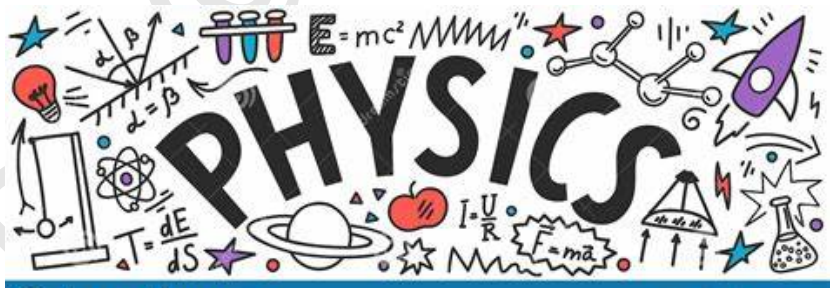
Graphics and visualization

Bin Gu

gubit@163.com
Department of Physics
Nanjing University of Information Science and Technology

September 28, 2023



1. Aims
2. Graphs
3. Scatter Plots
4. Density Plots
5. 3D Graphics
6. Animation

1. aims

How to produce the computer graphics used in physics, for visualizations of data/functions, and properties of physical systems?

`pylab` package: part of `matplotlib` package.

- Line graphs
- Scatter plots
- Density plots
- 3D graphics
- Animation

With `pylab`, you can also make contour plots, polar plots, pie charts, histograms, and more.
On-line document: matplotlib.org.

What's the difference between diagrams, charts and graphs?

2. Graphs

`plot(x,y)` and `show()` function.

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

```
#example
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()
```

useful buttons: view, save as, ……

2. Graphs

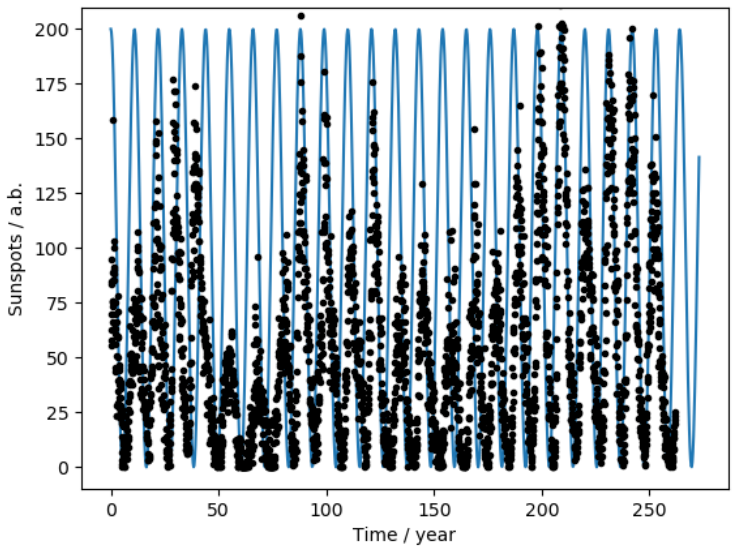
Graph of $y = f(x)$ and data in a file "sunspots.txt"

```
from pylab import plot,show
from numpy import linspace,sin,cos
from math import pi
#Generate Linearly Spaced Vector x
x = linspace(0,2022-1749,80*12)
y = (cos(x/11.*2*pi)+1)*100
plot(x,y)

from numpy import loadtxt
#load data from txt file to a 2D array
data = loadtxt("../source/sunspots.txt",
       float)
x = data[:,0]
y = data[:,1]
plot(x/12,y,"k.")
#plot(data[:,0]/12./11*pi,data[:,1])
show()
```

"sunspots.txt": the observed number of sunspots on the Sun for each month since January 1749.

```
0 58.0
1 62.6
.
3142 21.6
```



2. Graphs

Notes

1. **Blocking function**:
`show()`, `input()`, $\cdots$
Once you use show() to display a graph, the program will go no further (blocked) until you close the window containing the graph.
2. **Default style**: The plot consists of /point sets/ connected with straight lines.
3. **multi-plots**

3. x-, y- values for a figure are not known in advance, but need to be built/calculated step by step.

```
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()
```

2. Graphs

Customize feature and style: after plot before show

Default choice is reasonable. Art of science needs special care.

