Plotting in Madagascar

July 24, 2010

Madagascar development team

Outline

Principles

- All plotting programs take RSF files as input through stdin and write binary Vplot to stdout
- Device-dependent rendering is handled by Vplot pens
- Madagascar SCons build workflows will handle all conversions automatically

Command line

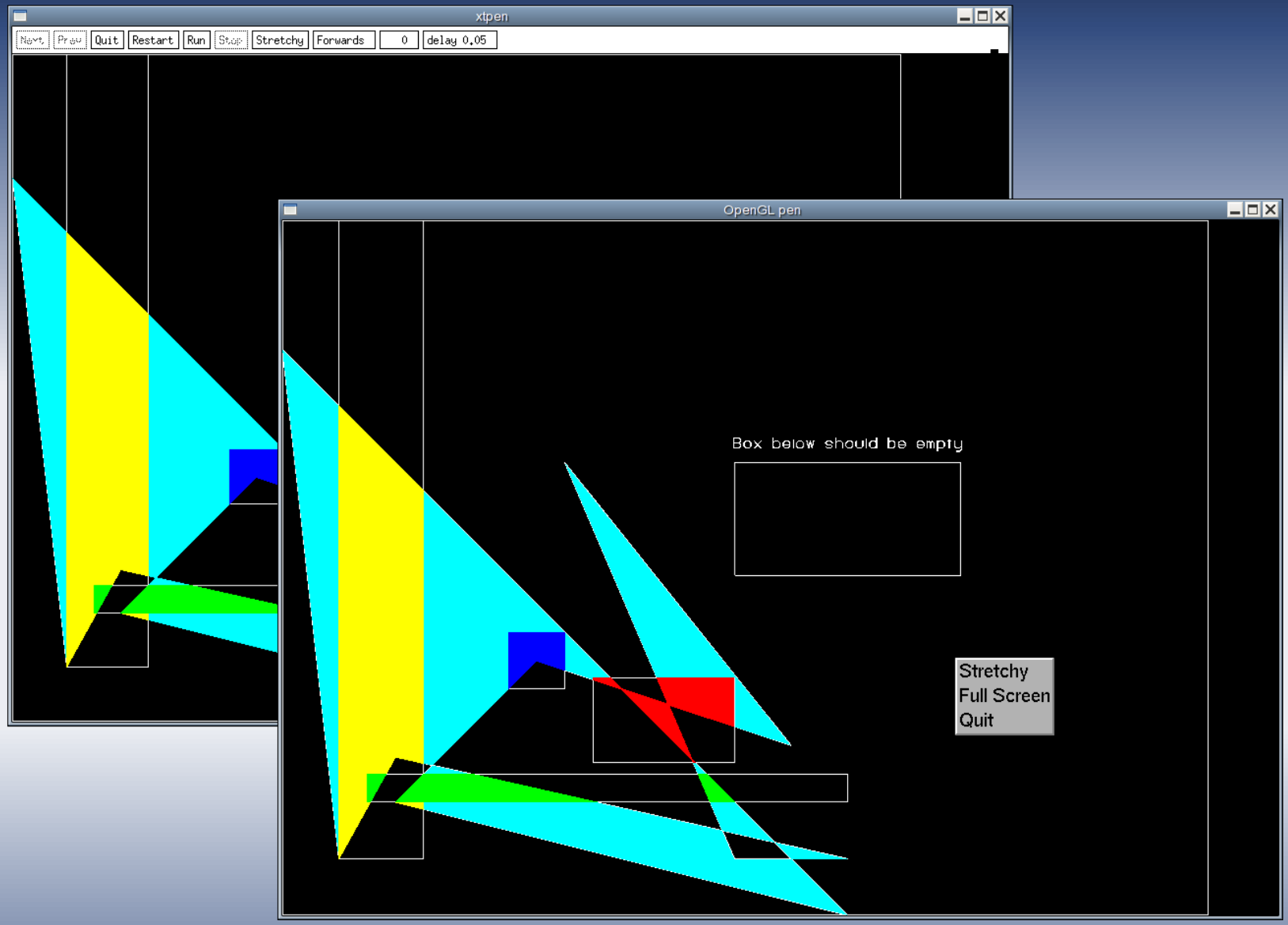
```
$ <input.rsf plotting_program [parameters] >output.vpl
$ <input.vpl vplot_pen [parameters]
$ <input.rsf some|processing|workflow|plotting|pen>image
```

Manual conversion

```
$ vpconvert file.vpl file.jpg
$ vpconvert format=tiff Fig/*.vpl
```

Pens

Pen name	Dependency	Output
xtpen	X11	Screen
oglpen	OpenGL	Screen
jpegpen	libjpeg	JPEG
tiffpen	libtiff	TIFF
ppmpen	NetPBM	PPM
gdpen	LibGD, FFmpeg	PNG,JPEG,GIF,MPEG
pngpen	Cairo	PNG
svgpen	Cairo	SVG
pdfpen	Cairo	PDF
pspen	None	Postscript
vppen	None	Vplot filter



RSFSRC $ ./configure SFPEN=oglpen

SConstruct functions

```
Plot(intermediate_plot[,source],plot_command)
Plot(intermediate_plot,intermediate_plots,combination)
Result(plot[,source],plot_command)
Result(plot,intermediate_plots,combination)
```

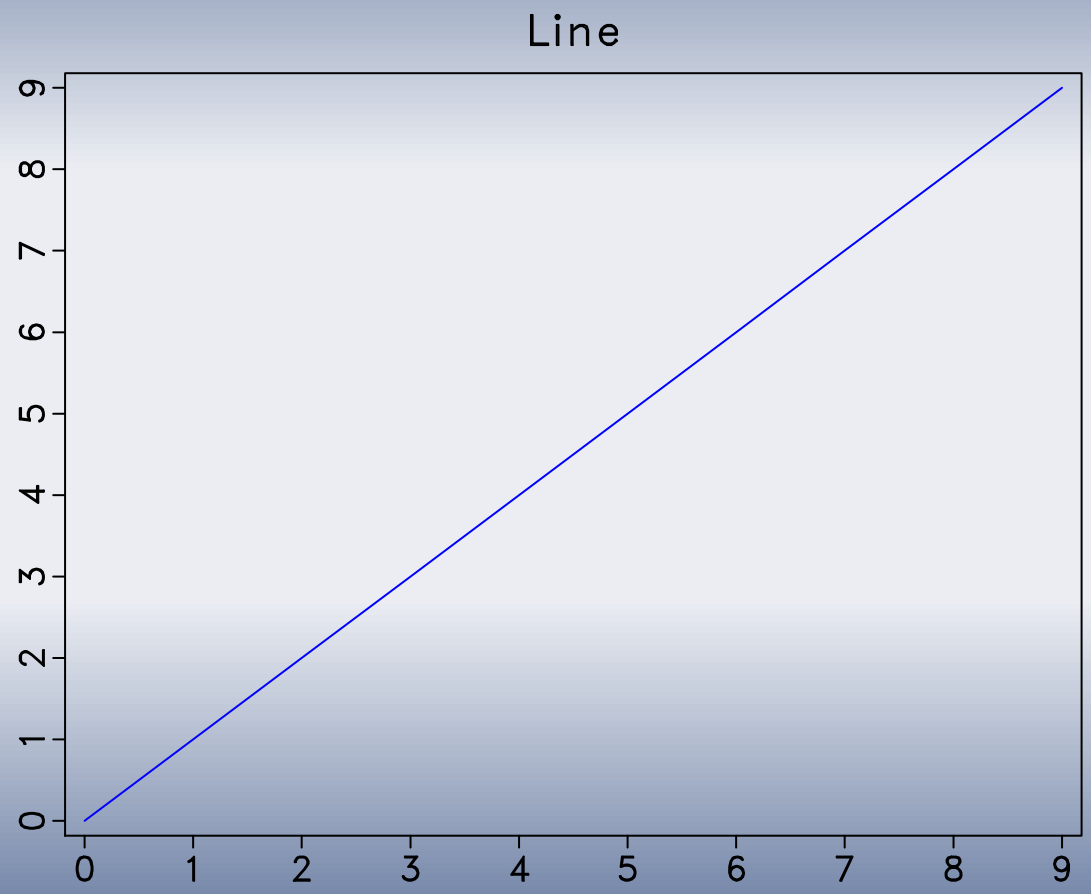
Command line

$ scons <target>.view # View result on the screen
$ scons <target>.print # Send to a printer
$ scons <target>.lock # Install figures into a separate location

