Divided Differences and Newton Interpolating Polynomials

Math 131: Numerical Analysis

J.C. Meza

2/27/24

Section 1

Divided Differences

Divided Differences

- Although an interpolating polynomial for a given set of points is unique, there are several algebraic representations we can use. We have one more form for interpolating polynomials that is frequently used.
- It has the advantage that the coefficients can be easily computed as new data becomes available.
- This could prove especially useful in an experimental setup where one may not know how much data will be available at the beginning of the experiment.

Notation

We'll need some new notation first in order to write down this new form for an interpolating polynomial.

- Let's first recall the shifted form for expressing a polynomial:

$$
\begin{gathered}
p_n(x) = c_0 + c_1(x - x_0) + c_2(x - x_0)(x - x_1) + ... \\
+ c_n(x - x_0)(x - x_1) ... (x - x_n - 1)
\end{gathered}
$$

Note that we could determine the coefficients through the following procedure:

$$
\begin{aligned}
&p_n(x_0) = c_0 = f(x_0) \\
&p_n(x_1) = c_0 + c_1(x_1 - x_0) = f(x_1) \implies c_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \\
&\quad ...
\end{aligned}
$$

Divided Differences

Definition

The ***zeroth divided difference*** of a function f with respect to x_i is defined as $f[x_i] = f(x_i)$. Using the procedure above as our guide, the ***first divided difference*** is defined as:

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

Higher order differences

Higher order divided differences can likewise be defined ***recursively*** in terms of lower order divided differences. For example the ***second divided difference*** can be expressed as:

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

so specifically for $i = 0$ we would have:

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

Higher order differences (general form)

In general, we can express the divided difference as:

$$f[x_0, x_1, x_2, \ldots, x_j] = \frac{f[x_1, x_2, \ldots, x_j] - f[x_0, x_1, \ldots, x_{j-1}]}{x_j - x_0} \tag{1}$$

Equation 1 is also known as the ***Newton divided difference*** form.

Useful Properties

Divided differences have some useful properties that we will use later on.

The first important one is that the ***order of the data points doesn't matter.***

In other words:

$$f[x_{i_0}, x_{i_1}, x_{i_2}, \ldots, x_{i_n}] = f[x_0, x_1, x_2, \ldots, x_n]$$

where$(i_0, i_1, i_2, \ldots, i_n)$ is a permutation of $0, 1, 2, \ldots, n$.

One can easily show that, for example:

$$f[x_1, x_0] = f[x_0, x_1].$$

Example

Compute the first and second divided differences for $f(x) = cos(x)$, using $x_0 = 0.2, x_1 = 0.3, x_2 = 0.4$.

$$
\begin{aligned}
f[x_0, x_1] &= \frac{f[x_1] - f[x_0]}{x_1 - x_0} \\
&= \frac{\cos(0.3) - \cos(0.2)}{0.3 - 0.2} \\
&= \frac{0.955336489 - 0.980066578}{0.1} \\
&= -0.24730089
\end{aligned}
$$

Example (cont.)

Likewise

$$
\begin{aligned}
f[x_1, x_2] &= \frac{f[x_2] - f[x_1]}{x_2 - x_1} \\
&= \frac{\cos(0.4) - \cos(0.3)}{0.1} \\
&= \frac{0.921060994 - 0.955336489}{0.1} \\
&= -0.34275495
\end{aligned}
$$

Example (cont.)

The second divided difference is therefore given by:

$$
\begin{aligned}
f[x_0, x_1, x_2] &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} \\
&= \frac{(-0.34275495) - (-0.24730089)}{0.4 - 0.2} \\
&= -0.477727030
\end{aligned}
$$

Section 2

Newton Interpolating Polynomials

Newton Interpolating Polynomials

- While divided differences have many applications, the one we will use here is as a means to write down an interpolating polynomial with an important computational property.
- Suppose as before that we have an interpolating polynomial of degree n such that $p_n(x) = f(x_i), i = 0, 1, \dots, n$ with distinct data points.
- Then an interpolating polynomial can be written using Newton divided differences as follows:

$$
\begin{aligned}
p_1(x) &= f(x_0) + f[x_0, x_1](x - x_0) \\
p_2(x) &= f(x_0) + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\
&\vdots \\
p_n(x) &= f(x_0) + f[x_0, x_1](x - x_0) + \dots \\
&\quad + f[x_0, x_1, \dots, x_n](x - x_0)(x - x_1) \cdots (x - x_{n-1})
\end{aligned}
$$

