MAE 3403 **Fall 2019** **Homework 4**

Due at 11:59 pm, Monday, Sept. 30th, in the Canvas Dropbox

You will be writing three different Python programs, and you will upload a single .zip file named Homework4.zip.

In this assignment, you must use variables, loops, if statements, your own function definitions and function calls to write the required functions. For now, you may not use any of the powerful functions available in python modules, with a few exceptions: You may import functions from the ***math***, ***copy***, ***matplotlib.plot*** and ***numpy***. *You may **NOT use numpy.linalg.lstsq(). You may use fsolve() and quad() from scipy, but nothing else.***

a) Write a program that demonstrates the Least Squares Curve Fitting method. You must write and call at least the following 3 functions:

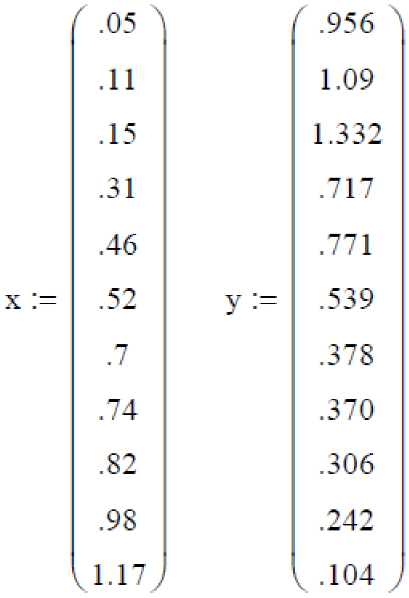
   def LeastSquares(x,y,power): # which calculates and returns an array containing the coefficients of the least squares polynomial

   def PlotLeastSquares(x,y,power, showpoints = True, npoints = 500): # which calls LeastSquares, generates datapoints and plots the least squares curve. If showpoints is True, also put the original data on the same plot.

   def main():

   A main program that uses the data given below to:

   1. Call LeastSquares to generate and print the coefficients of a Linear fit.
   2. Call PlotLeastSqares to display a plot for the Linear fit
   3. Call LeastSquares to generate and print the coefficients of a Cubic fit.
   4. Call PlotLeastSqares to display a plot for the Cubic fit
   5. Uses the results from 1 and 3 to plot the datapoints, the Linear fit and the Cubic fit, all on one graph. Use proper titles, labels and legends.



$$x := \begin{pmatrix} .05 \\ .11 \\ .15 \\ .31 \\ .46 \\ .52 \\ .7 \\ .74 \\ .82 \\ .98 \\ 1.17 \end{pmatrix} \qquad y := \begin{pmatrix} .956 \\ 1.09 \\ 1.332 \\ .717 \\ .771 \\ .539 \\ .378 \\ .370 \\ .306 \\ .242 \\ .104 \end{pmatrix}$$

b) Write a program that demonstrates the Cubic Spline method. You must write and call at least the following 3 functions:

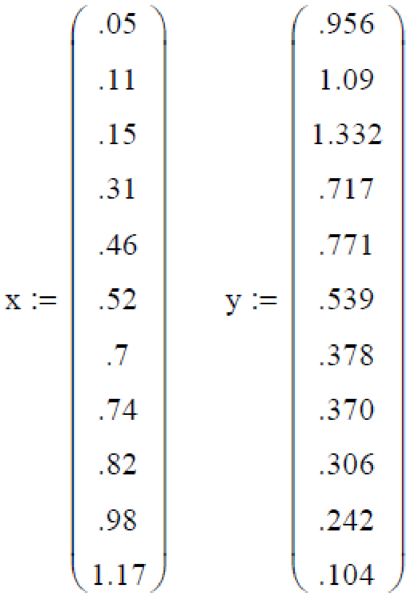
def CubicSpline(x,y,slope1 = 0, slope2 = 0): # which calculates and returns a matrix containing the coefficients of the cubic splines. Slope1 and slope2 are the slopes at the first and last points.

def PlotCubicSpline(x,y,slope1 = 0, slope2 = 0 , showpoints = True, npoints = 500): # which calls LeastSquares, generates datapoints and plots the least squares curve. If showpoints is True, also put the original data on the same plot.

def main():

A main program that uses the data given below to:

1. Call CubicSpline to generate and print the coefficients
2. Call PlotCubicSpline to display a plot of the cubic spline



$$x := \begin{pmatrix} .05 \\ .11 \\ .15 \\ .31 \\ .46 \\ .52 \\ .7 \\ .74 \\ .82 \\ .98 \\ 1.17 \end{pmatrix} \qquad y := \begin{pmatrix} .956 \\ 1.09 \\ 1.332 \\ .717 \\ .771 \\ .539 \\ .378 \\ .370 \\ .306 \\ .242 \\ .104 \end{pmatrix}$$

c) Write a program that demonstrates the Fourier Series method as defined on the MAE 3013 equation sheet. You must write and call at least the following 3 functions:

```
def FourierCoeffs(func, L, nterms)  # which calculates and returns the Fourier a's and b's
                                    #  the a's and b's are each numpy arrays of length nterms
                                    #  func is the name of the function.  L is the half-period.
                                    #  this function must call scipy.integrate.quad()

def PlotFourier(a, b, L, xmin, xmax  npoints=5000)  #  which plots the Fourier Series

def main():

    def Sharkfin(x):
        if x < 0: return (x+1)**3
        else: return 1-x**3

    def Squarewave(x):
        if x < 0: 1
        else: return -1



    L = 1
    a, b = FourierCoeffs(Sharkfin, L, 50)
    print(a, "\n",b,"\n")
    PlotFourier(a,b, L, -3*L, 3*L)

    L = 0.75
    a, b = FourierCoeffs(Squarewave, L, 50)
    print(a, "\n",b,"\n")
    PlotFourier(a,b, L, -3*L, 3*L)

    L = np.pi
    a, b = FourierCoeffs(lambda x: -x, L, 10)
    print(a, "\n",b,"\n")
    PlotFourier(a,b, L, -4*L, 4*L, npoints=10000)

main()
```