Instructor: Dr. S. Hughes

Due date: Sun, March 3 (midnight)

Lab 5 - Interpolation

Keywords: Monomial interpolation, Runge function, Langrange interpolation, Chebyshev nodes, Cubic Spline interpolation, Trigonometric interpolation

Task: Go through all the questions and coding exercises below and hand in a single modularized Python code where each part runs each part of the question(s). Remember and include your surname at the beginning of the code name.

Code Submissions on onQ: Attach a single (preferable, or no more than several) .py (called LabX_YourLastName.py) and single .py.txt (same code, same name, just a different extension).

Marks: Coding Skills, Efficiency, Correct Results and Clear Graphs and Good Presentation of Results (6); Good use of Comments (2); Good Coding Structure and Readability (2). Total: (10)

Reminder: All codes must run under Python 3.9.x. and the Spyder IDE (check this before submitting). Section off the code by using #%% (and label sections in code), and add useful comments, so we can step through it easily.

Acknowledgements: Please comment on any help that you received from your group members or others concerning this Lab assignment.

Some Background (in addition to lecture notes):

Polynomial interpolation - Wikipedia
Spline interpolation - Wikipedia
Trig interpolation - Wikipedia
Rubin Landau (video lecture)

Question 1

For this question, where deemed necessary, you can use your earlier Gaussian elimination code with pivoting, or simply use the `solve` command from `scipy.linalg`. We are going to explore some of the pros and cons of the various interpolation techniques we covered in the lecture notes, implemented in some nice Python code.

Implement a code to call the Runge function:

$$f^R(x) = \frac{1}{1+25x^2}, \tag{1}$$

from x between $[-1, 1]$, with n points, with a users choice of either equidistant spacing or a spacing using Chebyshev nodes. For example, a user might call something like:

```
dataxs, datays = generatedata(n, funR, "cheb")
```

Note that "nodes" are the data points we are using to interpolate to any other value of x, f. So we think of the nodes here as *known data*, and the other parts are unknown, even though we are using an analytical function here as a test case. Question 2 will use only data points (nodes) rather than a function to generate data point nodes.

Now, using $n = 15$ (number of points to interpolate to, let us call these: x_i), with both equidistant and Chebyshev nodes, compute Monomial and Lagrange interpolation using 100 equidistant x points (the new desired point). Show the solutions on two graphs, side by side, with a legend labelling the Lagrange, Monomial, and data points used for the interpolation. I obtain the result shown in Fig. 1, as a reference.

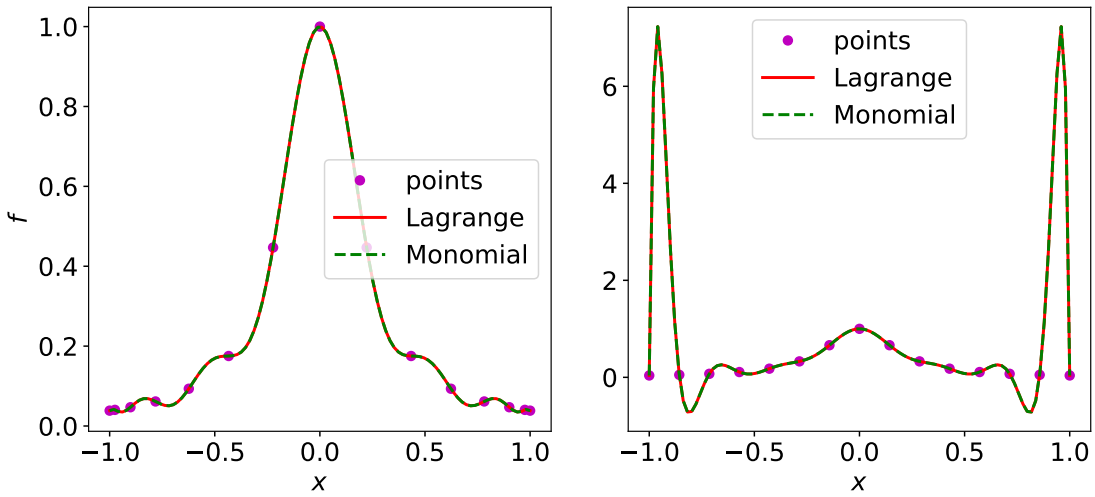

Figure 1: Lagrange and Monomial interpolation of the Runge function using $n = 15$, on Chebyshev (left) and equidistant nodes (right).