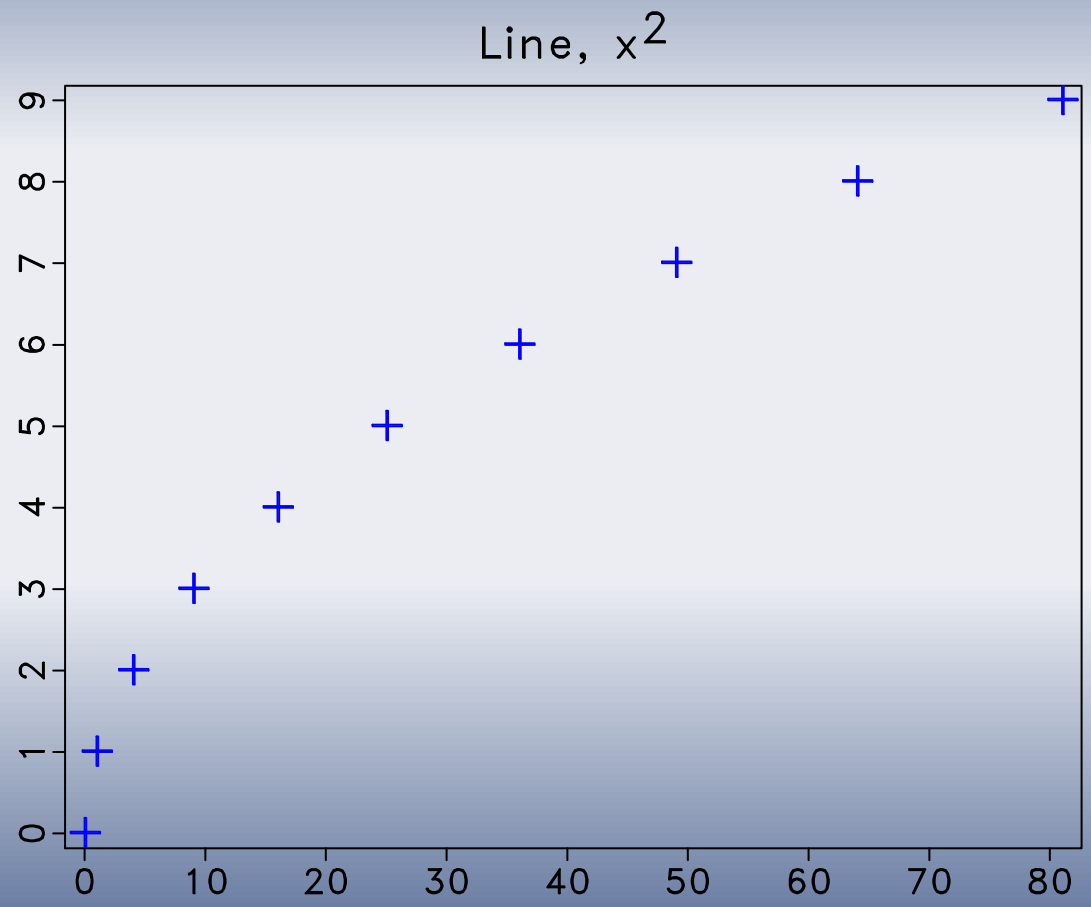
Environment variables

RSFFIGS - location for installation of figures
PSPRINTER - destination for lp command

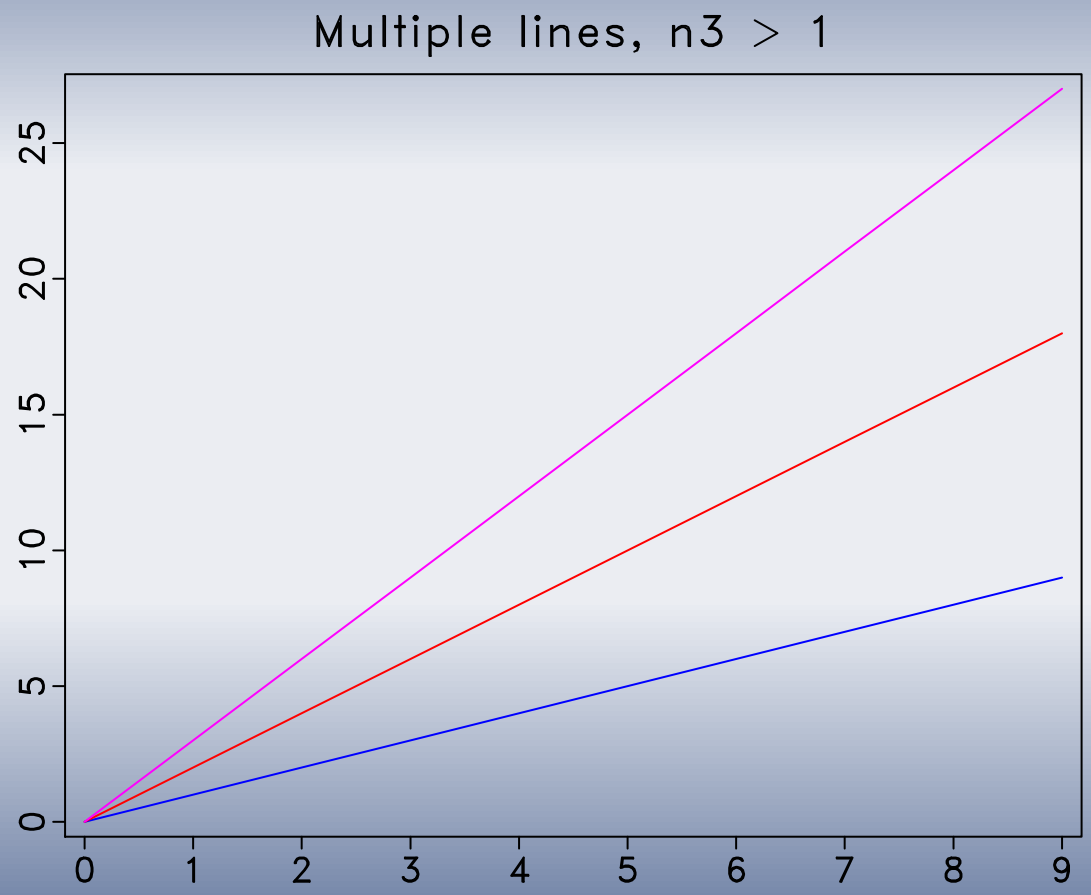
```
Flow('line',None,'math n1=10 output="x1"')
Result('line','graph title=Line')
```



