

```

      \\\///
      /  _  \
     (| (.) (.) |)
-----o00o--()--o00o-----
|                                     |
|  _ _ \ | | \ \ \ / _ _ \ | | _ _ | | | | | | |
| |_) | | | | > < | |_) | | ( ) | |
| _ _ / \ _ , / _ \ \ _ _ / | | \ _ |
|_|   | _ _ /   |_|                                     |
-----o0o0-----
      ( )   0ooo.
      \ (   ( )
      \_)   ) /
      (_/

```

Codeblock class: pyxplot

```
sudo apt-get install pyxplot
```

```
http://pyxplot.org.uk
```

Note:

- Imagine adds the following lines to the top of the script


```
set terminal {im_fmt}
set output {outfile}
```
- that means you cannot use set output in the pyxplot itself.
 There seems no way to convince `set output` to take a variable to
 indicate an output filename. It'll take anything quite literally..

runs:

```
> pyxplot {im_opt} <fname>.pyxplot
```

```
class->cmd
```

```
  pyxplot -> pyxplot
```

Metadata options

```
  imagine.im_out: img, fcb
```

```
  imagine.im_log: 4
```

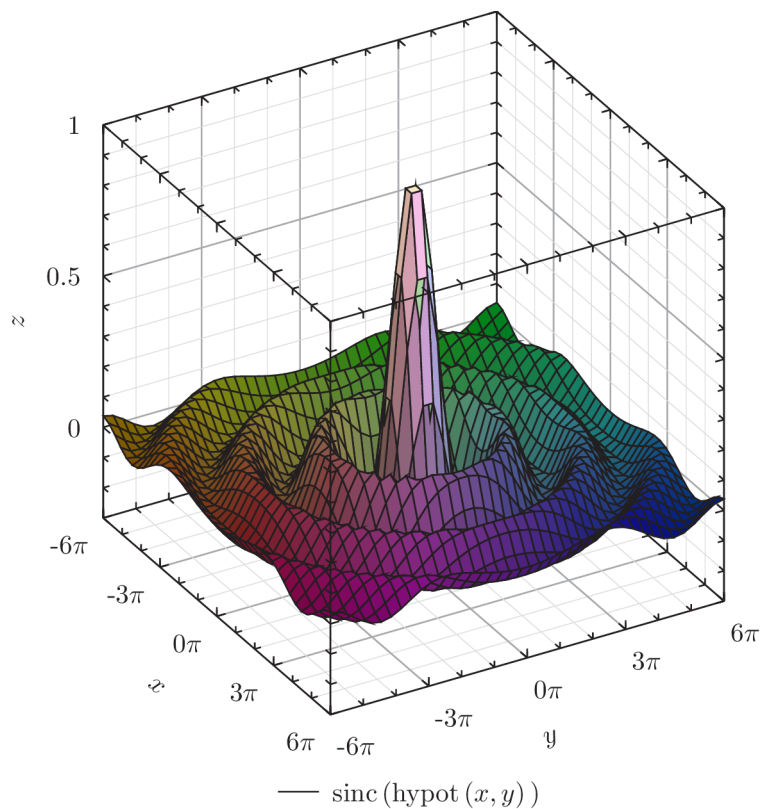
Note:

- no `sudo apt-get install pyxplot` on Ubuntu 18.04 bionic
- to install, download & unpack pyxplot, and:

```
virtualenv venv -p /usr/bin/python2 # installerscripts need python2
source venv/bin/activate
```

```
make  
sudo make install
```

