Python Graphique.

Thierry Dumont

Institut Camille Jordan
Université Lyon 1 & CNRS.

Petite liste des outils

- matplotlib `http:matplotlib.sourceforge.net/`
- veusz `http://home.gna.org/veusz/`
- TkPlotCanvas `http://wiki.python.org/moin/TkPlotCanvas`
- Gnuplot `http://www.gnuplot.info/`
- pgplot `http://www.astro.caltech.edu/~tjp/pgplot/`
- py-opendx `http://people.freebsd.org/~rhh/py-opendx/index.html`
- vtk `http://public.kitware.com/VTK/`
- mayavi `http://mayavi.sourceforge.net/`
- ...

- matplotlib
- mayavi.

matplotlib

- 2D.
- *matlab-like* par l'interface *pylab*.

matplotlib

- 2D.
- *matlab-like* par l'interface *pylab*.

Manifeste de l'auteur:
Besoin d'interaction que Matlab n'apporte pas...

matplotlib

Cahier des charges:

- haute qualité (niveau publication),
- sortir du Postscript (pour LaTeX),
- embarcable dans des applications,
- facile à comprendre,
- facile à utiliser.

matplotlib

3 niveaux:

1. interface *pylab*,
2. l'*api matplotlib*,
3. les *backends*.

matplotlib

3 niveaux:

1. interface *pylab*,
2. l'*api matplotlib*,
3. les *backends*.

Backends:

1. postscript,
2. png (sites web),
3. visualisation plus ou moins sophistiquée (antigrain...).

Customisation dans *matplotlibrc*.

```
from pylab import *
# create some data to use for the plot
dt = 0.001
t = arange(0.0, 10.0, dt)
r = exp(-t[:1000]/0.05)                    # impulse response
x = randn(len(t))
s = convolve(x,r,mode=2)[:len(x)]*dt  # colored noise
```

```
from pylab import *
# create some data to use for the plot
dt = 0.001
t = arange(0.0, 10.0, dt)
r = exp(-t[:1000]/0.05)                    # impulse response
x = randn(len(t))
s = convolve(x,r,mode=2)[:len(x)]*dt  # colored noise

# the main axes is subplot(111) by default
plot(t, s)
axis([0, 1, 1.1*amin(s), 2*amax(s) ])
xlabel('time (s)')
ylabel('current (nA)')
title('Gaussian colored noise')
```

```
# this is an inset axes over the main axes
a = axes([.65, .6, .2, .2], axisbg='y')
n, bins, patches = hist(s, 400, normed=1)
title('Probability')
setp(a, xticks=[], yticks=[])
# this is another inset axes over the main axes
a = axes([0.2, 0.6, .2, .2], axisbg='y')
plot(t[:len(r)], r)
title('Impulse response')
setp(a, xlim=(0,.2), xticks=[], yticks=[])
show()
```

