3D data Visualization

FOSSEE

Department of Aerospace Engineering
IIT Bombay

26 September, 2010
Day 2, Session 7

Outline

1. 3D Data Visualization
2. Tools available
    - mlab
    - Mayavi2

Outline

1. 3D Data Visualization

2. Tools available
   - mlab
   - Mayavi2

What is visualization?

Visual representation of data

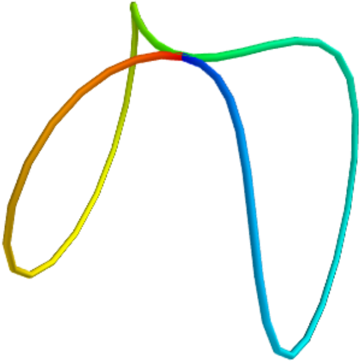
3D visualization

Harder but important

Is this Graphics?

Visualization is about data!

Examples: trajectory in space



Examples: Fire in a room

Demo of data

10 m

Outline

1 3D Data Visualization

2 Tools available

- mlab
- Mayavi2

Outline

1. 3D Data Visualization
2. Tools available
   - mlab
   - Mayavi2

Overview

- Simple
- Convenient
- Full-featured

Getting started

Vanilla:

```
$ ipython -wthread
```

with Pylab:

```
$ ipython -pylab -wthread
```

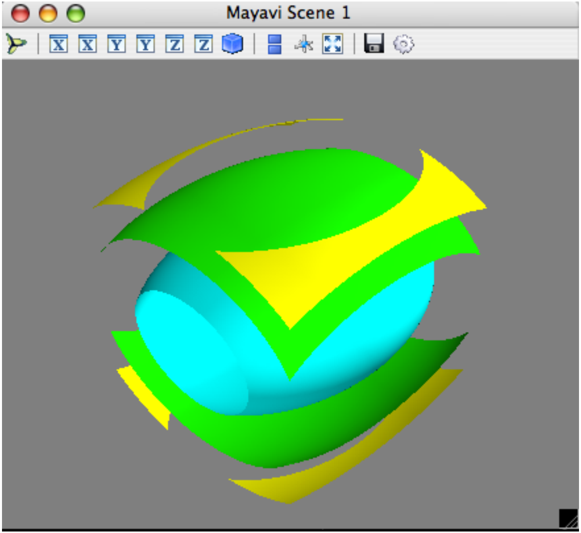
Using mlab

```
In []:from enthought.mayavi import mlab
```

Try these

```
In []: mlab.test_<TAB>
In []: mlab.test_contour3d()
In []: mlab.test_contour3d??
```

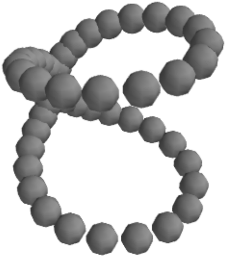
Exploring the view



- Mouse
- Keyboard
- Toolbar
- Mayavi icon

mlab plotting functions

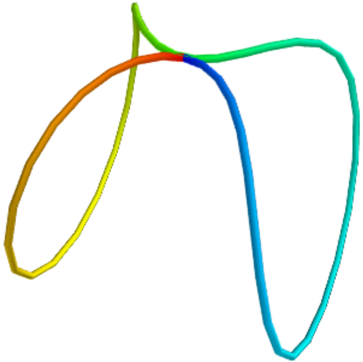
0D data



```
In []: t = linspace(0, 2*pi, 50)
In []: u = cos(t) * pi
In []: x, y, z = sin(u), cos(u), sin(t)
```

```
In []: mlab.points3d(x, y, z)
```

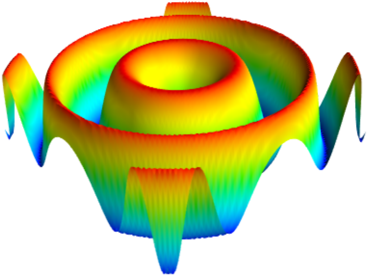
1D data



```
In []: mlab.plot3d(x, y, z, t)
```

Plots lines between the points

2D data



```
In []: x, y = mgrid[-3:3:100j,-3:3:100j]
In []: z = sin(x*x + y*y)
```

```
In []: mlab.surf(x, y, z)
```

