Python Workshop
Numpy Arrays

C.D. Langevin

U.S. Geological Survey
Reston, Virginia, USA

USGS National Groundwater Workshop, August 2012

Outline

What is Numpy

- Numpy is the main package for scientific computing using Python
- Provides an array object of type ndarray
- Many functions and methods available for fast array operations

Numpy Version

- Numpy can be obtained at `http://docs.scipy.org/doc/`
- Current version is 1.6.
- To determine the installed version

```
In [10]: import numpy

In [11]: print numpy.__version__
1.6.1
```

USGS science for a changing world

Data Types (dtype)

bool	Boolean (True or False) stored as a byte
int	Platform integer (normally either int32 or int64)
int8	Byte (-128 to 127)
int16	Integer (-32768 to 32767)
int32	Integer (-2147483648 to 2147483647)
int64	Integer (9223372036854775808 to 9223372036854775807)
uint8	Unsigned integer (0 to 255)
uint16	Unsigned integer (0 to 65535)
uint32	Unsigned integer (0 to 4294967295)
uint64	Unsigned integer (0 to 18446744073709551615)
float	Shorthand for float64.
float16	Half precision float: sign bit, 5 bits exponent, 10 bits mantissa
float32	Single precision float: sign bit, 8 bits exponent, 23 bits mantissa
float64	Double precision float: sign bit, 11 bits exponent, 52 bits mantissa
complex	Shorthand for complex128.
complex64	Complex number, represented by two 32-bit floats (real and imaginary components)
complex128	Complex number, represented by two 64-bit floats (real and imaginary components)

http://docs.scipy.org/doc/numpy-1.6.0/user/basics.types.html

USGS science for a changing world

Creating an Array

- Using the built-in array function

```
In [18]: import numpy

In [19]: a1d = numpy.array([0, 1, 2, 3, 4])

In [20]: a1d
Out[20]: array([0, 1, 2, 3, 4])

In [21]: a2d = numpy.array([ [1, 1, 1, 1], [2, 2, 2, 2] ])

In [22]: a2d
Out[22]:
array([[1, 1, 1, 1],
       [2, 2, 2, 2]])
```

- Using the arange built-in function

```
In [25]: a = numpy.arange(10)

In [26]: a
Out[26]: array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])

In [27]: a = numpy.arange(0, 100, 10)

In [28]: a
Out[28]: array([ 0, 10, 20, 30, 40, 50, 60, 70, 80, 90])
```

USGS science for a changing world

Creating an Array

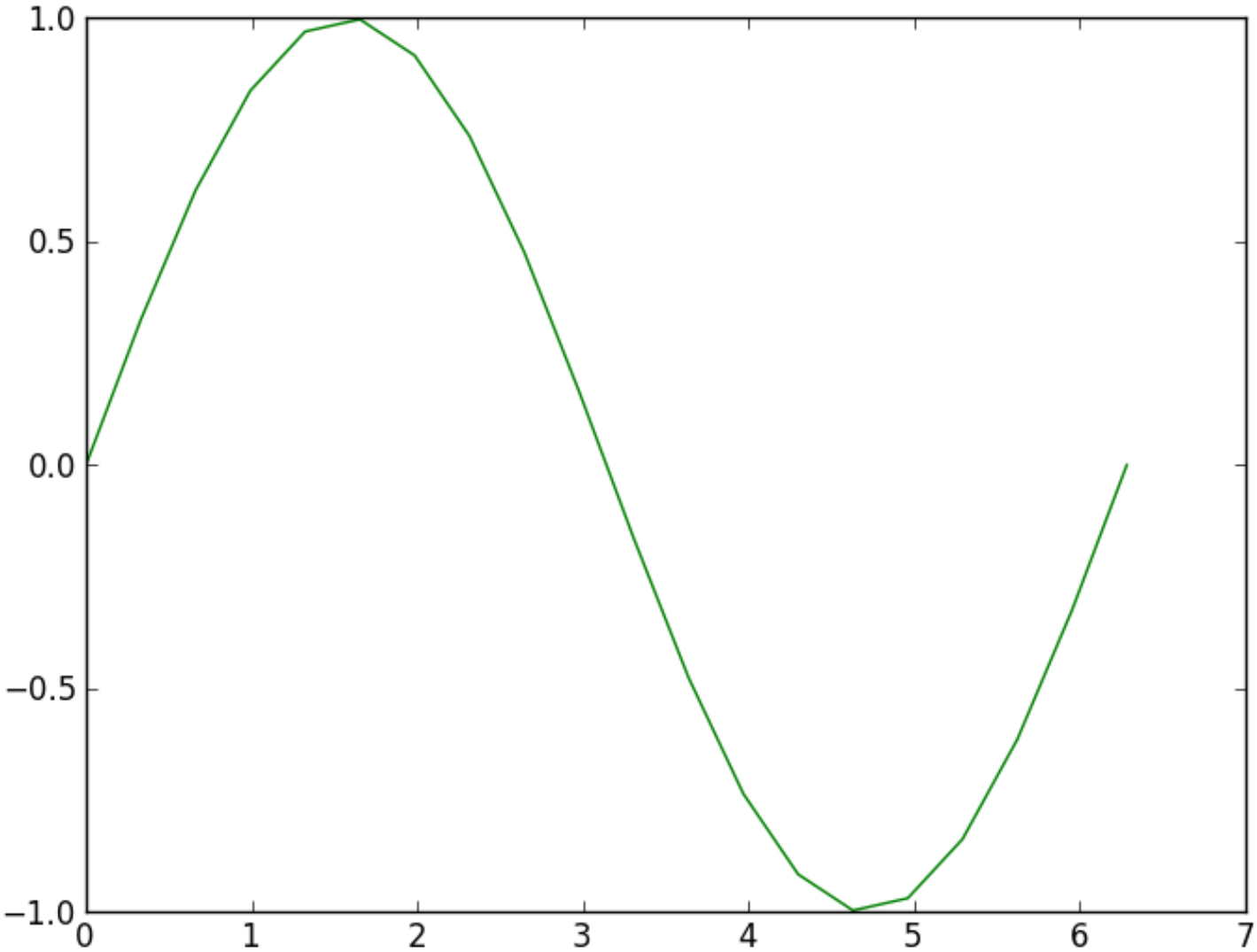
- Using the built-in linspace function

```
In [29]: a = numpy.linspace(0, 1, 7)

In [30]: a
Out[30]:
array([ 0.        ,  0.16666667,  0.33333333,  0.5       ,  0.66666667,
        0.83333333,  1.        ])
```

