3.1 Path & Interpolating Functions

In general, given a data set \( (x_1, y_1), \ldots, (x_n, y_n) \), we want to find a function \( f(x) \) such that \( f(x_i) = y_i \) for all \( i \in \mathbb{N} \).

![Diagram of interpolating function](image)

**Definition** The function \( y = P(x) \) interpolates the data \( (x_1, y_1), \ldots, (x_n, y_n) \) if \( P(x_i) = y_i \) for all \( i \in \mathbb{N} \).

Note that \( P \) is required to be a function (i.e., each input gets a single output). In general, because of its various desirable mathematical properties (i.e., low "complexity"), we generally aim to find \( P(x) \) where \( P(x) \) is a polynomial function.

Polynomial functions are the most fundamental functions for digital computer interpolation.

We can think of function as the reverse/inverse of function evaluation. In addition, function interpolation (using polynomials) is an instance of data compression.
**Def.** Given \( n \) data points \( \{(x_i, y_i)\} \), the Lagrange interpolation polynomial of degree \( \leq n-1 \) is defined:

\[
P_n(x) = y_1 L_1(x) + \cdots + y_n L_n(x)
\]

where:

\[
L_k(x) = \frac{(x-x_1)(x-x_2)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_n)}{(x_k-x_1)(x_k-x_2)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_n)}
\]

\( \forall 1 \leq k \leq n. \)

**Ex.** Given 3 data points: \( (x_1, y_1), (x_2, y_2), (x_3, y_3) \) we have:

\[
P_2(x) = y_1 \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)} + y_2 \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)} + y_3 \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}
\]

Claim: \( p_2(x) \) interpolates the given data.

Check \( p_2(x_1) = y_1 \cdot 1 + 0 + 0 = y_1 \), \( p_2(x_2) = 0 + y_2 \cdot 1 + 0 = y_2 \).

In general then, \( p_{n-1}(x) = y_1 L_1(x) + \cdots + y_n L_n(x) \)

If \( L_k(x_k) = 1 \), else \( L_k(x_j) = 0 \) for \( k \neq j \).

So the Lagrange polynomial of degree \( n-1 \) interpolates \( n \) given data points, in general.
Q: How many interpolating polynomials of degree n-1 exist for a data set of n points (no repeated x_i's)?

A: We seek to find coefficients 

\[ P(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1x + a_0 \]

so that \( P(x) \) interpolates the data.

This problem gives rise to \( n \) equations w/ \( n \) unknowns.

which means that any such solution is unique. Thus for \( n \) points, an interpolating polynomial of degree \( n-1 \) is unique!

Hence, the Lagrange interpolating polynomial is the unique \( n-1 \) degree polynomial that interpolates \( n \) data points.

Theorem: Fundamental Theorem of Polynomial Interpolation

Let \((x_1, y_1), \ldots, (x_n, y_n)\) be \( n \) points in the plane with distinct \( x_i \).

Then there exists a \textbf{unique} polynomial \( P \) of degree \( n-1 \) (or less) that satisfies \( P(x_i) = y_i \) at \( n \) points.

\textbf{Proof:} (Existence is proved by the explicit Lagrange interpolation formula given above).

\textbf{Uniqueness Proof:} (By contradiction) Suppose not, if \( g(x) \) and \( q(x) \) have degree \( n-1 \) and both interpolate the data.
5. \( P(x_i) = Q(x_i) = y_i \), \( P(x_2) = Q(x_2) = y_2 \), etc.

Define a new polynomial, \( H(x) = P(x) - Q(x) \).

Then \( \deg(H(x)) \leq n-1 \), clearly (why?). Also: \( H(x_i) = P(x_i) - Q(x_i) = 0 \),

and in general, \( H(x_i) = P(x_i) - Q(x_i) = 0 \quad \forall \quad i \in \{1, 2, \ldots, n\} \).

This shows that \( H \) has \( n \) distinct zeros. But, according to the FTA (Fundamental Theorem of Algebra), a non-constant polynomial of degree \( n-1 \) can have at most \( n-1 \) distinct zeros, unless the function is identically zero. Consequently, \( H(x) \equiv 0 \), which shows \( P(x) \equiv Q(x) \), proving uniqueness.

**Example:** Find the interpolating polynomial of degree 3 or less for \( (0,2), (1,1), (2,0) \), and \( (3,-1) \).

\[
P(x) = 2 \frac{(x-1)(x-2)(x-3)}{(0-1)(0-2)(0-3)} + 1 \frac{(x-0)(x-2)(x-3)}{(1-0)(1-2)(1-3)} + 0 \frac{(x-0)(x-1)(x-3)}{(2-0)(2-1)(2-3)} - 1 \frac{(x-0)(x-1)(x-2)}{(3-0)(3-1)(3-2)}
\]

\[
= -\frac{1}{3}(x^3 - 6x^2 + 11x - 6) + \frac{1}{2}(x^3 - 5x^2 + 6x) - \frac{1}{6}(x^3 - 3x^2 + 2x)
\]

\[
= -x + 2
\]

Our theorem says the interpolating polynomial of degree 3 or less is unique, so our result indicates the data set is co-linear.
Newton's Divided Difference

Lagrange interpolation, despite its seeming utility, is rarely used in practice. Instead, as we shall see, we prefer to use a recursive method known as Newton's Divided Difference.

**Def.** Denote by \( f[x_0, x_n] \) the coefficient of \( x^{x-1} \) term in the unique polynomial that interpolates \( (x_0, f(x_0)), \ldots, (x_n, f(x_n)) \).

