

Introduction to SciPy



Most of the materials are taken from [here](#)

Nastaran Okati

Outline

- ▷ Introduction
- ▷ Working with files
- ▷ Manipulating images
- ▷ Linear Algebra
- ▷ Integration
- ▷ Sparse matrices

1.

Introduction

What is SciPy?



- ▷ A framework built on top of NumPy
- ▷ Provides a large number of high-level scientific algorithms
- ▷ Provides implementations of a very extensive set of special functions

Some of the topics that SciPy covers:

- ▷ Special functions: `scipy.special`
- ▷ Integration: `scipy.integrate`
- ▷ Optimization: `scipy.optimize`
- ▷ Interpolation: `scipy.interpolate`
- ▷ Sparse matrices: `scipy.sparse`
- ▷ Signal Processing: `scipy.signal`
- ▷ Linear Algebra: `scipy.linalg`
- ▷ Statistics: `scipy.stats`
- ▷ Image Processing: `scipy.ndimage`
- ▷ File IO: `scipy.io`

2.

Working with Files

scipy.io

SciPy has many modules, classes, and functions available to read data from and write data to a variety of file formats.

Example:

Import scipy.io as sio

```
sio.loadmat('myfile.mat')
```

```
sio.savemat('myfile.mat', {'first':a, 'second':b})
```

3.

Working with Images

scipy.ndimage

This package contains various functions for multi-dimensional image processing.



Example:

Manipulating images:

```
from scipy import misc
from scipy import ndimage
Img = misc.imread('img.jpeg')
Shifted = ndimage.shift(img,(50,50,0))
Rotated = ndimage.rotate(img,50)
Img = ndimage.zoom(img,(2,2,1))
```

Exercise

Generate this figure using
Matplotlib and SciPy

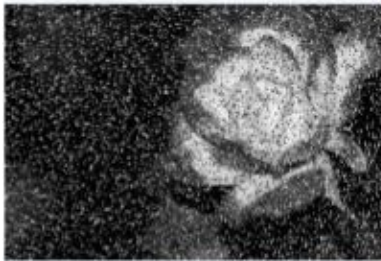


Blurring/Smoothing Effect

Modify the pixels in an image based on some function of a local neighborhood of each pixel.



Original



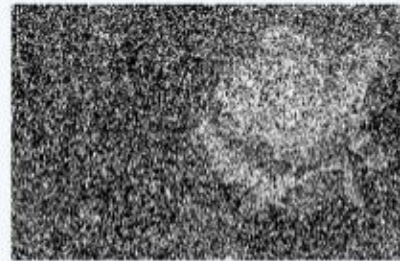
Salt & pepper
%20



De-noising by
Median filter



Original



Salt & pepper
%60



De-noising by
Median filter

Mean Filtering

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	0	0	0	0	0	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$F[x, y]$

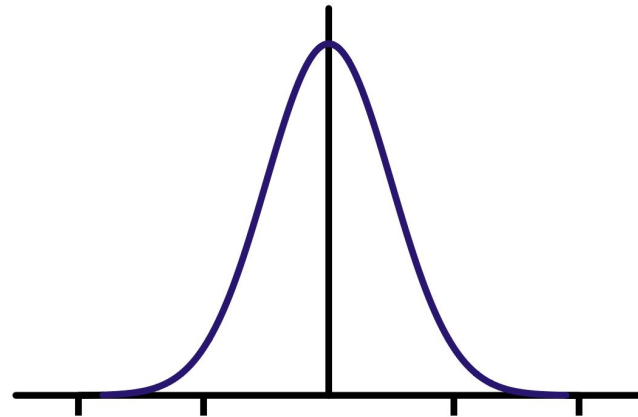
	0	10	20	30	30	30	20	10	
	0	20	40	60	60	60	40	20	
	0	30	60	90	90	90	60	30	
	0	30	50	80	80	90	60	30	
	0	30	50	80	80	90	60	30	
	0	20	30	50	50	60	40	20	
	10	20	30	30	30	30	20	10	
	10	10	10	0	0	0	0	0	

$G[x, y]$

Gaussian Filtering

$$\frac{1}{273}$$

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1



Blurring/Smoothing Effect

- ▷ `ndimage.guassain_filter`
- ▷ `ndimage.uniform_filter`
- ▷ `ndimage.minimum_filter`
- ▷ `ndimage.maximum_filter`
- ▷ `ndimage.median_filter`



4. Linear Algebra

scipy.linalg

The linear algebra module contains a lot of matrix related functions, including linear equation solving, eigenvalue solvers, matrix functions (for example matrix-exponentiation), etc.