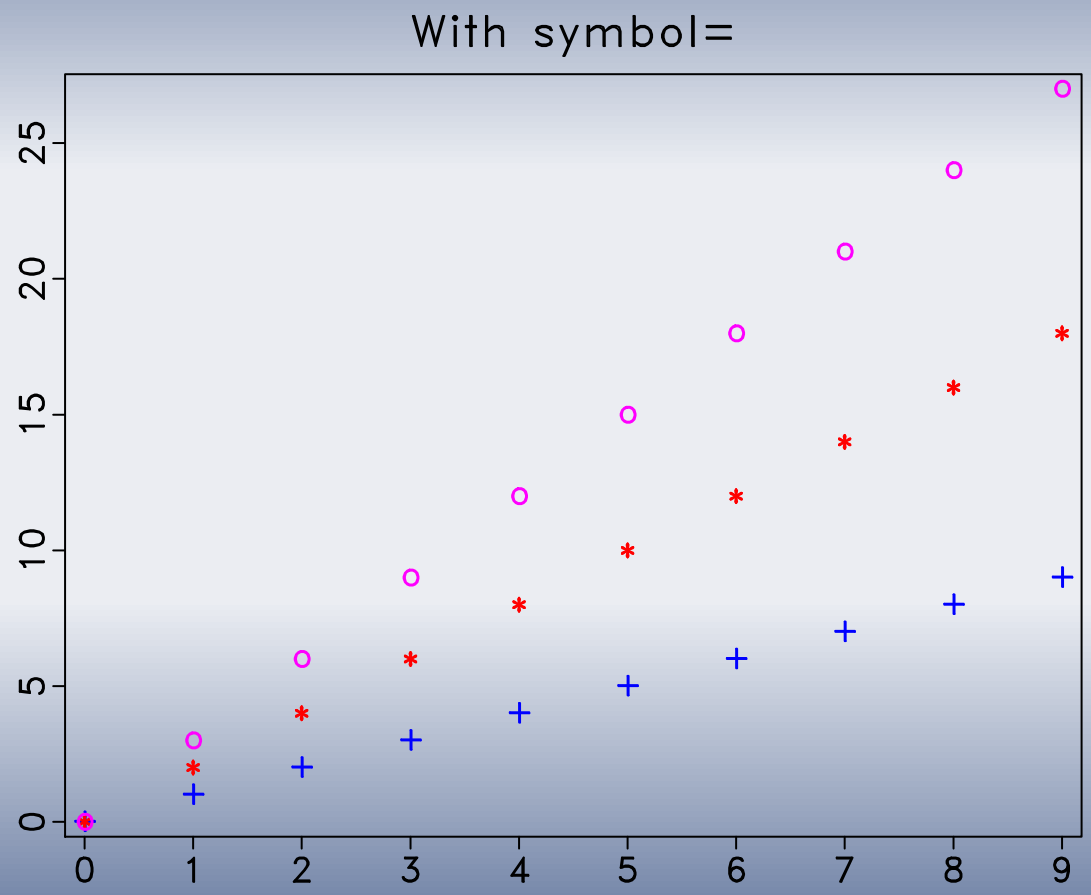
```
Flow('xsq',None,'math n1=10 output="x1*x1"')
Flow('sqline','xsq line','cmplx ${SOURCES[1]}')
Result('sqline','''
        graph symbol="+" title="Line, x\\^2" symbolsz=12''')
```



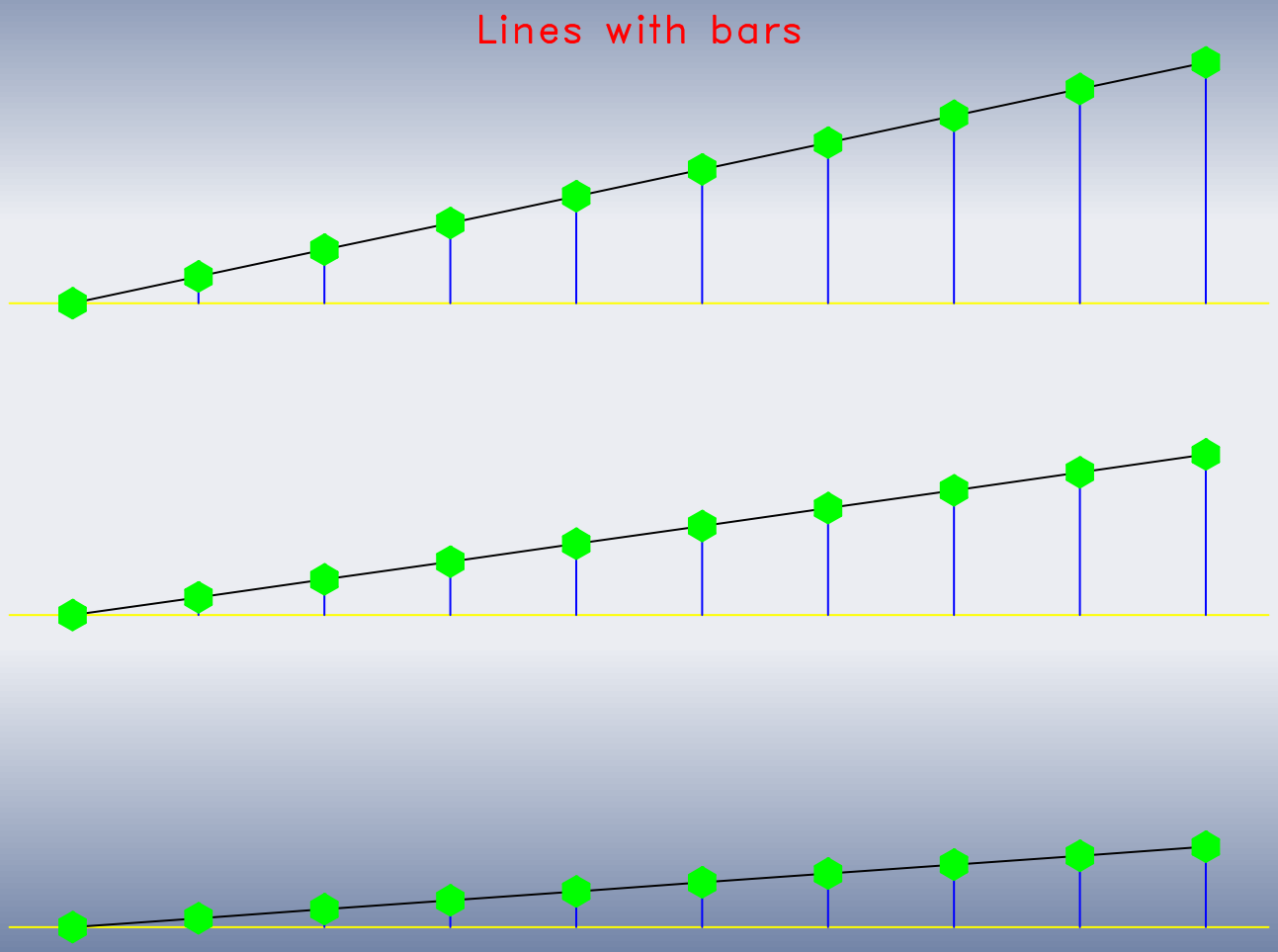
```
Flow('mline',None,'math n1=10 n2=3 output="(x2+1)*x1"')
Result('mlines','mline','graph title="Multiple lines, n3 > 1"')
```



