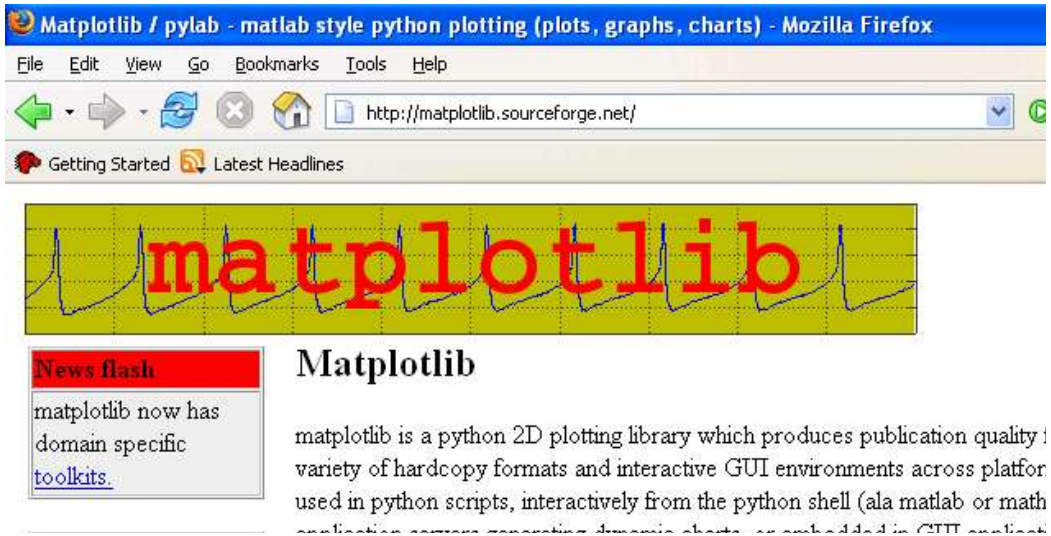
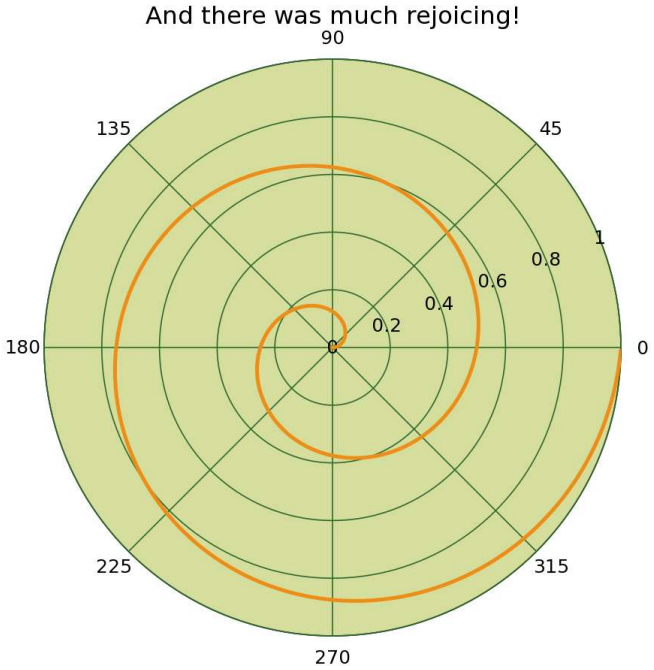
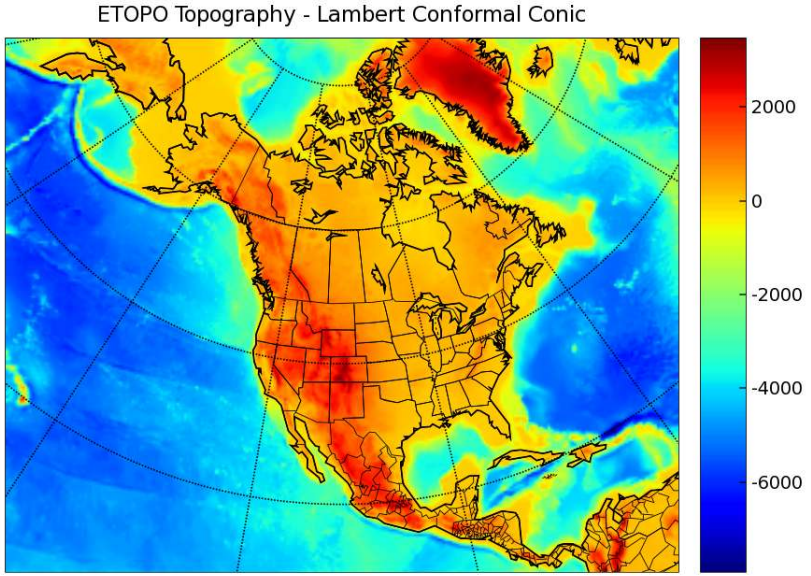
http://matplotlib.sourceforge.net