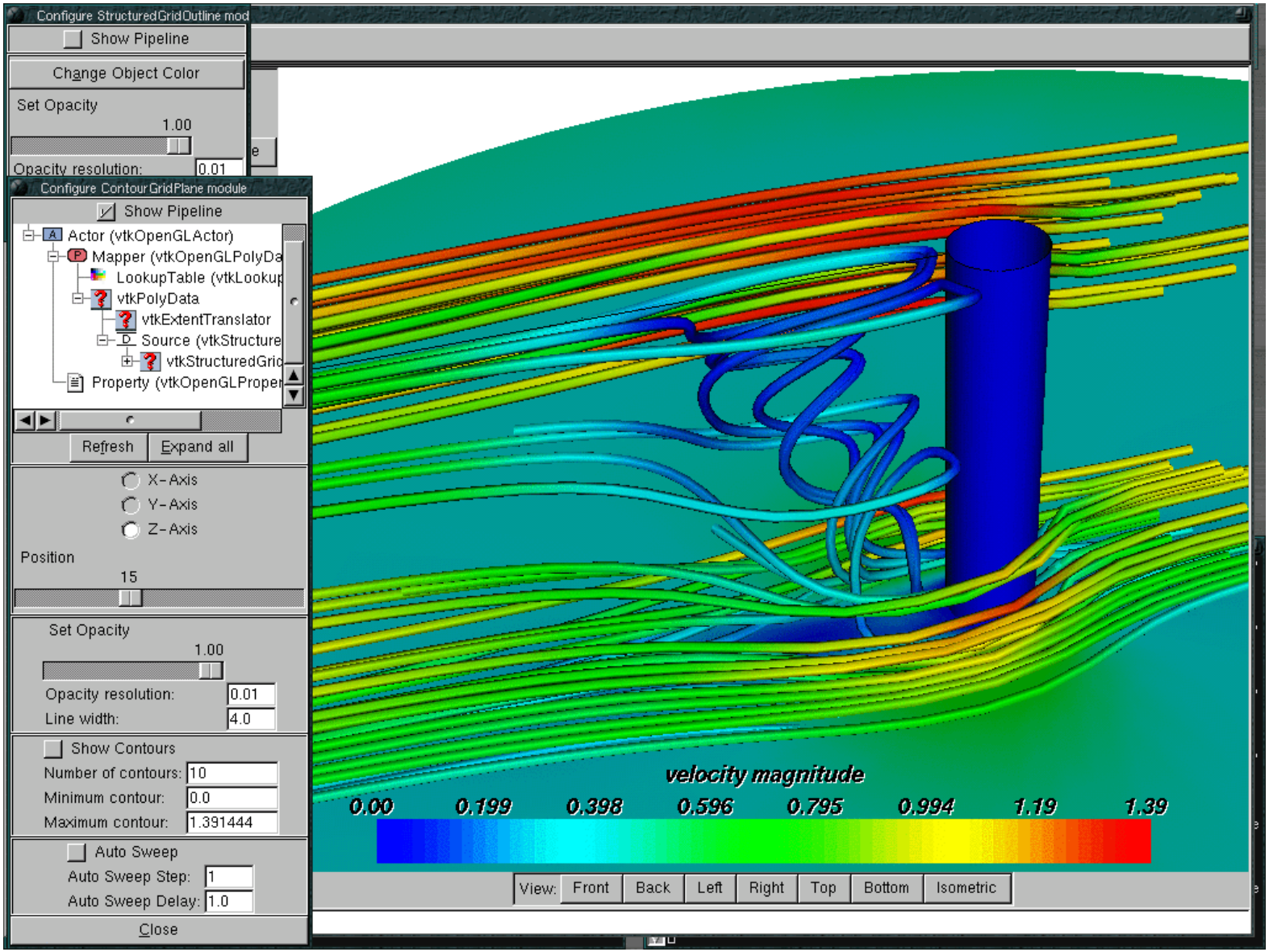
Autres aspects intéresants: textes à la TEX:

```
xlabel(r'\textbf{time (s)}')
ylabel(r'\textit{voltage (mV)}',fontsize=16)
title(r"\TeX\ is Number
 $\displaystyle\sum_{n=1}^\infty
 \frac{-e^{i\pi}}{2^n}$!",
 fontsize=16, color='r')
```

The MayaVi Data Visualizer

Visualisation de données scientifiques.