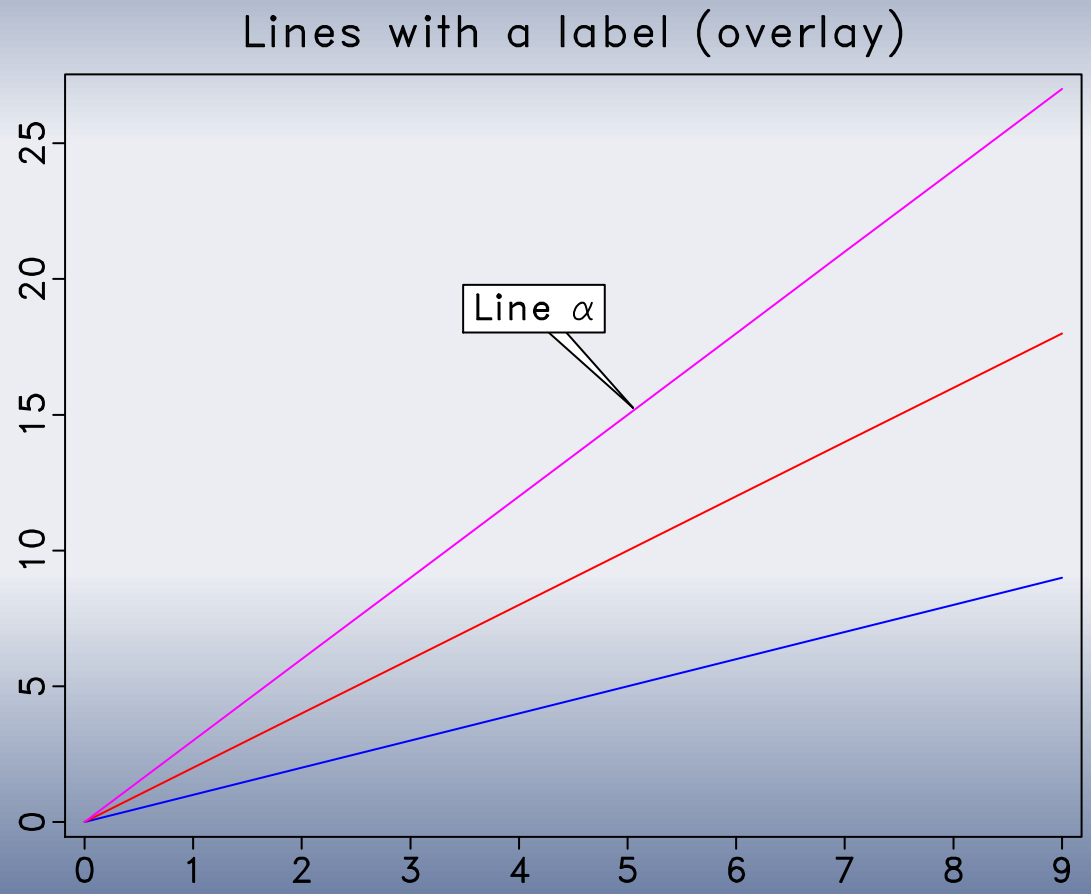
```
Result('mplines','mline',''' graph title="With symbol=" symbol=+*o symbolsz=8''')
```



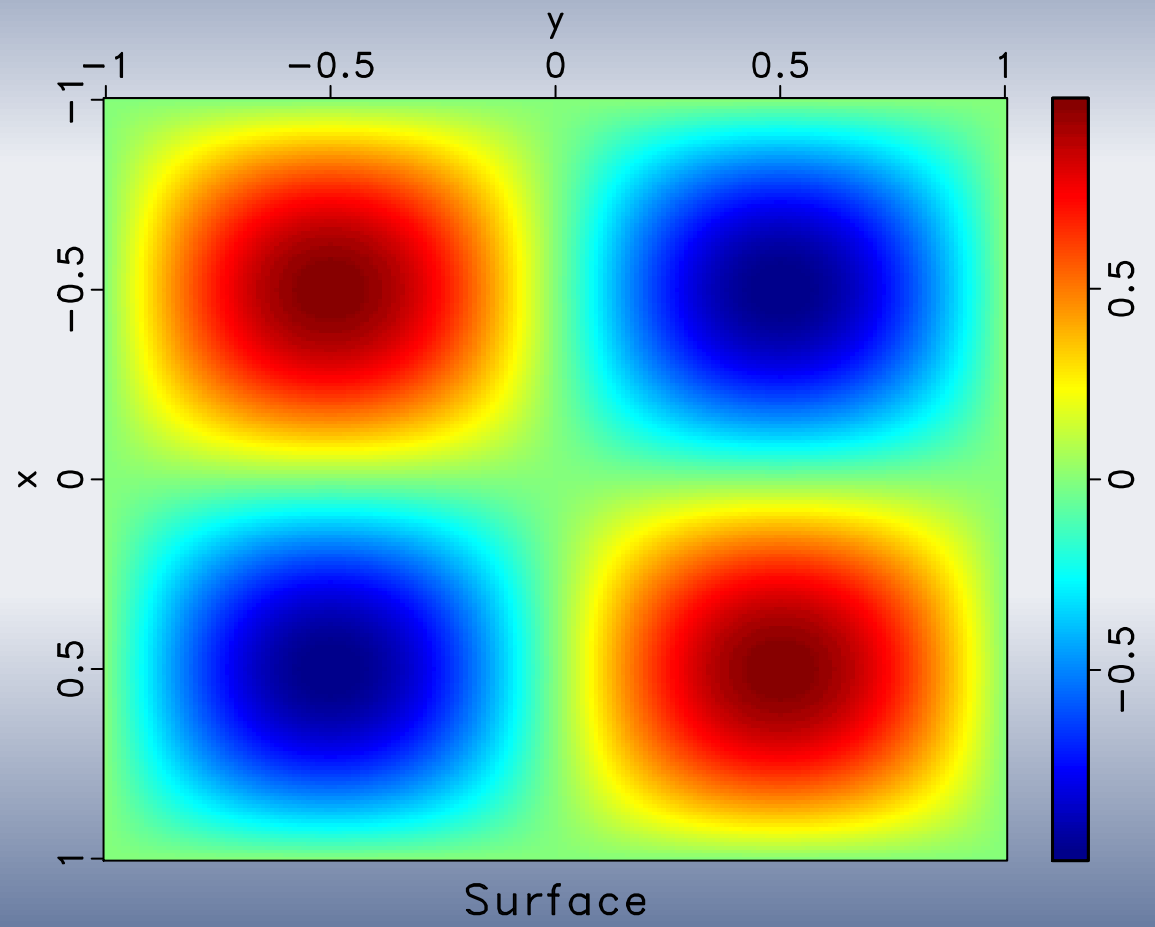
```
Result('dplines','mline','''
        dots title="Lines with bars" gaineach=n radius=0.15''')
```



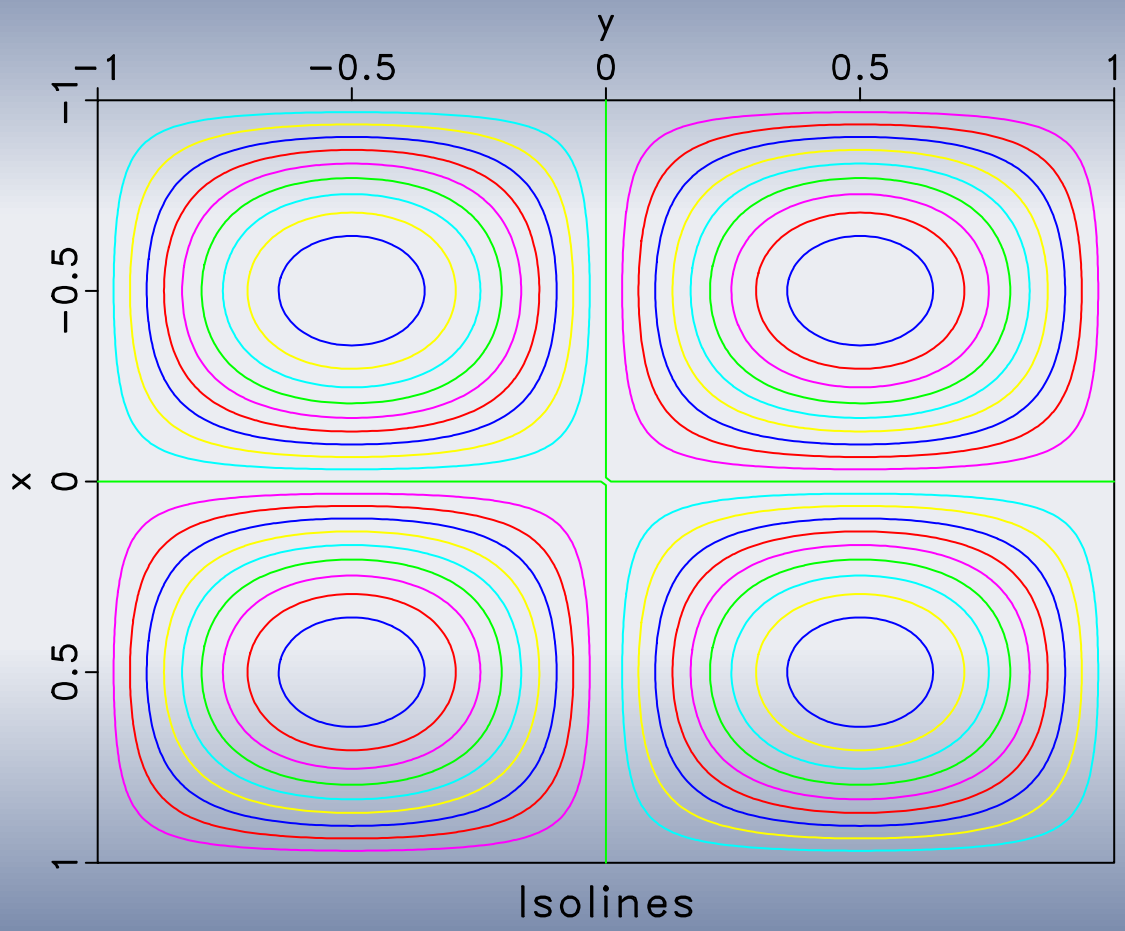
```
Plot('label1',None,'''
     box y0=5.6 x0=8 xt=-1 yt=1 label="Line \F10 a"''')
Plot('mline','mline','graph title="Lines with a label (overlay)"')
Result('mblines','mline label1','Overlay')
```