Assumes the points are rectilinear

mgrid

```
In []: mgrid[0:3,0:3]
Out[]:
array([[[0, 0, 0],
        [1, 1, 1],
        [2, 2, 2]],

       [[0, 1, 2],
        [0, 1, 2],
        [0, 1, 2]]])

In []: mgrid[-1:1:5j]
Out[]: array([-1., -0.5,  0.,  0.5,  1.])
```

Example

```
In []: x, y = mgrid[-1:1:5j, -1:1:5j]
In []: z = x*x + y*y

In []: z
Out[]:
array([[ 2.  , 1.25, 1.  , 1.25, 2.  ],
       [ 1.25, 0.5 , 0.25, 0.5 , 1.25],
       [ 1.  , 0.25, 0.  , 0.25, 1.  ],
       [ 1.25, 0.5 , 0.25, 0.5 , 1.25],
       [ 2.  , 1.25, 1.  , 1.25, 2.  ]])
```

2D data: mlab.mesh

```
In []: mlab.mesh(x, y, z)
```

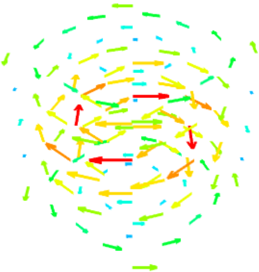
Points needn't be regular

```
In []: phi, theta = mgrid[0:pi:20j,
...                        0:2*pi:20j]
In []: x = sin(phi)*cos(theta)
In []: y = sin(phi)*sin(theta)
In []: z = cos(phi)
In []: mlab.mesh(x, y, z,
...             representation=
...             'wireframe')
```

3D data

```
In []: x, y, z = mgrid[-5:5:64j,
...                       -5:5:64j,
...                       -5:5:64j]
In []: mlab.contour3d(x*x*0.5 + y*y +
                      z*z*2)
```

3D vector data: `mlab.quiver3d`



```
In []: mlab.test_quiver3d()
```

```
obj = mlab.quiver3d(x, y, z, u, v, w)
```

30 m

Outline

1. 3D Data Visualization
2. Tools available
   - mlab
   - Mayavi2

Introduction to Mayavi

- Most scientists not interested in details of visualization
- Visualization of data files with a nice UI
- Interactive visualization of data (think Matlab)
- Embedding visualizations in applications
- Customization

The Goal

Provide a flexible library/app for all of these needs!

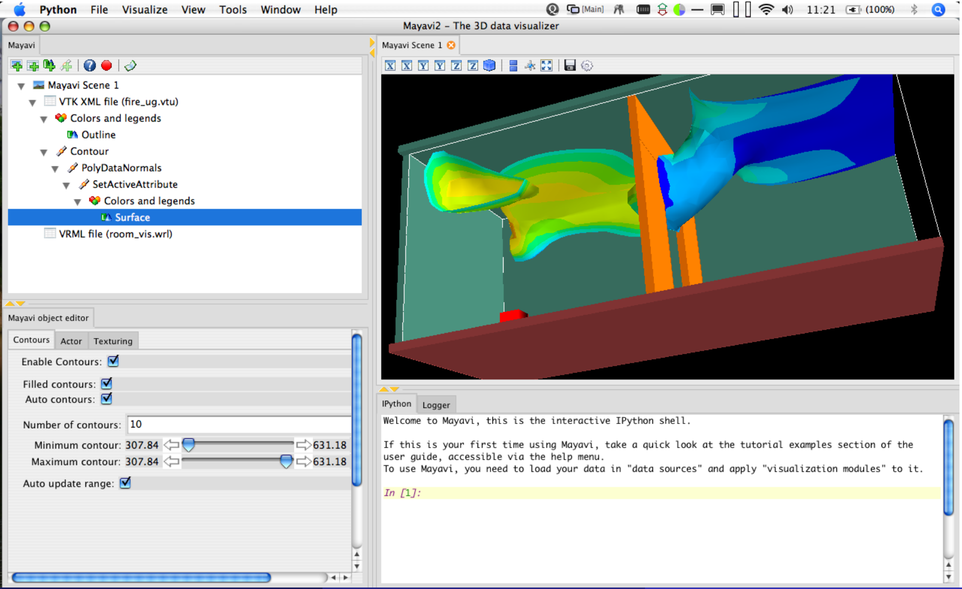
Introduction to Mayavi

- Most scientists not interested in details of visualization
- Visualization of data files with a nice UI
- Interactive visualization of data (think Matlab)
- Embedding visualizations in applications
- Customization

The Goal

Provide a flexible library/app for all of these needs!

