Project 5 due April 10 (or before). Computer lab parts must be demonstrated for me (make an appointment).

Project 5 involves numerical methods for piecewise polynomial interpolation and parametric equations.

1. A clamped cubic spline S for a function f is defined on $[1, 3]$ by

$$S(x) = \begin{cases} s_0(x) = 3(x-1) + 2(x-1)^2 - (x-1)^3 & 1 \le x < 2 \\ s_1(x) = a + b(x-2) + c(x-2)^2 + d(x-2)^3 & 2 \le x \le 3 \end{cases} \tag{1}$$

Given $f'(1) = f'(3)$, find a, b, c, and d.

2. A piecewise quadratic interpolant is defined in terms of the given data, $f(x_0)$, $f(x_1)$, …, $f(x_{2n})$, as follows:

$$Q(x) = \begin{cases} q_0(x) = a_0 + b_0(x-x_0) + c_0(x-x_0)^2 & x_0 \le x < x_2 \\ q_2(x) = a_2 + b_2(x-x_2) + c_2(x-x_2)^2 & x_2 \le x < x_4 \\ \cdots & \cdots \\ q_{2n-2}(x) = a_{2n-2} + b_{2n-2}(x-x_{2n-2}) + c_{2n-2}(x-x_{2n-2})^2 & x_{2n-2} \le x < x_{2n} \end{cases} \tag{2}$$

Given that

$$\begin{aligned} q_{2i}(x_{2i}) &= f(x_{2i}), \\ q_{2i}(x_{2i+1}) &= f(x_{2i+1}), \\ q_{2i}(x_{2i+2}) &= f(x_{2i+2}), \\ \text{for } i &= 0, \ldots, n-1, \end{aligned}$$

express the coefficients,

$$a_{2i}, b_{2i}, c_{2i}, \quad i = 0, \ldots, n-1,$$

in terms of the,

$$f(x_0), \ldots, f(x_{2n}).$$

3. Using Q defined in the previous exercise, find a bound on the error,

$$|Q(x) - f(x)|, \quad x_0 \le x \le x_{2n}.$$

Assume the interpolating nodes are evenly spaced with $h = x_{i+1} - x_i$. Your solution should look something like $|Q(x) - f(x)| \le \frac{Mh^r}{R}$ where M, r, and R are constants and M is a bound on the third derivative of f.

4. Find the parameterization for a triangle and a 4 leaf clover. (examples for a square and a 2 leaf clover are in the sample code: `piecewise_sample.cpp`).

5. Implement the piecewise cubic Hermite interpolation algorithm that we derived in class and compare the accuracy to the piecewise linear and piecewise cubic spline algorithms. The sample code, `piecewise_sample.cpp`, already contains the piecewise linear and piecewise cubic spline algorithms. Test your new piecewise cubic Hermite algorithm on the square (already in the code), two leaf clover (already in the code), triangle, and four leaf clover shapes. The rate of convergence with respect to increasing n should be consistent with the theory (derived in class) for smooth shapes.

6. The Crout factorization technique (page 408) can be used in the sample code instead of the currently implemented Gaussian elimination algorithm to make the cubic spline code more efficient. But, the book does not give the algorithm for the periodic version, only clamped or free boundary conditions. Implement a periodic version of the Crout algorithm for extra credit (challenging).