Last time

- Lagrange interpolation

For data \( (x_j, y_j), \quad j = 0, \ldots, n \)

there is a unique polynomial interpolant of degree \( n \) such that:

\[ p_n(x_j) = y_j \]

An explicit construction of \( p_n \) is the Lagrange form:

\[ p_n(x) = \sum_{k=0}^{n} L_k(x) y_k \]

with

\[ L_k(x) = \frac{\prod_{j=0}^{n} (x - x_j)}{\prod_{j \neq k} (x_k - x_j)} \]

\( L_k \) has the property that

\[ L_k(x_k) = 1 \]

\[ L_k(x_j) = 0 \quad \text{if} \quad j \neq k. \]
Polynomial interpolation is often used for function approximation.

\[ J_0(x) \] (Bessel function)

Error in polynomial interp is given by:

**Thm.** Let \( f \in C^{n+1}[a,b] \). For \( x \in [a,b] \), there exists \( \xi = \xi(x) \in (a,b) \) such that

\[
 f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{j=0}^{n} (x-x_j)
\]

This is a property of polynomials, not the particular Lagrange form (since the polynomial is unique).
To answer our 3rd question last time:

3) In floating point arithmetic, is the evaluation of $p_n$ stable?


Short story: The basic Lagrange form can be unstable (overflow, underflow, round-off error, etc.)
(slop analysis, tedious, ...)

The Barycentric Forms of Interpolation

The numerical stability & evaluating an interpolating polynomial can be fixed by rearranging its terms.

As motivation, examine barycentric coordinates on a triangle:

Ex: The barycentric coordinates of a point $P$ inside a triangle with vertices $A$, $B$, $C$ is given by:
\[ (x, \beta, \gamma) \leftarrow \text{Barycentric coordinates} \]

with \[ P = xA + \beta B + \gamma C \]

and \[ x + \beta + \gamma = 1. \]

I.E., "center" of triangle is given by \[ \begin{bmatrix} x \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \left( \text{center of mass} \right). \]

\[ \text{Idea: Replace } A, B, C \text{ with functions that sum to 1.} \]

\[ \text{Start with Lagrange form and rewrite} \]

\[ P_n(x) = \sum_{k=0}^{n} \left( \frac{1}{n} \left( x-x_j \right) \right) y_k \]

\[ = \sum_{k=0}^{n} \left( \frac{1}{n} \left( x-x_j \right) \right) \frac{1}{x-x_k} \left( \frac{1}{n} \sum_{j \neq k} \frac{1}{x_k-x_j} \right) y_k \]

\[ = \left( p(x) \right) \sum_{k=0}^{n} \frac{w_k}{x-x_k} y_k \]

\[ \text{Modified Lagrange} \]

\[ \text{First Barycentric Formula} \]

\[ \text{Stability analyzed in 2004 = backward stable. (Higham, 2004)} \]

when \[ q_{x_k} = \frac{1}{n} \sum_{j=0}^{n} \left( x-x_j \right), \quad w_k = \frac{1}{n} \sum_{j \neq k} \frac{1}{x_k-x_j} \]
This modified Lagrange form can be further simplified by "dividing by 1":

The polynomial interpolant of the function 1 at the same points \((x_j, y_j)\) is simply:

\[ 1 = q(x) \sum_{k=0}^{n} \frac{W_k}{x-x_k} \quad (\text{since } y_k = 1). \]

Then computing

\[ p_n(x) = q(x) \sum_{k=0}^{n} \frac{W_k y_k}{x-x_k} = q(x) \sum_{k=0}^{n} \frac{W_k y_k}{x-x_k} - \sum_{k=0}^{n} \frac{W_k y_k}{x-x_k} = q(x) \sum_{k=0}^{n} \frac{W_k y_k}{x-x_k} \]

This form is "forward stable for any reasonable choice of \(x_j\)." Basically, the errors in the top cancel out the errors in the bottom (i.e., exactly when \(x \approx x_k\)). (Also see Higham 2004).

This is the form that should almost always be used.
Cost?

Computing $W_k$: $O(n)$ only fair since solving for actual coefficients is $O(n^3)$.

Updating $W_k$'s with new data: $O(n)$

Evaluating $p_n(x)$: $O(n)$.

Convergence of Polynomial Interpolation

Let's examine the question of what happens as $n \to \infty$, i.e.,
\[
\lim_{n \to \infty} \max_{x} |f(x) - p_n(x)| = ? \quad \text{(error, supremum).}
\]

The pointwise error is
\[
\max_{x} |f^{(n+1)}(s)| \cdot \max_{x} \prod_{j=0}^{n} |x-x_j|
\]

It's not obvious if this decreases or increases...

(Show Matlab demo).

For equispaced and randomly distributed points, the phenomenon you are witnessing is known as the Runge effect ($\frac{1}{1+x^2}$ is the Runge function).