Polynomial Interpolation and Error Analysis

0.1 General Error Bound

Theorem 1
Assume that $f \in C^{n+1}[a,b]$ and that $x_0, x_1, \ldots x_n$ are distinct nodes in $[a,b]$. Let p_n denote the unique interpolating polynomial of degree $\leq n$ for the given $n+1$ nodes. If $x \in [a,b]$, then there exists a number $c = c(x) \in (a,b)$ such that

$$f(x) - p_n(x) = e_n(x), \tag{1}$$

where the error function is explicitly given by

$$e_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(c) \prod_{j=0}^{n} (x - x_j). \tag{2}$$

Note: This expression for the error e_n can be simplified if we choose to distribute our nodes so that they are equally spaced. That is, fix n and define the step length

$$h = \frac{b-a}{n},$$

then we can define the partition of the interval $[a,b]$ as follows:

$$x_0 = a, \quad x_i = a + ih, \quad \text{for } i = 1, 2, \ldots, n.$$

This results in $x_n = b$. See your textbook for more details that the following bound can be derived for the product term in (2),

$$\left| \prod_{j=0}^{n} (x - x_j) \right| \leq \prod_{j=0}^{n} |(x - x_j)| \leq \frac{n!}{4} h^{n+1}.$$

Then the error term in equation (2) can be bounded as follows

$$|e_n(x)| \leq \frac{h^{n+1}}{4(n+1)} \max_{c \in [a,b]} |f^{(n+1)}(c)|.$$

Hence, if we want to bound the error term, we need to be able to bound the derivative term in equation (2) for a given value of n.

0.2 Implications of Error Bounds

If the derivatives of f can be uniformly bounded by a constant, then we can choose n appropriately large in order to force the error term to be as small as we want. So, we can find a very accurate interpolating polynomial. In particular, if we look at the case when we choose to use equally spaced nodes (but without pre-determining n), then bounding the derivatives allows us to choose n large enough (alternatively h small enough) to give an accurate polynomial interpolant. See your **Homework 2** for an example. The above form of the error bound is not as useful as the following one for quick calculations.

REMARK: **If the derivatives of a function are not uniformly bounded, then we can get into trouble by choosing a large number of nodes at which to interpolate the function.** Theorem and example follow.

Theorem 2
Assume that $f \in C^{n+1}[a,b]$ and satisfies the bound $|f^{(n+1)}(x)| \le M$ for all $x \in [a,b]$. Let $p_n(x)$ denote the unique interpolating polynomial of degree $\le n$ that interpolates f at $n+1$ equally spaced nodes in $[a,b]$, including the endpoints. If $x \in [a,b]$, then

$$|f(x) - p_n(x)| \le \frac{1}{4(n+1)} M \left(\frac{b-a}{n}\right)^{n+1} \tag{3}$$

Example 1: See page 174 of your textbook Let $f(x) = \sin x$, and give an upper bound for the error if f is approximated by an interpolation polynomial with ten equally spaced nodes in $[0, 1.6875]$.

$$n = 9; \quad a = 0; \quad b = 1.6875;$$

AND

$$f^{(10)}(x) = -\sin x \quad \rightarrow \quad |f^{(10)}(x)| \le 1 \quad \forall\, x \in [a,b]$$

Hence, the interpolation error can be bounded by

$$|\sin x - p_n(x)| \le \frac{1}{40}(1)\left(\frac{1.6875}{9}\right)^{10} \approx 1.34 * 10^{-9}$$

for all $x \in [0, 1.6875]$.

Remark: And note that $f^{(n)}(x) = \pm \sin x$ for even n and $f^{(n)}(x) = \pm \cos x$ for odd n, so we have a UNIFORM BOUND on $f^{(n)}(x)$ FOR ALL n. That is, $|f^{(n)}(x)| \le 1$ for all x and for all n. *In such instances*, we can force interpolation error to 0 by increasing the number of interpolation nodes.