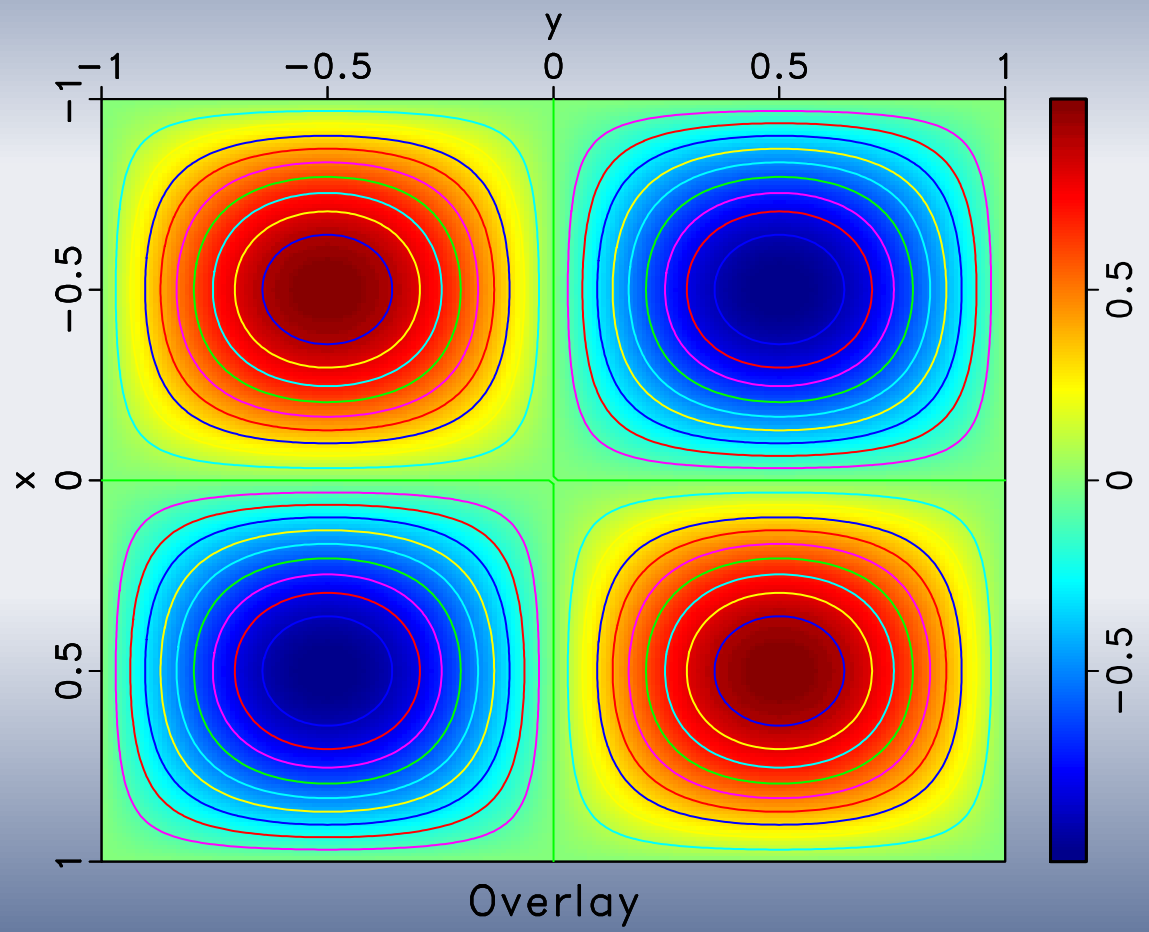
```
Flow('sinsq',None,'''
     math n1=201 n2=201 o1=-1.0 o2=-1.0 d1=0.01 d2=0.01
          output="sin(3.14*x1)*sin(3.14*x2)" label1=x label2=y''')
Result('sqsurf','sinsq','grey color=j title=Surface scalebar=y')
```



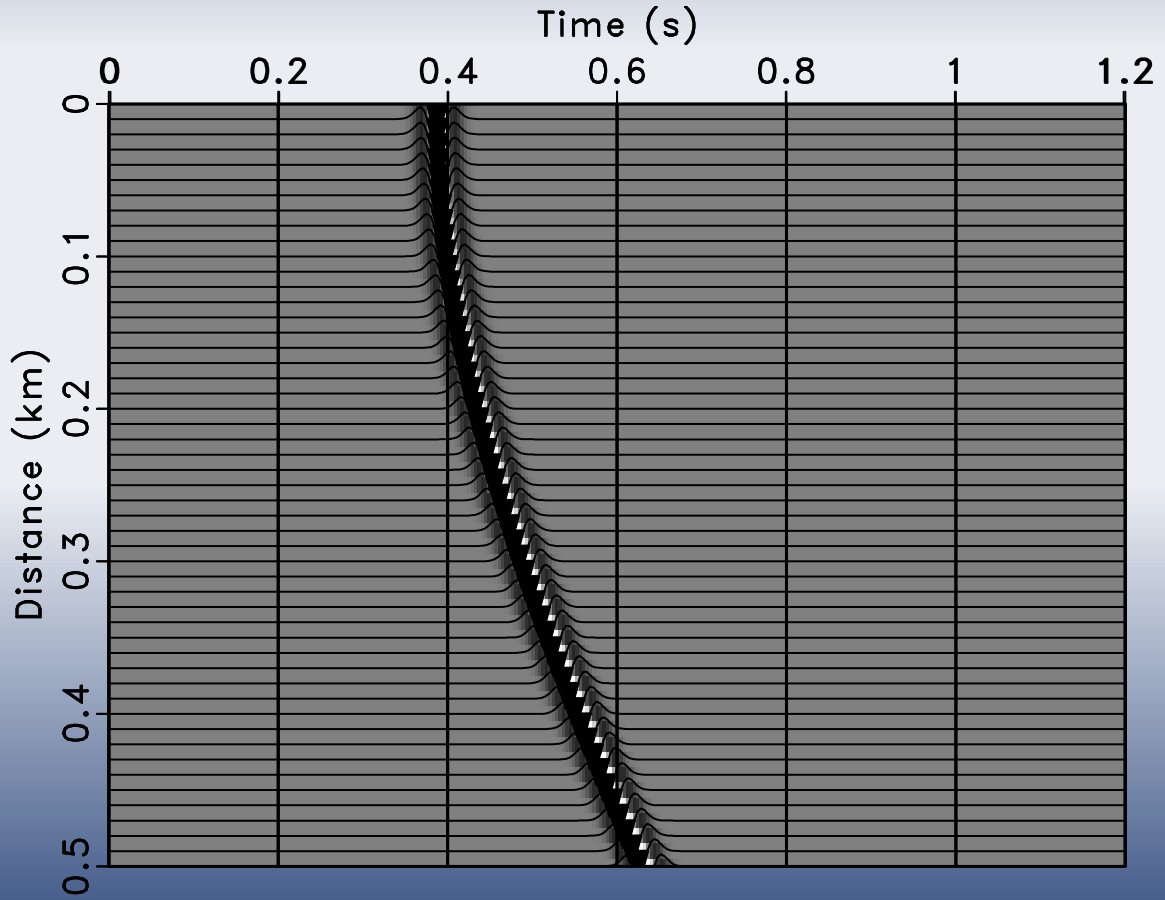
```
Result('csqsurf','sinsq','contour title=Isolines allpos=n')
```



