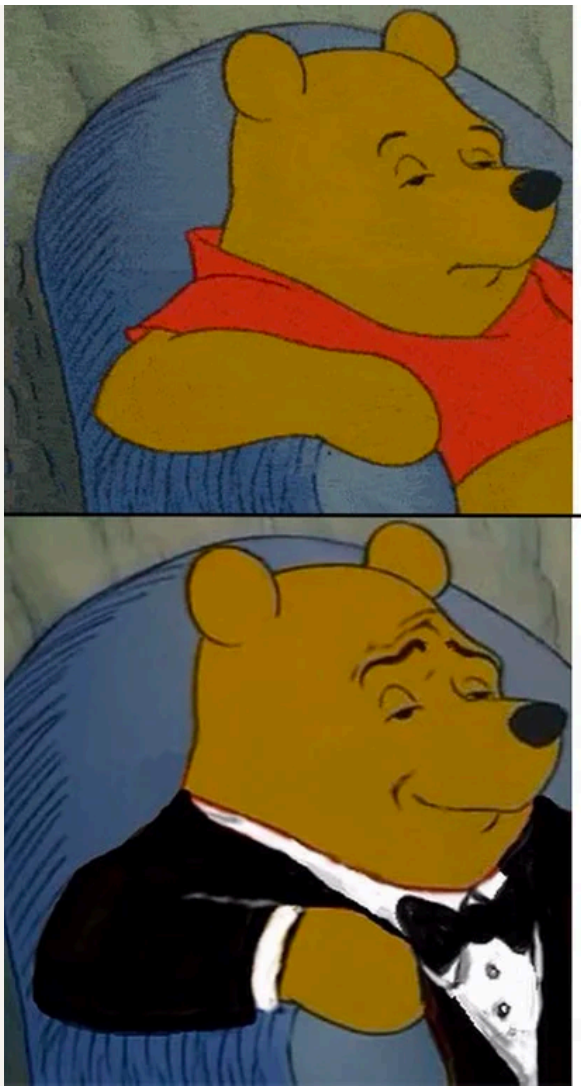
Basic Matrix Algebra in Python

CS 111: Introduction to Computational Science

Spring 2019 Lecture #2

Ziad Matni, Ph.D.



Using a scientific calculator

Using Python as a scientific calculator

Review of Some Linear Algebra

- N equations, N unknowns ($\mathbf{Ax} = \mathbf{b}$)
- Matrix view vs. Column view
- How to multiply matrices
- How to find inverse, transpose matrices
- Are matrices associative, distributive, commutative?
- Special matrices: I, U, L
- Determinants
- Eigenvalues and eigenvectors

N Linear Equations in *N* Unknowns

Example: How can we solve:

$$2x - y = 0$$

$$-x + 2y = 3$$

Using classical algebra?

Using matrix algebra? Column view (vector analysis)

Matrix Multiplication

Given:

$$U = \begin{bmatrix} 2 & 7 & 1 & 8 \\ 0 & 2 & 8 & 1 \\ 0 & 0 & 8 & 2 \\ 0 & 0 & 0 & 8 \end{bmatrix} \quad L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.5 & 1 & 0 & 0 \\ 0 & 0.5 & 1 & 0 \\ -0.5 & -0.5 & 0 & 1 \end{bmatrix}$$

Find A = L U

Is it the same as A = UL?

Use of Python

- We will be using the **numpy** module extensively
  - Numerical computing with arrays and matrices
  - https://docs.scipy.org/doc/numpy/reference/
  - http://www.numpy.org/

- Also, read more about **scipy** and **matplotlib**:
  - **scipy**: More advanced numerical computing, including sparse matrices
  - https://docs.scipy.org/doc/scipy/reference/
  - **matplotlib**: Plotting and visualization
  - https://matplotlib.org/contents.html

- A note on installs:
  - Don't have to install piece-wise, just install Anaconda (https://www.anaconda.com/distribution/)

```
# These are the standard imports for CS 111.

import os
import time
import math
import numpy as np
import numpy.linalg as npla
import scipy
from scipy import sparse
from scipy import linalg
import scipy.sparse.linalg as spla
import matplotlib.pyplot as plt
from matplotlib import cm
from mpl_toolkits.mplot3d import axes3d
%matplotlib tk
```

Examples in **numpy**

Create a 2x2 matrix $\begin{bmatrix} 2 & 3 \\ -1 & 2 \end{bmatrix}$

```
import numpy

A = numpy.array( [ [2, 3] , [-1, 2] ] )          # array()
print(numpy.linalg.matrix_rank(A))               # .linalg
MyX = numpy.round(10*numpy.random.rand(5))       # .round(), .rand()
b = A @ MyX                                  # @ operator: matrix multp.

X = numpy.linalg.solve(A,b)                      # solve()
```

Let's Do this In Python!

We turn to a demonstration using

Jupyter Notebook

You will get the entire transcript posted on our class' Main Website

A Note on Homework #1

- It's a Review Quiz
  - Concepts of Linear Algebra & Diff. Equations

- You will be given the document in BOTH **.pdf** and **.tex**
  - Use the **.tex** file as your template to create a LaTeX document
  - You will turn in your assignment as **.pdf** only on Gradescope
  - Section this week will review how you should do this

Your To-Dos

- Keep going with your reviews on Python & Linear Algebra & Diff. Eqs.

- Read **NCM** Sections 2.1 through 2.6
  (linear equations and Gaussian elimination)
  - Remember that this is written for MATLAB, so read/skim accordingly
  - Think about how it might be implemented in Python
    (we'll go over that in class too)

- Turn in homework #1 (Review Quiz) – due **Monday, April 8th**

</LECTURE>