MORE IMPORTANT REMARK: If the derivatives of all order for the function f are continuous but we CANNOT uniformly bound the derivatives, then increasing the # of interpolation nodes MAY NOT result in smaller errors. Example 2 gives us an example of *when bad things happen to good mathematicians.* (tongue-in-cheek cliche for entertainment purposes only)

Example 2: Runge's Function is defined by

$$f(x) = \frac{1}{1+x^2}.$$

It is the solid curve shown in Figure 1 below. Choose $x_0 = -5.0$, $h = 10/9$ and $x_j = -5 + jh$ for $j = 1, 2, \ldots 9$, so $x_9 = 5.0$. If we use a Lagrange Polynomial interpolant of degree 9, then we have the dashed curve (- -) in Figure 1. Notice the oscillations in the interpolating polynomial.

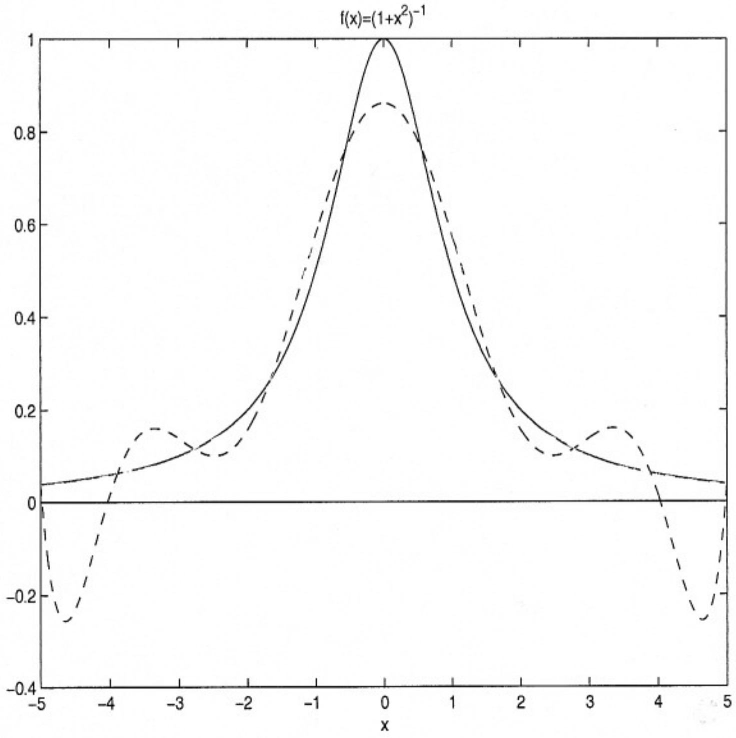

Figure 1: Runge's Function with Lagrange Interpolating Polynomial of degree 9.

How could we have predicted this behavior? If we examine the error bound for the Lagrange polynomial, we know that we are guaranteed a bound of

$$|f(x) - P(x)| \leq \frac{1}{40} M \left(\frac{10}{9}\right)^{10},$$

where $M = \max_{c\in[-5,5]} |f^{(10)}(c)|$. What is the maximum value that the tenth derivative of f takes on over the interval $[-5, 5]$? Check out graphs of the f', f'' and f''' in Figure 2. Notice the scale of the graph! The maximum absolute value of the third derivative is nearly 15, and the maximum value for $|f^{(10)}|$ is a *very large value*. Note that the term

$$\frac{1}{40}\left(\frac{10}{9}\right)^{10} \approx 2.868/40 \approx 0.0717,$$

so this term will not be small enough to guarantee a reasonable bound on the error. Furthermore, if we allow n to get larger, then the magnitude of the derivatives of f also get

