Steven Pav spav@math.ucsd.edu

§4.2 Errors in Polynomial Interpolation (Continued)

Recall that we bounded the error in a polynomial interpolation as follows:

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

We have no control over the function $f(x)$, its derivatives or the polynomial interpolant, thus the only way we can insure that the error $|f(x) - p(x)|$ is small is by judicious choice of the nodes x_i.

I've been advertising the Chebyshev nodes a bit; the Chebyshev nodes on $[a, b]$ have the remarkable property that

$$\left| \prod_{i=0}^{n} (t - x_i) \right| \leq 2^{-n}$$

for any $t \in [a, b]$. Moreover, it can be shown that for *any* choice of nodes x_i that

$$\max_{t \in [a,b]} \left| \prod_{i=0}^{n} (t - x_i) \right| \geq 2^{-n}.$$

In this sense the Chebyshev nodes are considered the best for polynomial interpolation.

Despite the proven superiority of Chebyshev Nodes, and the problems with the Runge Function, equally spaced nodes are frequently used for interpolation, since they are easy to calculate. We now consider bounding

$$\max_{x \in [a,b]} \prod_{i=0}^{n} |x - x_i|,$$

where

$$x_i = a + hi = a + \frac{(b-a)}{n} i, \qquad i = 0, 1, \ldots, n.$$

Start by picking an x. We can assume x is not one of the nodes, otherwise the product in question is zero. Then x is between some x_j, x_{j+1} We can show that

$$|x - x_j| \, |x - x_{j+1}| \leq \frac{h^2}{4}.$$

by simple calculus.

Now we claim that $|x - x_i| \leq (j - i + 1)\, h$ for $i < j$, and $|x - x_i| \leq (i - j)\, h$ for $j + 1 < i$. Then

$$\prod_{i=0}^{n} |x - x_i| \leq \frac{h^2}{4} \left[(j+1)! h^j \right] \left[(n-j)! h^{n-j-1} \right].$$

Through use of mathemagic, we get an overall bound

$$\prod_{i=0}^{n} |x - x_i| \leq \frac{h^{n+1} n!}{4}.$$

Note: When I tried to prove this in class, I was a little bit lazy, and only proved that $k!l! \leq (k+l)!$. Al pointed out that this would give us a bound of only $\frac{h^{n+1}(n+1)!}{4}$. However, the slightly stronger statement *is* true, though it is ugly to prove. See §4.2 problem number 3 in the textbook if you do not believe me.

The interpolation theorem then gives us

$$f(x) - p(x) \leq \frac{h^{n+1}}{4\,(n+1)} f^{(n+1)}(\xi).$$

The reason this does not seem to apply to Runge's Function is that $f^{(n)}$ for Runge's Function becomes unbounded as $n \to \infty$.

§4.3 Richardson Extrapolation

Suppose we have some blackbox function $f(x)$ and we wish to calculate $f'(x)$ at some given x. Not suprisingly, we start with Taylor's theorem:

$$f(x+h) = f(x) + f'(x)h + \frac{f''(\xi)h^2}{2}.$$

Rearranging we get

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{f''(\xi)h}{2}.$$

Remember that ξ is between x and $x+h$, but its exact value is not known. Normally we use facts about $f(\cdot)$ (or rather, about $f''(\cdot)$) to bound $f''(\xi)$. The approximation for $f'(x)$ should remind you of the definition of $f'(x)$ as a limit.

We call the error term the *truncation error*. The truncation error is the error that you accept when you say "$f'(x)$ is pretty much $\frac{f(x+h)-f(x)}{h}$." It has nothing to do with the kind of error that you get when you do calculations with a computer with limited precision; even if you worked in infinite precision, you would still have truncation error. Note also that if you actually try to compute the approximate $f'(x)$ in this case you may lose precision due to subtractive cancellation, as $f(x+h)$ is probably near $f(x)$ for small h.

