MATH 131: Numerical Methods for scientists and engineers - Assignment 3

Homework Assignment 3 due by 11:59 PM, March 15, 2020. Test your answers in Matlab
For this assignment you are allowed **5 submissions per exercise.**
Save each code as an .m file. We will use those codes later in class.

Exercises in Matlab Grader:

1. (20 pts) Write a `MATLAB` function, called `Lagrange_poly` that inputs a set of data points $(x, y) =$ `(datx, daty)`, a set x of numbers at which to interpolate, and outputs the Lagrange polynomials L and the polynomial interpolant, y, evaluated at x using Lagrange polynomial interpolation. Your function header should look something like:

```
function [y,L] = Lagrange_poly(x,datx,daty)
```

   See the template in Matlab Grader to help get you started.

2. (20 pts) Write a `MATLAB` function, called `Newtons_divided_differences` that inputs a set of data points $(x, y) =$ `(datx, daty)`, a set x of numbers at which to interpolate, and outputs the divided differences table F and the polynomial interpolant, y, evaluated at x using Newton's divided differences. Your function header should look something like:

```
function y = Newtons_divided_differences(x,datx,daty)
```

3. (20 pts) Write a `MATLAB` script to calculate the natrual cubic spline interpolation for a set of data points `datx` and `daty`, given below. The referenced code on Matlab Grader, `cubic_spline_coefs.m`, takes as input your set of datapoints, datx and daty, and calculates and returns the a_i, b_i, c_i, and d_i coefficients necessary for cubic spline interpolation. Interpolate over the points from 0 to 4 in increments of 0.01 (call this variable `x`). Make sure to calculate the cubic spline interpolation (call this variable `spline`). The data points are given by $(0, 1), (0.5, e^{0.5}), (2, e^2)$, and $(4, e^4)$. See the template to help you get started.

Exercises due on Catcourses:

4. (20 pts) *Show all code and work. I suggest checking out the video on Catcourses for using the 'Publish' feature in Matlab to ensure everything is included.* Consider the census population of the United States which is taken every 10 years. The following table displays the data from 1960 to 2010:

Year	1960	1970	1980	1990	2000	2010
Population (in thousands)	179,323	203,302	226,542	249,633	281,422	308,746

   (a) Use your `Lagrange_poly` function or your `Newtons_divided_differences` code to create an interpolating polynomial over the interval 1950 to 2020 in increments of 0.1 years.

   (b) Calculate the cubic spline interpolation over the interval 1950 to 2020 in increments of 0.1 years. Use your code from question 3 to help you figure out how to do it. No using built-in Matlab functions. For the years 1950-1960, use the interpolating cubic function calculated from 1960-1970 to extrapolate to the years 1950-1960. For the years 2010-2020, use the interpolating cubic function calculated from 2000-2010 to extrapolate to the years 2010-2020.

(c) On one figure, plot the true data points in black circles, the Lagrange/Newtons's interpolating polynomial in blue dashed line, and the cubic spline interpolating polynomial in red solid line. Make sure to label your axes and include a legend

(d) What are the predictions for the population in 1950 for your two interpolants? Calculate the relative error and comment on which is more accurate to the true 1950 population, which was 150,697,360. Why do you think the one method is more accurate than the other?

5. (20 pts) The fastest time ever recorded in the Kentucky Derby was by a horse named Secretariat in 1973. He covered the 1.25 mile track in 1 minute and 59.4 seconds. Times at the quarter-mile, half-mile, and mile poles were 25.2 seconds, 49.2 seconds, and 1:36.4 minutes.

   (a) Create an interpolating polynomial by hand (using either Lagrange Interpolating Polynomials or Newton's Divided Differences) to predict the time at the three-quarter mile pole and compare this to the actual time of 1:13.

   (b) Use the derivative of the interpolating polynomial to estimate the speed of Secretariat at the end of the race.