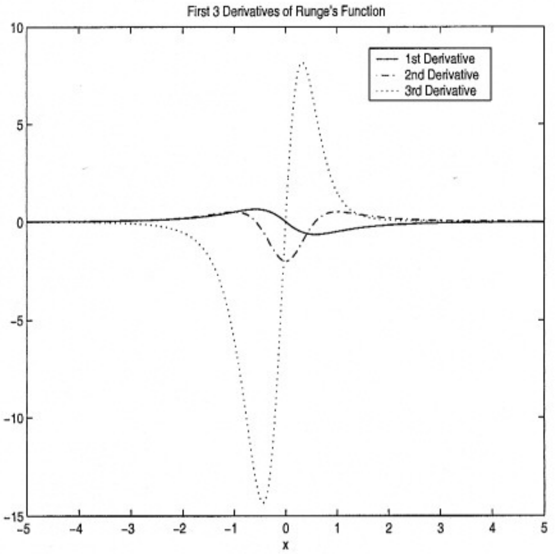

Figure 2: The first 3 Derivatives of Runge's Function

very large very fast. So, adding more interpolation points can increase the oscillation of the interpolating polynomial.
A different approach is a cubic spline interpolant. (We will talk about this one later.) In Figure 3, we see that the cubic spline interpolant does not show any of this oscillatory behavior. The cubic spline interpolant is the dashed and dotted (- .) curve. We get much better agreement with the function f over the entire interval $[-5, 5]$. Sometimes (that is, for some functions), we need better ways to interpolate, and we will talk about cubic splines soon.

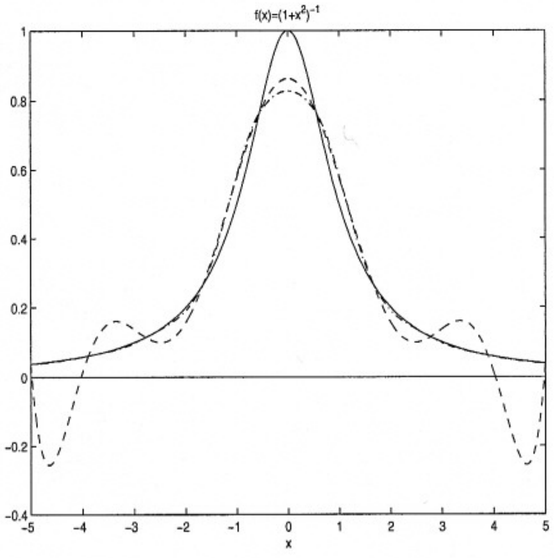

Figure 3: Lagrange and Cubic Spline Interpolating Polynomial.

Error Analysis:

Thm 1 (Error Expression for Interpolating Polynomials)

Assume that $f \in C^{n+1}[a,b]$ and that $x_0, x_1, \ldots x_n \in [a,b]$ are distinct nodes. If $x \in [a,b]$, then there exists a number $c = c(x)$ (c depends on x) such that

$$f(x) - P_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(c) \prod_{i=0}^{n} (x - x_i) \triangleq e_n(x)$$

where $P_n(x)$ is the Interpolating Polynomial using the nodes $x_0, x_1, \ldots, x_n$. The error term is given by

$$e_n(x) = \frac{f^{(n+1)}(c)}{(n+1)!} (x-x_0)(x-x_1)\ldots(x-x_n) = \frac{f^{(n+1)}(c)}{(n+1)!} \prod_{j=0}^{n} (x - x_j),$$

and c is a pt in the open interval $(a,b) = (x_0, x_n)$.

pf: First, note that if $x = x_j$ for $j = 0, 1, \ldots n$, then $E_n(x) = 0$, and $f(x) = P_n(x)$, so the result is true if x is a node. Now, choose arbitrary but fixed $x \neq x_j$ for all $j = 0, 1, \ldots n$.

① Define $w(t) = \prod_{i=0}^{n} (t - x_i)$
Note that $w(x)$ is a fixed, non-zero value. (since x is not a node)
Also note that $f(x)$ and $P_n(x)$ are fixed values.