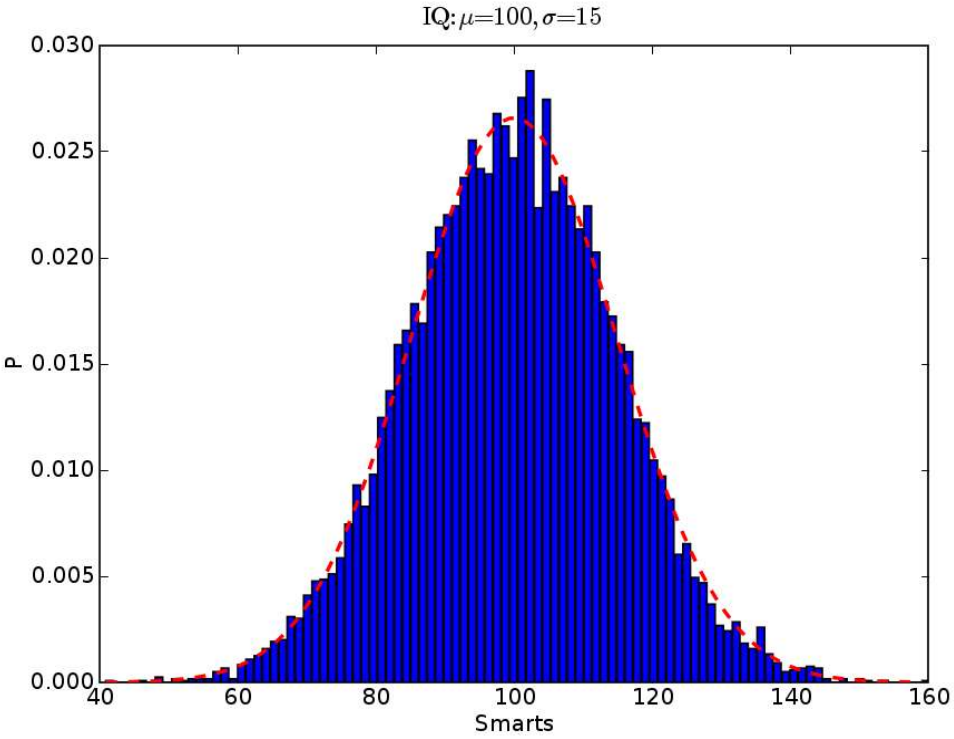


Matplotlib

- General purpose, matlab-like environment
- Embeddable in 6 GUI toolkits
- 3000 downloads/month
- Co-developed with NASA Jet Propulsion Labs, Hubble STScI, NOAA and others

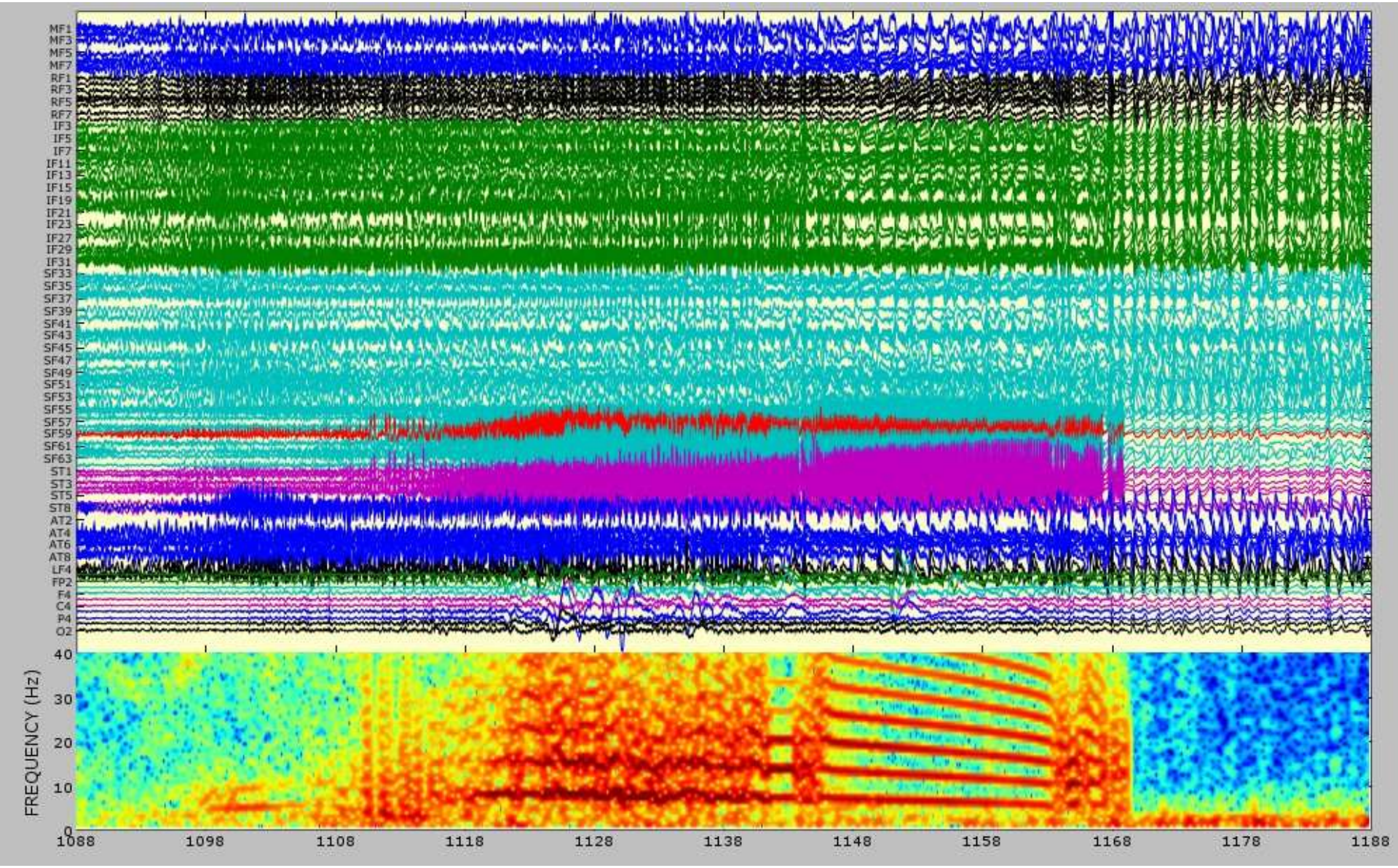


Easy things should be easy...




```
mu, sigma = 100, 15
x = mu + sigma*randn(10000)

# the histogram of the data
n, bins, patches = hist(x, 100, normed=1)

# add a 'best fit' line
y = normpdf( bins, mu, sigma)
l = plot(bins, y, 'r--', linewidth=2)
xlim(40, 160)

xlabel('Smarts')
ylabel('P')
title(r'$\rm{IQ:}\/ \mu=100,\/ \sigma=15$')
```