- Creating a new array from an existing array using a numpy function

```
In [16]: import numpy

In [17]: x = numpy.linspace(0, 2 * numpy.pi, 20)

In [18]: y = numpy.sin(x)

In [19]: plot(x, y)
Out[19]: [<matplotlib.lines.Line2D object at 0x04A7B8B0>]

In [20]: show()
```

Loading an Array from a File



Loading an Array from a File

- If we have the following table stored in an ascii text file

1	3	6	9
12	15	18	21
23	26	29	31
77	78	79	2

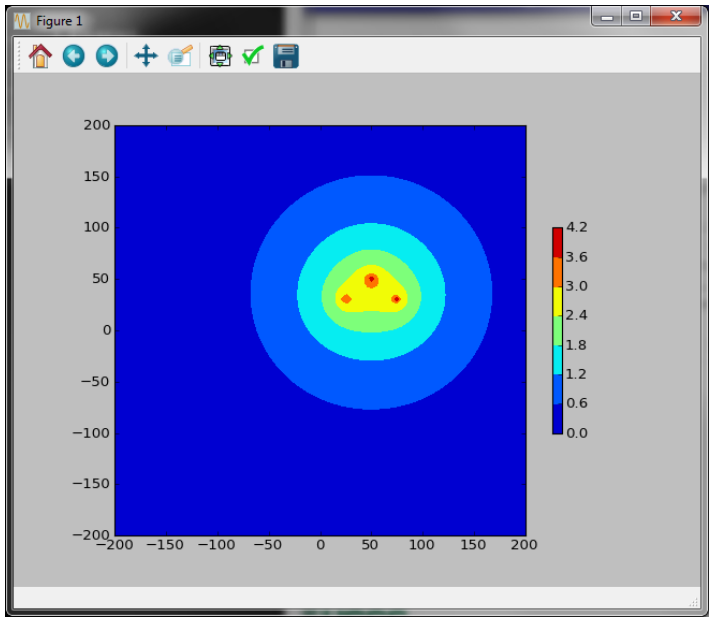
- The table can be loaded into a numpy array using the numpy loadtxt function

```
In [28]: a = numpy.loadtxt('data\\4by4array.txt')

In [29]: a
Out[29]:
array([[  1.,   3.,   6.,   9.],
       [ 12.,  15.,  18.,  21.],
       [ 23.,  26.,  29.,  31.],
       [ 77.,  78.,  79.,   2.]])
```

Theis Example

theis.py

```
1  import numpy as np
2  from scipy.special import expn
3
4  def theis(Q, T, S, t, r):
5      return Q / 4. / np.pi / T * expn(1, r ** 2 * S / 4. / T / t)
6
7  T = 1.; S = 0.001; t = 10.; rw = 0.01
8  xmin = -200; xmax = 200; nx = 100; ymin = -200; ymax = 200; ny = 100
9  x, y = np.meshgrid(np.linspace(xmin, xmax, nx), np.linspace(ymin, ymax, ny))
10 wells = [ [25, 30, 3], [50, 50, 3], [75, 30, 3] ]
11 ddn = np.zeros( x.shape, dtype=float)
12 for xw, yw, Qw in wells:
13     print 'processing well: ', xw, yw, Qw
14     r = np.sqrt((x - xw) ** 2 + (y - yw) ** 2)
15     np.where(r > rw, r, rw)
16     ddn = ddn + theis(Qw, T, S, t, r)
17
18 from matplotlib.pyplot import *
19 try:
20     close('all')
21 except:
22     pass
23 subplot(1, 1, 1, aspect='equal')
24 contourf(x, y, ddn)
25 colorbar(shrink=0.5)
26 show()
```



USGS science for a changing world

Indexing, Slicing, and Iterating

Shape Manipulation