1.4 Computing the polynomial interpolant

1.4.1 Synthetic Division

The Lagrange form of the polynomial interpolant has great theoretical importance. It is also useful for $n = 0, 1, 2$, say. It practice, for larger n, it is easy to write down the Lagrange form of the polynomial interpolant, but it is tedious (and computationally expensive) to work with. Evaluating each of the

$$L_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}$$

at a given value of x requires $2n - 1$ multiplications. So therefore computing

$$p_n(x) = \sum_{i=0}^{n} y_i L_i(x) = \sum_{i=0}^{n} \Big(y_i \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j} \Big),$$

for a given value of x requires $2n^2 + 2n$ multiplications. On the other hand, suppose we have p_n in the standard form

$$p_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n.$$

At first glance it seems that this takes $(n^2 - n)/2$ multiplications, which is only a little better. However there is a faster way of doing this called *synthetic division*. As an example we'll consider how this works for the case $n = 3$. For the general case, please have a look at [S96, Lecture 19].

`Take notes`:

In summary,

- The Lagrange form is easy to find and difficult to evaluate.
- The standard form (i.e., $p_n = a_0 + a_1 x + \cdots + a_n x^N$), is hard to find, but easy to evaluate.
- So we want to find a form for the polynomial is easy to construct *and* efficient to evaluate.

There are plenty of possibilities – too many to mention – and we'll consider one of them: the *Newton Form*.

1.4.2 The Newton Form of the Interpolant

This section is just for your information: it will not be covered in class or be part of any assessment.

These notes are based on §2.1.3 of [SB92] and on [S96, Lecture 19].

Suppose we were to write the interpolating polynomial p_n as

$$\begin{aligned} p_n(x) = a_0 &+ a_1(x - x_0) \\ &+ a_2(x - x_0)(x - x_1) \\ &+ a_3(x - x_0)(x - x_1)(x - x_2) \\ &+ \cdots + a_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}). \end{aligned}$$

Then it could actually be evaluated as

$$p_n(x) = a_0 + (x - x_0)\Big(a_1 + (x - x_1)\big(a_2 + (x - x_2)(a_3 + (\cdots + a_n(x - x_{n-1}))\cdots\big)\Big). \tag{1.9}$$

This is called the *Newton Form of the Interpolating Polynomial.* [8]

Let π_k be the nodal polynomial $\pi_k = \prod_{i=0}^{k}(x - x_i)$. Then the Newton form is

$$p_n(x) = a_0 + a_1\pi_0 + a_2\pi_1 + a_3\pi_2 + \cdots + a_n\pi_{n-1}.$$

So how do we find the coefficients? Let

$$F[x_i] = f(x_i),$$

$$F[x_i, x_{i+1}] = \frac{F[x_{i+1}] - F[x_i]}{x_{i+1} - x_i}$$

$$\vdots$$

$$F[x_i, x_{i+1}, \ldots, x_{i+k}] = \frac{F[x_{i+1}, x_{i+2}, \ldots x_{i+k}] - F[x_i, x_{i+1}, \ldots x_{i+k-1}]}{x_{i+k} - x_i}$$

Then

$$a_0 = f[x_0], \qquad a_1 = F[x_0, x_1], \quad \ldots$$
$$a_n = F[x_0, x_1, \ldots x_n].$$

Or, if you prefer:

$$p_n(x) = F[x_0] + \sum_{k=1}^{n} F[x_0, x_1, \ldots, x_k]\pi_k(x). \tag{1.10}$$

8 

Sir Issac Newton Isaac Newton, 1643 - 1727, England. Easily one of the greatest scientist of all time. This image is of a Polish stamp commemorating the scientific achievements. Source: `http://jeff560.tripod.com/stamps.html`

It is not hard to show that the formula given in (1.10) yields the interpolating polynomial. The argument is inductive. Before we do that, we will demonstrate that it works for a particular example.

Example 1.4.1. Write down the Newton form of the polynomial p_2 that interpolates e^x at $x_0 = 0$, $x_1 = 1$ and $x_2 = 2$.

