TAM 470 / CSE 450

Homework 1

Instructions: For problems that you are instructed to complete **by hand**, show all work and ensure that hand-written work is neat and easy to read; if you use computer code to make any calculations, submit a screenshot of your code. Problems hosted on **PrairieLearn** do not require any written work to be submitted unless the problem instructions indicate otherwise.

Problem 1 (10 points)

This problem is to be done by hand.

Consider the Lagrange polynomial of degree two, $p_2(x)$, that passes through (x_j, f_j) for uniformly spaced nodes on $[0, 1]$ and $f(x) = \sin(\pi x)$.

(a) (6 pts) Evaluate $p_2(\frac{1}{4})$.

(b) (4 pts) Compute the derivative of $p_2(x)$ at $x = \frac{1}{4}$.

Problem 2 (10 points):

Consider the following set of data points (x_i, y_i):

x	y
0	0
1	3
2	7
3	-1
4	0

(a) (6 pts) Set up the system of equations required to compute the spline parameters $g''(x_i)$ using natural spline (aka free run-out) end conditions. Then, solve the system of equations to obtain the parameters $g''(x_i)$. You can use Python or another tool to solve the linear system.

(b) (2 pts) Use your results from part (a) (i.e. **not** a `numpy` or `scipy` library) to compute the spline-interpolated value of the data at the coordinate $x = 2.4$. *Hint*: use Moin equation 1.6.

(c) (2 pts) Check your answer to part (b) by using `scipy.interpolate.CubicSpline` to compute the interpolated function value at $x = 2.4$. Submit a screenshot of your code and its output. *Hint*: be sure to read the `scipy` documentation (see hyperlink) to implement the correct spline end conditions.

Problem 3 (10 points):

For this problem:

- For Lagrangian interpolation, use `scipy.interpolate.lagrange`.
- For cubic spline interpolation, use `scipy.interpolate.CubicSpline` with the specified end conditions.

Using the same data from Problem 2, use Python to create a single plot that includes the discrete data points (x_i, y_i) as well as the following interpolating curves on the domain $x \in [0, 4]$:

(**a**) (2.5 pts) The interpolating Lagrange polynomial.

(**b**) (2.5 pts) Cubic spline interpolation using “not-a-knot” end conditions (the `scipy` default).

(**c**) (2.5 pts) Cubic spline interpolation using natural spline end conditions.

(**d**) (2.5 pts) Cubic spline interpolation using clamped end conditions $g'(0) = 1$ and $g'(4) = 0$.

All interpolating curves should be **smooth**. Be sure to add a figure title, axes labels, and a legend that identifies each curve on your plot. For this problem, submit the figure that you create as well as a screenshot of the code used to generate the figure.

Problem 4 (10 points): 4 credit-hour students only

Repeat **Problem 2** (all parts), but using the clamped end conditions $g'(0) = 1$ and $g'(4) = 0$.

Hint: You will have to derive the equations associated with the clamped boundary conditions in order to obtain equations that contain $g''(x_i)$ terms. Start from Moin equation 1.6.