Surface

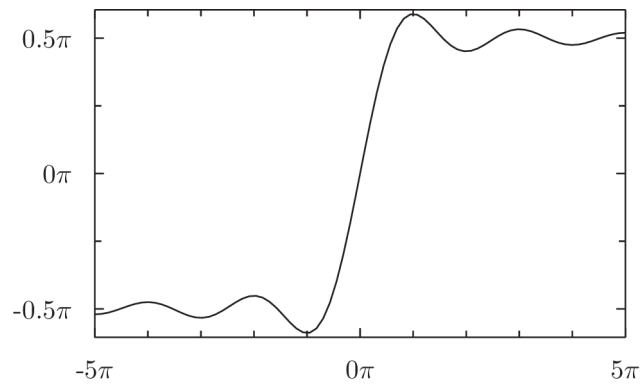


```

```pyxplot
set numerics complex
set xlabel r"x"
set ylabel r"y"
set zlabel r"z"
set xformat r"%sπ"%(x/pi)
set yformat r"%sπ"%(y/pi)
set xtics 3*pi ; set mxtics pi
set ytics 3*pi ; set mytics pi
set ztics
set key below
set size 6 square
set grid
plot 3d [-6*pi:6*pi][-6*pi:6*pi][-0.3:1] sinc(hypot(x,y)) \
 with surface col black \
 fillcol hsb(atan2($1,$2)/(2*pi)+0.5,hypot($1,$2)/30+0.2,$3*0.5+0.5)
```

```

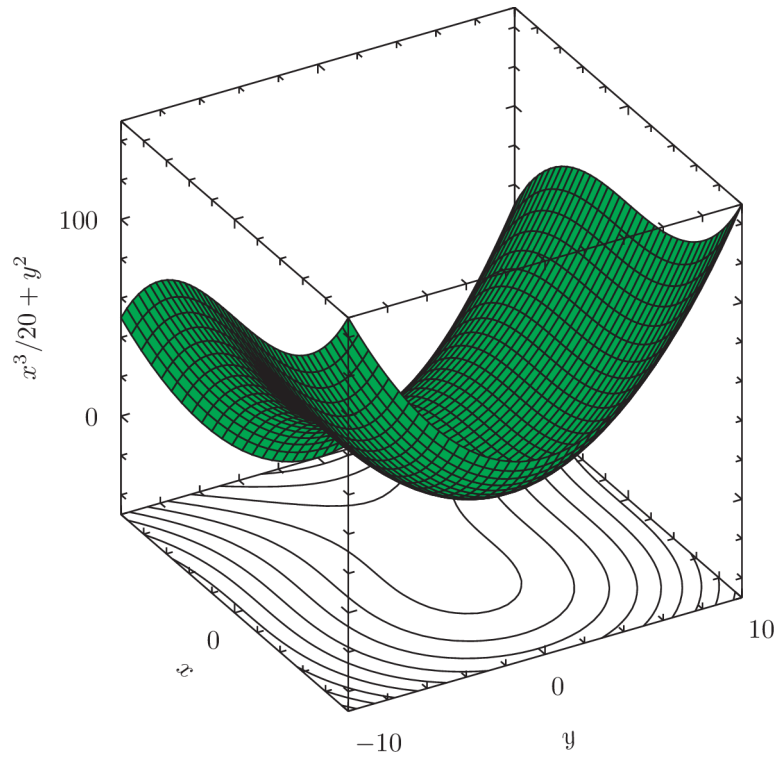
Numerical integration



$$\text{---} \int_0^x (\text{sinc}(z)) dz$$

```
```pyxplot
set samples 80
set key bottom right
set xformat r"%sπ"%(x/pi)
set yformat r"%sπ"%(y/pi)
set xrange [-5*pi:5*pi]
set width 7
set key below
plot int_dz(sinc(z),0,x)
```
```

Surface and contours



```

```pyxplot
set xlabel "x"
set ylabel "y"
set zlabel "$x^3/20+y^2$"
set nokey
set size 6 square
set nogrid
plot 3d x**3/20+y**2 with surface col black fillc green, \
 x**3/20+y**2 with contours col black
```

```

Documentation *PyxPlot*

pyxplot -h

Pyxplot 0.9.2

Usage: pyxplot <options> <filelist>

- h, --help: Display this help.
- v, --version: Display version number.
- q, --quiet: Turn off initial welcome message.
- V, --verbose: Turn on initial welcome message.
- c, --color: Use colored highlighting of output.
- m, --monochrome: Turn off colored highlighting.

