0 Instructions

This problem set is **due on Tuesday, Oct. 8 at midnight**. Submit your solutions on Canvas. Include the names of everyone you worked with on this problem set. Include any code you used to solve the problems as part of your submission.

1 Practice

Problem 1.1. Explain in your own words how to perform polynomial interpolation with both the Lagrange interpolation approach and the Newton interpolation approach.

Problem 1.2. Use both Lagrange interpolation and Newton interpolation to produce the interpolating polynomial for the four points $(0, 3), (1.2, 6), (3.5, 1), (7, 5)$. Do you expect your answers to be the same?

Problem 1.3. Explain in your own words how to interpret the error term formula for polynomial interpolation. Explain whether you can directly apply this error bound to the result of the previous problem to produce a numerical upper bound.

Problem 1.4. Explain the behavior of the polynomial term

$$(x - x_0)(x - x_1) \dots (x - x_n)$$

appearing in the interpolation error formula and how Chebyshev points help to mitigate this source of error.

2 Interpolation

In this problem, we will work with the function

$$f(x) = \frac{1}{1 + x^6}.$$

Recall from class the divided difference operators, defined inductively as follows. First, $f[x_i] = f(x_i)$. Then more generally, we define

$$f[x_i, \dots, x_{i+k}] = \frac{f[x_{i+1}, \dots, x_{i+k}] - f[x_i, \dots, x_{i+k-1}]}{x_{i+k} - x_i}.$$

As discussed in class, the divided differences are exactly the coefficients of the Newton interpolation at the points $x_0, \dots, x_n$

$$P_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + \dots + f[x_0, \dots, x_n](x - x_0) \dots (x - x_{n-1}).$$

Problem 2.1. Implement Newton interpolation using divided differences. Your function should take as input the function to interpolate along with the desired interpolation points and output in any reasonable format a representation of the interpolating polynomial. (You are free to choose how you represent them, but one standard way would be as a list of coefficients. For example, $[a, b, c]$ could correspond to the polynomial $ax^2 + bx + c$.)

Problem 2.2. For $n = 3, 5, 9, 17$, use your code to approximate $f(x)$ on the interval $[-2, 2]$ using

1. polynomial interpolation with $n + 1$ equally spaced points.
2. polynomial interpolation with $n + 1$ Chebyshev points.

Graph your approximation against the regular function on the interval $[-2, 2]$ and describe the effect of increasing the degree on the quality of the approximation. Using the error bound we found in class for interpolation, explain why you think this is the case.

3 Numerical Differentiation

Recall from the first homework that using the limit definition of the derivative directly, the best floating point approximation we are able to deliver has a minimum error of around $\sqrt{\varepsilon}$ due to floating point error. An alternative approach is to approximate the function first using polynomial interpolation and then to take a derivative of the polynomial.

In theory, using higher degree approximations to f should allow us to obtain better approximations for $f(x)$. In practice, using degrees that are too high quickly accumulates floating point rounding errors and costs time for function evaluation, so it's often better to stick to relatively low degree approximations (e.g. $n + 1 = 3$ and $n + 1 = 5$ are the most common).

Problem 3.1. Write down the polynomial interpolating $f(x)$ at the two points x_0 and $x_1 = x_0 + h$ in terms of f and h and give the error of the interpolation.

Problem 3.2. Take a derivative of the polynomial you found above to obtain an approximation for the derivative $f'(x_0)$. What error term do you obtain for this approximation, and how does this approximation compare to the limit approximation of the derivative?

While interpolation gives a meaningful approximation of the derivative, it is difficult to obtain an error bound from this method due to the presence of the function $\xi(x)$ appearing in the error term of polynomial interpolation. In particular, we cannot meaningfully evaluate the derivative $\xi'(x)$, so we will be unable to bound the error of the derivative approximation at points other than x_0 or x_1. Luckily, at these points, you should see from your formula that this term is cancelled out.

Problem 3.3. Estimate the approximation error for the derivative at an interpolation point without using $\xi'(x)$. More precisely, suppose P interpolates $f(x)$ at the points $x_0, \ldots, x_n$. Show that the error $|P'(x_j) - f'(x_j)|$ at an interpolation point can be bounded in terms of $f^{(n+1)}$ without needing the derivative $\xi'(x)$ (although you may still need $\xi(x)$).

If the interpolation polynomial is represented in Lagrange form, then we'll have

$$f(x) \approx \sum_k f(x_k) L_{n,k}(x)$$

and taking m derivatives gives the approximation

$$f^{(m)}(x) \approx \sum_k f(x_k) L_{n,k}^{(m)}(x).$$

In order to evaluate this approximation at a point x, we need to evaluate the m^{th} derivatives of the Lagrange interpolating polynomials. One way to approach this problem is by proving the identities

$$L_{n,k}^{(m)}(x) = \frac{m}{x_k - x_n} L_{n-1,k}^{(m-1)}(x) + \frac{x - x_n}{x_k - x_n} L_{n-1,k}^{(m)}(x)$$

for all n, k, m that make sense and then using these values in the approximation above above.

Problem 3.4. Verify that the Lagrange polynomials satisfy the recurrence relation

$$L_{n,k}^{(m)}(x) = \frac{m}{x_k - x_n} L_{n-1,k}^{(m-1)}(x) + \frac{x - x_n}{x_k - x_n} L_{n-1,k}^{(m)}(x)$$

where x_i is the i^{th} interpolation point.