Advantages of Newton Polynomial

Note that for all interpolating polynomial of degree $n>1$, they can all be constructed by using the previous interpolating polynomial of degree $n-1$, by simply adding one additional term. For example:

$$
\begin{aligned}
p_1(x) &= f(x_0) + f[x_0, x_1](x - x_0) \\
p_2(x) &= p_1(x) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\
&\vdots \\
p_n(x) &= p_{n-1}(x) + f[x_0, x_1, \ldots, x_n](x - x_0)(x - x_1) \cdots (x - x_{n-1})
\end{aligned}
$$

Adding new points

In general we can write:

$$p_{k+1}(x) = p_k(x) + f[x_0, x_1, \dots, x_k](x - x_0)(x - x_1) \cdots (x - x_k)$$

The beauty of this form is that we can easily go from a polynomial of degree k to degree $k+1$, by adding one ($k{-}th$ order) divided difference.

Remark

As before, it is not difficult to show that the Newton divided difference interpolating polynomial is exactly the same as the one we derived earlier (if perhaps in a slightly disguised form).

Exercise: Divided Difference Table

Table 1: Divided Difference Table

i	x_i	$f[\cdot]$	$f[\cdot,\cdot]$	$f[\cdot,\cdot,\cdot]$
0	1	1		
1	2	3		
2	4	3		

Compute the second degree Newton polynomial, i.e. $p_2(x)$.

Theorem

The following theorem will prove useful in future analysis.

Theorem

Let $n \geq 1$ and assume that $f(x) \in C^n[a,b]$. Let $x_0, x_1, \ldots, x_n$ be $n+1$ distinct points in $[a,b]$. Then

$$f[x_0, x_1, x_2, \ldots, x_n] = \frac{1}{n!} f^n(\xi),$$

for some ξ between the minimum and maximum of $x_0, x_1, \ldots, x_n$.

Remark: This might look familiar as it looks similar to the remainder term in the Taylor polynomial. As a result, it won't come as a surprise that it will be useful in our error analysis

Summary

We've now seen three different approaches for computing interpolating polynomials. Let's briefly summarize some of the most important features of each of them.

Table 2: Interpolating Polynomials Summary

Basis	$\phi_j(x)$	Constr.	Eval.	Feature
Monomial	x^j	$O(n^3)$	$O(n)$	Simple, easy
Lagrange	$L_j(x)$	$O(n^2)$	$O(n)$	$c_j = y_j$, most stable
Newton	$\prod_{i=0}^{j-1}(x - x_i)$	$O(n^2)$	$O(n)$	Adaptive (can add new points)

Section 3

Error Analysis

Error in Polynomial Interpolation

If we have an interpolating polynomial for a given function at a set of points, we know that by definition, it must match the function exactly at the nodes. However, this also brings up a related question - How does $p_n(x)$ behave at points other than the nodes $\{x_i\}$?

In order to consider this question we will make a few assumptions to help us with the analysis, specifically:

1. $f(x)$ is defined on $[a, b]$.
2. f has all needed derivatives and they are bounded.

Error Function

Let's first define the error function for the $n-th$ degree interpolating polynomial:

$$e_n(x) = f(x) - p_n(x), \quad x \in [a, b].$$

First step

The question we would like to ask is what does the error look like for a point in the interval $[a, b]$ that is not one of the node points.

Here we will use the simple observation that any x that is not a node can be treated as a new interpolation point. As such, we can use the Newton Interpolation formula for adding a new point, i.e.

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

where

$$\Psi_n(x) = (x - x_0)(x - x_1)\cdots(x - x_n).$$

Substituting this into the error function we get:

$$e_n(x) = f(x) - p_n(x) = f[x_0, x_1, \ldots, x_n, x]\Psi_n(x).$$

Alternate form for divided difference

Using previous theorem we can replace the divided difference with the derivative:

$$f[x_0, x_1, x_2, \dots, x_n, x] = \frac{f^{n+1}(\xi)}{(n+1)!},$$

to give us the following formula for the error function:

$$e_n(x) = f(x) - p_n(x) = \frac{f^{n+1}(\xi)}{(n+1)!}\Psi_n(x).$$

This leads us directly to the following theorem.

