Homework Set 1

ASTR 5900, Computational Physics & Astrophysics

Due 2026 February 11, 11:59pm

Carry out the following work and write up an informal report showing the outputs (e.g., plots/graphs, tables) and answers to questions. Prepare your report in LaTeX (e.g., use Overleaf), generate a pdf file, and ***upload your report as a single pdf file*** on Canvas. Your report should ***include links to your project code*** file(s) in your GitHub repository. Code should have adequate comments, be human-readable, and have at least 3 git-commits (more is better).

Watch the 3-part video series on interpolation, linear and cubic spline:
https://www.youtube.com/watch?v=RpxoN9-i7Jc
https://www.youtube.com/watch?v=LaolbjAzZvg
https://www.youtube.com/watch?v=4VpE9Tbie14

1. Complete the derivations skipped in the videos:

   (a) [0.5 points] for a_i in the linear interpolation in the first video (very simple);

   (b) [0.5 pts] for equations labeled "3" and "4" for the coefficients of the cubic spline in the second video (more algebra);

   (c) [0.5 pts] for the final equation all in terms of just the b coefficients, shown in the first part of the third video (more algebra; you don't need to put into matrix form).

2. Write your own code for a function to do a linear interpolation from an input set of datapoints with (x, y) coordinates, to return the value of y at any requested value of x that is within the domain of the input dataset. *Extrapolation* is when you try to predict values of y at values of x that fall *outside* the domain of your dataset. Consider whether or when you would think that's ok to do. Adopt some assumption and write your code to deal with extrapolation (e.g., you could forbid users from doing this; you could assume values continue with a constant slope from the last points; you could assume a constant value from the last points; or you could make other assumptions).

   (a) [2 pts] Consider the dataset in the file `HW01_data.txt` (provided separately). You may input these values in your code by hand or write a routine to read in this file. Use your code to interpolate the data to at least 10 times higher resolution (i.e., 10 times more points than the dataset, within the same domain). Plot the dataset, and overplot your own linear-interpolation points. Make a distinction between the original points and your interpolated points.

   (b) [1 pts] Use an off-the-shelf code for doing a cubic spline interpolation of the data (e.g., using a library in Python or MATLAB). Plot the cubic spline interpolation on top of the dataset, again making a distinction between the dataset and the interpolation (here you *can* draw the spline as a continuous line, if you want).

(c) [0.5 pts] Notice that the spline can predict values that could be sometimes higher or lower than the surrounding points. I.e., it can create local minima/maxima in between the given datapoints. Is this OK? Give an example of a physical situation (and describe a hypothetical dataset) where this is probably the correct behavior, and one where it is probably incorrect.

(d) [0.5 pts] Describe some of the advantages and disadvantages of the linear and cubic spline interpolations (both in terms of their performance and complexity of calculation involved).

3. Consider the following function

$$y = \sin\left(\frac{\pi}{2}x\right) + \frac{x}{2}.$$

Create a dataset that samples this function at every integer from 0 to 10.

(a) [1 pts] Plot your dataset, and overplot the data interpolated to 10 times higher resolution, using both your linear code and an off-the-shelf cubic spline.

(b) [1.5 pts] Plot the relative error between your interpolated values (both linear and cubic spline on the same plot) and the true function, in the domain $x = [0, 10]$. The relative error is the difference between interpolated values and the true value from the function, divided by the true value. Comment on what the plot shows.

4. [2 pts] Imagine you have precise measurements of the position of a ***simple harmonic oscillator*** at discrete points in time. If you use a linear interpolation of those data in between the measured points, do you expect the interpolated values to exhibit conservation of mechanical (potential and kinetic) energy? Why or why not? Would the answer change, if you used a cubic spline?
(*Optional if extra time*: code up such a scenario and demonstrate your answer with numerical results.)

<u>Optional Exercises</u> (or possible ideas for group projects later in the semester):

Learn and implement a different type of interpolation. Describe the advantages and disadvantages compared to the linear or cubic spline interpolations.

Find a dataset that is relevant to your own research or interests (or anything else), and perform a task/project that requires interpolation of this dataset.

So far, we've only considered interpolation in one dimension. A common simple method for 2D interpolation is called bilinear interpolation (essentially just doing 1D linear interpolation in one direction followed by the other). Learn this method, or another of your choosing, and code up a function that performs this. Images are a common form of 2D data. Can you use your code to interpolate between pixels in an image? Or design your own task/project requiring 2D interpolation.