Polynomial interpolation:

The Polynomial Interpolation Problem: Find a polynomial $p(x) = a_0 + a_1x + ... + a_nx^n$ which satisfies the $n+1$ conditions $p(x_0) = y_0$, $p(x_1) = y_1, ..., p(x_n) = y_n$.

It is easy to see that the "unknown" coefficients $a_0, .., a_n$ are to satisfy a system of $n+1$ equations in $n+1$ unknowns, which we write in shorthand as $p(x_i) = y_i \;,\; i = 0, ..., n$

I. The "homogeneous" system $p(x_i) = 0 \;,\; i = 0, ..., n$ has $p(x) \equiv 0$, i.e. $a_0 = a_1 = ... = a_n = 0$, as its only solution. This is obvious from algebra since a polynomial of degree n which is zero at the $n+1$ points $x_0, .., x_n$ must be the zero polynomial. This implies, from the theory of linear equations, that the general system $p(x_i) = y_i \;,\; i = 0, ..., n$ has a unique solution for any given choice of $y_0, .., y_n$. Thus the solution of the polynomial interpolation problem exists, and is unique.

II. Direct construction of an interpolating polynomial: The **Lagrange Form** , the Lagrange fundamental polynomials.

Example: Consider this mysterious looking polynomial

$$p(x) =$$

$$y_0\frac{(x-1)(x-2)(x-3)}{(0-1)(0-2)(0-3)} + y_1\frac{(x-0)(x-2)(x-3)}{(1-0)(1-2)(1-3)} + y_2\frac{(x-0)(x-1)(x-3)}{(2-0)(2-1)(2-3)} + y_3\frac{(x-0)(x-1)(x-2)}{(3-0)(3-1)(3-2)}$$

We can easily see that $p(0) = y_0, p(1) = y_1, p(2) = y_2, p(3) = y_3$. Generalizing this construction to a general set of interpolation points $x_0,..,x_n$ we obtain Lagrange form of the interpolating polynomial:

$$p(x) = y_0L_0(x) + y_1L_1(x) + ... + y_nL_n(x)$$

where $L_i(x)$ is the **Lagrange fundamental polynomial** satisfying

$L_i(x) = \{1 \text{ if } x = x_i \text{ ; } 0 \text{ if } x = x_j \text{ for } j \neq i\}$ and is given explicitly by

$$L_i(x) = \frac{(x-x_0)\bullet\bullet(x-x_{i-1})\ (x-x_{i+1})\bullet\bullet(x-x_n)}{(x_i-x_0)\bullet\bullet(x_i-x_{i-1})(x_i-x_{i+1})\bullet\bullet(x_i-x_n)}$$

We have more efficient ways of computing than using the polynomial in this form, but there is a very important theoretical & practical importance to $L_i(x)$; it represents the sensitivity of the value of the interpolating polynomial p at x to changes in the datavalue y_i , so that if y_i changes by Δy_i then $p(x)$ changes by $(\Delta y_i)L_i(x)$.

Newton form of the interpolating polynomial:

Here is another way of constructing the interpolating polynomial that leads to the so-called Newton form the and theory of divided differences. It's the best algorithm for constructing and evaluating the interpolating polynomial.

For notation purposes, it helps to think of the y-values as coming from some function $f(x)$, so that we are interpolating the values $f(x_i)$ at the points $\{x_i\}$, i.e. satisfying the equations $p(x_i) = f(x_i), i = 0,..,n$. Note the following:

$p_0(x) = f(x_0)$ interpolates the single value $f(x_0)$ with a polynomial of degree zero

$p_1(x) = f(x_0) + a_1(x - x_0)$ interpolates at $x = x_0$ and with the right choice of

$a_1 = \dfrac{f(x_1) - f(x_0)}{x_1 - x_0}$ also interpolates at $x = x_1$

Now consider

$p_2(x) = p_1(x) + a_2(x - x_0)(x - x_1)$

Because of the form of the last term we added, $p_2(x)$ still interpolates at x_0 and x_1. If we now choose a_2 the right way we can make $p_2(x)$ interpolate at x_2 as well. We could write $a_2 = \dfrac{f(x_2) - p_1(x_2)}{(x_2 - x_0)(x_2 - x_1)}$ but there is a better way to calculate a_2 that we'll see later.

For now, all we need to notice is that we can calculate the unique a_2 for which $p_2(x)$

interpolates at x_0, x_1, x_2 .