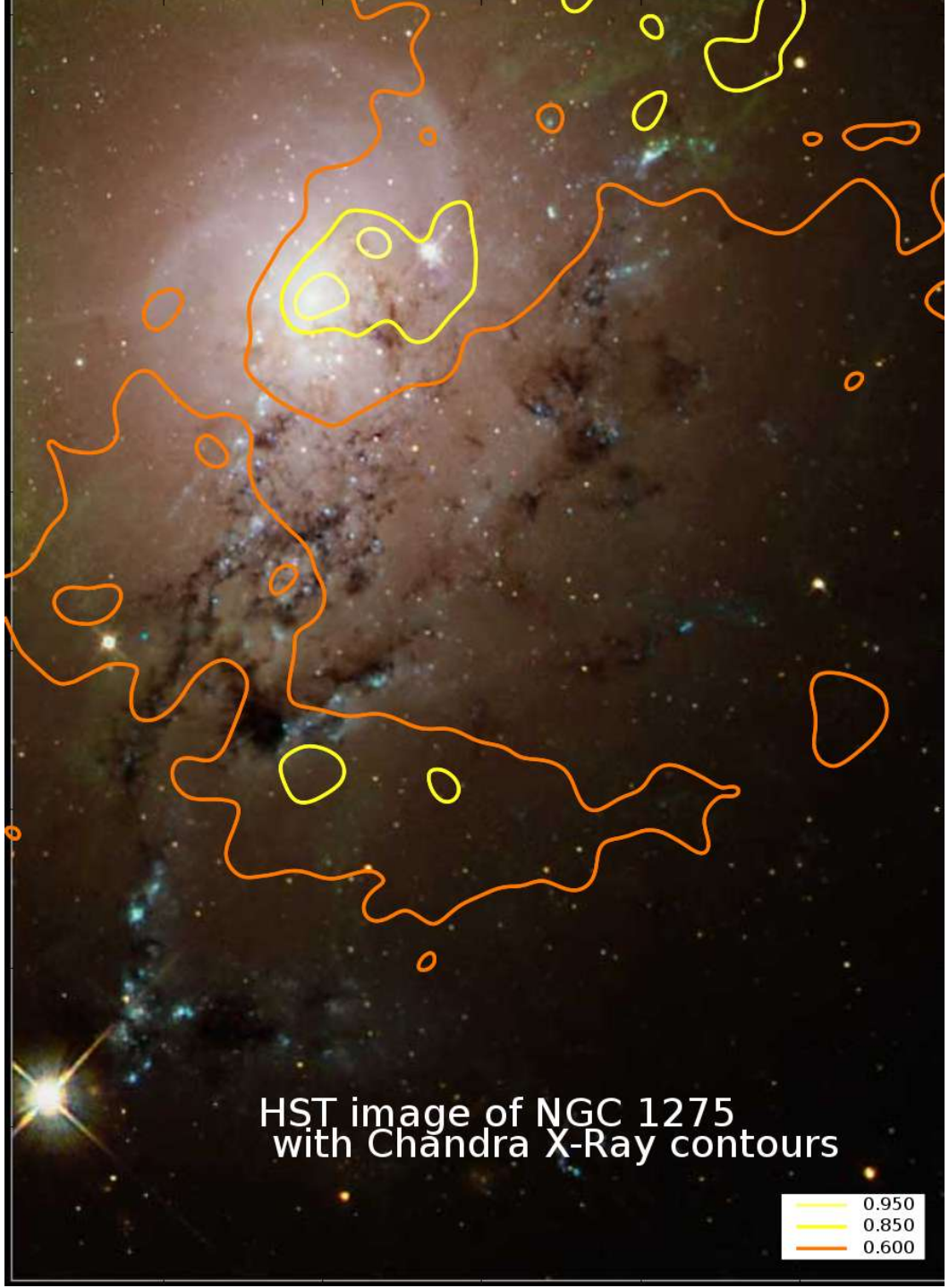
Hard things should be possible



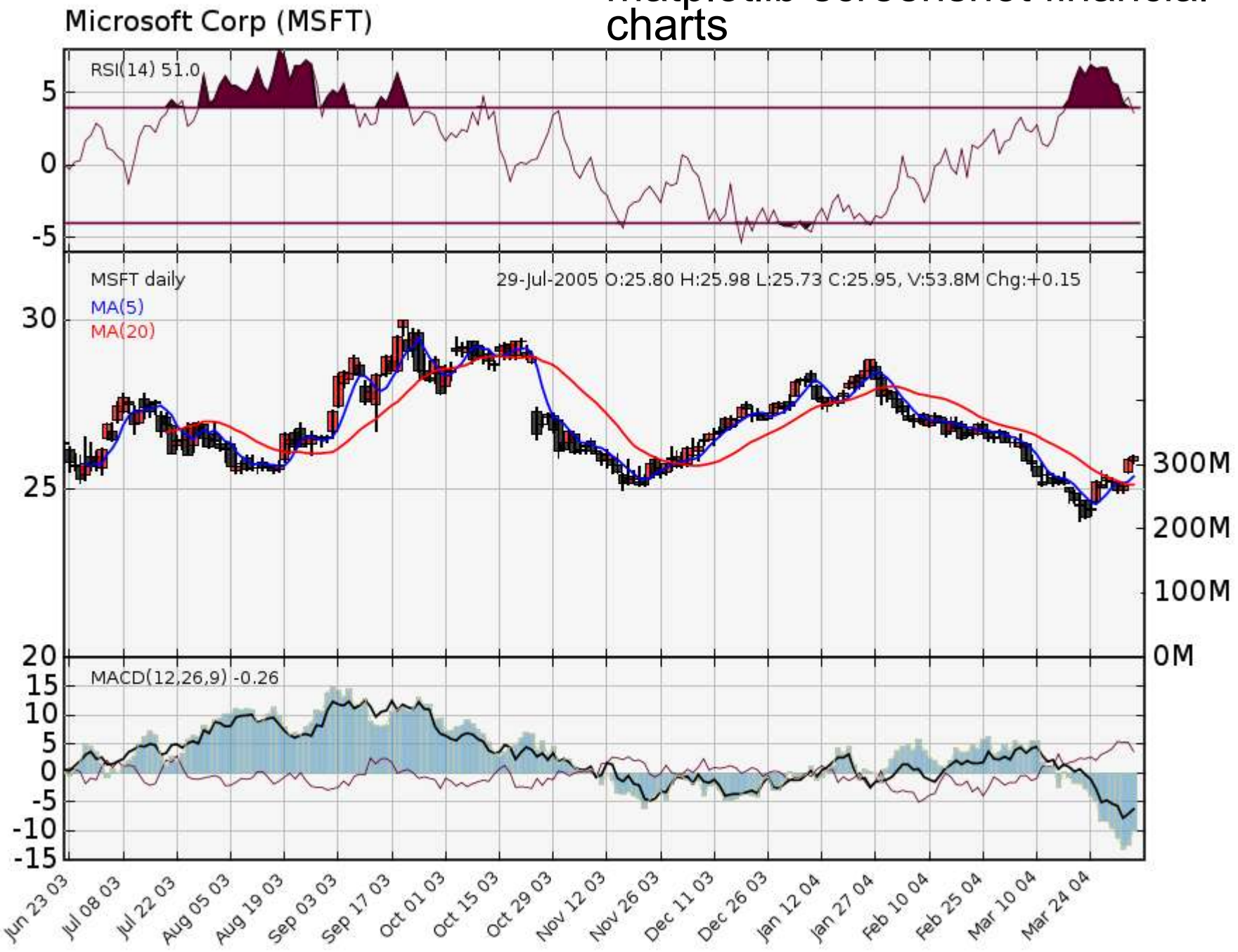
Overview of features

- Most basic charts supported: psd, hist, plot, scatter, polar, pie, bar, errorbar, images, pseudo-colors, ...
- Full Numeric/numarray compatibility at python and extension code layer.
- Matlab compatible “pylab” interface and object oriented API
- Embeddable in 6 GUI toolkits: Tk, Qt, GTK, WX, Cocoa, FLTK
