MATLAB Assignment 2

Spring 2018, Section B

January 24, 2018

This problem set will cement your understanding of array operations and go over several important built in functions. These operations are especially helpful in avoiding loops, we will explore the differences in speed and elegance. Homework is due on January 31st, 2018 to ccnezin+ece210@cooper.edu.

1. Vector? I barely know her! Here we will look at some applications of built in vectorized functions.

(a) Create a vector of 100 evenly spaced samples of the exponential function from 0 to 1.

(b) Approximate the integral using the trapezoid method (use *trapz* and multiply by the interval) and rectangular method (sum over all points and multiply by the interval).

(c) Approximate the cumulative integral using the trapezoidal method (use *cumtrapz*) and rectangular method (use *cumsum*).

(d) Approximate the derivative by taking the difference between all adjacent elements and dividing by the time spacing. Similarly, approximate the second derivative. What are the lengths of each derivative vector?

(e) Given the vector [0 1 2 3 4 5], create the vector [3 4 5 0 1 2] using *circshift*.

2. Array Foray Perform the following matrix operations.

(a) Use *reshape* to create a 10×10 matrix A where $A = \begin{bmatrix} 1 & 11 & ... & 91 \\ 2 & 12 & ... & 92 \\ \vdots & \vdots & \ddots & \vdots \\ 10 & 20 & ... & 100 \end{bmatrix}$.

(b) Use *magic* to create a 10×10 magic matrix B. Use B to create a matrix C which has the same diagonal values of B and is zero elsewhere. **Note**: You might want to look up *diag* to see how to do this elegantly.

(c) Flip the second column of B such that the column is inverted up down.

(d) Flip the matrix A from left to right.

(e) Make cSum the column-wise sum of every column of AB (normal matrix multiplication). The result should be a row vector.

(f) Make cMean the row-wise mean of every row of AB (element-wise matrix multiplication). The result should be a column vector.

(g) Delete the last column of A.

3. Gotta Go Fast Generate a 300×500 matrix with entries $a_{i,j} = \frac{i^2+j^2}{i+j+3}$ using the following methods and use *tic toc* to time the speed of each and report the times in a table (using *table* function).

(a) Using for loops and no pre-allocation.

(b) Using for loops and pre-allocating memory with *zeros*.

(c) Using only elementwise matrix operations. **Note**: *repmat* and *meshgrid* will be useful here.