- `xlim(0,10)`, `ylim(-1.2,1.2)`: set limits of range of the axes.
- `xlabel("time / y")`, `ylabel="Sunspots / a.b."`: set x,y labels.
- `plot(x,y,"b-")`; set draw style of $y(x)$. First letter: color; the remainder: line style.
  r, g, b, c, m, y, k, and w, for red, green, blue, cyan, magenta, yellow, black, and white, respectively.

  "-" solid line (default), "–" dashed line, "o" points with isolated circles (no connecting lines), and "s" points with squares.
- notation, background, axes style, ···

3. Scatter Plots

For functions, independent variable is represented with the horizontal axis, while dependent variable with vertical axis.

For two characters (properties) of a set of independent cases, we often do scatter plot, with dots for pair of measurements.

Example: Scores of each students (math,phys); Star (temperature, magnitude-星等)

- plot(x,y,"k.") with dot or circle style.
- scatter(x,y)

Hertzsprung – Russell(赫罗) diagram. A scatter plot of the magnitude (i.e., brightness) of stars against their approximate surface temperature (which is es- timated from the color of the light they emit). Each dot on the plot represents one star out of a catalog of 7860 stars that are close to our solar system

4. Density Plots

`imshow()` presents data on 2D grids with color or brightness

Examples: Condensed matter physics; fluid dynamics; nuclear physics.

" circular.txt"

```
0.0050 0.0233 0.0515   ...
0.0233 0.0516 0.0798   ...
0.0515 0.0798 0.1080   ...
..   ...   ...   ...   ...
```

```
from pylab import imshow,show
from numpy import loadtxt
data = loadtxt("../circular.txt", float)
imshow(data)
colorbar()
show()
```

style list:

1. jet default heat map (red blue)
2. gray
3. hot black-red-yellow-white
4. spectral 7 colors
5. bone gray-scale / blue hint.
6. hsv rain bow with red$\cdots$red

```
imshow(data,origin="lower")
imshow(data,extent[0,10,0,5])
imshow(data,aspect=2.0
```

Mixture of plot,imshow,scatter.

4. Density Plots

Example 3.1 Wave interference

$$f(x,y) = f_0 sin(kr_1) + f_0 sin(kr_2).$$

```
from math import sqrt, sin, pi
from numpy import empty
from pylab import imshow,gray,show
wavelength = 5.0
k = 2*pi/wavelength
xi0 = 1.0
separation = 20.0
# Separation of centers in cm
side = 100.0
# Side of the square in cm
points = 500
# Number of grid points along each side
spacing = side/points
# Spacing of points in cm
# Calculate the positions of the centers
      of the circles
x1 = side/2 + separation/2
y1 = side/2
x2 = side/2 - separation/2
y2 = side/2
```

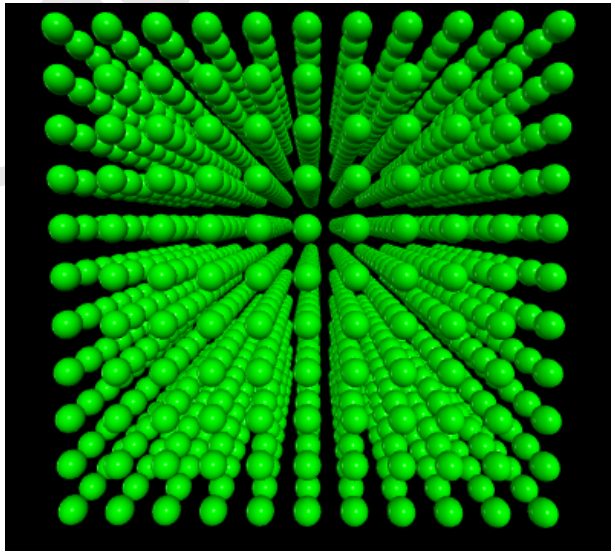
```
# Make an array to store the heights
xi = empty([points,points], float)
# Calculate the values in the array
for i in range(points):
    y = spacing*i
    for j in range(points):
        x = spacing*j
        r1 = sqrt((x-x1)**2+(y-y1)**2)
        r2 = sqrt((x-x2)**2+(y-y2)**2)
        xi[i,j] = xi0*sin(k*r1) + xi0*sin
            (k*r2)
# Make the plot
imshow(xi, origin="lower",extent=[0,side
      ,0, side])
gray()
show()
```