- GUI neutral event handling, drawing, and widgets
- Mathematical expressions with self-contained parser/layout engine or TeX/LaTeX integration
- Interactive support from ipython
- Raster and Vector outputs (PNG, SVG, PS, ...)
- WC3 compliant cross platform font management, unicode support
- Antialiasing, alpha transparency
- Active developer community / mailing lists

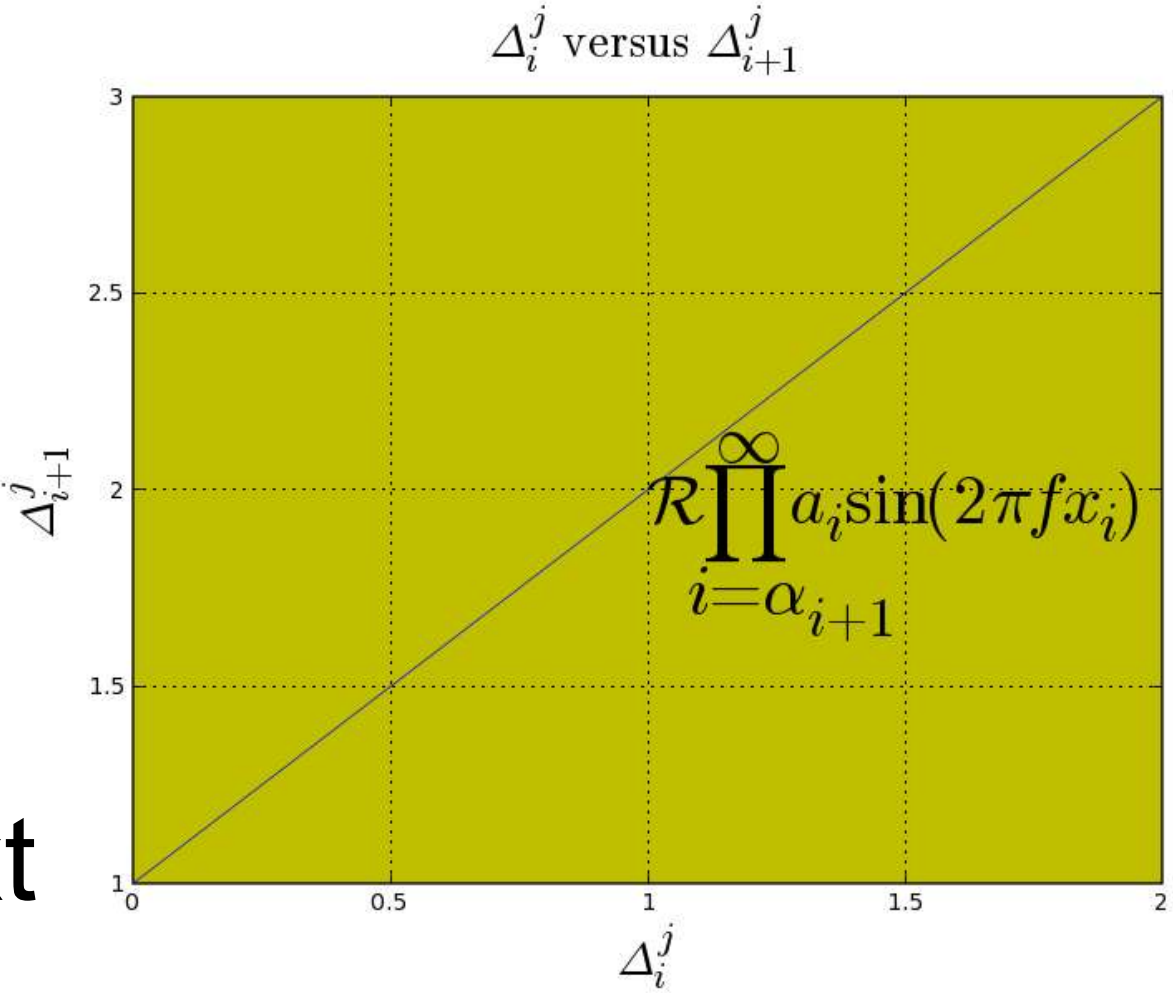
matplotlib screenshot Hubble Space Telescope (courtesy of STScI)