Overview of features



Mayavi2 - The 3D data visualizer

Mayavi Scene 1
- VTK XML file (fire_ug.vtu)
  - Colors and legends
    - Outline
  - Contour
    - PolyDataNormals
      - SetActiveAttribute
        - Colors and legends
          - Surface
- VRML file (room_vis.wrl)

Mayavi object editor

Contours | Actor | Texturing

Enable Contours: ☑

Filled contours: ☑

Auto contours: ☑

Number of contours: 10

Minimum contour: 307.84 ... 631.18

Maximum contour: 307.84 ... 631.18

Auto update range: ☑

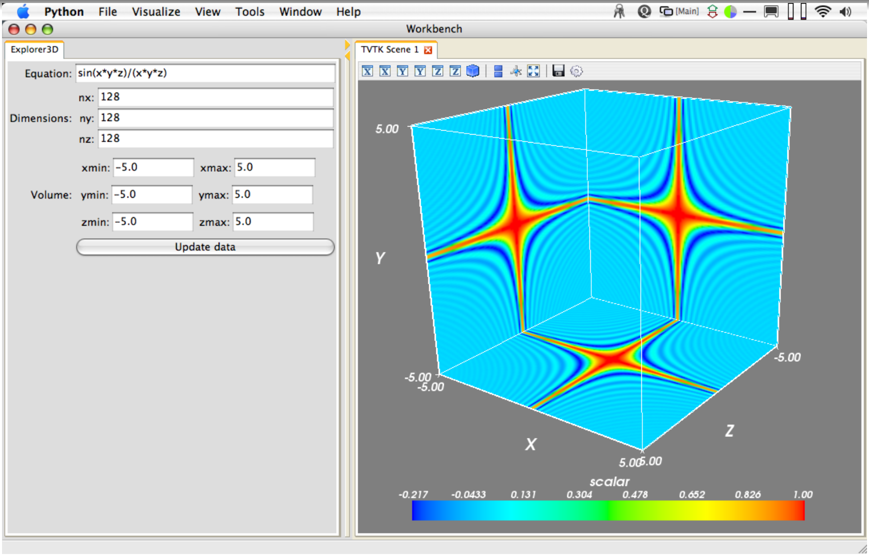
IPython | Logger

```
Welcome to Mayavi, this is the interactive IPython shell.

If this is your first time using Mayavi, take a quick look at the tutorial examples section of the
user guide, accessible via the help menu.
To use Mayavi, you need to load your data in "data sources" and apply "visualization modules" to it.

In [1]:
```

Mayavi in applications

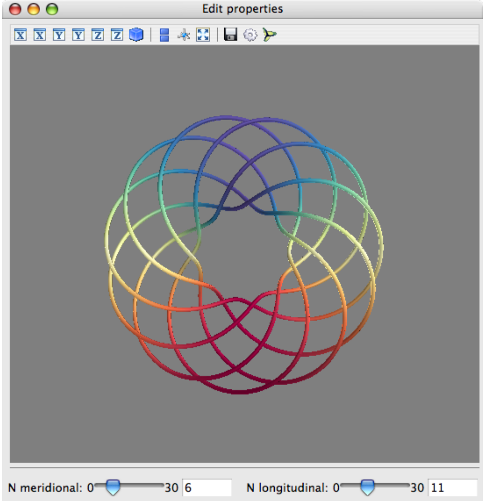


Equation: sin(x*y*z)/(x*y*z)

