AMATH 301 – Winter 2023
Homework 1

Due: 11:59pm, January 13, 2022.

Instructions for submitting:

- Coding problems are submitted to Gradescope as a python script (`.py` file) or a Jupyter Notebook (`.ipynb` file) (warning, this may not work as well as a python script). You have **10** attempts (separate submissions to Gradescope) for the coding problems.
- Writeup problems are submitted to Gradescope as a single `.pdf` file that contains text and plots. Put the problems in order and label each writeup problem. When you submit, **you must indicate which problem is which on Gradescope**. Failure to identify pages on Gradescope will lead to a **5% grading penalty. All code you used for this part of the assignment should be included at the end of your .pdf file. Failure to do so results in a 25% penalty.**

Coding problems

This section is worth 5 points, with each variable having equal weight.

1. **Parts (a) - (d) of this problem are completed for you to show you how to save variables to make sure that they are checked by the autograder.** Define the following variables in python:
   (a) Define $x = 3.1$ and save it to the variable `A1`.
   (b) Define $y = -29$ and save it to the variable `A2`.
   (c) Define $z = 9e$ (where e is the number such that $\ln(e) = 1$) and save it to the variable `A3`.
   (d) Define $w = e^4$ (where e is the number such that $\ln(e) = 1$) and save it to the variable `A4`.
   (e) Compute $\sin(\pi)$ in python using `numpy`. Save the result to the variable `A5`.
2. Create/calculate the array $[\cos(0), \cos(\pi/4), \cos(\pi/2), \cos(3\pi/4), \cos(\pi)]$ and save the result to the variable `A6`. **(Hint: I have given you code that creates the array $\mathbf{x} = [0, \pi/4, \pi/2, 3\pi/4, \pi]$, how can you use this to help you solve this problem?**

3. Use `linspace` and `arange`, in the `numpy` package, to create the vectors $\mathbf{u}$ and $\mathbf{v}$ where $\mathbf{u}$ is the vector with 6 evenly spaced elements beginning with 3 and ending with 4 and where

$$\mathbf{v} = \begin{pmatrix} 0 & 0.75 & 1.5 & 2.25 & 3.0 & 3.75 \end{pmatrix}. \tag{1}$$

   (a) Save the array $\mathbf{u}$ to the variable `A7`.

   (b) Save the array $\mathbf{v}$ to the variable `A8`.

   (c) Save the array $\mathbf{w} = \mathbf{v} + 2\mathbf{u}$ to the variable `A9`.

   (d) Using python commands, create a vector $\mathbf{w}$ whose entries are the entries of $\mathbf{u}$ cubed (i.e., to the third power). Save the result to the variable `A10`.

   (e) Using python commands, create a vector $\mathbf{x}$ whose entries are $\tan(\mathbf{u}) + e^{\mathbf{v}}$. Save the result to the variable `A11`.

   (f) Use **indexing** to save the 3rd number in $\mathbf{u}$ to the variable `A12`. *Hint: Remember that indexing begins at 0 for python.*

4. Create an array z with elements that have spacing $1/100$ between them and that begins at -6 and ends at 3.

   (a) Save the array z to the variable `A13`.

   (b) Use indexing to create another array that consists of every other entry in z. Save that array to the variable `A14`.

   (c) Use indexing to create another array that consists of every *third* entry in z. Save that array to the variable `A15`.

   (d) Use indexing to create an array consisting of *the last 5* entries in z. Save that array to the variable `A16`.

Writeup problems

This section is worth 5 points.

You may want to start a new python script for the writeup portion of the assignment, or you may want to use a Jupyter notebook.

1. The following code will create a straight line. Copy it, save the plot, and add it to your writeup (**NOTE: You may not be able to copy and paste directly. Instead, just recreate this code**).

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


# First create the array "x" between -5 and 5
x = np.arange(-5, 5+0.5, 0.5)
```

```
# Then we want to plot y = x, so we define that
y = x

# Now that we have both arrays plotted we can plot using "plt.plot"
plt.plot(x, y, 'k') # This creates a black line.

# Suppose we also want to highlight a few points, the points at which we have data,
# we can do so with blue markers.
plt.plot(x, y, 'bo')

plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

Explain the connection between the blue circles in the plot and the black curve. How are they related?

2. Create another figure similar to the previous figure except representing the function $f(x) = x^2$. Add a title to the figure explaining what is being plotted.