Python + Emacs in Scientific computing

Scipy 2013

2013-12-14 Sat

1 Mathematical operations in Python

1.1 Square root

In mathematics, a square root of a number a is a number y such that y2 = a, in other words, a number y whose square (the result of multiplying the number by itself, or y × y) is a.[1] For example, 4 and 4 are square roots of 16 because 42 = (4)2 = 16.

Every non-negative real number a has a unique non-negative square root, called the principal square root, which is denoted by a, where is called the radical sign or radix. For example, the principal square root of 9 is 3, denoted 9 = 3, because 32 = 3 × 3 = 9 and 3 is non-negative. The term whose root is being considered is known as the radicand. The radicand is the number or expression underneath the radical sign, in this example 9.

```python
# ---------------------
import numpy as np
print np.sqrt(2)
# ---------------------
```

1.41421356237

1.2 Logarithm

The logarithm of a number is the exponent to which another fixed value, the base, must be raised to produce that number. For example, the logarithm of 1000 to base 10 is 3, because 1000 is 10 to the power 3: 1000 = 10×10×10 = 103. More generally, if x = by, then y is the logarithm of x to base b, and is written y = logb(x), so log10(1000) = 3.

[1] DEFINITION NOT FOUND: 1

The logarithm to base 10 (b = 10) is called the common logarithm and has many applications in science and engineering. The natural logarithm has the constant e (2.718) as its base; its use is widespread in pure mathematics, especially calculus. The binary logarithm uses base 2 (b = 2) and is prominent in computer science.

```python
# ----------------------------------
import numpy as np
print np.log(10)
print np.log10(10)   # base10
# ----------------------------------
```

2.30258509299
1.0

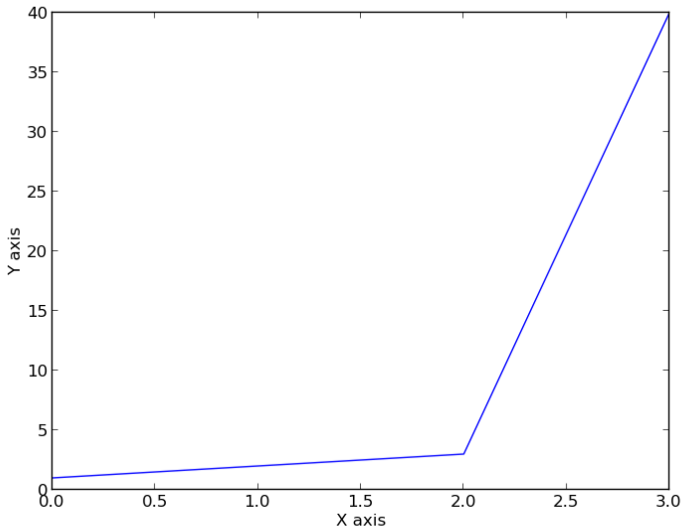
1.3 Plots

matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. matplotlib can be used in python scripts, the python and ipython shell (ala MATLAB®* or Mathematica®†), web application servers, and six graphical user interface toolkits.

- Example-1

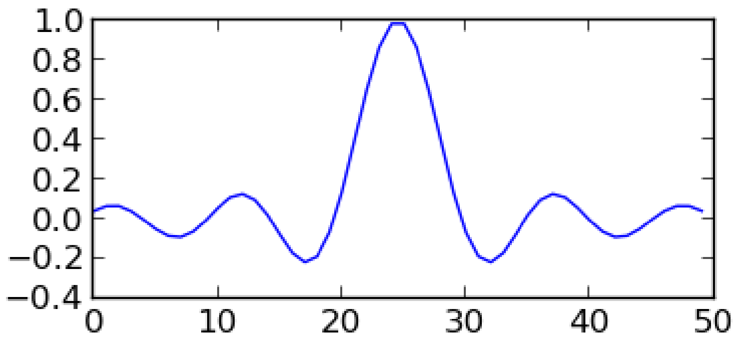
```python
# -------------------------------------------------------------
import matplotlib.pyplot as plt
plt.plot([1,2,3,40])
plt.ylabel('Y axis')
plt.xlabel('X axis')
plt.savefig(fname)

return fname                     # return filename to org-mode
# -------------------------------------------------------------
```



- Example-2

```
# -----------------------------------------------------------
import matplotlib, numpy
matplotlib.use('Agg')
import matplotlib.pyplot as plt
fig=plt.figure(figsize=(4,2))
x=numpy.linspace(-15,15)
plt.plot(numpy.sin(x)/x)
fig.tight_layout()
plt.savefig('images/python-matplot-fig.png')
return 'images/python-matplot-fig.png' # return filename to org-mode
# -----------------------------------------------------------
```



2 Data

2.1 Table: Student marks

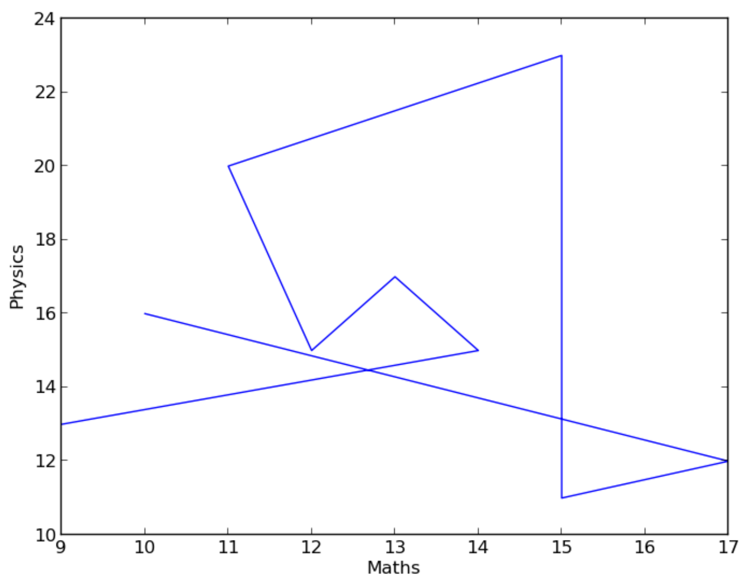
Student	Maths	Physics	Mean
Bertrand	13	09	
Henri	15	14	
Arnold	17	13	
Sam	15	12	
Emmy	20	11	
Roy	23	15	
Victor	11	15	
Robert	12	17	
Harper	16	10	
Mean			0

```
# -----------------------------------------------------------
maths=[]
physics=[]
mean=[]
for i in marks[1:-1]:
    maths.append(i[1])
    physics.append(i[2])
    mean.append(i[3])

import matplotlib.pyplot as plot
plot.plot(physics,maths)
plot.ylabel('Physics')
```

```
plot.xlabel('Maths')
plot.savefig('marks.png')

return 'marks.png'
# ------------------------------------------------------------
```



2.2 Table: VI characteristics of diode marks

V(volts)	I(mA)	V/I
0.21	0.21	.
0.41	0.41	.
0.61	0.61	.
0.81	0.81	.
1.09	1.09	.
1.20	1.20	.

```
# ------------------------------------------------------------
v=[]
i=[]
for reading in readings[1:]:
    v.append(reading[1])
    i.append(reading[2])
```

```
import matplotlib.pyplot as plt
plt.plot(i,v)
plt.ylabel('I')
plt.xlabel('V')
plt.savefig('iv.png')

return 'iv.png'
# -----------------------------------------------------------
```