(0,0,0)*2,3,1))
```

No es preciso que los argumentos de near sean constantes. Por ejemplo, se obtiene un plano entero de vectores usando `near(X,0,0)`. Truco: Establecer un valor de `graph3D.resolution` de modo que `near(X,Y,Z)` sea cierta en un punto de la malla. Por ejemplo, si sólo estamos interesado en el vector cerca de `(0,0,0)` podemos establecer `graph3D.resolution:=3`

Mover y cambiar las gráficas

Para ampliar (zoom) y girar las gráficas, se pueden utilizar las teclas `Home (Inicio)`, `PageUp (RePág)`, `PageDown (AvPág)`, y las teclas con flechas.

DpGraph también permite ver como cambia una superficie, cuya ecuación depende de parámetros, al modificar los valores de los parámetros. En la ecuación de la superficie pueden intervenir hasta cuatro parámetros: Las constantes `a, b, c,`

`d` cuyos valores iniciales, y rango de variación, se suelen dar en las primeras líneas del fichero ***.dpg. Para activar el cambio de valor de un parámetro hay que seleccionar la letra correspondiente en el cuadro de diálogo que se abre con la opción `Scrollbar` del menú. Entonces el valor del de ese parámetro se puede modificar con la barra de desplazamiento. Se pueden explorar los efectos de cambiar los valores de estas constantes en el ejemplo Ejemplo8.dpg), con el que se visualiza un elipsoide de semiejes variables a,b,c usando el comando

```
GRAPH3D((x^2/a^2+y^2/b^2+z^2/c^2=d^2))
```

En este ejemplo intervienen los siguientes comandos para fijar los valores iniciales de a,b,c, así como los valores máximo y mínimo que corresponden a los extremos de la barra de desplazamiento.

```
a:=2
a.minimum:=0.1
a.maximum:=2
b:=3
b.minimum:=0.1
b.maximum:=3
c:=2
c.minimum:=0.1
c.maximum:=4
d:=0.6
d.minimum:=0.1
d.maximum:=1
```

Además, la opción `scrollbar` permite observar de forma dinámica (usando la barra de desplazamiento que aparece a la derecha de la ventana del programa) los cortes de la superficie con los planos X=constante, Y=constante y Z=constante. Para ello hay que marcar una de esas letras en el cuadro de diálogo de la opción.

Errores comunes

Los errores más comunes cuando se crean gráficas en la ventana de edición son:
- No utilizar asteriscos (*) para la multiplicación;
- No escribir los símbolos de igualdad (=) o desigualdad (<>);
- Olvidar los paréntesis para las listas de ecuaciones;
- Tratar de utilizar igualdades en las intersecciones (DpGraph sólo puede representar intersecciones de desigualdades, es decir, regiones del espacio);
- Poner o dejar un espacio en blanco entre los dos símbolos :=, >=, o <=, que se usan para asignar parámetros o escribir desigualdades;
- Usar los parámetros variables u, v para gráficas en implícitas;
- Usar las variables x, y, z, r, theta, rho, phi, para gráficas en paramétricas.

Equivocado:	Correcto:
`graph3d( 3x = z^2 )`	`graph3d( 3*x = z^2 )`
`graph3d( x^2+y^2 )`	`graph3d( z = x^2+y^2 )`
`graph3d( x=1, y=1 )`	`graph3d( ( x=1, y=1 ) )`
`graph3d( x=1 & y=1 )`	`graph3d( x >1 & y >= 1 )`
`graph3d( z^2 >= x^2 )`	`graph3d( z^2 >= x^2 )`
`graph3d( z=u^2+v^2)`	`graph3d( z=x^2+y^2 )`
`graph3d( rectangular(x,y,sin(x+y)))`	`graph3d( rectangular(u,v,sin(u+v)))`

Gestión de gráficas

Copiar gráficas al portapapeles: Haga "click" en la opción `clipboard` del menú, para copiar la gráfica al portapapeles. Para copiar la ventana entera (incluyendo los bordes), presione simultáneamente las teclas `Alt` y `ImprPant`. Para copiar toda la pantalla del monitor presione sólo la tecla `ImprPant` .
Copiar gráficas a otra aplicación: Si la otra aplicación soporta la acción de pegar (paste), primero copie la gráfica al portapapeles y luego haga "click" en la opción `pegar (paste)` del menú de `Edición (Edit)` de la otra aplicación.
Imprimir gráficas: Comience por exportar el gráfico a una aplicación, como por ejemplo Word o Paint, que permita imprimir, y utilice esta aplicación para `imprimir` o guardar la gráfica en un formato transportable como `GIF, JPG, PS, PDF`.