matplotlib screenshot financial charts



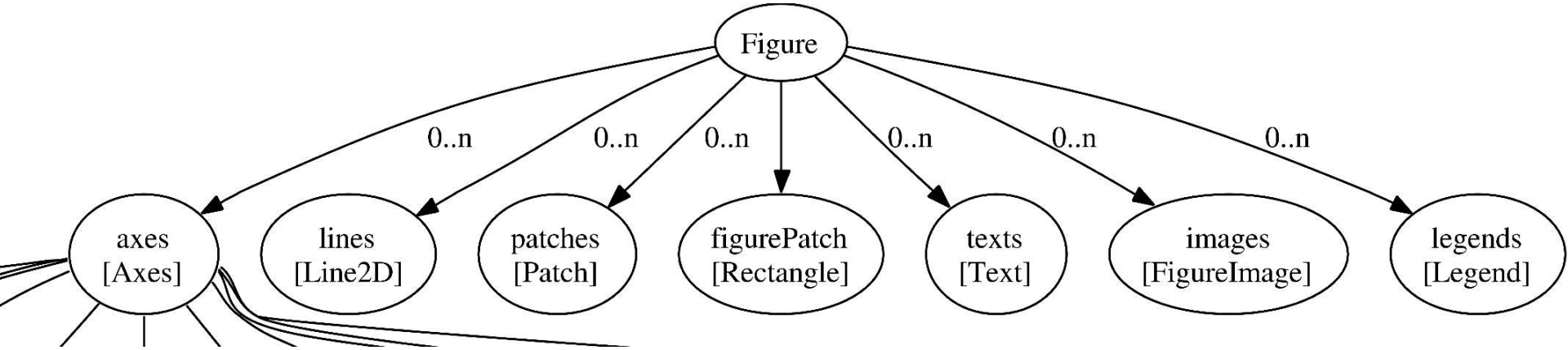
```
tex = r'$\cal{R}\prod_{i=\alpha_{i+1}}^\infty a_I  …
          \rm{sin}(2 \pi f x_i)$'
text(1, 1.6, tex, fontsize=30)
```

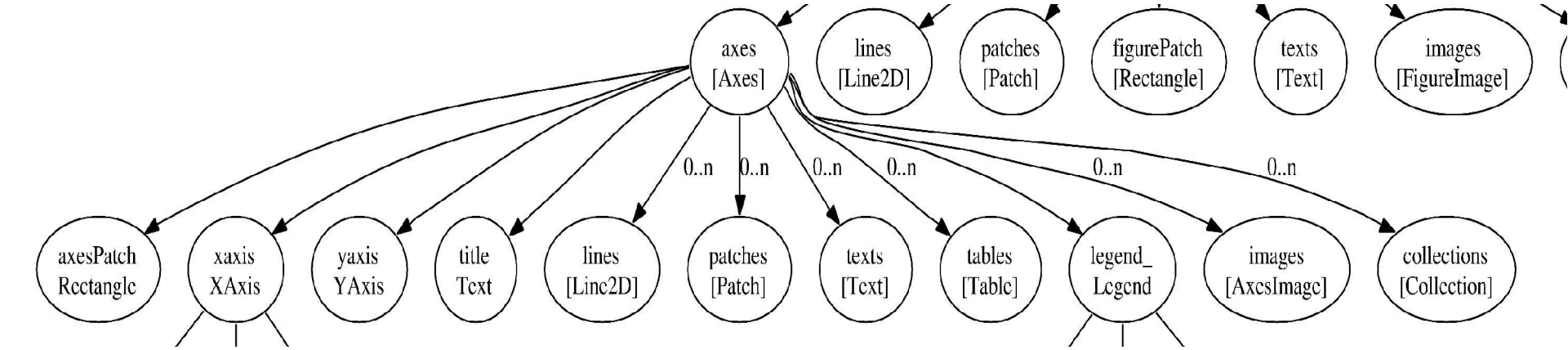


mathtext demo

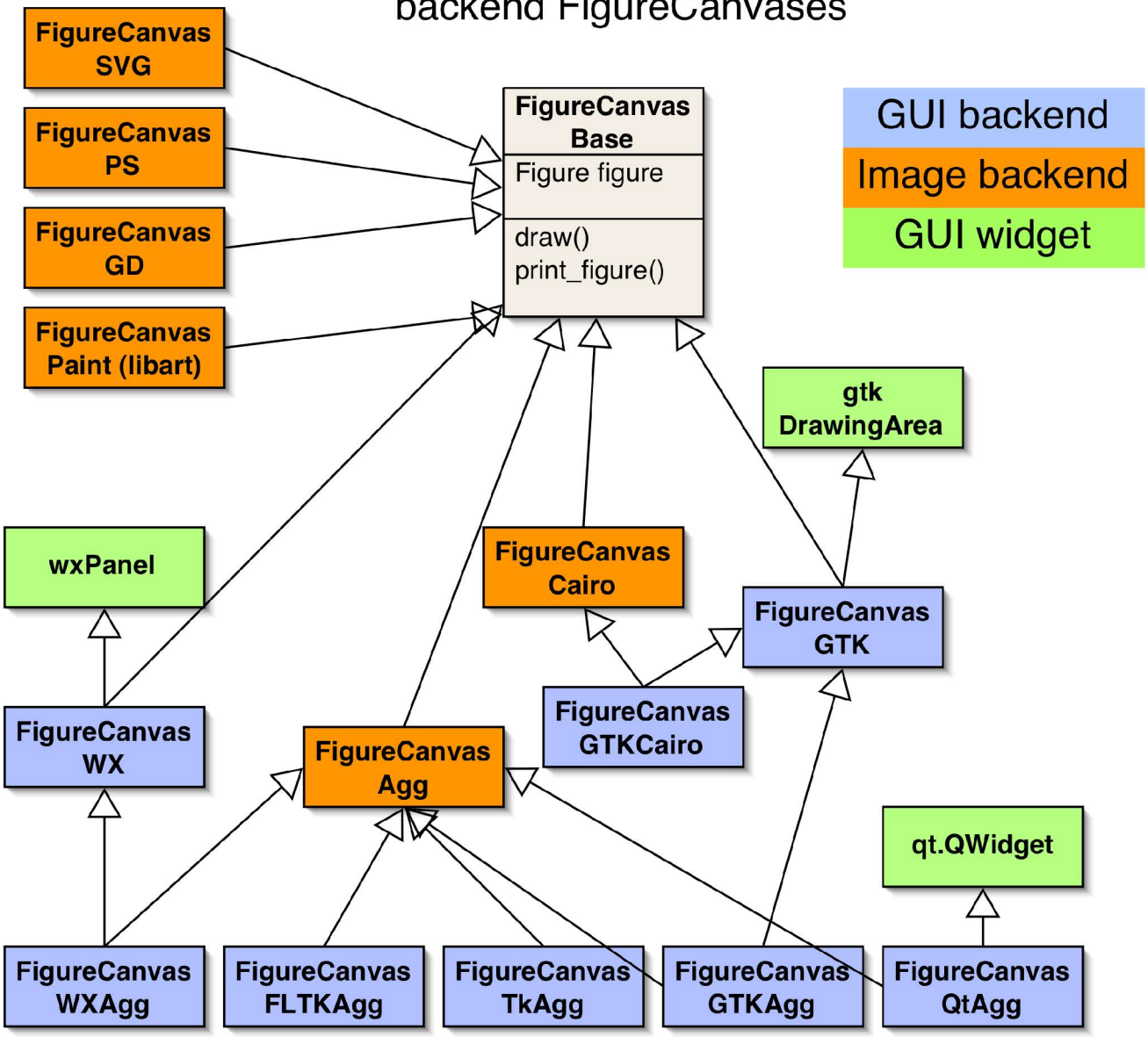
Off the beaten path:
The Matplotlib OO interface