**Ex.** (one more Lagrange)

Interpolate: \((0, 1)\), \((2, 2)\) \& \((3, 4)\)

\[
P_2(x) = 1 \cdot \frac{(x-2)(x-3)}{(0-2)(0-3)} + 2 \cdot \frac{(x-0)(x-3)}{(2-0)(2-3)} + 4 \cdot \frac{(x-0)(x-2)}{(3-0)(3-2)}
\]

\[
= \frac{1}{2}x^2 - \frac{1}{2}x + 1
\]

Using our new notation, then: \[ f[0 \ 2 \ 3] = \frac{1}{2}. \]

We now give a remarkable alternative interpolation formula:

\[
P(x) = f[x_0] + f[x_1, x_2] (x-x_0) + f[x_1, x_2, x_3] (x-x_1)(x-x_2)
\]

\[
+ f[x_1, x_2, x_3, x_4] (x-x_1)(x-x_2)(x-x_3)
\]

\[
\ldots + f[x_1, x_2 \ldots x_n] (x-x_1) \ldots (x-x_{n-1})
\]

Newton's Divided Difference.
Moreover, the coefficients $f[x_1, \ldots, x_n]$ can be calculated recursively.

First, list the data points in a Table:

<table>
<thead>
<tr>
<th>$x_i$</th>
<th>$f(x_i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>$f(x_1)$</td>
</tr>
<tr>
<td>$x_2$</td>
<td>$f(x_2)$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
<tr>
<td>$x_n$</td>
<td>$f(x_n)$</td>
</tr>
</tbody>
</table>

Now define the divided differences, which are real numbers:

\[
\begin{align*}
 f[x_1] &= f(x_1) \\
 f[x_1, x_2] &= \frac{f[x_1, x_2] - f[x_1]}{x_2 - x_1} \\
 \vdots \\
 f[x_1, x_2, \ldots, x_n] &= \frac{f[x_1, x_2, \ldots, x_n] - f[x_1, x_2, \ldots, x_{n-1}]}{x_n - x_{n-1}} \\
 f[x_1, x_2, \ldots, x_n, x_{n+1}] &= \frac{f[x_1, x_2, \ldots, x_n, x_{n+1}] - f[x_1, x_2, \ldots, x_n, x_{n+2}]}{x_{n+3} - x_n} \\
 \end{align*}
\]

In general, we write this recursively as a Table:

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$f[x_1]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_2$</td>
<td>$f[x_1, x_2]$</td>
</tr>
<tr>
<td>$x_3$</td>
<td>$f[x_1, x_2, x_3]$</td>
</tr>
<tr>
<td>$\vdots$</td>
<td>$\vdots$</td>
</tr>
</tbody>
</table>

The interpolating polynomial is:

\[
P(x) = \sum_{i=1}^{n} f[x_1, \ldots, x_i] (x-x_1) \cdots (x-x_{i-1})
\]

**FACT:** The unique interpolating polynomial is given by the \textit{divided difference} form.

The unique polynomial is given by \textit{efficiently} expanded polynomial.
Integrate \( y \) D.D. : \((0,1), (2,2), (3,4)\)

\[
\begin{array}{c|c}
0 & 1 \\
2 & 2 \\
3 & 4 \\
\end{array}
\]

\[
\begin{array}{c}
= \frac{1}{2}x^2 - \frac{1}{2}x + 1
\end{array}
\]

\[
\begin{array}{c}
\Rightarrow y(x) = 1 + \frac{1}{2}(x-0) + \frac{1}{2}(y-0)(x-2)
\end{array}
\]

Computation:

\[
f[x_1] = 1
\]

\[
f[x_2] = 2
\]

\[
f[x_3] = 4
\]

\[
f[x_2, x_3] = \frac{f[x_2] - f[x_1]}{x_2 - x_1} = \frac{2-1}{2-0} = \frac{1}{2}
\]

\[
f[x_2, x_3] = \frac{f[x_3] - f[x_2]}{x_3 - x_2} = \frac{4-2}{3-2} = 2
\]

\[
f[x_1, x_2, x_3] = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1} = \frac{2 - \frac{1}{2}}{3-0} = \frac{1}{2}
\]

(4) A key advantage of using Newton's divided difference form stems from the fact that if we add a point to our data set, only a small number of updated computations are required (because of recursion).

(6) On the other hand, if we computed \( P_n(x) \) for \( n \) data points, and then add a new datum to our set (now \( n = 4 \)), computing \( P_n(x) \) for this data would require \( n^2 \) to start. This process over from scratch (very inefficient).
Example: Add a fourth point: (1, 0) to the given data set and compute $P_3(x)$, using Newton’s Divided Diff.

\[
P_3(x) = 1 + \frac{1}{2} (x-0) + \frac{1}{2} (x-0)(x-2) + \frac{1}{2} (x-0)(x-2)(x-3)
\]

\[
= P_2(x) + (\frac{1}{2}) (x-0)(x-2)(x-3)
\]

- Newton’s D.D. is often used for data that is updated in “real-time.”

Q: How many interpolating polynomials of degree 4 interpolate a data set consisting of 4 points?

A: An infinite number! Why?

By Newton’s D.D.

$P_4(x) = P_3(x) + c_4 (x-x_0) (x-x_1) (x-x_2) (x-x_3) (x-x_4)$

uniquely defined by 5 points

Polynomial interpolation is an example of compression in numerical analysis.

Calculator: What is $\sin(123)$ = ?

Mechanism:

Small data set of points on $y = \sin(x)$

Interpolate data to get $\sin(x) \approx P(x)$

Output: $P(123)$

error (see 5.2)