# Lecture E. Homework 5

Covered Contents: Polynomial Interpolation, Numerical Differentiation, (Lec 11 - 15)
Deadline: 11/5/2021, 23:59 PST
Total points: Pen-and-Paper (20 + 20 + 20 + 10 = 70) + Coding (30) = 100.
Submit "hw5.zip"

## Pen and Paper

**E.1.** Let $f(x) = \sin(2\pi x)$ and let $[a, b] = [0, 1]$.
(a) Construct a piecewise-linear polynomial that interpolates $f$ at $\{x_0, x_1, x_2\} = \{0, \frac{1}{2}, 1\}$. Let's call this object $P_{1,2}$.
(b) Construct a piecewise linear polynomial that interpoltes $f$ at $\{x_0, x_1, x_2, x_3, x_4\} = \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$. Let's call this object $P_{1,4}$.
(c) Draw a graph (by hand, or if you'd like, with MATLAB) for $x \in [0, 1]$ of (i) $f(x)$, (ii) the answer to part (a), (iii) the answer to part (b), and (iv) a piecewise-linear polynomial that interpolates $f$ at $x = \{0, \frac{1}{8}, \frac{1}{4}, \frac{3}{8}, \frac{1}{2}, \frac{5}{8}, \frac{3}{4}, \frac{7}{8}, 1\}$ (no need to derive a formula). Let's call this last object $P_{1,8}$.

For this example, will the pointwise error $|f(x) - P_{1,n}| \to 0$ as $n \to \infty$?

**E.2.** Construct the natural cubic spline $s(x)$ for the data $(0.1, -0.29004996)$, $(0.2, -0.56079734)$, $(0.3, -0.81401972)$. These values correspond to the function $f(x) = x^2 \cos(x) - 3x$.
(a) Approximate $f(0.18)$ and $f'(0.18)$ using $s(x)$ and $s'(x)$, respectively, and list the relative errors.
(b) Approximate $f'(0.2)$ using $s'(x)$. Do the values $f'(0.2)$ and $s'(0.2)$ agree? Based on the definition of a cubic spline, should they agree?

**E.3.** Let $f \in C^4([a, b])$.
(a) Use Taylor's theorem to write $f$ as a third order (cubic) Taylor polynomial plus a fourth order (quartic) remainder term. Expand about the point $x_0$.
(b) Use the result from (a) to evaluate $f(x)$ at the points $x = x_0 + h$ and $x = x_0 - h$. Add the two results together to derive the centered difference approximation to the second derivative:

$$\frac{f(x_0 + h) - 2f(x_0) + f(x_0 - h)}{h^2} = f''(x_0) + \frac{h^2}{4!}(f^{(4)}(\xi_1) + f^{(4)}(\xi_2))$$

**E.4.** Let $f(x) = 3xe^x - \cos(x)$. Use the data below and your answer to exercise problem 3 to approximate $f''(1.3)$ with $h = 0.1$ and $h = 0.01$. Compare your results to the true $f''(1.3)$.

Data: (1.20,11.59006), (1.29,13.78176), (1.30,14.04276), (1.31,14.30741), (1.40,16.86187)

# Coding

**What to submit:** (30 points).

An important skill of a numerical analyst is to be able to utilize functions and routines previously written by others by carefully reading that codes documentation and giving these functions/routines the proper inputs. This is especially true for large scale problems and/or projects, in which writing your own code from scratch would be too time-consuming. In this exercise you will practice this skill by calling three separate MATLAB routines.

In the script 'runge_phenom.m' you will find code that plots the function $f(x) = 1/(1 + 25x^2)$ from $x = -1$ to $x = 1$. From class we know that using a polynomial interpolant with equispaced nodes to approximate this function can produce wild oscillations and hence a poor approximation. Your job is to use the functions 'lagrange.m', 'chebyshev_coefficients.m' and 'chebyshev_interpolant.m', as well as the function 'spline' (built in to MATLAB) to produce interpolants for $f$ on $[-1, 1]$.

- 'lagrange.m' produces the Lagrange polynomial for $f(x)$ at the specified interpolation nodes
- 'chebyshev_coefficients.m' and 'chebyshev_interpolant.m' will produce the so-called Chebyshev polynomial that interpolates $f(x)$ at the nodes $x_j = \cos(\theta_j)$, where $\theta_j = \pi j/(N-1)$ and $j = 0, \ldots, N-1$
- 'spline' produces a piecewise-defined cubic spline for $f(x)$ at the specified interpolation nodes

To receive full credit for this problem, you must give these 4 functions the correct inputs and store the results in the vectors 'ylagrange', 'ycheb' and 'yspline', and then plot each interpolant against the original $f(x)$. The plotting code is already written for you, but you must uncomment the 'plot' commands. Save the plot to a file (a 'jpg' or 'png') for the cases N=7, N=11, and N=20. Here N is the number of interpolation points. For each N, which interpolant gives the best results? (It may be helpful to only plot 1 or 2 interpolants at a time by commenting/uncommenting different plot commands)

Some notes:

- The scripts for Lagrange and Chebyshev interpolation must be in the same directory as 'runge_phenom.m'
- To understand how to properly utilize each function, it is helpful to type 'help [function name]' into the Command Window, for example:

  ```
  help lagrange
  ```

  will describe what the function does, i.e., what it outputs, and importantly, what inputs are needed. Since the 'spline' function is native to MATLAB, documentation can be found easily online as well.