Chapter 3

Graphics and Visualization

Soon-Hyung Yook

March 20, 2017

Table of Contents I

Introduction

- Two Main Types of Computer Graphics Used in Physics
  1. Depiction of numerical data displayed on calibrated axes.
  2. Scientific diagram and animation

Graphics

pylab package

- The name pylab is a reference to the Matlab, whose graph-drawing features pylab is intended to mimic.
- The pylab package is a part of a larger package called matplotlib.
- pyplot has features for generating graph of many different types.
- We will focus only on:
  - line graphs
  - scatter plots
  - density (or heat) plots
- http://matplotlib.org for matplotlib or full feature of pylab.

Graphics

`plot` and `show` function

- Two basic functions to create an ordinary graph.

```
from pylab import plot,show
y=[1.0,2.4,1.7,0.3,0.6,1.8]
plot(y)
show()
```

Plotting (x, y) pair

```
from pylab import plot,show
x=[0.5,1.0,2.0,4.0,7.0,10.0]
y=[1.0,2.4,1.7,0.3,0.6,1.8]
plot(x,y)
show()
```

Graphics

- plot $\sin x$ from $x = 0$ to $x = 10$

```
from pylab import plot, show
from numpy import linspace, sin

x=linspace(0.0,10.0,100)
y=sin(x)
plot(x,y)
show()
```

- Here we use `sin` function from `numpy`.
- `sin` function in `numpy` is a special version of sine that works with arrays.
- similar with `map(sin,x)` (here `sin` is from `math` package)

Graphics: Read Data from File

```
from numpy import loadtxt
from pylab import plot, show

data=loadtxt(”values.dat”, float)
x=data[:,0]
y=data[:,1]
plot(x,y)
show()
```

- we use `loadtxt` function to load data from text file.
- `data[:,0]`, `data[:,1]` are array slicing.

More concisely:

```
from numpy import loadtxt
from pylab import plot, show

data=loadtxt(”values.dat”, float)
plot(data[:,0],data[:,1])
show()
```

Graphics: Playing with List

```
from pylab import plot,show
from math import sin
from numpy import linspace

xpoints=[]
ypoints=[]
for x in linspace(0,10,100):
  xpoints.append(x)
  ypoints.append(sin(x))
plot(xpoints,ypoints)
show()
```

Graphics: xlim, ylim

```
from pylab import plot, show, ylim
from numpy import linspace, sin

x=linspace(0.0,10.0,100)
y=sin(x)
plot(x,y)
ylim(-1.1,1.1)
show()
```

Graphics: `xlabel, ylabel`

```
from pylab import plot, show, ylim,xlabel,ylabel
from numpy import linspace, sin

x=linspace(0.0,10.0,100)
y=sin(x)
plot(x,y)
ylim(-1.1,1.1)
xlabel(r'$x$',fontsize=20)
ylabel(r'$\sin␣x$',fontsize=20)
show()
```

Graphics: Plot with Symbols, Color, etc.

```
from pylab import plot, show, ylim,xlabel,ylabel
from numpy import linspace, sin,cos

x=linspace(0.0,10.0,100)
y=sin(x)
plot(x,y,'g--')
y=cos(x)
plot(x,y,'ro')
ylim(-1.1,1.1)
xlabel(r'$x$',fontsize=20)
ylabel(r'$\sin x$, $\cos x$',fontsize=20)
show()
```

- LaTeX is working with python.
- Similar to the math editor in HWP.

Scatter Plots

- Use symbols for `plot`.
    - `plot(x,y,"ko")`, `plot(x,y,"k.")`
    - Alternatively, `pylab` provides the function `scatter`.

```
from pylab import scatter, show,xlim,ylim,xlabel,ylabel
from numpy import loadtxt

data=loadtxt("stars.txt",float)
x=data[:,0]
y=data[:,1]
xlabel=("Temperature")
ylabel=("Magnitude")
xlim(0,13000)
ylim(-5,20)
scatter(x,y)
show()
```

Density Plots I

- Use imshow function.

```
from pylab import imshow, show
from numpy import loadtxt

data=loadtxt("circular.txt",float)
imshow(data)
show()
```

- Note that the numerical labels on the axes reflect the array index (i, j) for `data[i,j]`
- The origin of the figure is at top left corner.
- The vertical axis increasing downwards.
- The array element `data[i,j]` are written with the row (vertical) index first and the column (horizontal) index second, like the matrix: i.e.,(i, j) corresponds to (y, x) pair.
  - Convert the coordinate into (x, y) pair.

Density Plots II

- Use an additional argument for the `imshow` function to flip the density plot top-to-bottom:

  `imshow(data,origin="lower")`

```
from pylab import imshow, show
from numpy import loadtxt

data=loadtxt("circular.txt",float)
imshow(data)
show()
```

- the function `gray()` changes the color mode into gray mode.

```
from pylab import imshow, show
from numpy import loadtxt

data=loadtxt("circular.txt",float)
imshow(data)
gray()
show()
```

- `jet, gray, hot, spectral, bone, hsv` with color scheme `redblue, redwhiteblue, inversegray`, etc.

Density Plots III

- for details see the textbook or visit `matplotlib.org`.
- more options for `imshow` function:
    - Change the beginning and end of the horizontal and vertical scale:

      ```
      imshow(data,origin="lower",extent=[0,10,0,5])
      ```
    - Change the aspect ratio:

      ```
      imshow(data,origin="lower",extent=[0,10,0,5],aspect=2.0)
      ```
    - `imshow` also combined with `xlim, ylim`

3D Graphics

See the textbook but it has some problem...