Matplotlib API: Artist containment





backends


Inheritance diagram for backend FigureCanvases
GUI backend
Image backend
GUI widget
FigureCanvas SVG
FigureCanvas PS
FigureCanvas GD
FigureCanvas Paint (libart)
FigureCanvas Base
Figure figure
draw()
print_figure()
gtk DrawingArea
wxPanel
FigureCanvas Cairo
FigureCanvas GTK
FigureCanvas WX
FigureCanvas GTKCairo
FigureCanvas Agg
qt.QWidget
FigureCanvas WXAgg
FigureCanvas FLTKAgg
FigureCanvas TkAgg
FigureCanvas GTKAgg
FigureCanvas QtAgg

Matplotlib API: Embedding in a GUI

```
import gtk

win = gtk.Window()
win.connect("destroy", lambda x: gtk.main_quit())
win.set_default_size(400,300)
win.set_title("Embedding in GTK")

vbox = gtk.VBox()
win.add(vbox)

fig = Figure(figsize=(5,4), dpi=100)
ax = fig.add_subplot(111)
t = arange(0.0,3.0,0.01)
s = sin(2*pi*t)

ax.plot(t,s)

canvas = FigureCanvas(fig)  # a gtk.DrawingArea
vbox.pack_start(canvas)

toolbar = NavigationToolbar(canvas, win)
vbox.pack_start(toolbar, False, False)

win.show_all()
gtk.main()
```

Event handling and GUI neutrality

Contouring and basemap

Plotting data on maps with Matplotlib: the Basemap toolkit

What it is:

- An add-on for matplotlib that makes it easy to plot data on map projections (with an emphasis on the geosciences).

What it isn't:

- A fully featured GIS toolkit.

Example

```
>>> # import toolkit
>>> from matplotlib.toolkits.basemap import Basemap
>>> # create a class instance. This sets up
>>> # a coordinate system for a given map projection
>>> # and converts boundary data (coastlines, rivers,
>>> # states and countries) to that coord. System.
>>> m = Basemap(projection='ortho',\
          lat_0=50,lon_0=-100,resolution='l')
>>> """
* 17 map projections supported (via Proj4 lib)
* Resolution keyword specifies boundary dataset to
  use ('c' crude is crudest, 'h' is most detailed).
* __call__ converts to/from projection coords."""
>>> print m(-118,34) # lon,lat for Pasadena.
>> (4738832.3099009357, 4812941.9920444777)
>>> print m(4738832.31,4812941.99,inverse=True)
>> (-117.99999999482151, 33.99999998152694)
```

Example (cont)

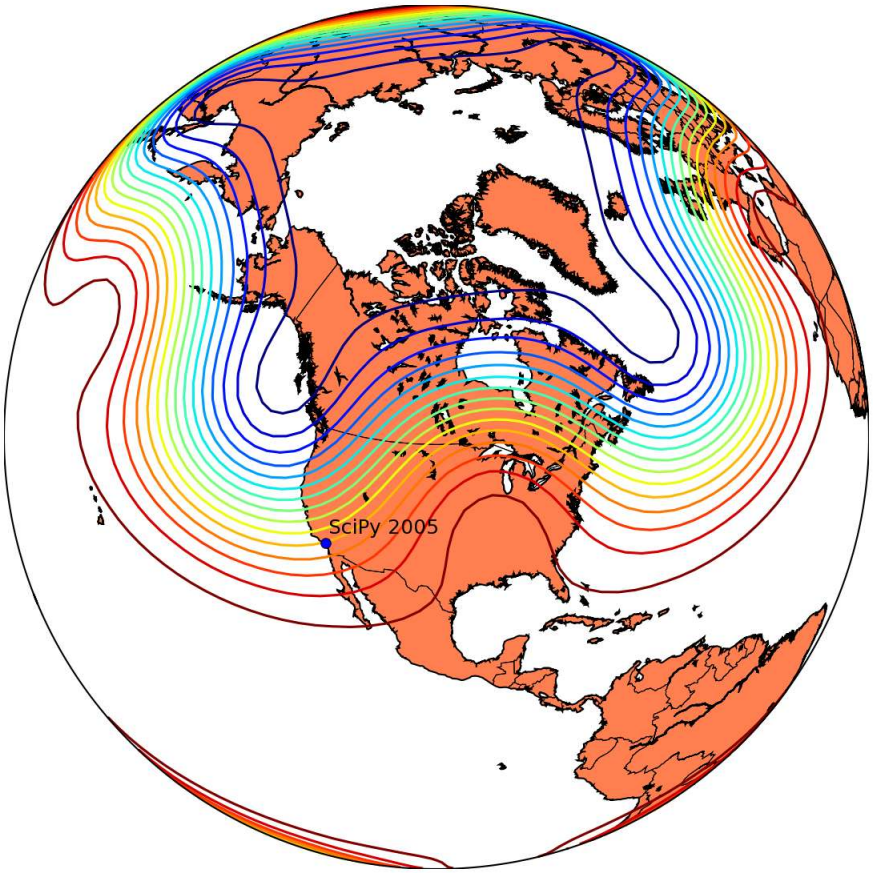
```
>>> import pylab as P
>>> # set up square figure, add axes (without frame)
>>> fig=P.figure(figsize=(8,8))
>>> fig.add_axes([0.05,0.05,0.9,0.9],frameon=False)
>>> m.drawcoastlines() # draw coastlines
>>> m.drawcountries()  # draw country boundaries
>>> # fill continents.
>>> m.fillcontinents(\
>>>     color='coral')
>>> m.drawmapboundary()
```

