UNIVERSITY OF DUNDEE

# Introduction to

python

# Why Python?

## Highly expressive:

```
friends = ['john', 'pat', 'gary', 'michael']
for i, name in enumerate(friends):
    print("iteration {iteration} is {name}".format(iteration=i, name=name))
```

## Object oriented

## Ease:

Easy to learn, change from IDL or Matlab.

## Very active community

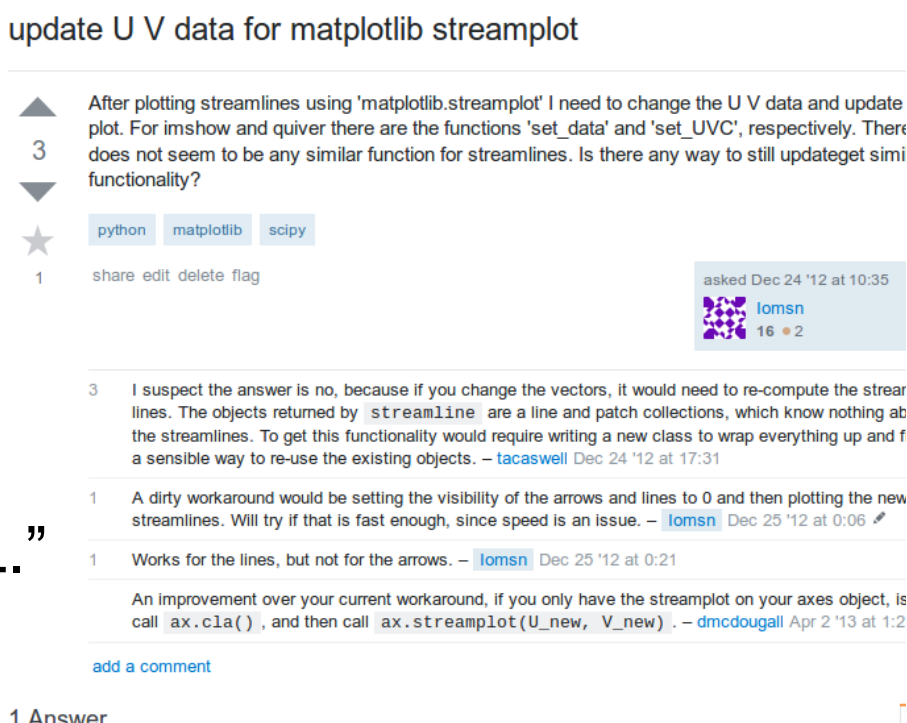
## Extendability:

libraries for numerics, data analysis, plotting, financing, ...

## Interoperability:

Import IDL and Matlab code.

“But all my routines are written in IDL.”



update U V data for matplotlib streamplot

3

1

After plotting streamlines using 'matplotlib.streamplot' I need to change the U V data and update plot. For imshow and quiver there are the functions 'set_data' and 'set_UVC', respectively. There does not seem to be any similar function for streamlines. Is there any way to still updateget simi functionality?

python matplotlib scipy

share edit delete flag

asked Dec 24 '12 at 10:35

lomsn

16 • 2

3 I suspect the answer is no, because if you change the vectors, it would need to re-compute the strean lines. The objects returned by `streamline` are a line and patch collections, which know nothing ab the streamlines. To get this functionality would require writing a new class to wrap everything up and f a sensible way to re-use the existing objects. – tacaswell Dec 24 '12 at 17:31

1 A dirty workaround would be setting the visibility of the arrows and lines to 0 and then plotting the new streamlines. Will try if that is fast enough, since speed is an issue. – lomsn Dec 25 '12 at 0:06

1 Works for the lines, but not for the arrows. – lomsn Dec 25 '12 at 0:21

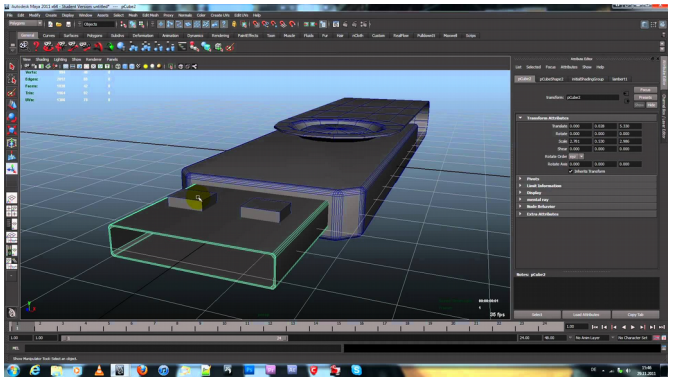
An improvement over your current workaround, if you only have the streamplot on your axes object, is call `ax.cla()`, and then call `ax.streamplot(U_new, V_new)`. – dmcdougall Apr 2 '13 at 1:2