In this way, we can construct, in principle, an interpolating polynomial of degree n of the form

$$p_n(x) = a_0 + a_1(x - x_0) + \ldots + a_n(x - x_0) \bullet\bullet(x - x_{n-1}) = p_{n-1}(x) + a_n(x - x_0) \bullet\bullet(x - x_{n-1})$$

whose partial sums $p_k(x) = a_0 + a_1(x - x_0) + \ldots + a_k(x - x_0) \bullet\bullet(x - x_{k-1})$ are interpolating polynomials of lower degree and which solve the interpolation problem at $x_i, i = 0, .., k$. Note that each a_k can be described as the coefficient of x^k in the interpolating polynomial $p_k(x)$ - this characterization does not depend on the form of the polynomial representation or the order in which the interpolation points are used.

This is called the Newton form of the interpolating polynomial. The polynomial, once the coefficients have been determined, can be efficiently evaluated recursively using nested multiplication. The coefficients themselves are important in how they depend on the function values $f(x_i)$ and the x_i themselves. They are called divided differences, and there is a nice theory that applies to them.

Divided differences:

Definition:

Let $[t_0, .., t_k]$ be a vector of (for now, distinct) values. Given a function f , we define the divided difference of f at the values $[t_0, .., t_k]$ to be the coefficient of x^k in the (interpolating) polynomial p of degree k that satisfies $p(t_i) = f(t_i)$, $i = 0, .., k$. We write $f[t_0, .., t_k]$ to denote this divided difference. A divided difference over $k + 1$ points is referred to as a k^{th} order divided difference.

Using the divided difference notation, we can write the Newton form of the interpolating polynomial as

$$p_n(x) = f[x_0]+f[x_0,x_1](x-x_0) + \ldots + f[x_0,\ldots,x_n](x-x_0)\bullet\bullet(x-x_{n-1})$$

We will see that this is a discrete version of the Taylor polynomial P_n, in the sense that we obtain P_n as the polynomial that interpolates the derivatives of f at $x = a$, and we add an n^{th} degree term on to P_{n-1} to obtain P_n. Also the k^{th} order divided difference $f[t_0,..,t_k]$ will be seen to be closely related to the k^{th} derivative of f.

Next we have a nice theorem which provides a recursive formula for calculating divided differences.

Theorem: $f[x_0,x_1,\ldots,x_{n-1},x_n] = \dfrac{f[x_0,x_1,\ldots,x_{n-1}] - f[x_1,\ldots,x_{n-1},x_n]}{x_n - x_0}$

This theorem espresses an n^{th} order divided difference as a (first-order) divided difference of $n-1^{st}$ order divided differences.

Proof: Suppose the polynomial $P(x)$ interpolates f at $x_0,x_1,\ldots,x_{n-1}$ so that $P(x_i) = f(x_i)$, $i = 0,..,n-1$; suppose the polynomial $Q(x)$ interpolates f at $x_1,\ldots,x_{n-1},x_n$ so that $Q(x_i) = f(x_i)$, $i = 1,..,n$ Now we create as follows a polynomial $p(x)$ that interpolates f at $x_0,x_1,\ldots,x_{n-1},x_n$:

$$p(x) = \frac{(x_n-x)P(x) + (x-x_0)Q(x)}{(x_n-x_0)}$$

Clearly $p(x_0) = P(x_0) = f(x_0)$ and $p(x_n) = Q(x_n) = f(x_n)$ and for $i = 1,..,n-1$ we have

$$p(x_i) = \frac{(x_n-x_i)P(x_i) + (x_i-x_0)Q(x_i)}{(x_n-x_0)} = \frac{(x_n-x_i)f(x_i) + (x_i-x_0)f(x_i)}{(x_n-x_0)} = f(x_i)$$

Now in $p(x)$ the coefficient of x^n is, by definition, the divided difference $f[x_0,x_1,\ldots,x_{n-1},x_n]$; on the right, in the expression $p(x) = \dfrac{(x_n-x)P(x) + (x-x_0)Q(x)}{(x_n-x_0)}$ the coefficient of x^n is the highest power coefficient of $Q(x)$ minus the highest power coefficient of

$P(x)$, divided by $(x_n - x_0)$, or $\dfrac{f[x_0, x_1, \ldots, x_{n-1}] - f[x_1, \ldots, x_{n-1}, x_n]}{x_n - x_0}$, and equating the coefficient of x^n on both sides, we obtain

$$f[x_0, x_1, \ldots, x_{n-1}, x_n] = \frac{f[x_0, x_1, \ldots, x_{n-1}] - f[x_1, \ldots, x_{n-1}, x_n]}{x_n - x_0} .$$

Let's use this result for a practical purpose: Computing the coefficients $a_i = f[x_0, .., x_i]$ of the Newton polynomial.