- ▷ Linear Equation Systems
- ▷ Eigenvalues and eigenvectors
- ▷ Matrix operations

Linear Equation System

$$Ax = b$$

```
from scipy.linalg import *  
A = array([1,2,3],[4,5,6],[7,8,9])  
b = array([1,2,3])  
X = solve(A,b)
```

Eigenvalues and eigenvectors

- ▷ `evals = eigvals(A)`
- ▷ `evals, evects = eig(A)`

Matrix Operations

- ▷ `inv(A)`
- ▷ `det(A)`
- ▷ `norm(A, ord=2)`

5. Integration

scipy.integrate

- ▷ `Scipy.integrate.quad(f, x_lower, x_upper)`

Exercise:

1. Compute the integral of $f(x) = x^3 + 2x + 100$ in $[-100, 100]$
2. Write a program that gets n from the user and computes the Integral of $f(x) = x^n + nx$ in $[-100, 100]$

6.

Sparse Matrices

How to store sparse matrices?

- ▷ Coordinate Form (COO)
- ▷ List of List (LIL)
- ▷ Compressed Sparse Column (CSC)

Often a sparse matrix is initially created in COO or LIL format (where we can efficiently add elements to the sparse matrix data), and then converted to CSC before used in real calculations.

Example

From `scipy.sparse` import *

- ▷ `csr_matrix()`
- ▷ `todense()`
- ▷ `lil_matrix()`

7. Optimization

Finding a minima

```
from scipy import optimize
```

```
▷ X_min = optimize.fmin_bfgs(f,-2)
```

Exercise: Find the minimum of $f(x) = x^3 + (x-2)^2 + x^4$

Finding a solution for a function

```
from scipy import optimize
```

```
▷ ans = optimize.fsolve(f, 0.1)
```

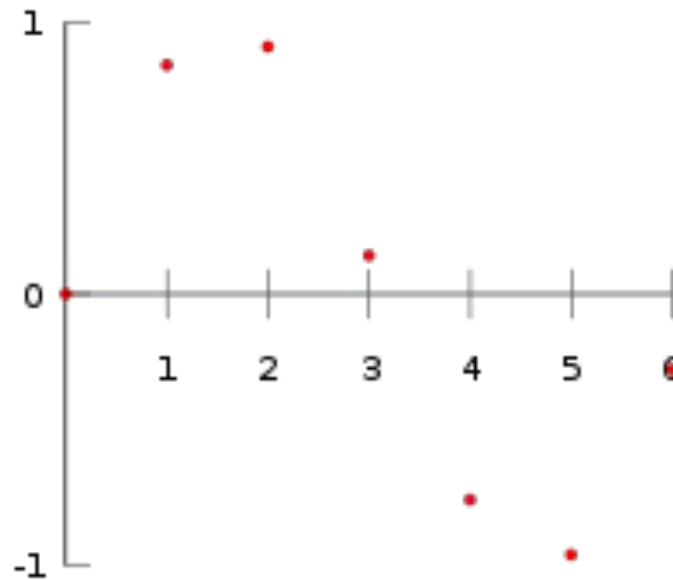
Exercise: Find a solution for the function in the previous slide

8.

Interpolation

What is Interpolation?

Interpolation is a method of constructing new data points within the range of a discrete set of known data points.



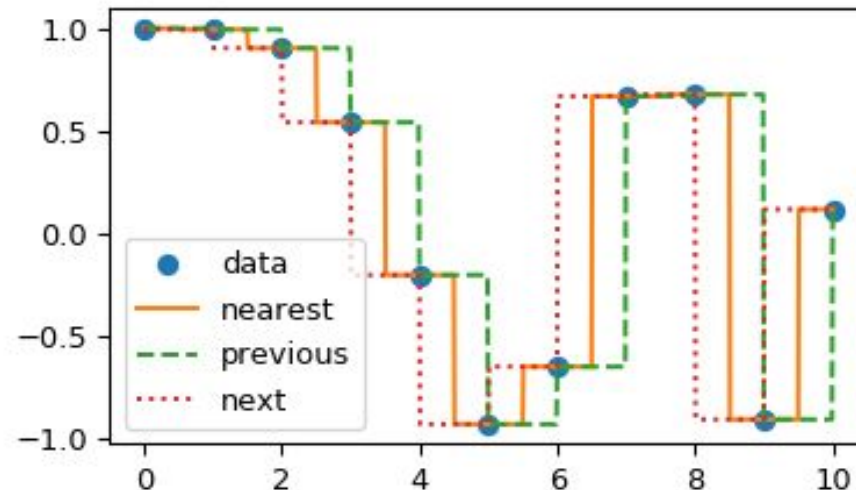
Piecewise constant interpolation

```
from scipy.interpolate import *
```

```
interp1d(x, y, kind='nearest')
```

```
f2 = interp1d(x, y, kind='previous')
```