add a comment

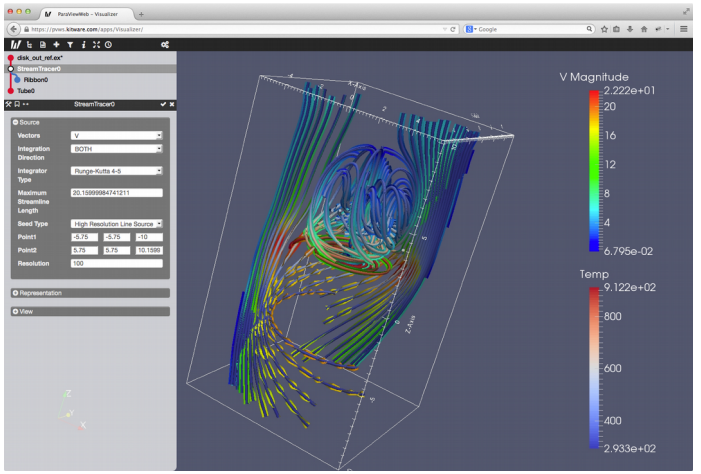
1 Answer

# Why Python?

"Python is everywhere, it is all around us, even now in this very room."



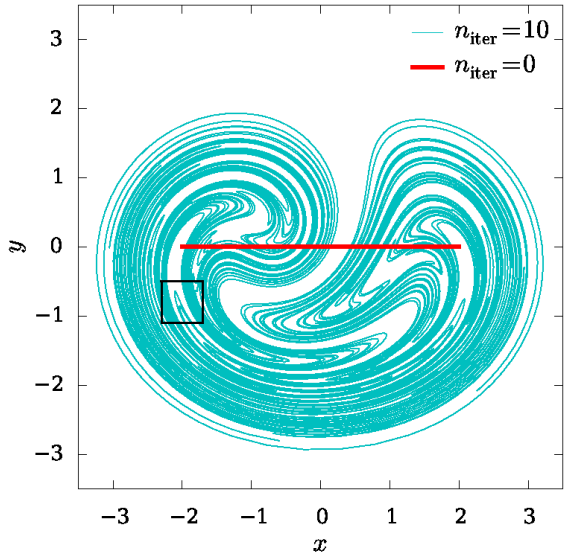
3ds Max
Maya
Blender
Cinema 4D
...