Dimensions: nx: 128 ny: 128 nz: 128

Volume: xmin: -5.0 xmax: 5.0 ymin: -5.0 ymax: 5.0 zmin: -5.0 zmax: 5.0

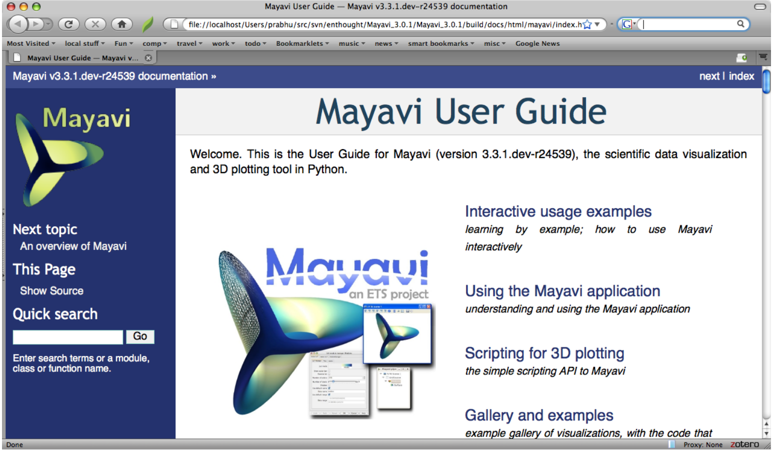
Live in your dialogs



N meridional: 0 30 6

N longitudinal: 0 30 11

Exploring the documentation



Mayavi User Guide — Mayavi v3.3.1.dev-r24539 documentation
file://localhost/Users/prabhu/src/svn/enthought/Mayavi_3.0.1/Mayavi_3.0.1/build/docs/html/mayavi/index.h
Most Visited
local stuff
Fun
comp
travel
work
todo
Bookmarklets
music
news
smart bookmarks
misc
Google News
Mayavi User Guide — Mayavi v...
Mayavi v3.3.1.dev-r24539 documentation »
next | index
Mayavi
Next topic
This Page
Quick search
Go
Mayavi
an ETS project
Done
Proxy: None
zotero

An overview of Mayavi

Show Source

Enter search terms or a module, class or function name.

Mayavi User Guide

Welcome. This is the User Guide for Mayavi (version 3.3.1.dev-r24539), the scientific data visualization and 3D plotting tool in Python.

Interactive usage examples
learning by example; how to use Mayavi interactively

Using the Mayavi application
understanding and using the Mayavi application

Scripting for 3D plotting
the simple scripting API to Mayavi

Gallery and examples
example gallery of visualizations, with the code that

Summary

- http://code.enthought.com/projects/mayavi
- Uses VTK (www.vtk.org)
- BSD license
- Linux, win32 and Mac OS X
- Highly scriptable
- Embed in Traits UIs (wxPython and PyQt4)
- Envisage Plugins
- Debian/Ubuntu/Fedora
- Pythonic

40 m

Getting hands dirty!

Motivational problem

Atmospheric data of temperature over the surface of the earth. Let temperature (T) vary linearly with height (z):

$$T = 288.15 - 6.5z$$

Simple solution

```
lat = linspace(-89, 89, 37)
lon = linspace(0, 360, 37)
z = linspace(0, 100, 11)
```

```
x, y, z = mgrid[0:360:37j,-89:89:37j,
                0:100:11j]
t = 288.15 - 6.5*z
mlab.contour3d(x, y, z, t)
mlab.outline()
mlab.colorbar()
```

Simple solution

```
lat = linspace(-89, 89, 37)
lon = linspace(0, 360, 37)
z = linspace(0, 100, 11)
```

```
x, y, z = mgrid[0:360:37j,-89:89:37j,
                0:100:11j]
t = 288.15 - 6.5*z
mlab.contour3d(x, y, z, t)
mlab.outline()
mlab.colorbar()
```

Exercise: Lorenz equation

$$\begin{aligned}\frac{dx}{dt} &= s(y-x)\\ \frac{dy}{dt} &= rx-y-xz\\ \frac{dz}{dt} &= xy-bz\end{aligned}$$

Let $s = 10$,
$r = 28$,
$b = 8./3.$

Region of interest

```
x, y, z = mgrid[-50:50:20j,-50:50:20j,
                -10:60:20j]
```

60 m

Solution

```
def lorenz(x,y,z,s=10.,r=28.,b=8./3.):
    u = s*(y-x)
    v = r*x-y-x*z
    w = x*y-b*z
    return u,v,w
x,y,z = mgrid [-50:50:20j,-50:50:20j,
                     -10:60:20j ]
u,v,w = lorenz( x , y , z )
# Your plot here
#
mlab.show()
```

We have covered:

- Need of visualization.
- Using mlab to create 3 D plots.
- Mayavi Toolkit.