

Interpolation

Interpolating means finding a polynomial that passes through a discrete set of points:

Interpolation

Interpolating means finding an order n polynomial $P_n(x) = a_0 + a_1x + a_2x^2 + \dots + a_nx^n$ such that $P_n(x_i) = f(x_i) \forall i = 1, 2, \dots, n$, for a discrete set of points:

$$(x_0, f_0), (x_1, f_1), \dots, (x_n, f_n),$$

where $x_i = f(x_i)$.

The solution to this problem *always exists* and *is always unique*. It can be found by defining and solving a linear system. However, this method is not very popular, as the system can be very ill-conditioned. Other methods have been developed to interpolate polynomials much more effectively. We will see *Newton's Interpolation* and *Lagrange's Interpolation*.

Newton's Interpolation

The general form of *Newton's Polynomial* going through a set of $n + 1$ points is:

$$f(x) = a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \dots + a_n(x - x_0)(x - x_1) \dots (x - x_{n-1})$$

which can be written as:

$$f(x) = \sum_{i=0}^n a_i n_i(x)$$

where $n_i(x) = \prod_{j=0}^{i-1} (x - x_j)$.

This polynomial can be rewritten also in the form:

$$f(x) = a_0 + (x - x_0)(a_1 + (x - x_1)(a_2 + (x - x_2)(a_3 \dots + a_n(x - x_{n-1}))))$$

In this way, when we evaluate at x_i , all the terms multiplying the $(x - x_i)$ will vanish, and we will have:

$$f(x_i) = a_0 + (x_i - x_0)(a_1 \dots + (x_i - x_{i-3})(a_{i-2} + (x_i - x_{i-2})a_i))$$

Then, evaluating at x_0, x_1, \dots , we obtain:

$$f(x_0) = a_0 \rightarrow a_0 = y_0$$

$$f(x_1) = y_0 + (x_1 - x_0)a_1 \rightarrow a_1 = \frac{y_1 - y_0}{x_1 - x_0}$$

$$f(x_2) = y_0 + (x_2 - x_0) \left(\frac{y_1 - y_0}{x_1 - x_0} + (x_2 - x_1)a_2 \right) \rightarrow$$

$$\rightarrow a_2 = \frac{\frac{y_2 - y_0}{x_2 - x_0} - \frac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_1} = \dots = \frac{\frac{y_2 - y_1}{x_2 - x_1} - \frac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_0}$$

And if so on. These coefficients are expressed as what we call **divided differences**. For example:

$$a_1 = \frac{y_1 - y_0}{x_1 - x_0} = f[x_1, x_0]$$

$$a_2 = \frac{\frac{y_2 - y_1}{x_2 - x_1} - \frac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_0} = \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0} = f[x_2, x_1, x_0]$$

$$a_3 = \frac{\frac{\frac{y_3 - y_2}{x_3 - x_2} - \frac{y_2 - y_1}{x_2 - x_1}}{x_3 - x_1} - \frac{\frac{y_2 - y_1}{x_2 - x_1} - \frac{y_1 - y_0}{x_1 - x_0}}{x_2 - x_0}}{x_3 - x_0} =$$

$$= \frac{f[x_3, x_2, x_1] - f[x_2, x_1, x_0]}{x_3 - x_0} = f[x_3, x_2, x_1, x_0]$$

In general:

The k -th coefficient of the *Newton Polynomial* is defined by the following *divided differences* formula:

$$a_k = f[x_k, x_{k-1}, \dots, x_1, x_0] =$$

$$= \frac{f[x_k, x_{k-1}, \dots, x_2, x_1] - f[x_{k-1}, x_{k-2}, \dots, x_1, x_0]}{x_k - x_0}$$

The good thing about this method is that it is easy to add new data points, as they do not affect the previous coefficients.

Lagrange Interpolation

For the problem of finding an interpolating polynomial of degree n for a set of $n + 1$ points x_0, x_1, \dots, x_n and their associated set of values f_0, f_1, \dots, f_n , let us suppose that the polynomial we are looking for can be written as:

$$P_n(x) = \sum_{k=0}^n f_k l_k(x)$$

where $\{l_k\}$ for $k = 0, 1, \dots, n$ are a basis of polynomials of degree n that form a basis of \mathbb{P}_n

If we find a set of polynomials such that

$$l_k(x) = \begin{cases} 1, & i = k \\ 0, & i \neq k \end{cases}$$

then our problem is solved.

In order to find the polynomials we are looking for, we must remember that if l_k is a polynomial with x_i as a root, it verifies $l_k = (x - x_i)p_{n-1}(x)$. Then, as $l_k(x)$ has roots $x_1, x_2, x_{k-1}, x_{k+1}, \dots, x_n$. Then, the

polynomials we are looking for are:

$$l_k(x) = C_k \prod_{i=0, i \neq k}^n (x - x_i)$$

where C_k is a constant which is found after requiring $l_k(x_k) = 1$:

$$C_k = \prod_{i=0, i \neq k}^n \frac{1}{x_k - x_i}$$

Finally:

The polynomials we are trying to find are:

$$l_k(x) = \prod_{i=0, i \neq k}^n \frac{x - x_i}{x_k - x_i}$$

and thus the **interpolating polynomial** is given by the expression:

$$P_n(x) = \sum_{k=0}^n f_k \prod_{i=0, i \neq k}^n \frac{x - x_i}{x_k - x_i}$$

The interpolation polynomial is only an approximation:

The error of the lagrange interpolation polynomial is:

$$E_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^n (x - x_i), \quad x_0 < \xi < x_n$$

where $w_{n+1}(x) = \prod_{i=0}^n (x - x_i)$ is the *nodal polynomial* of order $n + 1$.

Runge's Phenomenon and Spline Interpolation

Interpolating with high order polynomials in a certain interval improves the results inside most of the interval. However, the errors tend to become very large at the extremes. This is known as *Runge's Phenomenon*, and it is the reason why it is not a good idea to interpolate with high order polynomials.

An additional related problem is that high order interpolation polynomials tend to fluctuate between nodes. To avoid these problems we can use **spline interpolation**.

Spline Interpolation

Spline interpolation consists on dividing at the nodes the interval where we are interpolating, then joining each pair of nodes with a low order interpolation polynomial.

We typically make the interpolation using a polynomial of order 3. In each spline interval (each defined by two data points), we need four equations in order to determine the four coefficients of our interpolation

polynomial. If we have n data points, we have $n - 1$ spline intervals, so we need $4(n - 1)$ equations to solve the interpolation:

- We can get $2(n - 1)$ equations from requiring the interpolation polynomials to contain the data points.
- We can get $n - 2$ equations from requiring continuity of the first derivative at the data points.
- We can get $n - 2$ equations from requiring continuity of the second derivative at the data points.
- We can get the last two equations from either of the following conditions:
 - *Natural condition*: imposing that the second derivative is zero at the endpoints.
 - *Knot-a-knot condition*: imposing that the third derivative at the end nodes is equal to the third derivative at the neighboring nodes.