Read BF Chapter 3.3, 3.5

Exercises

Chapter 3.3 # 1*, 3*, 7a, 10, 11, 16, 17

* You should know how to do this sort of problem, but I recommend you do A, B, C below and then skip 1 and 3, since the numbers in 1 and 3 are horrible.

Problem A: Find the Newton form of the fourth degree polynomial interpolating $n!$ at $n = 0, 1, 2, 3, 4$.

Problem B: This example was first computed by James Gregory in 1670. The Newton forward-difference formula is sometimes known as Newton-Gregory interpolation.

Use the forward difference method to interpolate $f(x) = x^3$ at $x = 10, 15, 20, 25, 30$ and use it to compute 23^3.

Problem C: Compute $\ln 1.09$ correct to five decimal places by interpolating natural logarithms for $1, 1.1, 1.2, 1.3$.

Chapter 3.5 # 3a, 13

Problem D: Compute the natural cubic spline through $(0, 0)$, $(1, 10)$, and $(2, 4)$.

Problem E: Explain why the following function is not a spline:

$$f(x) = \begin{cases} -x^2 + 1 & \text{if } 0 \le x \le 1 \\ (x-1)^3 + 2(x-1) & \text{if } 1 \le x \le 2 \\ 2(x-2) + 3 & \text{if } 2 \le x \le 3 \end{cases}$$

MATLAB/Octave

1. In this question, you investigate the interpolating polynomial for $f(x) = \frac{1}{x}$ with nodes at $1, 2, \ldots, n$.

   (a) Find (by hand, with exact fractions) the Newton form of the interpolating polynomial for $f(x) = 1/x$ with nodes at $1, 2, 3, 4$. Repeat with nodes $1, 2, 3, 4, 5$.

   (b) (Optional, challenging)
   Prove the interpolating polynomial for $1/x$ with nodes $1, 2, 3, \ldots, n$ takes the form:

$$\begin{aligned} p_n(x) = 1 - \frac{1}{2!}(x-1) + \frac{1}{3!}(x-1)(x-2) - \frac{1}{4!}(x-1)(x-2)(x-3) + \cdots \\ \cdots \pm \frac{1}{n!}(x-1)(x-2)\cdots(x-(n-1)) \end{aligned}$$

   (c) From part (b), rewrite $p_n(x)$ in the nested form:

$$p_n(x) = 1 - \frac{1}{2}(x-1)\left[1 - \frac{1}{3}(x-2)\left[1 - \frac{1}{4}(x-3)\left[\cdots \frac{1}{n}(x-(n-1))\right]\cdots\right]\right]$$

   Write a Matlab function `harmonicpoly(n,x)` that computes $p_n(x)$ using the nested form.

   (d) The interpolating polynomial $p_n(x)$ converges to $1/x$ as $n \to \infty$. However, it makes a big difference how you compute it:

      i. Use `polyinterp` to graph $p_n(x)$ for $x \in [1, 4]$. Increase n until it starts to go wrong.

      ii. Use `newtoninterp` to graph $p_n(x)$ for $x \in [1, 4]$. Increase n until it starts to go wrong.

      iii. Use `harmonicpoly` to graph $p_n(x)$ for $x \in [1, 4]$. Increase n, it should stay near $1/x$.

   For this part, you can write down the values of n at which things break down, and describe what each graph does. Or, print out pictures of the graphs when they begin to fail.

2. Using Matlab's `spline` function, create a script that draws a (rough) picture of the Billiken.