Example (cont)

```
>>> # Plot labelled dot at Pasadena.
>>> x, y = map(-118,34)
>>> m.plot([x],[y],'bo')
>>> P.text(x+50000,\
    y+50000,'SciPy 2005'
```

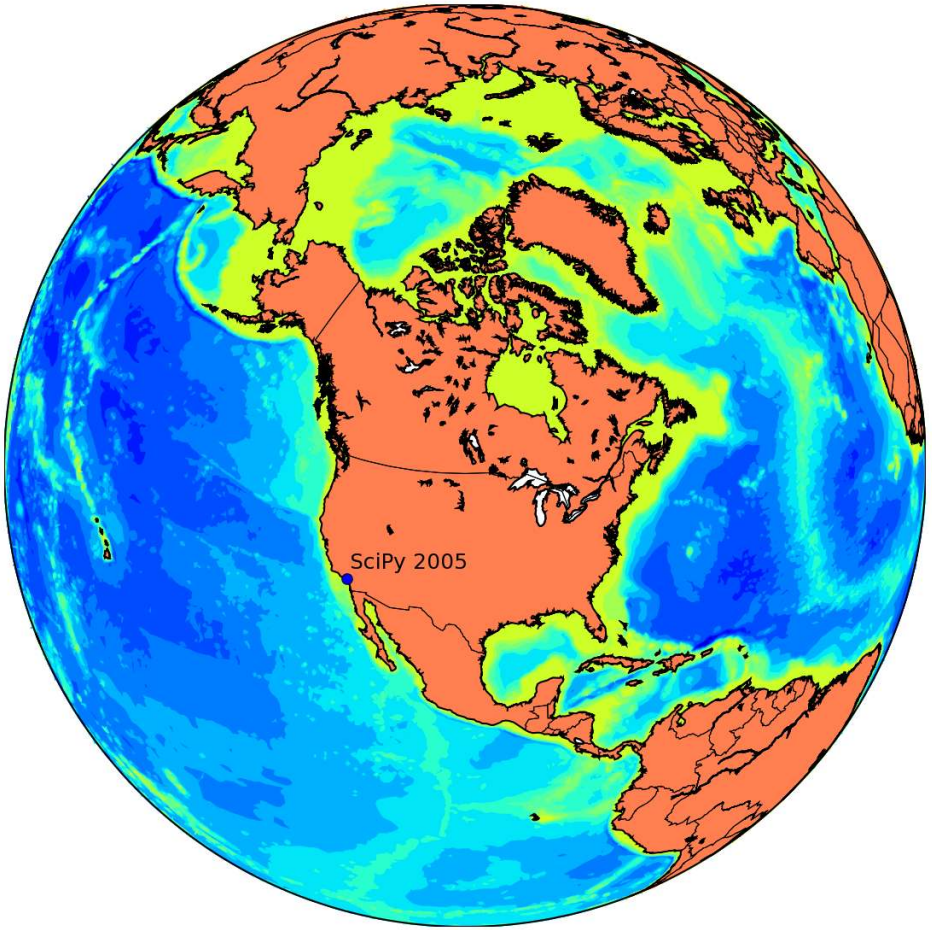
Example (cont)

```
>>> # contour some fake data on a
>>> # regular lat/lon grid over the map.
>>> x, y = m(lons, lats)
>>> levs, colls = \
  map.contour(x,y,fakedata,\
       15,linewidths=1.5)
```