5. 3D Graphics

The **visual** package, which is named as **vpython**, works by creating specified objects on the screen, such as spheres, cylinders, cones, and so forth, and then, if necessary, changing their position, orientation, or shape to make them move around.

Example:

```
# use vpython package to make a crystal
      lattice of atoms. Note: the name "
      visual" has been changed to vpython
#https://vpython.org/%contents/
      VPythonArchitecture.pdf
from vpython import sphere, vector, color
from numpy import array
L = 5
R = 0.3
for i in range(-L,L+1):
    for j in range(-L,L+1):
        for k in range(-L,L+1):
            sphere(pos=vector(i,j,k),
                   radius=R,color=color.
                   green)
```



The figure will be displayed on your Web Browser. Change the direction the the camera is looking in with your mouse.

5. 3D Graphics

The "object" type, and its attributions: pos, color, size, $\cdots$

```
from vpython import sphere, vector, color, box
from numpy import empty

s=sphere()
s.radius=0.5
s.color=color.green
s.pos=vector(1.,5,5)
type(s)

sa=empty(10,sphere)
# type of the element is sphere
t=box(pos=vector(0,0,0),length=5 ,height=7,width=5)
t.color=color.red
for n in range(10):
    print(n)
    sa[n]=sphere(pos=vector(0,n*0.5,n*0.5), radius=n*0.1)
```

5. 3D Graphics

```
from vpython import box,cone, cylinder ,pyramid,arrow. vector
box(pos=vector(x,y,z), axis=vector(a,b,c), \
    length=L, height=H, width=W, up=vector(q,r,s))
cone(pos=vector(x,y,z), axis=vector(a,b,c), radius=R)
cylinder (pos=vector(x,y,z), axis=vector(a,b,c), radius=R)
pyramid(pos=vector(x,y,z), size=vector(z,b,c))
arrow(pos=vector(x,y,z), axis=vector(a,b,c), \
    headwidth=H, headlength=L, shaftwidth=W)
```

About the screen window :

```
from vpython import vector ,canvas,box
canvas(x=100,y=100,width=600,height=600, \
      center=vector (5,0,0, forward=vector(0,0,-1), \
    background=color.blue,foreground=color. yellow )

d = canvas()
d.background=color.white
```

Notes: vpython-doc; Classic VPython" (VPython 6 or earlier)

6. Animation

```
from vpython import sphere, rate, vector, color, canvas
from math import cos, sin, pi
from numpy import arange

cv = canvas()
cv.background=color.white
s = sphere(pos=vector(1,0,0), radius=0.1, color=color.yellow)
for theta in arange(0,10*pi,0.1):
    rate(30)
    #30 times a second
    x = cos(theta)
    y = sin(theta)
    s.pos = vector(x,y,0)
```

– change the position of an object repeatedly and rapidly, you can make the object appear to be moving and you have an animation.

Homework

Exercise 3.5: Visualization of the solar system

Exercise 3.8:计算物理实践 1

用最小二乘法，分析密立根油滴实验数据，计算出普朗克常量（Nobel Prize work）

画出流程图，写出 Python 程序，绘制拟合图，并总结分析数据。