

# AMS 595: Fundamentals of Computing: Part II

## Lecture 6: Array Computation and Curve Plotting

Xiangmin Jiao

Stony Brook University

# Outline

1 Array Computations with NumPy

2 Curve Plotting

3 File I/O for Arrays

# Introduction to NumPy

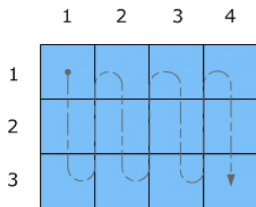
- NumPy provides multidimensional arrays
  - ▶ All elements must be the same data type
  - ▶ Many different data types supported
  - ▶ Size is fixed (memory is allocated for the size specified)
- Arithmetic operations work on arrays
- Provides MANY functions that operate on whole arrays
  - ▶ These operations are written in a compiled language, and run fast
  - ▶ Generally speaking, you want to avoid loops to get the best performance, although it may make code unreadable

# Intro to NumPy: Array Operations

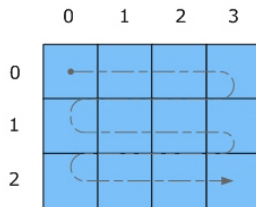
- Arithmetic operator (+, -, /, \*) work element-wise
  - ▶  $A * B$  is not a matrix product, but is element-wise multiplication, similar to `.*` in MATLAB
  - ▶ instead multiplies the corresponding elements in each array together
  - ▶ `dot(A, B)` does a dot product
- Universal functions (sin, cos, exp, ...) work element-wise
- NumPy has a matrix class, which is a subclass of array
  - ▶ `*` are matrix-matrix multiplication
- See [Numpy for Matlab users](#) for mapping from MATLAB to NumPy
- Many linear-algebra functions are provided SciPy (next lecture)
- See [Jupyter notebook on NumPy](#)

# Key Differences Between NumPy and MATLAB

- 1-based (MATLAB) vs. 0-based (Python) indexing
- Pass by value (MATLAB) vs. pass by reference (Python)
  - ▶ In Python, slicing creates a “view” into an array
- Column major vs. row major
  - ▶ MATLAB (like Fortran) uses *column major*
  - ▶ Python (like C) uses *row major*
  - ▶ This is important to understand for efficiency and for passing data between different languages



A: Column-major order (Fortran-style)



B: Row-major order (C-style)

# Outline

1 Array Computations with NumPy

2 Curve Plotting

3 File I/O for Arrays

# Plotting using Matplotlib

- [Matplotlib](#) is standard plotting library for scientific python
  - ▶ Mostly for 2-d data
  - ▶ Can use LaTeX commands for equations
  - ▶ Can produce “publication-quality” plots; see [gallery](#) for examples
- Importing Matplotlib
  - ▶ matplotlib is the entire package
  - ▶ matplotlib.pyplot is a module within matplotlib that provides easy access to the core plotting routines
  - ▶ pylab combines pyplot and numpy into a single namespace to provide MATLAB-like interface
- See [Jupyter notebook on Matplotlib](#) with [this sample data](#)

# Outline

1 Array Computations with NumPy

2 Curve Plotting

3 File I/O for Arrays



# Text and Binary File

- `numpy.loadtxt()` provides easy way to load rows of data from text file
- SciPy's `scipy.io` module provides methods to read and write MATLAB's MAT files, which are binary files