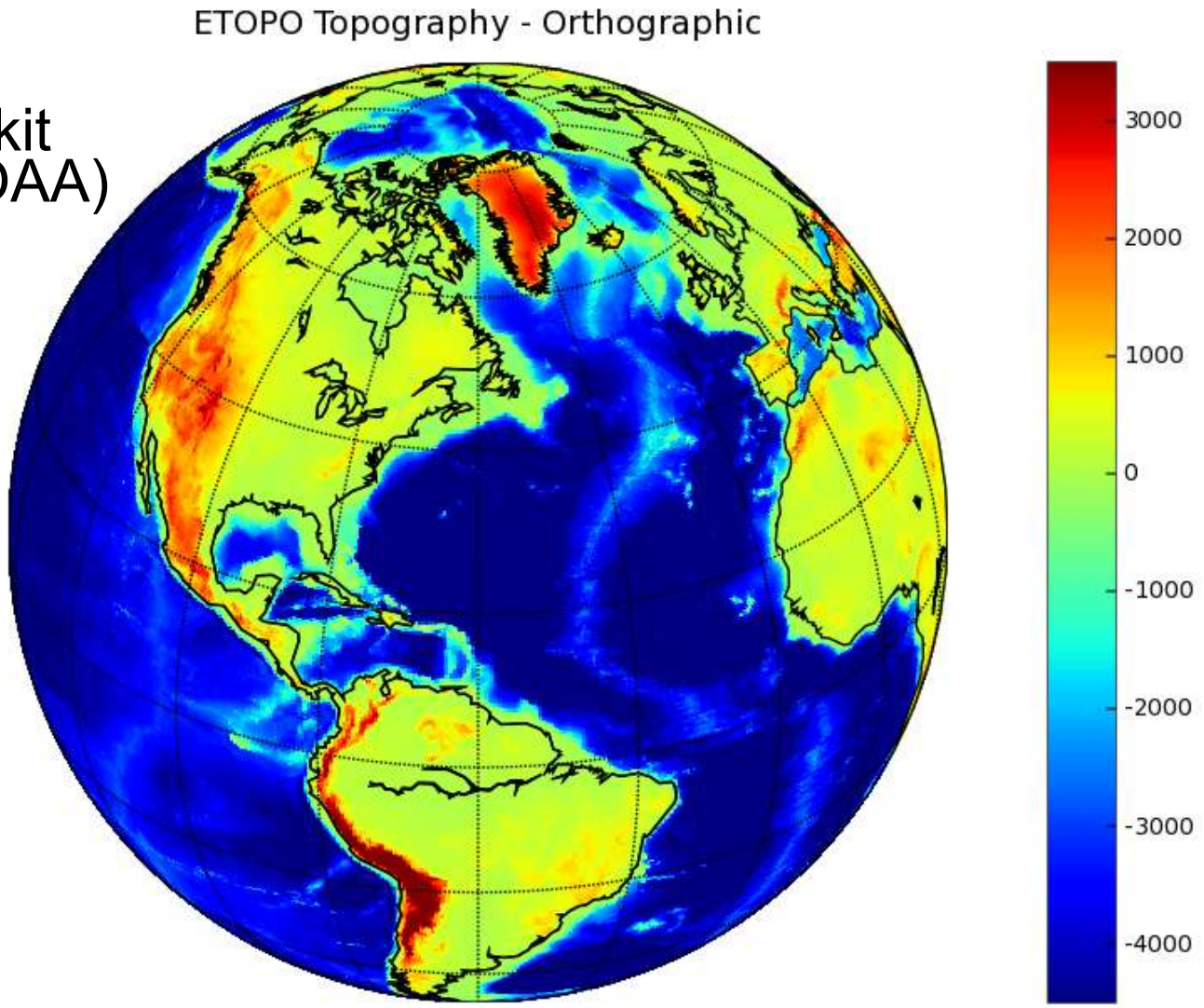


Example (cont)

```
>>> # or even some real data (ETOPO bathymetry)
>>> levs, colls = \
  map.contour(x,y,etopo20,20,cm=P.cm.jet,colors=None)
```



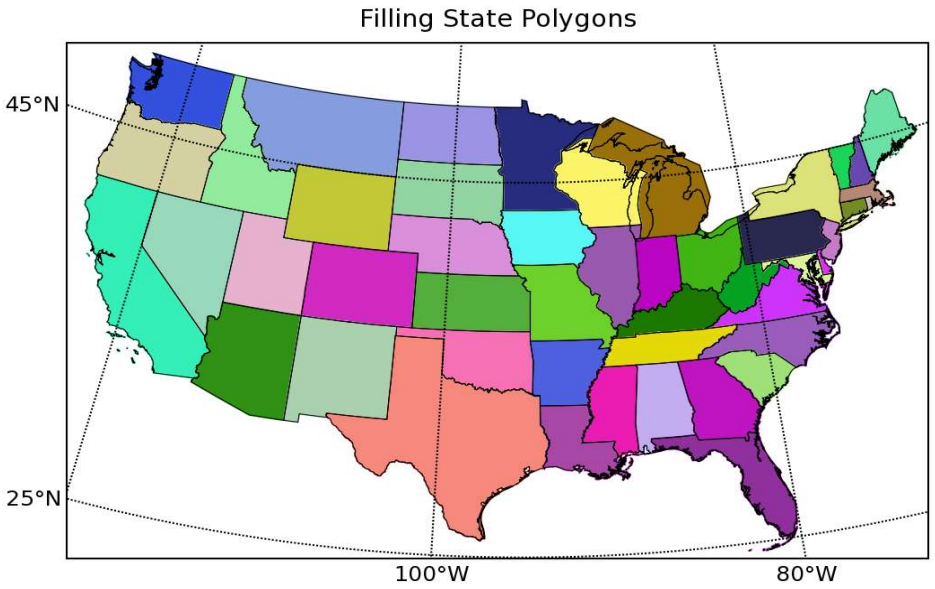
matplotlib
screenshot
basemap toolkit
(courtesy of NOAA)



Other Useful features

- Class instance can be pickled an re-used (good when creating lots of plots with the same map using high resolution boundaries).
- Can import and plot vecator geospatial data in ESRI shapefile format (using Bernhard Herzog's pyshapelib, which is included).
- Raster geospatial data can be plotted with the help of gdal module (***http://gdal.maptools.org***).

- Can handle non-spherical ellipsoids.
- Can compute distances over the earth's surface using geodetic formulas.



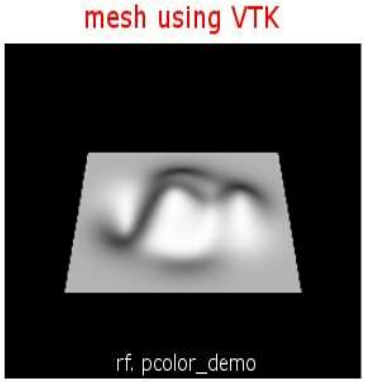
Animation :
how to make it easy and how to make it faster

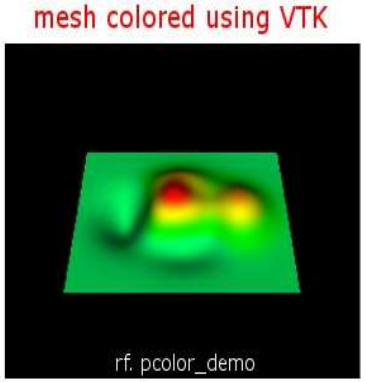
The animation pipeline

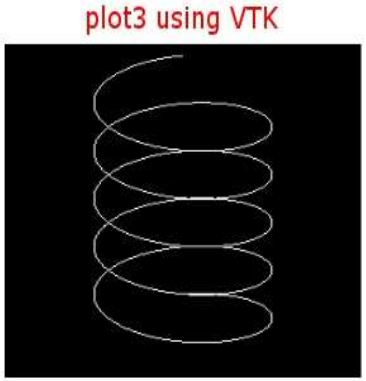
```
Draw a background and copy it into a buffer

Create some objects you want to animate

while 1:
    restore the background

    update the data and properties of your
       animated objects

    redraw the portion of your screen where
       something has changed
```

3D

VTK->matplotlib
http://sda.iu.edu/matplot.html







Matplotlib->VTK
(see matplotlib wiki)