Question 2

You are given the following set of "data" points, with x and y values, where $y(x)$ represents some functional dependence:

```
datax = np.array([2.5,2.8,3.0,3.3,3.7,4.1,4.3,4.7,
                  5.,5.2,5.6,6.1,6.5,6.7,7.1,7.2,7.5])

datay = np.array([1.084e-17, 6.7041e-11, 1.125e-07, 2.359e-04,
       5.749e-02, 5.188e-01, 7.865e-01, 9.919e-01,
       1.000e+00, 9.984e-01, 8.784e-01, 2.312e-01,
       6.329e-03, 2.359e-04, 3.579e-09, 6.704e-11,
       1.084e-17])
```

First obtain a Lagrange interpolation to the data using 500 equidistant x (desired) points, between $x = 2.5$ and $x = 7.5$ (same limits as the data nodes), so your code might look something like:

```
itot = 500 # number of points to compute for x_i
xmin = 2.5
xmax = 7.5

xs = np.zeros(itot)
ys = np.zeros(len(xs))
xs = xmin+(xmax-xmin)*np.arange(itot)/(itot-1)

# call the Lagrange interpolation
for i in range(len(xs)):
    ys[i] = lagrange(datax, datay, xs[i])
```

Next, implement a subroutine to do a cubic spline interpolation, and obtain a the cubic spline interpolation of the same data (given 15 node points). Show both interpolation solutions graphically, alongside the data. An example solution is shown on Fig. 2.

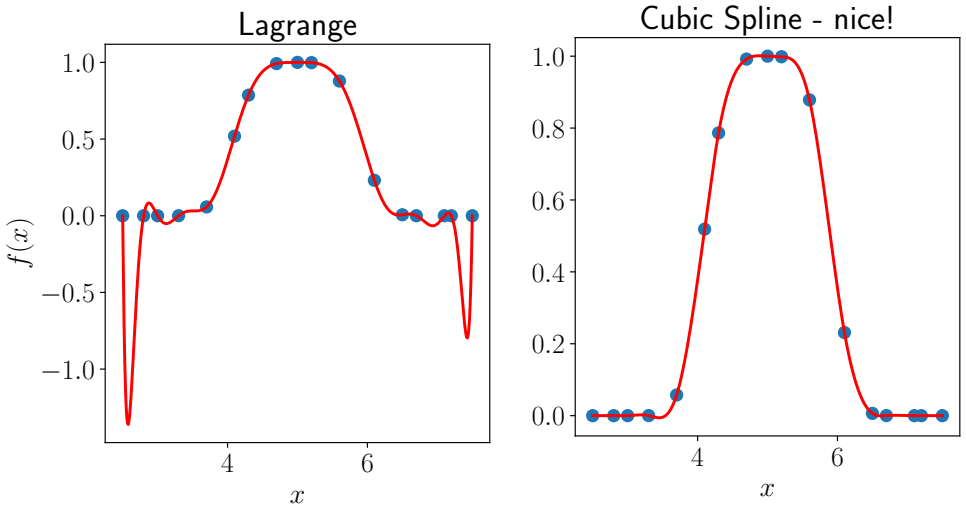

Figure 2: Lagrange and Cubic Spline interpolation using evenly spaces 15 data points, using Lagrange and Cubic Spline. Note how well cubic spline works, even though the data is not in Chebyshev form, which is more typical of experimental data. The red line is using 500 computed points (x_i)

Question 3

Use trigonometric interpolation of the following periodic function:

$$f(x) = e^{\sin(2x)}, \tag{2}$$

with $n = 11$ and $n = 51$ (points to interpolate to), taken from an equidistant grid of points with x between $[0, 2\pi]$. Show a graph of the interpolated line of points (say 100 or 500 x points), in the same range, versus the data used for the interpolation, and plot these graphs side by side. You can see an example solution from the lecture notes.

Bonus marks [1]

Full disclosure, I generated the Q2 "data" myself. Bonus mark of 1 if you can tell me the precise mathematical function? Print the function to the screen. As a hint, it is a "super-Gaussian."