Problemas conocidos:
1.`Microsoft WordPad` bajo Windows NT 4.0 al incluir gráficos producidos por DpGraph o gráficos copiados desde el portapapeles, los construye demasiado grandes. También recorta estos gráficos por el lado derecho.
2.`Microsoft Paint` utilizando 256 colores sólo usa 16 colores con los gráficos producidos por DpGraph.

Modo de visionado

Los siguientes comandos definen parámetros para el aspecto de la gráfica, color, etc..
Los valores predeterminados aparecen en MAYÚSCULAS

`graph3d.box := TRUE/false`

Para ver o no las aristas de la caja de visión

`graph3d.mesh := TRUE/false`

Para dibujar o no las líneas que unen los puntos que están en la superficie y en la malla determinada por los puntos considerados en cada lado de la caja.

`graph3d.view := STANDARD`

Punto de vista, cuyos posibles valores son: STANDARD, top, side, front, textbook (estandar, desde arriba, desde un lado, desde el frente, o como en los libros de texto).

`graph3d.perspective := TRUE/false`

Para ver o no en perspectiva.

graph3d.background := WHITE

Color de fondo, cuyos posibles valores son: black, gray, white, brown, red, yellow, green, cyan, blue, magenta.

`graph3d.color := BYHEIGHT`

Color de la gráfica, que puede tomar los valores: byheight, (por el valor de z), bysteepness, (por las curvas de la malla) black, gray, white, brown, red, yellow, green, cyan, blue, magenta, y también una expresión que depende de `x, y, z` o de `time` (tiempo), por ejemplo `graph3d.color := sin(x)*sin(y)*sin(z)+time/6`.

Para convertir la expresión en un color DpGraph usa solamente la mantisa del valor de la expresión en el punto e instante correspondiente. Al valor que resulta, entre 0 y 1 le asigna un color de la escala empezando con rojo (=0) y terminando con negro.

`graph3d.contrast := 1/3`

Valor por predeterminado 1/2. Puede tomar valores entre 0 y 1.

`graph3d.transparency := 0`

Valor predeterminado 1/2. Puede tomar valores entre 0 y 1.

`graph3d.highlight:= 1/3`

Valor predeterminado 0. Puede tomar valores entre 0 y 1.

`graph3d.shading:= 1/2`

Valor predeterminado 0. Puede tomar valores entre 0 y 1.

`graph3d.resolution:=50`

Con este comando se fija el número de puntos considerados en el lado más largo de la caja en la que se representa la superficie. Para gráficas en forma implícita el valor predeterminado, por defecto, es de 21 x 21 x 21, que utiliza una malla con 21 puntos en cada eje. (Cuando los lados de la caja no son iguales, se les asigna una resolución proporcional a la longitud del lado). La manera más sencilla de mejorar el aspecto de las gráficas es aumentando su resolución (a mayor resolución más detalle pero más lento). Para cambiar la resolución utilice la ventana de edición `Edit` para insertar o modificar la línea `graph3d.resolution := 30`

`graph3d.stepsu:=50,  graph3d.stepsv:=50`

Resolución en forma paramétrica. El valor predeterminado es de 40 x 40.

`graph3d.vectorcolor := red`

Puede tomar los valores: `BYHEIGHT (por el valor de z), bysteepness, (por las curvas de la malla) black, gray, white, brown, red, yellow, green, cyan, blue, magenta` y también una expresión que depende de `x, y, z`, o de `time`.

`graph3d.vectorhighlight:= 1/2`

Valor predeterminado 0. Puede tomar valores entre 0 y 1.

`graph3d.vectorshading:= 1/2`

Valor predeterminado 0. Puede tomar valores entre 0 y 1.

`graph3d.vectoralign:= TAIL`

Valores: `tail, center, tip` (Alinear el vector desde base, centro, punta)

`graph3d.vectorarrowhead:= true/FALSE`

Sirve para dibujar vectores con o sin punta de flecha.

Resumen de los valores para las opciones

`true`	para mesh, perspective, box, y vectorarrowhead.
`false`	para mesh, perspective, box, y vectorarrowhead.
`standard`	para view (x será el eje más horizontal).
`textbook`	para view (y será el eje más horizontal).
`top`	para view (desde arriba)
`front`	para view (desde el frente)
`side`	para view (desde el lateral)
`byheight`	para color o vectorcolor (por altura)
`bysteepness`	para color o vectorcolor (por pasos)
`black`	para color, vectorcolor, o background.