A brief introduction to Pyxplot can be obtained by typing 'man pyxplot'; the full Users' Guide can be found in the file:
/usr/local/share/doc/pyxplot/pyxplot.pdf

Extensive documentation is also available at <<http://www.pyxplot.org.uk>>, which also provides contact information for bug reports.

man page

PYXPLOT(1)

General Commands Manual

PYXPLOT(1)

NAME

pyxplot - a multi-purpose command-line data processing, vector graphics and graph-plotting tool.

SYNOPSIS

pyxplot [file ...]

DESCRIPTION

Pyxplot is a multi-purpose graph plotting tool, scientific scripting language, vector graphics suite, and data processing package. Its interface is designed to make common tasks -- e.g., plotting labelled graphs of data -- accessible via short, simple, intuitive commands.

But these commands also take many optional settings, allowing their output to be fine-tuned into styles appropriate for printed publications, talks or websites. Pyxplot is simple enough to be used without prior programming experience, but powerful enough that programmers can extensively configure and script it.

A SCIENTIFIC SCRIPTING LANGUAGE

Pyxplot doesn't just plot graphs. It's a scripting language in which variables can have physical units. Calculations automatically return results in an appropriate unit, whether that be kilograms, joules or lightyears. Datafiles can be converted straightforwardly from one set of units to another. Meanwhile Pyxplot has all the other features of a scripting language: flow control and branching, string manipulation, complex data types, an object-oriented class structure and straightforward file I/O. It also supports vector and matrix algebra, can integrate or differentiate expressions, and can numerically solve systems of equations.

A VECTOR GRAPHICS SUITE

The graphical canvas isn't just for plotting graphs on. Circles, polygons and ellipses can be drawn to build vector graphics. Colors are a native object type for easy customisation. For the mathematically minded, Pyxplot's canvas interfaces cleanly with its vector math environment, so that geometric construction is easy.

A DATA PROCESSING PACKAGE

Pyxplot can interpolate data, find best-fit lines, and compile histograms. It can Fourier transform data, calculate statistics, and output results to new datafiles. Where fine control is needed, custom code can be used to process every data point in a file.

Examples of the graphical output produced by Pyxplot can be found on the Pyxplot website: [<http://www.pyxplot.org.uk/>](http://www.pyxplot.org.uk/).

Full documentation can be found in: `/usr/local/share/doc/pyxplot/pyxplot.pdf`

COMMAND LINE OPTIONS

| | |
|--------------------------------|-------------------------------------|
| <code>-h, --help:</code> | Display this help. |
| <code>-v, --version:</code> | Display version number. |
| <code>-q, --quiet:</code> | Turn off initial welcome message. |
| <code>-V, --verbose:</code> | Turn on initial welcome message. |
| <code>-c, --color:</code> | Use colored highlighting of output. |
| <code>-m, --monochrome:</code> | Turn off coloured highlighting. |

AUTHORS

Dominic Ford [<coders@pyxplot.org.uk>](mailto:coders@pyxplot.org.uk)

BUGS AND USER FORUMS

To report bugs in Pyxplot, or to meet other Pyxplot users in our forums, please visit us on sourceforge: [<http://sourceforge.net/projects/pyxplot/>](http://sourceforge.net/projects/pyxplot/).

CREDITS

Matthew Smith, Michael Rutter, Zoltan Voros and John Walker have all contributed code to Pyxplot. We welcome bug reports, which can be submitted to our project page on Sourceforge, and thank the many testers who have already made significant contributions to the project by helping us to track down bugs.

SEE ALSO

`pyxplot_watch(1)`, `gnuplot(1)`

PYXPLOT(1)