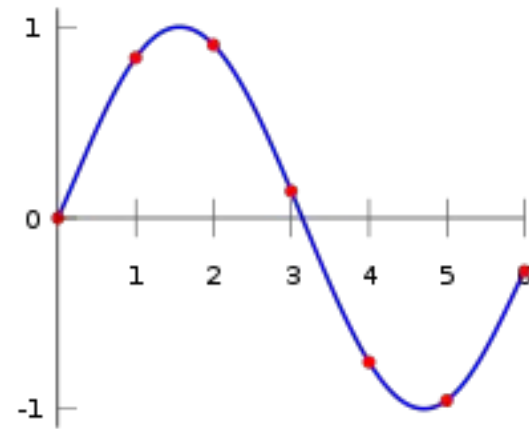
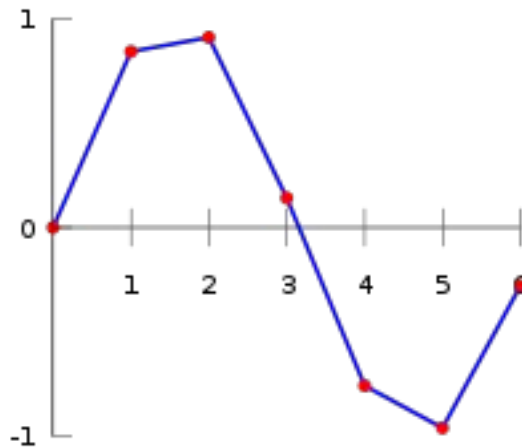
```
f3 = interp1d(x, y, kind='next')
```



Linear and polynomial interpolation

`f = interp1d(x, y)`

`f = interp1d(x, y, kind='cubic')`



Exercise

From the function $f(x) = \sin(x)$ get several data points and add small noise to the data points. Use `interp1d` with different kinds and plot the results using Matplotlib.

9. Statistics

Statistics

```
From scipy import stats
```

```
X = stats.norm()
```

```
X = stats.poisson()
```

```
X.pdf()
```

```
X.cdf()
```

```
X.rvs(size=1000,bins=50)
```

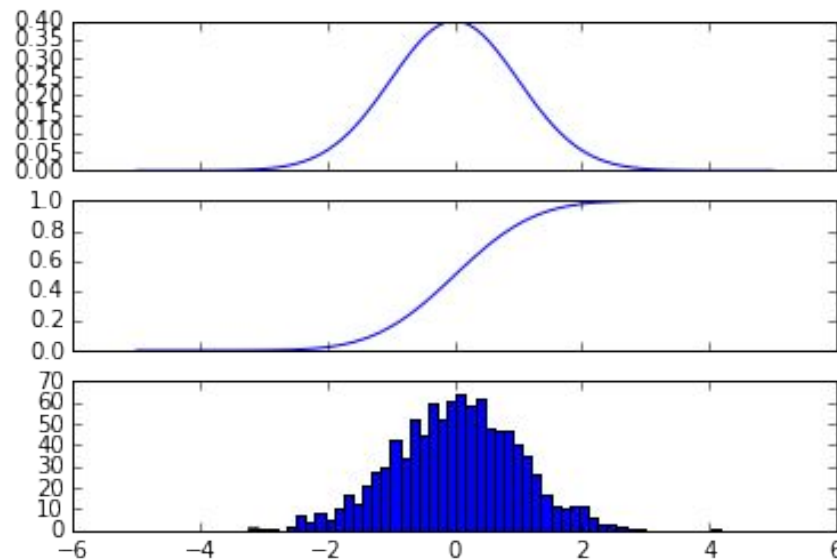
```
X.mean()
```

```
X.std()
```

```
X.var()
```

Exercise

Create a continuous random variable with normal distribution and plot the pdf, cdf and the histogram of 1000 random realizations of the variable



References

- ▷ <https://github.com/jrjohansson/scientific-python-lectures/blob/master/Lecture-3-Scipy.ipynb>
- ▷ <https://docs.scipy.org/doc/scipy/reference/>
- ▷ <https://data-flair.training/blogs/image-processing-with-scipy-and-numpy/>

Thanks!

Any questions?