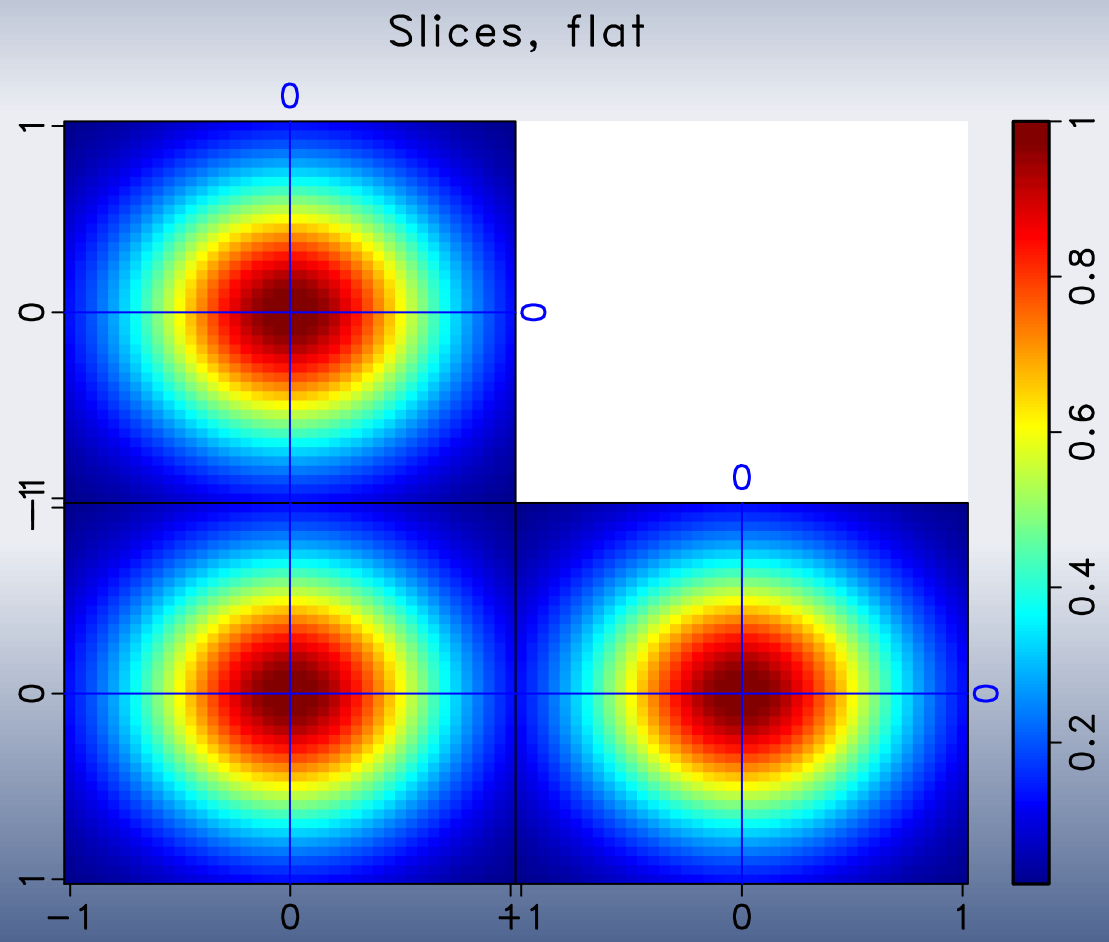
```
Plot('psinsq','sinsq',''' grey color=j title=Overlay scalebar=y
                           min1=-1 max1=1 min2=-1 max2=1''')
Plot('pcsinsq','sinsq','contour title=Overlay allpos=n scalebar=y')
Result('csurf','psinsq pcsinsq','Overlay')
```



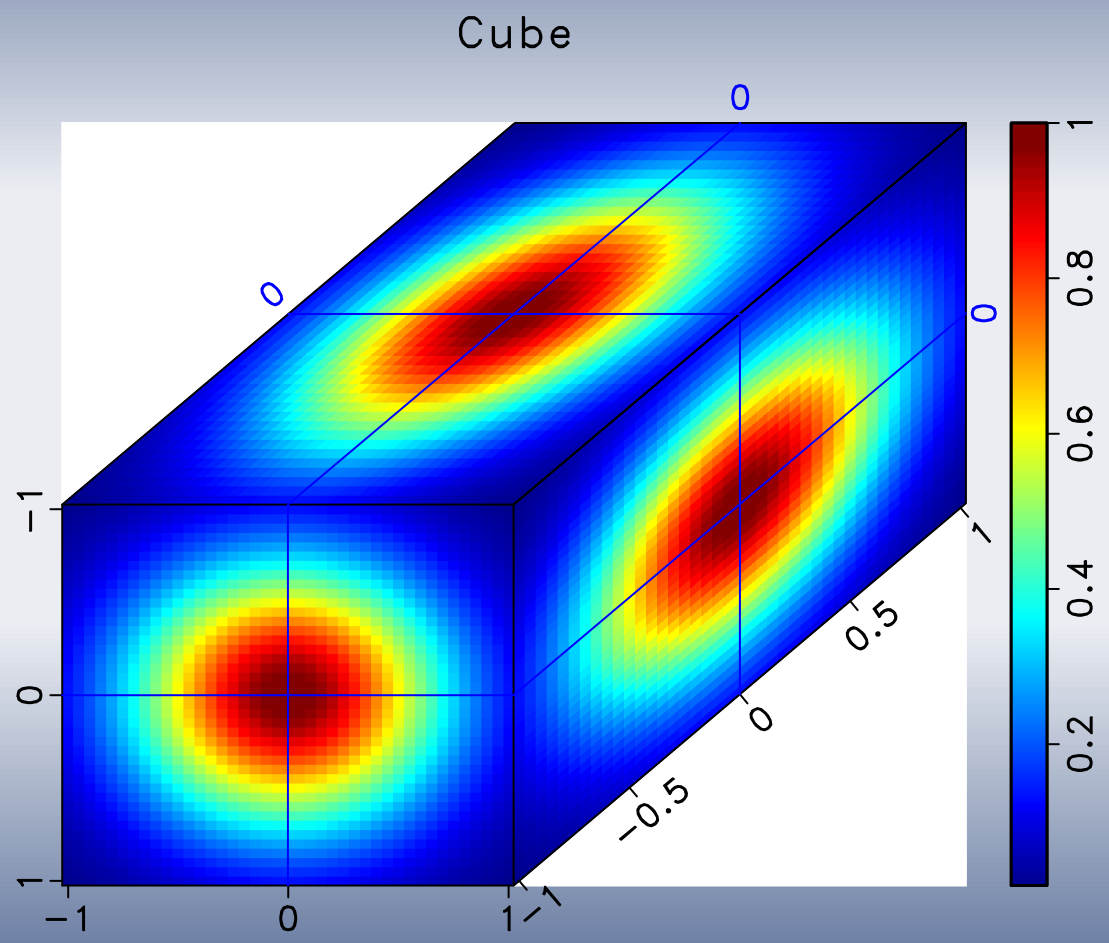
```
Flow('gath',None,'''aspread n1=301 n2=51 d2=0.01
                    nsp=1 pos1="sqrt(0.15+x2^2)" title=''')
Plot('wgath','gath','''wiggle poly=y min2=0 max2=0.5 plotcol=7
                        yreverse=y wherexlabel=top gridcol=7''')
Plot('rgath','gath','grey transp=n min2=0 max2=0.5')
Result('sgath','rgath wgath','Overlay')
```