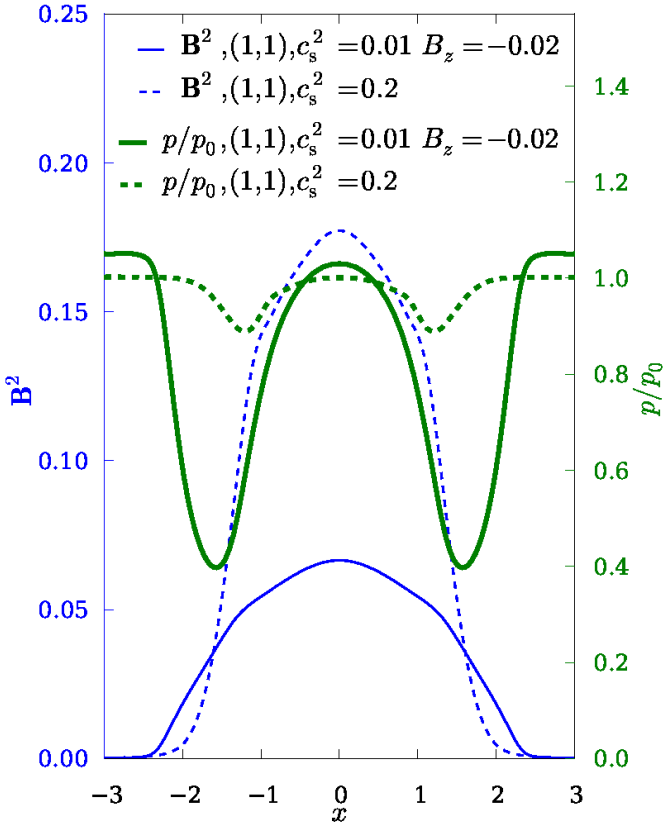


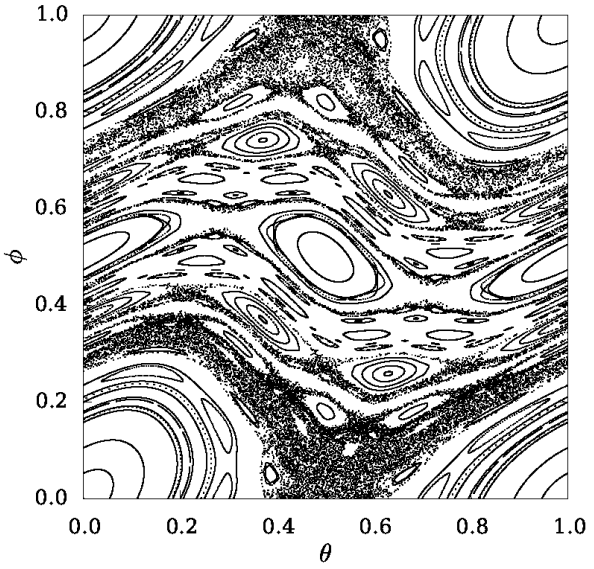

Paraview, Visit, Vapor, ...

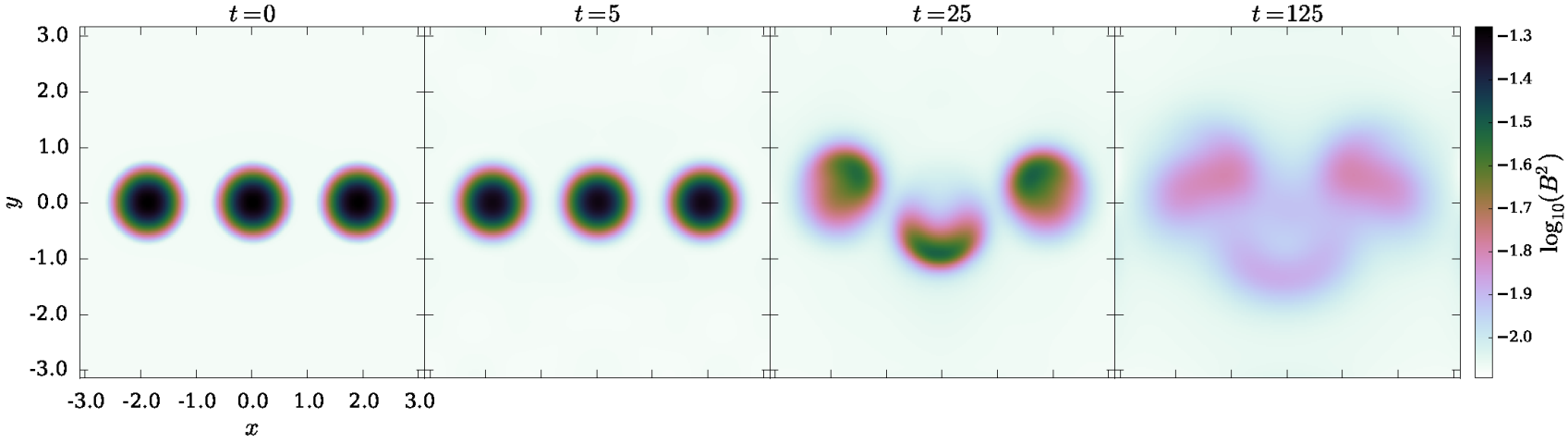
Civilization IV, Battlefield 2, World of Tanks, ...