For this approximation, our truncation error is $\frac{f''(\xi)h}{2}$ or, more loosely, $\mathcal{O}(h)$. We may want a more precise approximation. By now, you should know that any calculation starts with Taylor's Theorem:

$$\begin{aligned} f(x+h) &= f(x) + f'(x)h + \frac{f''(x)}{2}h^2 + \frac{f'''(\xi_1)}{3!}h^3 \\ f(x-h) &= f(x) - f'(x)h + \frac{f''(x)}{2}h^2 - \frac{f'''(\xi_2)}{3!}h^3 \end{aligned}$$

By subtracting these two lines, we get

$$f(x+h) - f(x-h) \quad = \quad 2f'(x)h + \frac{f'''(\xi_1) + f'''(\xi_2)}{3!}h^3.$$

Thus

$$\begin{aligned} 2f'(x)h &= f(x+h) - f(x-h) - \frac{f'''(\xi_1) + f'''(\xi_2)}{3!}h^3 \\ f'(x) &= \frac{f(x+h) - f(x-h)}{2h} - \left[\frac{f'''(\xi_1) + f'''(\xi_2)}{2}\right]\frac{h^2}{6} \end{aligned}$$

If $f'''(x)$ is continuous, then there is some ξ between ξ_1, ξ_2 such that $f'''(\xi) = \frac{f'''(\xi_1)+f'''(\xi_2)}{2}$. (This is the MVT at work.) Assuming some uniform bound on $f'''(\cdot)$, we get

$$f'(x) \quad = \quad \frac{f(x+h) - f(x-h)}{2h} + \mathcal{O}\left(h^2\right).$$

Improving Accuracy

This gives us a truncation error of $\mathcal{O}\left(h^2\right)$, which is better than $\mathcal{O}\left(h\right)$. Can we do better? (Always ask yourself that) Let's define

$$\phi(h) = \frac{1}{2h}\left[f(x+h) - f(x-h)\right].$$

Had we expanded the Taylor's Series for $f(x+h), f(x-h)$ to more terms we would have seen that

$$\phi(h) \quad = \quad f'(x) + a_2h^2 + a_4h^4 + a_6h^6 + a_8h^8 + \ldots$$

The constants a_i are a function of $f^{(i)}(x)$ only. (In fact, they should take the value of $2\frac{f^{(i)}(x)}{i!}$.) What happens if we now calculate $\phi(h/2)$?

$$\phi(h/2) \quad = \quad f'(x) + \frac{1}{4}a_2h^2 + \frac{1}{16}a_4h^4 + \frac{1}{64}a_6h^6 + \frac{1}{256}a_8h^8 + \ldots$$

But we can combine this with $\phi(h)$ to get better accuracy. We have to be a little tricky, but we can get the $\mathcal{O}\left(h^2\right)$ terms to cancel by taking the right multiples of these two approximations:

$$\begin{aligned} \phi(h) - 4\phi(h/2) &= -3f'(x) + \frac{3}{4}4h^4 + \frac{15}{16}a_6h^6 + \frac{63}{64}a_8h^8 + \ldots \\ \frac{4\phi(h/2) - \phi(h)}{3} &= f'(x) - \frac{1}{4}4h^4 - \frac{5}{16}a_6h^6 - \frac{21}{64}a_8h^8 + \ldots \end{aligned}$$

This approximation has a truncation error of $\mathcal{O}\left(h^4\right)$.

This technique of getting better approximations is known as the *Richardson Extrapolation*, and can be repeatedly applied. We will also use this technique later to get better quadrature rules–that is, ways of approximating the definite integral of a function.

Abstracting Richardson's Method

We now discuss Richardson's Method in a more abstract framework. Suppose you want to calculate some quantity L, and have found, through theory, some approximation:

$$\phi(h) = L + \sum_{k=1}^{\infty} a_{2k} h^{2k}.$$

Let

$$D(n, 0) = \phi\left(\frac{h}{2^n}\right).$$

Now define

$$D(n, m) = \frac{4^m D(n, m-1) - D(n-1, m-1)}{4^m - 1}.$$

We will be interested in calculating $D(n, n)$ for some n. We will show that

$$D(n, n) = L + \mathcal{O}\left(h^{2(n+1)}\right).$$

First we examine the recurrence for $D(n, m)$. As in divided differences, we use a pyramid table:

$$\begin{array}{lllll}
D(0,0) & & & & \\
D(1,0) & D(1,1) & & & \\
D(2,0) & D(2,1) & D(2,2) & & \\
\vdots & \vdots & \vdots & \ddots & \\
D(n,0) & D(n,1) & D(n,2) & \cdots & D(n,n)
\end{array}$$

By definition we know how to calculate the first column of this table; every other entry in the table depends on two other entries, one directly to the left, and the other to the left and up one space. Thus to calculate $D(n, n)$ we have to compute this whole lower triangular array.