

Python Scipy

Panchatcharam Mariappan

Assistant Professor

**Department of Mathematics and Statistics,
IIT Tirupati**

- NumPy/SciPy – numerical and scientific function libraries.
- Collection of mathematical algorithms
- NumPy extension
- Interactive python session by providing the user with high-level commands
- Manipulating and visualizing data
- Data processing
- System prototyping like MATLAB, SciLab

Methods for Integrating Functions given a function object:

Method	Explanation
quad	General purpose integration
dblquad	General purpose double integration
tplquad	General purpose triple integration
fixed_quad	Integrate function $f(x)$ using Gaussian quadrature of order n
quadrature	Integrate with given tolerance using Gaussian quadrature
romberg	Integrate function using Romberg integration

Methods for Integrating Functions given a fixed samples:

Method	Explanation
<code>trapz</code>	Use trapezoidal rule to compute integral from samples
<code>cumtrapz</code>	Use trapezoidal rule to cumulatively compute integral
<code>simps</code>	Use Simpson's rule to compute integral from samples
<code>romb</code>	Use Romberg Integration to compute integral from $(2^{**k} + 1)$ evenly-spaced samples

Simple Integral example

$$\int_a^b \sin x \, dx$$

scipy.integrate!

Methods for Integrating Functions given a fixed samples:

```
#Integration
import numpy as np
import scipy.integrate
print(scipy.integrate.quad(np.sin,0,np.pi))
print(scipy.integrate.quad(np.sin,-np.inf,np.inf))
#Integration Sampling
x=np.linspace(0,np.pi,10000000)
y=np.sin(x)
print(scipy.integrate.trapz(x,y))
```

<https://docs.scipy.org/doc/scipy/reference/tutorial/integrate.html>

```
from scipy import constants
print(constants.liter)
for i,j in constants.physical_constants.items():
    print(i,j)
```

Linear Algebra

```
import numpy as np
from scipy import linalg
A = np.array([[1, 2], [3, 4]])
b = np.array([[5], [6]])
print("A: ",A)
print("b: ",b)
print("Inverse: of ",linalg.inv(A))
print("Solution Ax=b",np.linalg.solve(A,b))
print("Determinant of A: ", linalg.det(A))
print("Column Sum norm: ",linalg.norm(A,1))
print("Row sum norm: ",linalg.norm(A,np.inf))
```


Linear Algebra

```
#1D Interpolation
from scipy.interpolate import interp1d
import numpy as np
x = np.linspace(0, 10, num=21, endpoint=True)
y = np.exp(-x**2/9.0)
f = interp1d(x, y)
f2 = interp1d(x, y, kind='cubic')
xnew = np.linspace(0, 10, num=51, endpoint=True)
import matplotlib.pyplot as plt
plt.plot(x, y, 'o', xnew, f(xnew), '-', xnew, f2(xnew), '--')
plt.legend(['data', 'linear', 'cubic'], loc='best')
plt.show()
```

Submodules	Description
scipy.special	Special Functions
scipy.fft	Fast Fourier Transforms
scipy.signal	Signal Processing
scipy.csgraph	Compressed Sparse Graph
scipy.spatial	Spatial Data structures, Delaunay, Simplices
scipy.stats	Statistics
scipy.ndimage	Multidimensional Image Processing
scipy.io	Input Output
scipy.sparse.linalg	Eigenvalues for large sparse matrices



End of Python Scipy



IP[y]:
IPython