`gray`	para color, vectorcolor, o background.
`white`	para color, vectorcolor, o background.
`brown`	para color, vectorcolor, o background.
`red`	para color, vectorcolor, o background.
`yellow`	para color, vectorcolor, o background.
`green`	para color, vectorcolor, o background.
`cyan`	para color, vectorcolor, o background.
`blue`	para color, vectorcolor, o background.
`magenta`	para color, vectorcolor, o background.
`tail`	para vectoralign.
`center`	para vectoralign.
`tip`	para vectoralign.

Constantes, variables, funciones y símbolos

Constantes

`pi`	(π)
`e`	(base del logaritmo natural)

Variables

`x`	(x variable para representaciones 3D)
`y`	(y variable para representaciones 3D)
`z`	(z variable para representaciones 3D)
`r`	(radio polar, r=sqrt(xˆ2+yˆ2))
`theta`	(ángulo polar, 0<= theta< 2 π, theta=angle(-x,-y)+pi)
`rho`	(radio esférico, rho=sqrt(xˆ2+yˆ2+zˆ2))
`phi`	(ángulo esférico, 0<=phi<=pi, phi=angle(z,r))
`a`	(variable a para la barra de desplazamiento)
`b`	(variable b para la barra de desplazamiento)
`c`	(variable c para la barra de desplazamiento)
`d`	(variable d para la barra de desplazamiento)
`time`	(va tomando el valor del reloj del ordenador para gráficos que cambian con el tiempo)

Símbolos

`=`	(igualdad, para escribir ecuaciones)
	(menor que, para escribir desigualdades)
$>$	(mayor que, para escribir desigualdades)
`<=`	(menor o igual que, para escribir desigualdades))
`>=`	(mayor o igual que, para escribir desigualdades))
`&`	(intersección)
`:=`	(asignación, para asignar variables)
`(...)`	(las listas de argumentos o ecuaciones van entre paréntesis)
`+`	(suma o signo para números positivos)
`-`	(resta o signo para números negativos)
`*`	(multiplicación)
	(división)
`^`	(potencias)
`|..|`	(valor absoluto, = abs(x), la última expresión también es válida)
`!`	(factorial)

Funciones de tres argumentos:

`rectangular`	(para gráficos de superficies paramétricas, usando coordenadas cartesianas)
`cylindrical`	(para gráficos de superficies paramétricas, usando coordenadas cilíndricas
`spherical`	(para gráficos de superficies paramétricas, usando coordenadas esféricas)
`near`	(1 si (x,y,z) esta cerca del near current grid, en otro caso permanece indefinida)
`vector`	(dibuja un campo de vectores)

Funciones de dos argumentos:

`angle`	($-\pi \leq$ angle $\leq \pi$, argumento principal de $z = x + iy$)
`beta`	(la función beta)

Funciones con un argumento:

`abs`	(valor absoluto, abs(x)=$\|x\|$, la última expresión también es válida)
`acos`	(arcocoseno)
`acosh`	(arcocoseno hiperbólico)
`acot`	(arcocotangente)
`acoth`	(arcocotangente hiperbólico)
`acsc`	(arcocosecante)
`acsch`	(arcocosecante hiperbólico)
`asec`	(arcosecante)
`asech`	(arcosecante hiperbólica)
`asin`	(arcoseno)
`asinh`	(arcoseno hiperbólico)
`atan`	(arcotangente)
`atanh`	(arcotangente hiperbólica)
`ceiling`	(menor entero mayor o igual que el argumento)
`cos`	(coseno)
`cosh`	(coseno hiperbólico)
`cot`	(cotangente)
`coth`	(cotangente hiperbólica)
`csc`	(cosecante)
`csch`	(cosecante hiperbólica)
`floor`	(mayor entero menor o igual que el argumento)
`gamma`	(function gamma)
`ln`	(log base e)
`log`	(log base 10)
`nonneg`	(1 si x>=0, en otro caso no está definido)
`one`	(1 si 0<=x<=1, en otro caso no está definido)
`sec`	(secante)
`sech`	(secante hiperbólico)
`sign`	(signo, sign(x)= -1 si x es negativo, 1 si es positivo, indefinido si es 0 o no está defini
`sin`	(seno)
`sinh`	(seno hiperbólico)
`sqr`	(cuadrado, sqr(x)=x*x)
`sqrt`	(raiz cuadrada)
`tan`	(tangente)
`tanh`	(tangente hiperbólica)