# Why Python?

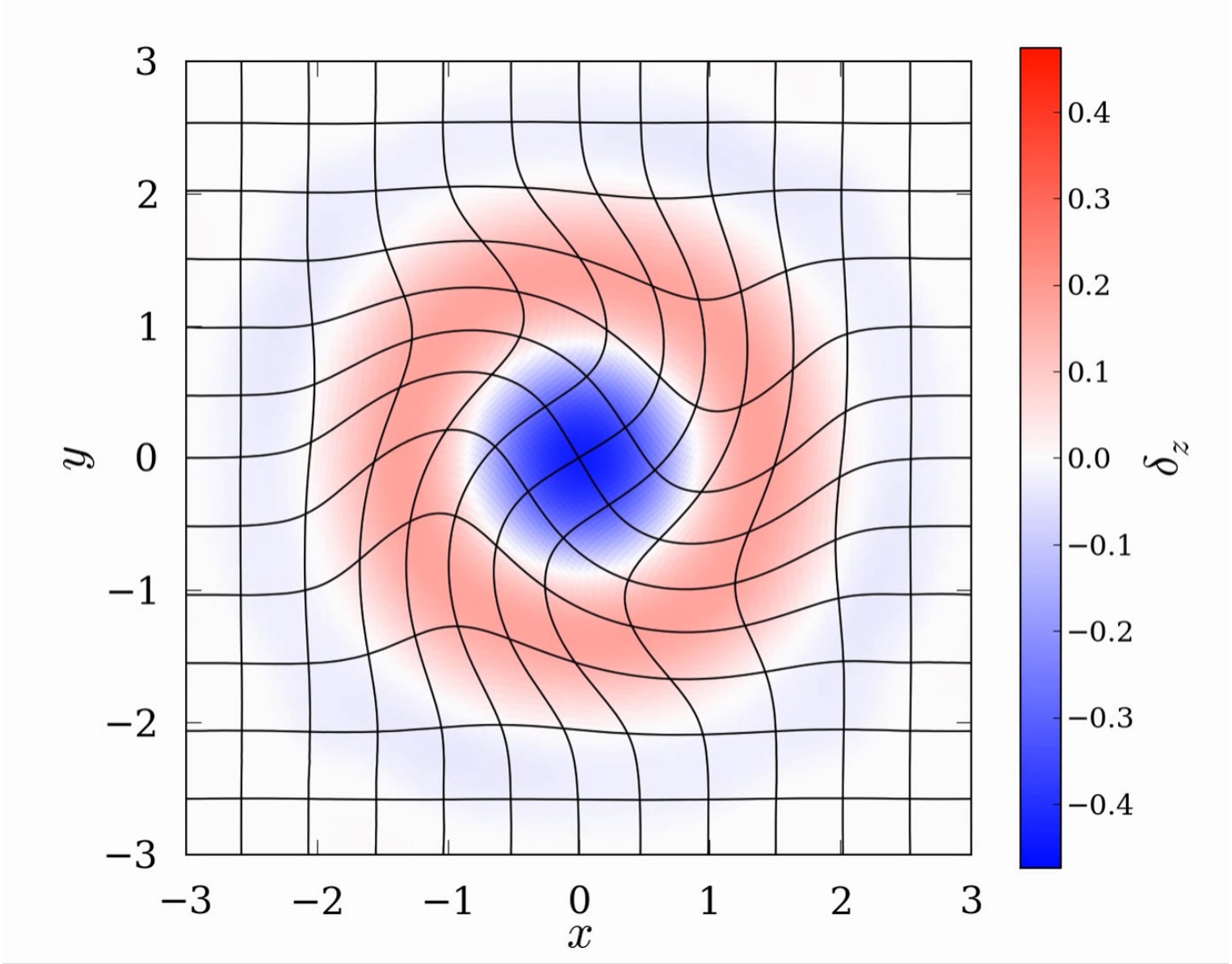

$n_{\rm iter}=10$
$n_{\rm iter}=0$
$x$
$y$


$\mathbf{B}^2,(1,1),c_s^2=0.01\ B_z=-0.02$
$\mathbf{B}^2,(1,1),c_s^2=0.2$
$p/p_0,(1,1),c_s^2=0.01\ B_z=-0.02$
$p/p_0,(1,1),c_s^2=0.2$
$\mathbf{B}^2$
$p/p_0$
$x$


