ECIV521 - Homework 1, Matthew Hatami

Attached with this file you can find 3 python codes as below - with their description:

exercise1.py:
In this python file I have answered exercise 1 in homework. Running this code you will get 4 real number arrays that I built with different methods. The first one, I compiled it manually, the second one, I used 'np.arange()' method, for third one I used 'np.linspace()', and for the last one, I generated an array of 10 random numbers with 'np.random.rand()'

In Exercise 1, we were asked to discuss similarities and differences between the methods above as well, here are what I think:

Similarities:

- In all methods used in my codes, or even the ones that I haven't, we can generate an array of numbers and after assigning the array to a variable, they won't be different from each other, we can treat all of them similarly, no matter which method we used to generate that specific array.

Differences:

- Each method has specific features, and they can become handy in different situations. For example, when we have the first and last value and we need to have values evenly spaced between them, we can use 'np.linspace()', or or 'np.arange()'. The difference between these two is that in 'np.arange()', we can specify the step size, but in 'np.linspace()', we can specify the number of values we want to have between the first and last value. And the last method, 'np.random.rand()', is used to generate random numbers between 0 and 1 when we need to do so.

exercise2.py:
This code is my answer to exercise 2. By running this code, you will have a pop-up window of two plots as it was asked in the question. I have also saved and attached the plots in a pdf file as well.

Exercise2_plot.pdf:
This is the output of exercise2.py saved in pdf format.

Code.1py:
This file includes the codes required to plot the solution for code 1 exercise in homework 1. In this code I'm approximating the sine function with taylor expansion. The taylor series expansion breaks the original function in the sum of polynomial terms. I have used 6 different values for 'm' to calculate the taylor series approximate up to that number, and then defined x from -5 to 5. Then I calculated the exact Sine values, as well as its approximations to show how increasing m affects the approximation. As the plot suggests, for m values larger than 5, the approximation and the sine function are almost identical, in the given period.