Solution: (For the Newton form)

	$F[x_i]$	$F[x_i x_{i+1}]$	$F[x_i x_{i+1} x_{i+2}]$
x_0	1		
		$e-1$	
x_1	e		$\frac{1}{2}(e^2-2e+1)$
		e^2-e	
x_2	e^2		

This gives

$$\begin{aligned} p_2(x) &= F[x_0] + (x-x_0)\big(F[x_0x_1] + (x-x_1)F[x_0x_1x_2]\big), \\ &= 1 + x\big((e-1) + (x-2)\frac{1}{2}(e^2-2e+1)\big). \end{aligned}$$

Written in the form of (1.10) above, this is

$$p_2(x) = 1 + (e-1)(x) + \frac{1}{2}(e^2-2e+1)(x)(x-1).$$

One can quickly check that this

1. is a polynomial of degree n,
2. interpolates e^x at $x=0$, $x=1$ and $x=2$.

It is not hard to show that the formula given in (1.10) yields the interpolating polynomial. The argument is inductive. Clearly its true for $n = 0$, because in that case the interpolant is just the constant polynomial $p_0 = F[x_o] = f(x_0)$.

Next, suppose it is true for n points. Then let

$$\begin{aligned} p_n &\text{ interpolate } f(x) \text{ at } x = x_0, x_1, \dots x_n, \\ q_{n-1}(x) &\text{ interpolate } f(x) \text{ at } x = x_0, x_1, \dots x_{n-1}, \\ r_{n-1}(x) &\text{ interpolate } f(x) \text{ at } x = x_1, x_2 \dots x_n. \end{aligned}$$

We must convince ourselves that

$$p_n(x) = q_{n-1}(x) + \frac{x-x_0}{x_n-x_0}\big(r_{n-1}(x) - q_{n-1}(x)\big),$$

First, since both q_{n-1} and r_{n-1} are polynomials of degree $n-1$, it follows that the expression on the right is indeed a polynomial of degree n.

Next, check that it is the interpolant of f at x_0, x_1, ..., x_n. For the first point, x_0:

$$\begin{aligned} p_n(x_0) &= q_{n-1}(x_0) + \frac{x_0-x_0}{x_n-x_0}\big(r_{n-1}(x_0) - q_{n-1}(x_0)\big) \\ &= q_{n-1}(x_0) = f(x_0). \end{aligned}$$

For the last point, x_n:

$$\begin{aligned} p_n(x_n) &= q_{n-1}(x_n) + \frac{x_n-x_0}{x_n-x_0}\big(r_{n-1}(x_n) - q_{n-1}(x_n)\big) \\ &= q_{n-1}(x_n) + \big(r_{n-1}(x_n) - q_{n-1}(x_n)\big) \\ &= r_{n-1}(x_n) = f(x_n). \end{aligned}$$

And for every other point, x_i, $i = 1, \dots, n-1$,

$$\begin{aligned} p_n(x_i) &= q_{n-1}(x_i) + \frac{x_i-x_0}{x_n-x_0}\big(r_{n-1}(x_i) - q_{n-1}(x_i)\big) \\ &= f(x_i) + \frac{x_i-x_0}{x_n-x_0}\big(f(x_i) - f(x_i)\big) = f(x_i). \end{aligned}$$

Finally, we need to compare the coefficient of x^n on the left- and right-hand sides of this equations.

The coefficient of x^{n-1} in q_{n-1} is $F[x_0, x_1, \dots, x_{n-1}]$. The coefficient of x^{n-1} in r_{n-1} is $F[x_1, x_2, \dots, x_n]$. So the coefficient of x^n in p_n is

$$\frac{1}{x_n-x_0}\big(F[x_1, x_2, \dots, x_n] - F[x_0, x_1, \dots, x_{n-1}]\big),$$

as required.

1.4.3 Exercise

Exercise 1.12. Do Exercise 6.3 from Süli and Mayers, *An Introduction to Numerical Analysis*.