$\phi$
$\theta$


$t=0$
$t=5$
$t=25$
$t=125$
$x$
$y$
$\log_{10}(B^2)$

# Why Python?



# History

1991, Guido van Rossum

Python 2.0
2000

Python 3.0
2008

Python 3.52
2016

Python
Course

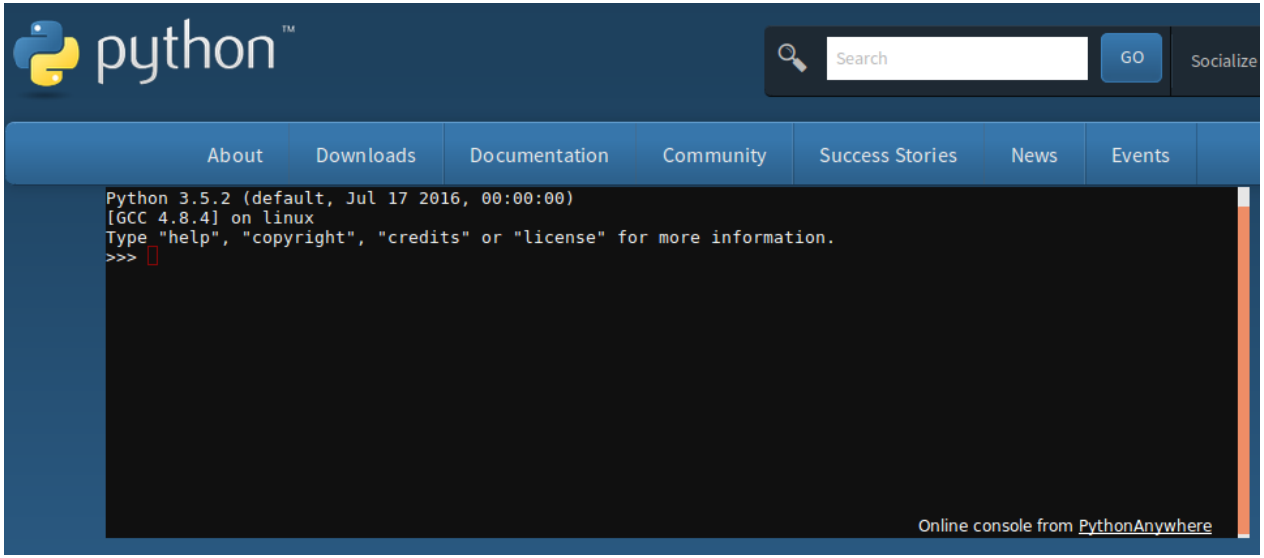
Namesake:

# Ways of Using Python

python/ipython

```
iomsn@a523:~/dundee/simon/teaching/python16$ ipython
Python 2.7.12 (default, Jul  1 2016, 15:12:24)
Type "copyright", "credits" or "license" for more information.

IPython 2.4.1 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.
paraview version 5.0.1

In [1]:
```

