

Math with Python

Programming Lecture 2

Nicholas Dwork

1

Numpy

A Python package that provides numerical routines

Vectors

Matrices

Matrix Methods

You must import Numpy before you use it

```
import numpy as np
```

```
# Can then use numpy packages with np later
```

2

Arrays

An array is an ordered set of numbers

```
import numpy as np  
array1 = np.array( [2 1 4] ) # makes array with values  
array2 = np.ndarray( [5,4] ) # makes array of size 5x4  
  
array3 = np.array( [ [ 2 1 4 ], [ 8 9 8 ] ] )  
# Makes two dimensional array with values  
  
array4 = np.zeros( [5, 4] ) # makes array of all zeros  
array4[1,1] = 8; # puts an 8 in the 1,1 position
```

3

Arrays

An array is an ordered set of numbers

```
import numpy as np  
array5 = np.linspace(0, np.pi, 100 )  
# array of 100 points evenly spaced between 0 and pi
```

4

Size of Array

The `shape` method provides the size of the array or matrix

Suppose `A` is a matrix

`A.shape[0]` is the number of rows

`A.shape[1]` is the number of columns

5

Solving Linear Systems

Suppose you want `x` such that $A x = b$

```
import numpy as np
xHat = np.linalg.solve( A, b )
```

```
# to get the Pseudo-inverse of A
A_dagger = np.linalg.pinv( A )
```

6

Compute the Norm

```
import numpy as np
normX = np.linalg.norm( x )
```

7

Plotting

**Plotting is encapsulated in the matplotlib package
We will use the pyplot subpackage**

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, np.pi, 100 )
    # array of 100 points evenly spaced between 0 and pi
y = np.sin(x)
plt.(x,y)
plt.show()
```

8

Save and Load Data

Also part of numpy

```
import numpy as np
```

```
a = np.linspace( 0, 1, 1000 )  
np.save( 'aFile.npy', a ) # save data as binary file  
np.savetxt( 'aFile.txt', a ) # save data as text file
```

```
np.load( 'aFile.npy', loadedBinaryA )  
np.load( 'aFile.txt', loadedTxtA )
```

9

Save and Load Images

Part of the image subpackage of matplotlib

```
import matplotlib.pyplot as plt  
import matplotlib.image as img  
import numpy as np
```

```
imgArray = img.imread( 'imgFile.png' )  
imgArray = imgArray[1:50,1:50]  
img.imsave( 'imgFile.png', imgArray )
```

10