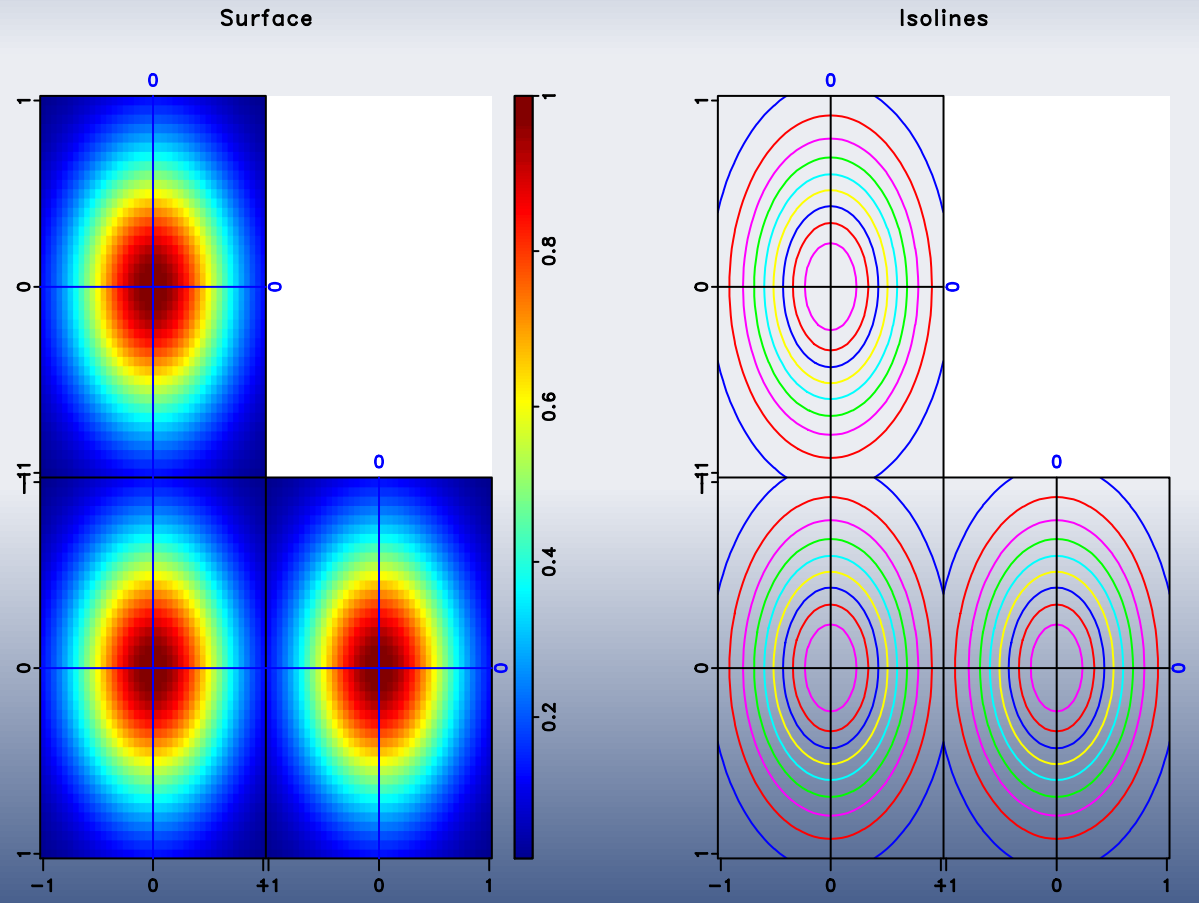
```
Flow('ganom',None,'''math n1=41 n2=41 n3=41 o1=-1 o2=-1 o3=-1
     d1=0.05 d2=0.05 d3=0.05 output="exp(-2*(x1^2+x2^2+x3^2))" ''')
Result('fganom','ganom','''byte gainpanel=all bar=gbar.rsf
       allpos=y | cubeplot color=j scalebar=y bar=gbar.rsf
       title="Slices, flat" frame1=20 frame2=20 frame3=20''')
```



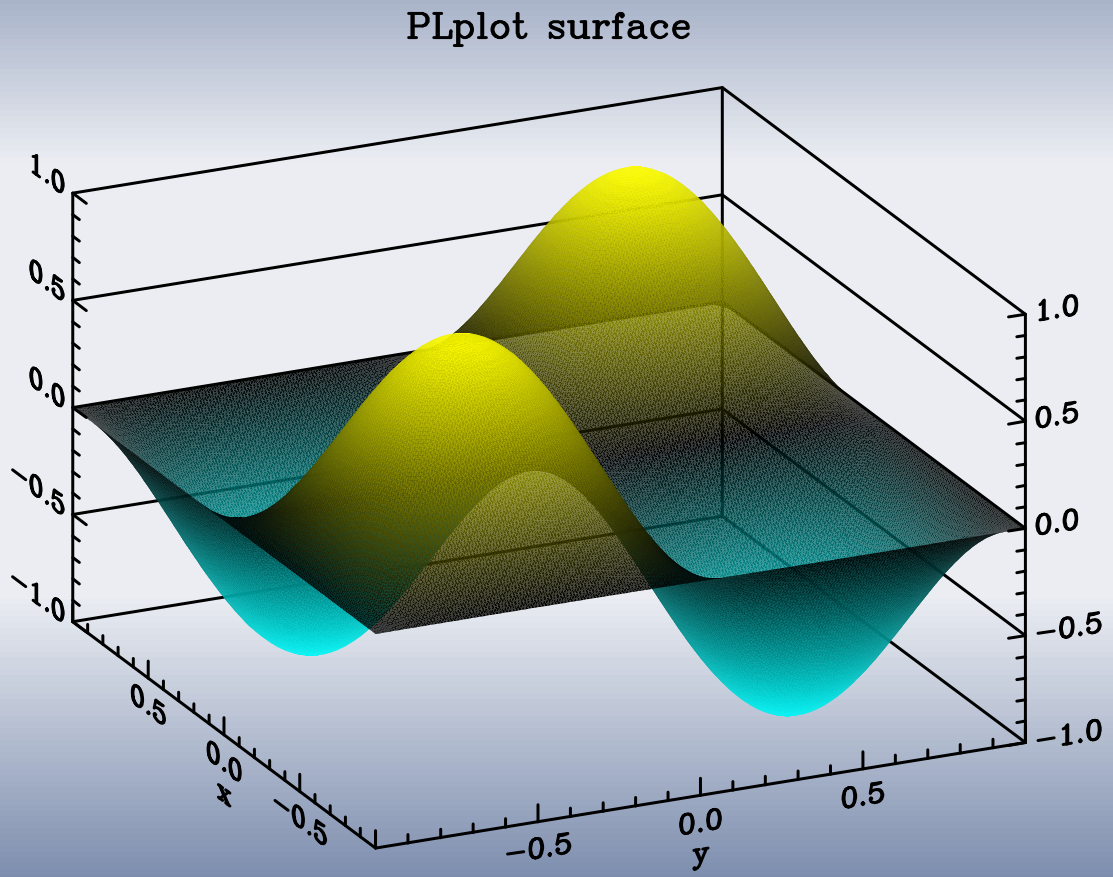
```
Result('ganom3','ganom',''byte gainpanel=all bar=gbar.rsf
       allpos=y | cubeplot color=j scalebar=y bar=gbar.rsf
       title="Cube" frame1=20 frame2=20 frame3=20 flat=n''')
```



```
Plot('pfganom','ganom',''byte gainpanel=all bar=gbar.rsf
     allpos=y | cubeplot color=j scalebar=y bar=gbar.rsf
     title="Surface" frame1=20 frame2=20 frame3=20'')
Plot('pcganom','ganom',''contour3 color=j scalebar=y
     title=Isolines nc=30 frame1=20 frame2=20 frame3=20'')
Result('sganom','pfganom pcganom','SideBySideAniso')
```



```
Result('sqplsin','sinsq','''
       plsurf mesh=n color=e title="PLplot surface"''')
```