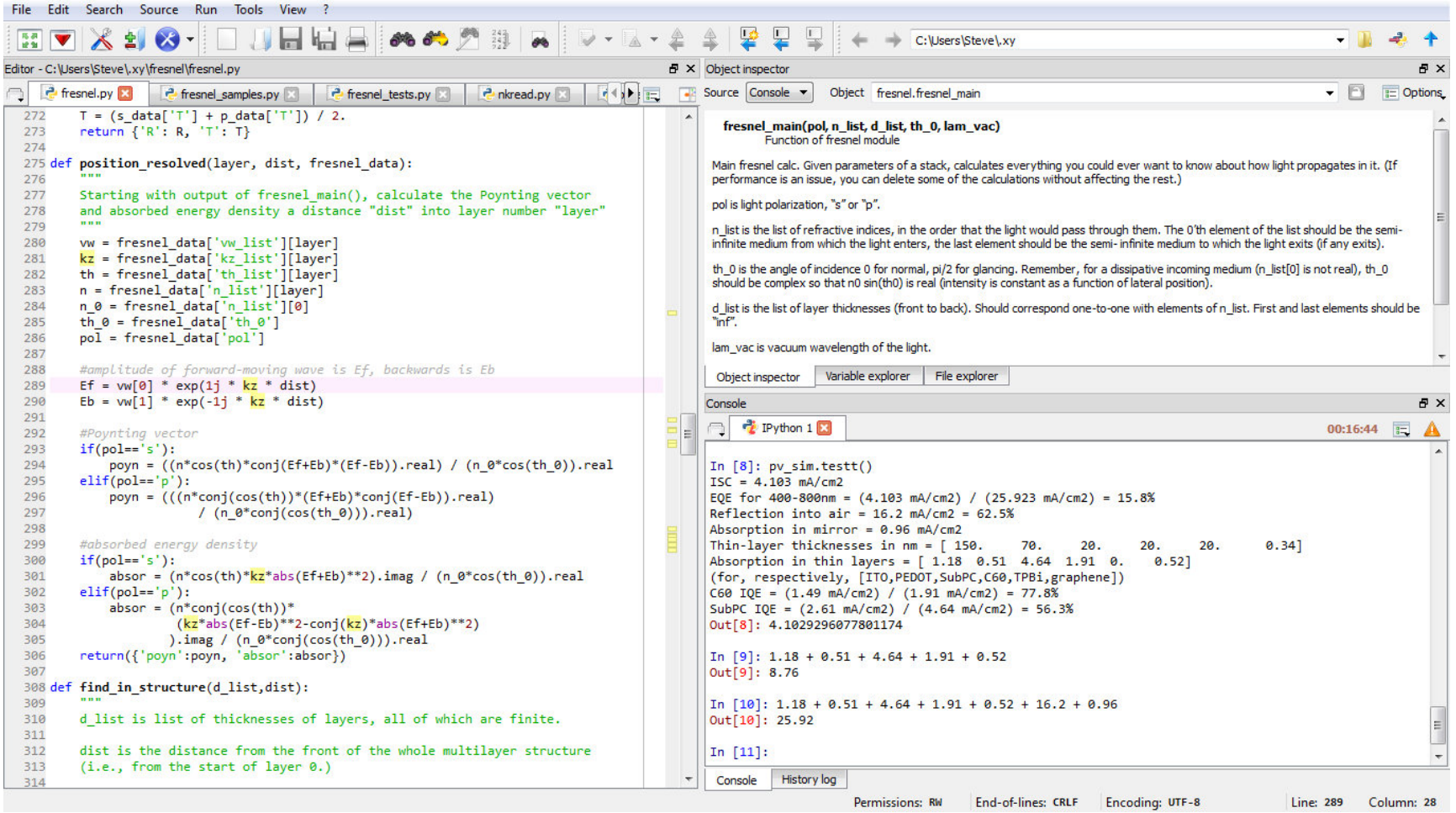
https://www.python.org/



# Ways of Using Python

ANACONDA®

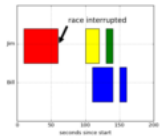
spyder

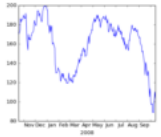


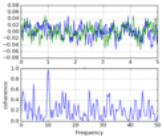

```
    T = (s_data['T'] + p_data['T']) / 2.
    return {'R': R, 'T': T}

def position_resolved(layer, dist, fresnel_data):
    """
    Starting with output of fresnel_main(), calculate the Poynting vector
    and absorbed energy density a distance "dist" into layer number "layer"
    """
    vw = fresnel_data['vw_list'][layer]
    kz = fresnel_data['kz_list'][layer]
    th = fresnel_data['th_list'][layer]
    n = fresnel_data['n_list'][layer]
    n_0 = fresnel_data['n_list'][0]
    th_0 = fresnel_data['th_0']
    pol = fresnel_data['pol']

    #amplitude of forward-moving wave is Ef, backwards is Eb
    Ef = vw[0] * exp(1j * kz * dist)
    Eb = vw[1] * exp(-1j * kz * dist)

    #Poynting vector
    if(pol=='s'):
        poyn = ((n*cos(th)*conj(Ef+Eb)*(Ef-Eb)).real) / (n_0*cos(th_0)).real
    elif(pol=='p'):
        poyn = (((n*conj(cos(th))*(Ef+Eb)*conj(Ef-Eb)).real)
                    / (n_0*conj(cos(th_0))).real)

    #absorbed energy density
    if(pol=='s'):
        absor = (n*cos(th)*kz*abs(Ef+Eb)**2).imag / (n_0*cos(th_0)).real
    elif(pol=='p'):
        absor = (n*conj(cos(th))*
                 (kz*abs(Ef-Eb)**2-conj(kz)*abs(Ef+Eb)**2)
                ).imag / (n_0*conj(cos(th_0))).real
    return({'poyn':poyn, 'absor':absor})

def find_in_structure(d_list,dist):
    """
    d_list is list of thicknesses of layers, all of which are finite.

    dist is the distance from the front of the whole multilayer structure
    (i.e., from the start of layer 0.)
```

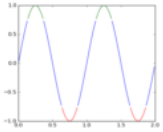
# Outlook

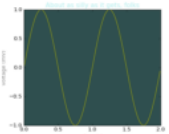
## Matplotlib Gallery

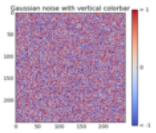

broken_barh

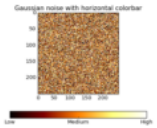

centered_ticklabels

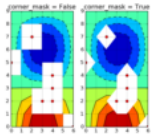

cohere_demo

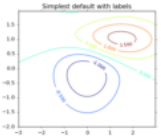

color_by_yvalue

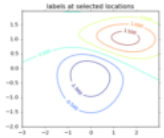

color_demo

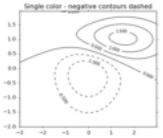

colorbar_tick_labelling_demo


colorbar_tick_labelling_demo


contour_corner_mask


contour_demo


contour_demo


contour_demo

## Scipy Functionalities

### Tutorial

Tutorials with worked examples and background information for most SciPy submodule