We begin with an array of x-values $x_0, \ldots, x_n$ and an array of function values $f(x_0), \ldots, f(x_n)$. We compute the first order differences between adjacent points, then the second order differences, and so on. In tabular form, this produces the divided difference table

$$
\begin{array}{llclclcl}
\underline{x_i} & \underline{f[x_i]} & & \underline{f[x_i, x_{i+1}]} & & \underline{f[x_i, x_{i+1}, x_{i+2}]} & \cdots & \\
x_0 & f(x_0) & \searrow & & & & & \\
 & & & f[x_0, x_1] & \searrow & & & \\
x_1 & f(x_1) & \begin{smallmatrix}\nearrow \\ \searrow\end{smallmatrix} & & & f[x_0, x_1, x_2] & & \\
 & & & f[x_1, x_2] & \begin{smallmatrix}\nearrow \\ \searrow\end{smallmatrix} & & \searrow & \\
\vdots & \vdots & & \vdots & & \vdots & & f[x_0, \ldots, x_n] \\
 & & & f[x_{n-2}, x_{n-1}] & \begin{smallmatrix}\nearrow \\ \searrow\end{smallmatrix} & & \nearrow & \\
x_{n-1} & f(x_{n-1}) & \begin{smallmatrix}\nearrow \\ \searrow\end{smallmatrix} & & & f[x_{n-2}, x_{n-1}, x_n] & & \\
 & & & f[x_{n-1}, x_n] & \nearrow & & & \\
x_n & f(x_n) & \nearrow & & & & &
\end{array}
$$

The coefficients of the Newton interpolating polynomial appear on the top downward-sloping diagonal of the table. One might also note that any path through the table provides the coefficients of $p(x)$ in (a different) Newton form. If one wishes to interpolate at an additional interpolation point, x_{n+1}, then this may be added to the bottom of the table and the bottom upward-sloping diagonal computed to find the coefficient $a_{n+1} = f[x_0, .., x_n, x_{n+1}]$ of the additional term $a_{n+1}(x - x_0) \bullet\bullet (x - x_n)$.

Now we demonstrate the connection between divided differences and derivatives:

Theorem: Assuming that f has an n^{th} continuous derivative, $f[x_0, ..., x_n] = \frac{1}{n!} f^{(n)}(\xi)$ where ξ is between the smallest of the x_i and the largest.

Proof: Let $p(x)$ interpolate $f(x)$ at $x_0, ..., x_n$ and consider the difference $f(x) - p(x)$. This function is zero at the n+1 points $x_0, ..., x_n$. Now if a function is zero at two points, its derivative is zero somewhere in between (Rolle's theorem). If a function is zero at three points, then the first derivative is zero at two points (in between) and the second derivative is then zero at one point in between the three. Continuing, a function that is zero at $n+1$ points has its n^{th} derivative zero at one point in between. Also note that the n^{th} derivative of $p(x)$ is $n! * coeff\ of\ x^n = n! f[x_0, ..., x_n]$. We now have:

$(f - p)^{(n)}(\xi) = f^{(n)}(\xi) - p^{(n)}(\xi) = 0$, $f^{(n)}(\xi) - n! f[x_0, ..., x_n] = 0$ and the theorem follows.

Finally, we provide an important theorem analogous to Taylor's Theorem which says how well a polynomial interpolant $p_n(x)$ of degree n approximates the function $f(x)$

Theorem: The error in polynomial interpolation satisfies

$$f(x) - p_n(x) = f[x_0, .., x_n, x](x - x_0) \bullet\bullet (x - x_n) = \frac{1}{(n+1)!} f^{(n+1)}(\xi)(x - x_0) \bullet\bullet (x - x_n)$$

where ξ is between the smallest and largest values of the x_i.

Proof: We are given the points $x_0, .., x_n$ and values $f(x_1), .., f(x_n)$, and $p_n(x)$ interpolates the values of f at the given x-values. Now pick some other x-value, say $x = t$ and interpolate at $x_0, .., x_n, t$ with the polynomial $p_{n+1}(x)$ written in Newton form as $p_{n+1}(x) = p_n(x) + f[x_0, .., x_n, t](x - x_0) \bullet\bullet (x - x_n)$. Now since p_{n+1} interpolates at $x = t$ we have $f(t) = p_{n+1}(t) = p_n(t) + f[x_0, .., x_n, t](t - x_0) \bullet\bullet (t - x_n)$. Now simply replace t by x on both sides and obtain $f(x) - p_n(x) = f[x_0, .., x_n, x](x - x_0) \bullet\bullet(x - x_n)$ and the latter expression can be replaced by $\frac{1}{(n+1)!} f^{(n+1)}(\xi)(x - x_0) \bullet\bullet(x - x_n)$ by the previous theorem.

We sometimes say that the remainder in polynomial interpolation is given by $\frac{1}{(n+1)!} f^{(n+1)}(\xi)(x - x_0) \bullet\bullet(x - x_n)$. This can be estimated in a manner similar to the remainder in Taylor's theorem.