② Define the constant (well-defined since $w(x) \neq 0$)

$$\lambda = \frac{f(x) - P_n(x)}{w(x)} \qquad (\Rightarrow \lambda w(x) = f(x) - P_n(x))$$

③ Define $\phi(t) = f(t) - P_n(t) - \lambda w(t)$
Make the following observations:

Observations

A. $\Phi(x) = 0$

B. $\phi(x_j) = 0$ for $j = 0, 1, 2, \ldots$

C. $\phi \in C^{n+1}[a,b]$

Since ϕ has $n+2$ roots in $[a,b]$, Rolle's Thm. (bottom pg. 172) guarantees that ϕ' has $n+1$ roots in (a,b). Applying Rolle's Thm to ϕ', it follows that ϕ'' has n roots in (a,b). By successively applying Rolle's Thm (OR by invoking the "Generalized Rolle's Thm"), one arrives at the conclusion that $\phi^{(n+1)}$ must have a root in (a,b). Hence, $\exists$ $C = C(x)$ so that $C \in (a,b)$ and

$$\phi^{(n+1)}(c) = 0$$

So, $$\Phi^{n+1}(c) = \frac{d^{n+1}}{dt^{n+1}} \Phi(t) \Big|_{t=C}$$

And $$\frac{d^{n+1}}{dt^{n+1}} \left[\Phi(t) \right] = f^{(n+1)}(t) - \frac{d^{n+1}}{dt^{n+1}} [P_n(t)] - \lambda \frac{d^{n+1}}{dt^{n+1}} [w(t)]$$

- $\deg(P_n(t)) \leq n \implies \frac{d^{n+1}}{dt^{n+1}} [P_n(t)] = 0 \quad \forall t$

- $w(t) = \prod_{i=0}^{n} (t - x_i) = t^{n+1} + g(t)$
  with $\deg(g(t)) \leq n$.

$$\frac{d^{n+1}}{dt^{n+1}} [w(t)] = (n+1)!$$

$$\implies \Phi^{(n+1)}(t) = f^{(n+1)}(t) - \lambda (n+1)!$$

Hence $\quad \phi^{(n+1)}(c) = 0$

gives us

$$f^{(n+1)}(c) - \lambda(n+1)! = 0$$

$$f^{(n+1)}(c) - \left[\frac{f(x) - P_n(x)}{w(x)}\right](n+1)! = 0$$

$$\left[f(x) - P_n(x)\right]\frac{(n+1)!}{w(x)} = f^{(n+1)}(c)$$

$$f(x) - P_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(c)\, w(x)$$

$$f(x) - P_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(c) \prod_{i=0}^{n}(x - x_i)$$

<u>Drawbacks:</u>

1. Requires $f \in C^{n+1}[a,b]$ (lots of smoothness), and we have to be able to bound the derivatives of f to get an explicit error bound.

2. If we only have data pts to use for interpolation (with no explicit representation of the derivatives of f), then we must assume the function has enough derivatives that are well-behaved so that the interpolation error is reasonably small. (May Require info about your particular data set.)

Interpolation Errors Are Related To Divided Differences

Theorem 3

Let p_n interpolate the function f at nodes $x_0, x_1, \ldots, x_n$. For any x that's not a node,

$$f(x) - p_n(x) = f[x_0, x_1, \ldots, x_n, x] \prod_{i=0}^{n} (x - x_i)$$

proof: Let x be any point that's not a node, and assume that $f(x)$ is defined. Let $q(t)$ be the poly. with $\deg(q(t)) \leq n+1$ which interpolates f at $x_0, x_1, \ldots, x_n, x$. Using the Newton form of the poly., we write

$$q(t) = p_n(t) + f[x_0, x_1, \ldots, x_n, x] \prod_{i=0}^{n} (t - x_i)$$

But $q(x) = f(x)$, so we have

$$f(x) = p_n(x) + f[x_0, x_1, \ldots, x_n, x] \prod_{i=0}^{n} (x - x_j)$$

$$f(x) - p_n(x) = f[x_0, x_1, \ldots, x_n, x] \prod_{i=0}^{n} (x - x_j)$$

//