Polynomial Interpolation Error

Theorem (Polynomial Interpolation Error)

Let $n \geq 0$ and $f \in C^n[a,b]$. Suppose we are given $x_0, x_1, \ldots, x_n$ distinct points in $[a,b]$. Then

$$f(x) - p_n(x) = \frac{f^{n+1}(\xi)}{(n+1)!}\Psi_n(x), \tag{2}$$

with $\Psi_n(x) = \prod_{i=0}^{n}(x - x_i).$, for $a \leq x \leq b$, where $\xi(x)$ is an unknown between the min and max of $x_0, x_1, \ldots, x_n$ and x.
Furthermore

$$\max |f(x) - p_n(x)| \leq \frac{1}{(n+1)!} \max_{a \leq t \leq b} \left| f^{(n+1)}(t) \right| \cdot \max_{a \leq s \leq b} \prod_{i=0}^{n} |s - x_i|$$

Interpretation

- This theorem is nice, but in order to proceed much further, we would need to know more about both f and $\Psi_n(x)$.
- Nonetheless, it does suggest that in order to minimize the error, it would be good to keep the new point x close to one of the interpolating points.
- A final note is in order. We used the Newton form to obtain this error bound, but the bound will apply to other polynomial forms since we know we have a unique polynomial, and the various forms we have used are merely disguised versions of each other.

Example

To Do:

- Include one example here for theorem, e.g. $f(x) = \exp(x)$
- Include one example here for previous theorem, e.g. $f(x) = \cos(x)$

Practical Tips

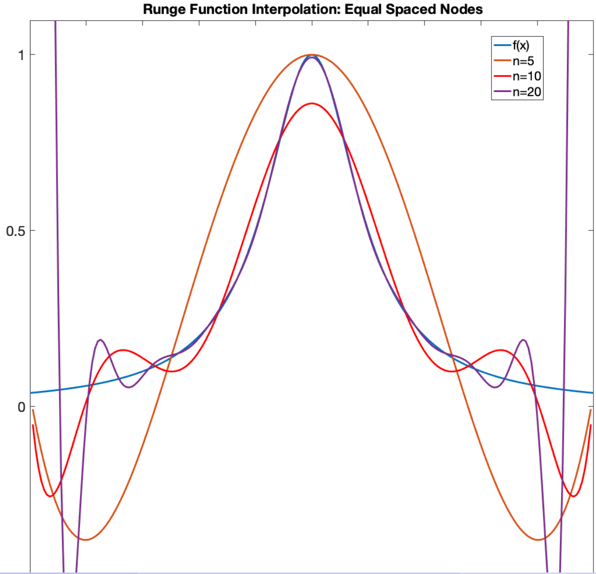
1. In understanding the error we might encounter when using interpolation, we can use Equation 2, to give us a sense of how the error behaves throughout the interval. Clearly, at the node points themselves, the error will be zero, but what about the rest of the points in $[a, b]$?
2. When considering the error bounds above, the interpolation error is likely to be smaller when evaluated at points close to the middle of the domain.
3. In practice, high degree polynomials with equally spaced nodes are not suitable for interpolation because of this oscillatory behavior.
4. However, if a set of suitably chosen data points that are not equally spaced may be useful in obtaining polynomial approximations of some functions.

Example Runge Function

- An excellent example of the type of behavior that can occur when using equally spaced nodes was described by Runge in 1901.
- If one uses equally spaced nodes, for example $x_i = \frac{2i}{n} - 1$ on the interval $[-1, 1]$, then it can be shown that the interpolation error grows without bound at the ends of the interval.
- For example consider the seemingly innocuous function:

$$f(x) = 1/(1 + x^2), \qquad -5 \leq x \leq 5.$$

Example (cont.)



High-Degree Polynomials - DON'T!

Important

In general, one should be wary of using a high-degree polynomial as they can oscillate drastically and care must be taken whenever used. Other approaches will prove to be more useful in situations where more accuracy is required or more data points are given.

Chebyshev Points

The Chebyshev points are defined on the interval $[-1, 1]$ by:

$$x_i = \cos\Big(\frac{2i+1}{2(n+1)}\pi\Big), \quad i = 0, \dots n. \tag{3}$$

To generate the desired points for a general interval $[a, b]$, one then uses the transformation on Equation 3:

$$\tilde{x}_i = a + \frac{b-a}{2}(x_i + 1), \quad i = 0, \dots n. \tag{4}$$

Advantages

These new points have the feature that they are clustered near the end points of the interval rather than being uniformly spaced across the interval.

If one uses, for example, the Lagrange polynomial form with the Chebyshev points defined by Equation 4 then it can be shown that the interpolation error is greatly reduced.

> **Note**
>
> Chebyshev points and interpolation have many interesting properties and many different applications. Unfortunately, we will not have time to go over most of these advanced topics.