

MAT360 Section Summary: 3.2

Divided Differences

(Newton wins again)

1. Summary

We start with the interpolating polynomial of degree n given in Newton form:

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

where the coefficients a_i are to be determined. If all the $x_i = 0$, then we have the Maclaurin expansion of the polynomial; if all $x_i = c$ for some constant $c \neq 0$, then we have the Taylor series expansion of P_n about c . In either event, the coefficients will be scaled derivatives at a single point.

We're now interested in the case where the x_i are different (possibly simply equally spaced points); the coefficients will still be related to derivative information, but that information will be distributed across the x_i , rather than focused on a point. The key to computing them will be divided differences.

Divided differences are basically approximations to derivatives, as one can see from Theorem 3.6.

This is a somewhat "classical" subject (old-fashioned?): one used to consult tables to evaluate functions, whereas today we've got computers which have been programmed to do the job for us. Textbooks used to include lots of tables (e.g. of trigonometric functions), and if you wanted $\sin(1.543)$, you'd look in the table, find the values of $\sin(1.54)$ and $\sin(1.55)$, and interpolate! Nowadays, this is rarer, but you may have still had to do such things in a statistics class, for example, where tables of normal probabilities or t-distribution values are still used....

2. Definitions

- zeroth divided difference: $f[x_i] = f(x_i)$

- first divided difference:

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

- second divided difference:

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

- k th divided difference:

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

-
- **Newton's interpolatory divided difference formula:** using these divided differences, we can demonstrate that

$$P_n(x) = f[x_0] + \sum_{k=1}^n \underbrace{f[x_0, x_1, \dots, x_k]}_{\text{divided difference}} (x - x_0) \cdots (x - x_{k-1})$$

is the n^{th} degree interpolating polynomial.

3. Theorems/Formulas **Theorem 3.6:** Suppose that $f \in C^n[a, b]$

and x_0, x_1, \dots, x_n are distinct numbers in $[a, b]$. Then $\exists \xi \in [a, b]$ /

$$\overline{f[x_0, x_1, \dots, x_n]} = \frac{f^{(n)}(\xi)}{n!}.$$

Derivation of the coefficients of the Newton polynomial: Rather than define the divided differences as above, we could generate a recursive definition for them.

Define the term $f[x_0, x_1, \dots, x_n]$ as the leading coefficient of $P_n(x)$:

$$P_n(x) = f[x_0, x_1, \dots, x_n](x - x_0) \cdots (x - x_{n-1}) + P_{n-1}(x)$$

Since

$$P_n(x) = \frac{(x_n - x)P_{n-1}(x) + (x - x_0)Q_{n-1}(x)}{x_n - x_0}$$

where $P_{n-1}(x)$ is the $(n-1)^{\text{th}}$ interpolating polynomial to x_0, \dots, x_{n-1} , and $Q_{n-1}(x)$ is the $(n-1)^{\text{th}}$ interpolating polynomial to x_1, \dots, x_n , we can compute the leading coefficient of P_n as a function of the leading coefficients of P_{n-1} and Q_{n-1} :

$$\text{leading}(P_n(x)) = \frac{-\text{leading}(P_{n-1}) + \text{leading}(Q_{n-1})}{x_n - x_0}$$

Hence,

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

When you have a recursive definition, you need a basement: it is the leading coefficient of the constant interpolating function:

$$\overline{f[x_0]} = \overline{f(x_0)}$$

Conclusion: the Newton interpolating polynomial is given by

$$P_n(x) = f[x_0] + \sum_{k=1}^n f[x_0, x_1, \dots, x_k](x - x_0) \cdots (x - x_{k-1})$$

To get the polynomial's coefficients, you simply look along the diagonal of the divided difference table. Computation of the coefficients costs

$$3 \sum_{i=1}^n i = 3 \frac{n(n+1)}{2}$$

operations, whereas evaluation involves $4n - 1$ operations. While this seems expensive compared to Horner's method, we generate a succession of estimates (using the 0th through nth degree polynomials). So there is bang for the buck...

4. Properties/Tricks/Hints/Etc. This form of the interpolating polynomial is nice because we can easily increase the degree by adding an

additional knot, without much additional work. We can reuse the interpolating polynomial of previous degree. Furthermore, the knots don't have to be added at the end or beginning: this process was independent of the order of the x_i .

By following the table of divided differences up from the function values, we get a look at the estimated value of the function using higher and higher powered polynomials. We hope that if the values are settling down, that we're doing a pretty good job of approximating it (and that we needn't proceed to even higher degree). If we're not settling down, however, we know that with the Newton formulation, we can do even better.

If you're at the "left hand side" (near x_0) of the table, then it makes sense to make use of the forward-differences; if you're at the right of the table, then use the backward-differences. The only advantage of using one side versus the other is in watching the value of $P(x)$ stabilize as additional degreed polynomials are used; using either form gives the same value for the highest degreed approximation. So the authors' injunction that "The Newton formulas are not appropriate for approximating $f(x)$ when x lies near the center of the table..." are a little draconian.... Use them!

x	$f(x_0)$	$f[x_0, x_1]$	$f[x_0, x_1, x_2]$	$f[x_0, x_1, x_2, x_3]$
	$f(x_1)$	$f[x_1, x_2]$	$f[x_1, x_2, x_3]$	$f[x_1, x_2, x_3, x_4]$
	$f(x_2)$	$f[x_2, x_3]$		
	$f(x_3)$			