18.330 Lecture Notes: Richardson Extrapolation

Homer Reid

March 3, 2015

Suppose we are carrying out some sort of numerical procedure that involves an adjustable parameter Δ that tunes the accuracy of method at the expense of computational cost. As we shrink Δ toward 0, the accuracy of our calculation improves, but the calculation becomes more expensive. (Alternatively, we might characterize computational cost in terms of $N \propto \frac{1}{\Delta}$, in which case the accuracy improves as $N \to \infty$.) A good example to have in mind is numerical quadrature via the trapezoidal rule: here the adjustable parameter Δ is just the width of the trapezoids, and $N = \frac{b-a}{\Delta}$ is the total number of trapezoids we need to use to integrate over an interval $[a, b]$.

Let $F(\Delta)$ denote the value returned by our numerical procedure for a given choice of Δ. Ideally we would like to compute the quantity $F(\Delta = 0)$, but this is generally impossible as it would require an infinite amount of computation. Instead, we will have to make do with computing F at finite values of Δ.

We will concern ourself here with the case in which we know *a priori* how the accuracy of our numerical procedure depends on Δ. More specifically, we will assume that we know our method is a p-th order method – that is, that the error incurred by our numerical procedure is given by a polynomial in Δ whose leading term has degree p, i.e.

$$F(\Delta) - F(0) = A\Delta^p + O\left(\Delta^{p+1}\right). \tag{1}$$

where A is some unknown constant. For example, for the trapezoidal rule we have $p = 2$, while for the rectangular rule we have $p = 1$.

To summarize the situation in symbols, we have

$$\boxed{\underbrace{F(0)}_{\text{what we want}} = \underbrace{F(\Delta)}_{\text{what we can compute}} - \underbrace{A\Delta^p}_{\text{dominant error term}} + \underbrace{O(\Delta^{p+1})}_{\text{higher-order error terms}}} \tag{2}$$

The quantity p determines how hard we have to work to improve the accuracy of a given estimate of our quantity. To see this, suppose we have computed $F(\Delta)$ for some value of Δ, and suppose we now want to refine this estimate by adding roughly one digit of precision—that is to say, we want to decrease the error by a factor of 10. If $p = 1$, then to reduce the error by 10 we must decrease Δ by

10. For something like rectangular-rule integration, this means we have to do 10 times more work just to earn that extra digit! In contrast, if $p = 2$, then we only need to do $\sqrt{10} \approx 3$ times more work. Clearly the higher the value of p the better.

Richardson extrapolation is a technique for increasing the effective value of p. The idea is to compare two evaluations of $F(\Delta)$, at two different values of Δ, and use what we know about the Δ dependence of the to eliminate the leading-order error term. To see how it works, suppose we have evaluated F at Δ and at $\Delta/2$. Applying equation (1) twice, we express the numbers we have obtained in the form

$$F(\Delta) = F(0) + A\Delta^p + O(\Delta^{p+1}) \tag{3}$$

$$F\left(\frac{\Delta}{2}\right) = F(0) + A\left(\frac{\Delta}{2}\right)^p + O(\Delta^{p+1}) \tag{4}$$

Now multiply the second line here by 2^p, subtract the first line from it, and do a little algebra to obtain[1]

$$F(0) = \frac{2^p F\left(\frac{\Delta}{2}\right) - F(\Delta)}{2^p - 1} + O(\Delta^{p+1}) \tag{5}$$

The point is that the error term proportional to Δ^p in (3) and (4) has *cancelled* out of the combination in (5), leaving us with an estimate of our quantity whose error decays more rapidly with Δ.

The first term on the LHS of (5) defines the Richardson-extrapolated version of our numerical method at convergence parameter Δ:

$$F^{\text{Richardson}}(\Delta) \equiv \frac{2^p F\left(\frac{\Delta}{2}\right) - F(\Delta)}{2^p - 1} \tag{6a}$$

or, written in terms of the parameter $N \propto \frac{1}{\Delta}$,

$$F^{\text{Richardson}}(N) \equiv \frac{2^p F(2N) - F(N)}{2^p - 1} \tag{6b}$$

If $F(\Delta)$ converges to the exact answer like Δ^p, then $F^{\text{Richardson}}(\Delta)$ converges to the exact answer like Δ^{p+1}. (But note that each invocation of $F^{\text{Richardson}}$ requires you to do $3N$ work, instead of the N work you need to do for F.)

In other words, to summarize the situation in symbols again,

$$\boxed{\underbrace{F(0)}_{\text{what we want}} = \underbrace{\frac{F(\Delta) - 2^p F\left(\frac{\Delta}{2}\right)}{1 - 2^p}}_{\text{what we can compute}} + \underbrace{O(\Delta^{p+1})}_{\text{dominant error term}}} \tag{7}$$

[1] If you are following along with the algebra at home, you will notice that the $O(\Delta^{p+1})$ term in equation (5) is a linear combination of the $O(\Delta^{p+1})$ terms in (3) and (4). The point is that any linear combination of two quantities that are each $O(\Delta^{p+1})$ yields a third quantity that is itself $O(\Delta^{p+1})$, no matter what coefficients we choose in the linear combination (as long as none of them depend on Δ). This is a feature of the $O(\cdot)$ notation: it completely ignores multiplicative coefficients and only keeps track of the leading Δ dependence.

The quantity labeled "what we can compute" in this equation is the Richardson-extrapolated version of our numerical method at convergence parameter Δ. Comparing this to equation (2), we see that we have effectively improved the rate of convergence of our numerical approximation scheme.

Terminology

In some cases, the application of Richardson extrapolation to an existing numerical method is assigned a new name, even though the underlying method is really the same. For example, the application of Richardson extrapolation to Newton-Cotes quadrature rules is called *Romberg integration*. On the other hand, in the world of ODE integrators the combination of Richardson extrapolation with the midpoint method (which you considered in PSet 3) is known as the *Bulirsch-Stoer* algorithm.