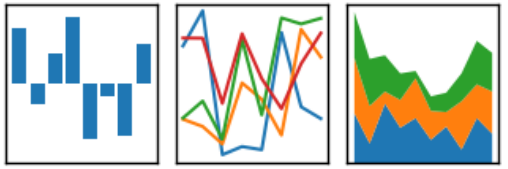
- SciPy Tutorial
  - Introduction
  - Basic functions
  - Special functions (**scipy.special**)
  - Integration (**scipy.integrate**)
  - Optimization (**scipy.optimize**)
  - Interpolation (**scipy.interpolate**)
  - Fourier Transforms (**scipy.fftpack**)
  - Signal Processing (**scipy.signal**)
  - Linear Algebra (**scipy.linalg**)
  - Sparse Eigenvalue Problems with ARPACK
  - Compressed Sparse Graph Routines (**scipy.sparse.csgraph**)
  - Spatial data structures and algorithms (**scipy.spatial**)
  - Statistics (**scipy.stats**)
  - Multidimensional image processing (**scipy.ndimage**)
  - File IO (**scipy.io**)
  - Weave (**scipy.weave**)

# Outlook

## Python Data Analysis Library

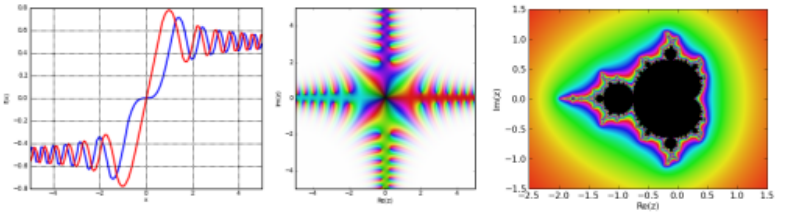
pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