Combinations

Movie
Overlay
SideBySideAniso
OverUnderAniso
SideBySideIso
OverUnderIso
TwoRows
TwoColumns

Command line help for common parameters

$ sfdoc stdplot

Use functions to reduce code redundancy

```
def grey(title,transp='n',bias=bias):
    return '''sfgrey title="%s" transp=%s bias=%g
           ''' % (title,transp,bias)
Plot('plot',grey('Functions are useful'))
```

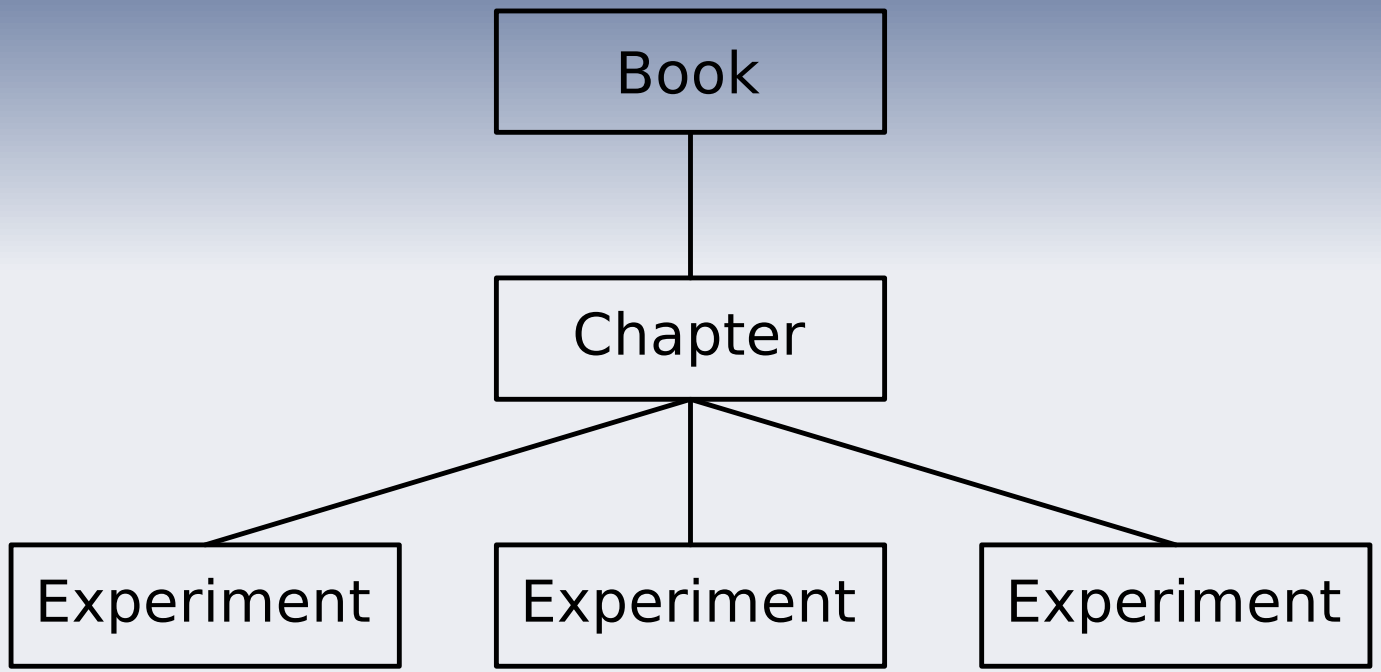


Figure: Directory structure for reproducible documents

paper/sample1/SConstruct

```
from rsf.proj import *

Flow('sin',None,'math n1=101 d1=0.1 output="sin(x1)"')
Result('sin','graph label1=x title="sin(x)"')

End()
```

paper/sample2/SConstruct

```
from rsf.proj import *

Flow('sin2',None,'''math n1=95 n2=95 d1=0.1 d2=0.1
                     output="sin(x1)*sin(x2)"''')
Result('sin2','''grey color=h label1=x label2=y
                   scalebar=y title="sin(x)*sin(y)"''')

End()
```

paper/sample.tex

```
\author{Aye-aye}
%%%%
\title{Sample reproducible paper}
\begin{abstract}
  This is a reproducible paper.
\end{abstract}

\subsection{Plots}
Figure~\ref{fig:sin} and Figure~\ref{fig:sin2} are
examples of reproducible figures.

\inputdir{sample1}
\plot{sin}{width=0.95\textwidth}{First figure.}
\inputdir{sample2}
\plot{sin2}{width=0.95\textwidth}{Second figure.}
```

paper/SConstruct

```
from rsf.tex import *

Paper('sample',color='sin sin2',hires='sin2')

End()
```

Command line

```
$ sftour scons lock
$ scons sample.pdf
$ scons sample.html
$ scons sample.install
```

Sample reproducible paper

Aye-aye

ABSTRACT

This is a reproducible paper.

Plots

Figure 1 and Figure 2 are examples of reproducible figures.

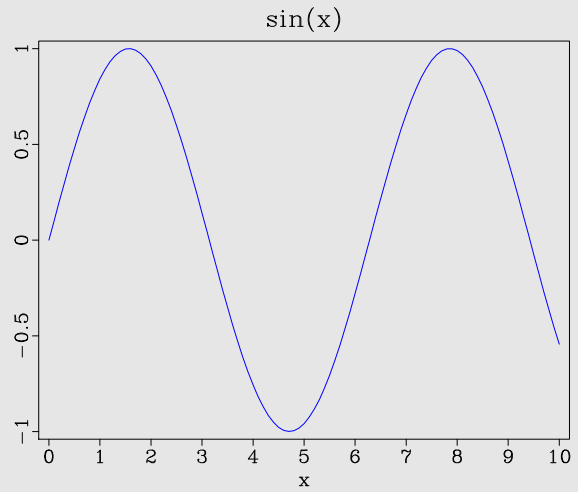


Figure 1: First figure.

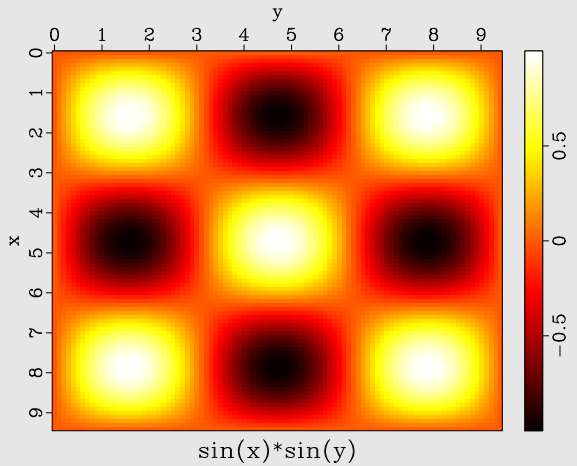


Figure 2: Second figure.

Takeaway message

You do not have to convert anything manually, if you follow the workflow above.