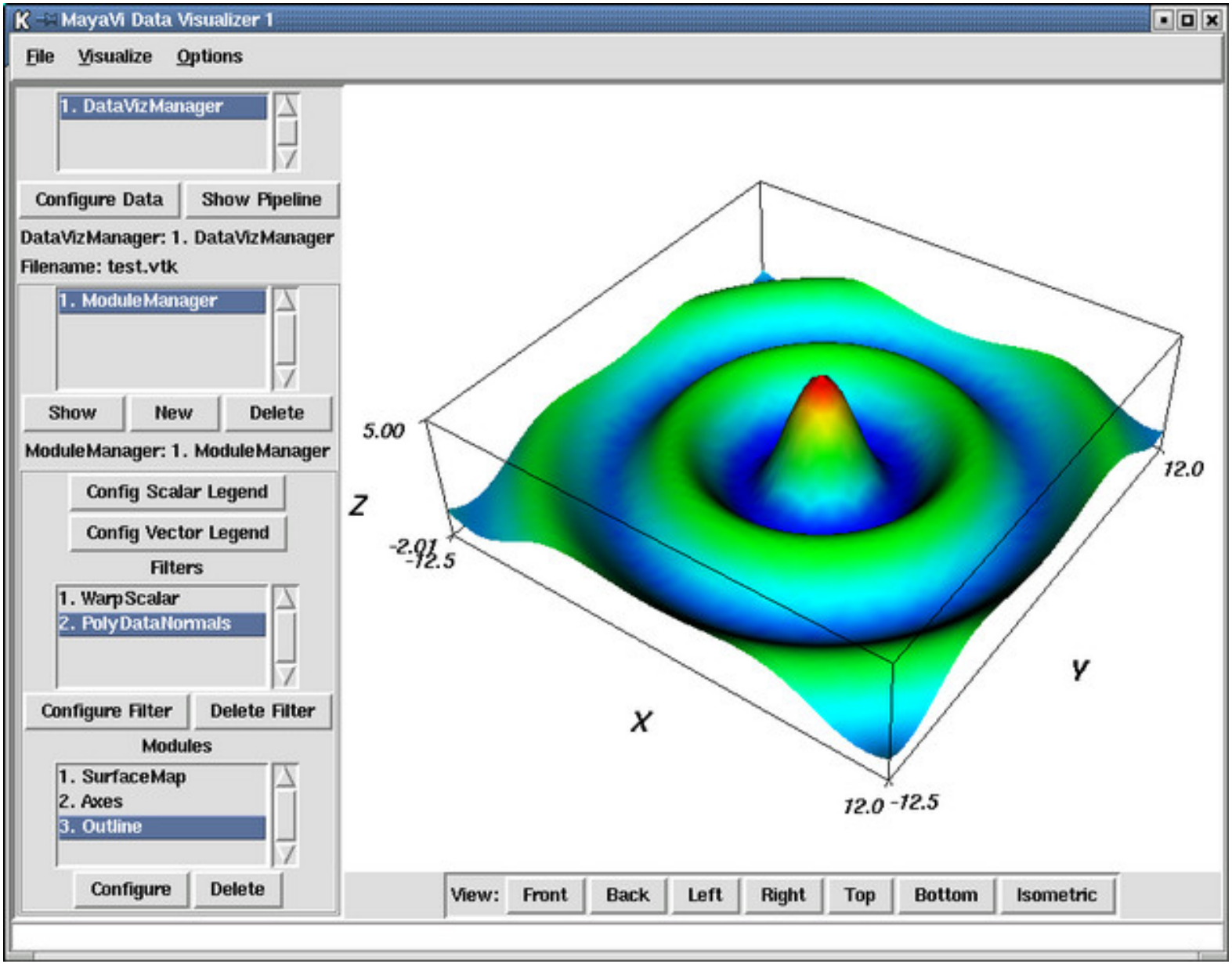
- construit avec VTK (et PYTHON-VTK).
- lecture (entre autres) des données en “VTK data format” (xml).
- “pipe line” d’outils et de filtres.


Configure StructuredGridOutline mod
Show Pipeline
Change Object Color
Set Opacity
1.00
Opacity resolution:
0.01
Configure ContourGridPlane module
Show Pipeline
Actor (vtkOpenGLActor)
Mapper (vtkOpenGLPolyDa
LookupTable (vtkLookup
vtkPolyData
vtkExtentTranslator
Source (vtkStructure
vtkStructuredGric
Property (vtkOpenGLProper
Refresh
Expand all
X-Axis
Y-Axis
Z-Axis
Position
15
Set Opacity
1.00
Opacity resolution:
0.01
Line width:
4.0
Show Contours
Number of contours:
10
Minimum contour:
0.0
Maximum contour:
1.391444
Auto Sweep
Auto Sweep Step:
1
Auto Sweep Delay:
1.0
Close
velocity magnitude
0.00
0.199
0.398
0.596
0.795
0.994
1.19
1.39
View:
Front
Back
Left
Right
Top
Bottom
Isometric

Utilisation depuis Python

1. engendrer des fichiers vtk: `import pyvtk`...
2. “scripter” mayavi: `import mayavi`.

```
import mayavi
v = mayavi.mayavi() # create a MayaVi window.
d = v.open_vtk('/tmp/test.vtk', config=0) # open th
# The config option turns on/off showing a GUI con
# load the filters.
f = v.load_filter('WarpScalar', config=0)
n = v.load_filter('PolyDataNormals', 0)
n.fil.SetFeatureAngle (45) # configure the normals
# Load the necessary modules.
m = v.load_module('SurfaceMap', 0)
a = v.load_module('Axes', 0)
a.axes.SetCornerOffset(0.0) # configure the axes m
o = v.load_module('Outline', 0)
v.Render() # Re-render the scene.
```


MayaVi Data Visualizer 1
File
Visualize
Options
1. DataVizManager
Configure Data
Show Pipeline
DataVizManager: 1. DataVizManager
Filename: test.vtk
1. ModuleManager
Show
New
Delete
ModuleManager: 1. ModuleManager
Config Scalar Legend
Config Vector Legend
Filters
1. WarpScalar
2. PolyDataNormals
Configure Filter
Delete Filter
Modules
1. SurfaceMap
2. Axes
3. Outline
Configure
Delete
5.00
Z
-2.01
-12.5
12.0
Y
X
12.0
-12.5
View:
Front
Back
Left
Right
Top
Bottom
Isometric

Projet en sommeil quelque temps, mais Mayavi2 en route.