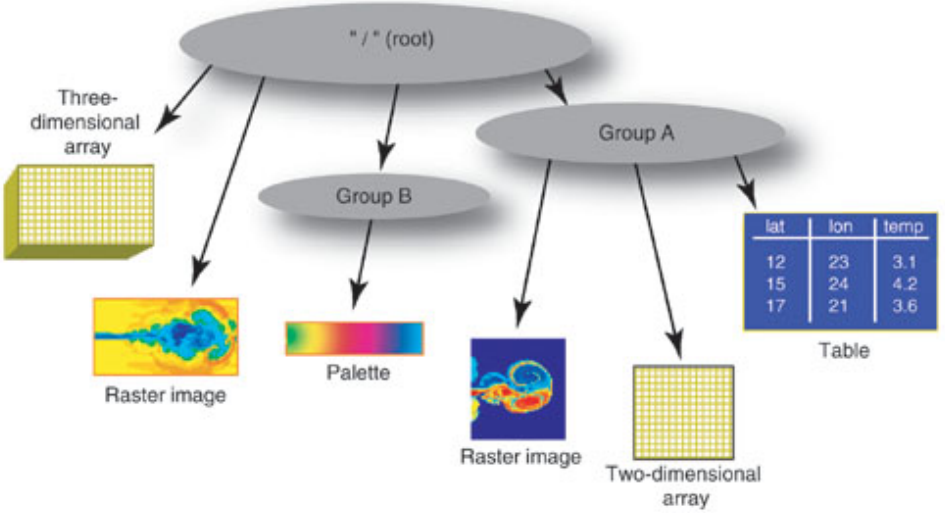
http://pandas.pydata.org

## mpmath
## floating-point arithmetic with arbitrary precision

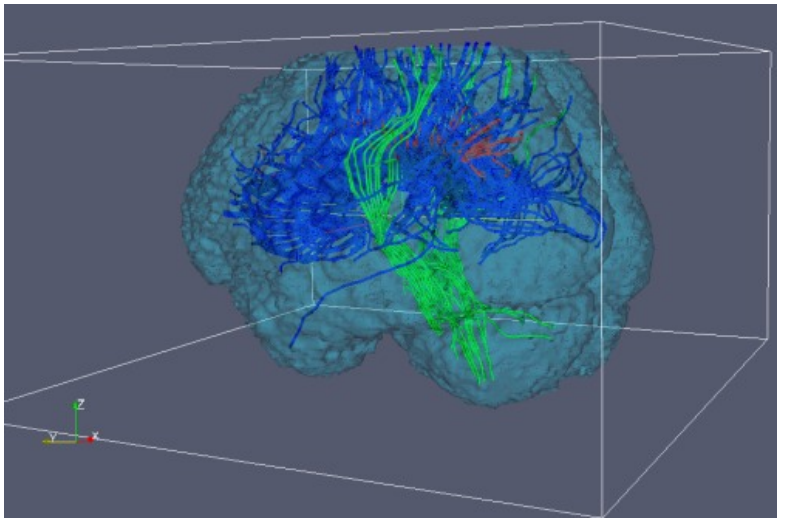


http://mpmath.org

## Hierarchical Data Format




## vtk Data Format



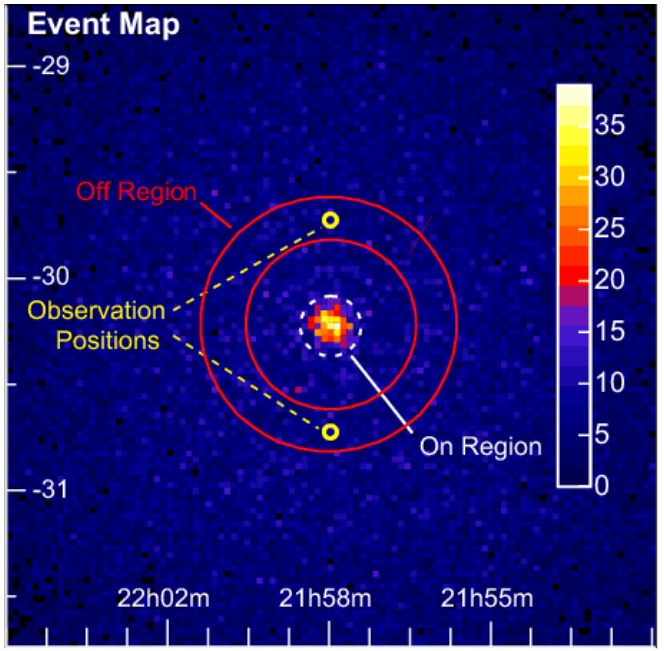
# Outlook

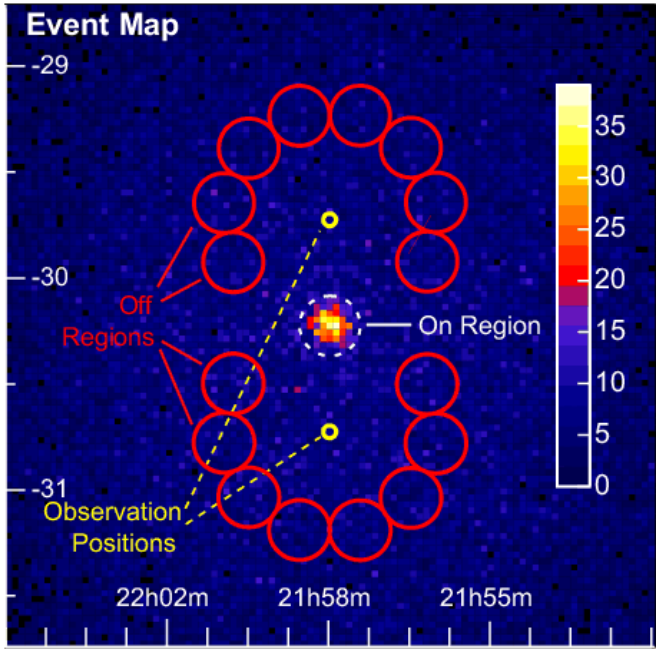
SunPy

http://sunpy.org

astropy

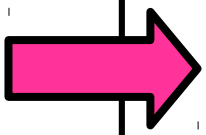

http://www.astropy.org



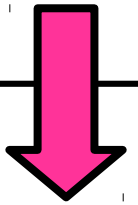



# Help!

```
# Python Documentation
help(plt.plot)
plt.plot?
source(plt.plot)
plt.plot??
```



Google



stackoverflow

## update U V data for matplotlib streamplot

3

1

After plotting streamlines using 'matplotlib.streamplot' I need to change the U V data and update the plot. For imshow and quiver there are the functions 'set_data' and 'set_UVC', respectively. There does not seem to be any similar function for streamlines. Is there any way to still updateget similar functionality?

python matplotlib scipy

share edit delete flag

asked Dec 24 '12 at 10:35

lomsn

16 ● 2

3 I suspect the answer is no, because if you change the vectors, it would need to re-compute the stream lines. The objects returned by `streamline` are a line and patch collections, which know nothing about the streamlines. To get this functionality would require writing a new class to wrap everything up and finding a sensible way to re-use the existing objects. – tacaswell Dec 24 '12 at 17:31

1 A dirty workaround would be setting the visibility of the arrows and lines to 0 and then plotting the new streamlines. Will try if that is fast enough, since speed is an issue. – lomsn Dec 25 '12 at 0:06

# Practice!