LECTURE 26: *Richardson extrapolation*

3.5 Richardson extrapolation, Romberg integration

Throughout numerical analysis, one encounters procedures that apply some simple approximation (e.g., linear interpolation) to construct some equally simple algorithm (e.g., differentiate the interpolant to get a finite difference formula (Section 1.7), integrate the interpolant to get the trapezoid rule (Section 3.2). An unfortunate consequence is that such approximations often converge slowly, with errors decaying only like h or h^2, where h is some discretization parameter (e.g., the spacing between interpolation points).

In the case of integration, you might prefer using a higher order method, like Clenshaw–Curtis or Gaussian quadrature. What we talk about here is an alternative to such approaches.

In this lecture we describe a remarkable, fundamental tool of classical numerical analysis. Like alchemists who sought to convert lead into gold, so we will take a sequence of slowly convergent data and extract from it a highly accurate estimate of our solution. This procedure is *Richardson extrapolation*, an essential but easily overlooked technique that should be part of every numerical analyst's toolbox. When applied to quadrature rules, the procedure is called *Romberg integration*.

We begin in a general setting: Suppose we seek some abstract quantity, $\xi \in \mathbb{R}$, which could be the value of a definite integral, a derivative, the solution to a differential equation at a certain point, or something else entirely. Further suppose we cannot compute ξ exactly; we can only access numerical approximations to it, generated by some function (an algorithm) Φ that depends upon a mesh parameter h. We compute $\Phi(h)$ for several values of h, expecting that $\Phi(h) \to \Phi(0) = \xi$ as $h \to 0$. To obtain good accuracy, one naturally seeks to evaluate Φ with increasingly smaller values of h. There are two reasons not to do so:

- Often Φ becomes increasingly expensive to evaluate as h shrinks;
- The numerical accuracy with which we can evaluate Φ may deteriorate as h gets small, due to rounding errors in floating point arithmetic. (For an example of the latter, try computing estimates of $f'(\alpha)$ using the formula $f'(\alpha) \approx (f(\alpha + h) - f(\alpha))/h$ as $h \to 0$.)

For example, computing $\Phi(h/2)$ often requires at least twice as much work as $\Phi(h)$. In some cases, $\Phi(h/2)$ could require 4, or even 8, times as much work at $\Phi(h)$, i.e., the expense of Φ could grow like $1/h$ or $1/h^2$ or $1/h^3$, etc.

Assume that Φ is infinitely continuously differentiable *as a function of* h, thus allowing us to expand $\Phi(h)$ in the Taylor series

$$\Phi(h) = \Phi(0) + h\Phi'(0) + \tfrac{1}{2}h^2\Phi''(0) + \tfrac{1}{6}h^3\Phi'''(0) + \cdots .$$

The derivatives here may seem to complicate matters (e.g., what are the derivatives of a quadrature rule with respect to h?), but we shall not need to compute them: they key is that the function Φ behaves

smoothly in h. Recalling that $\Phi(0) = \xi$, we can rewrite the Taylor series for $\Phi(h)$ as

$$\Phi(h) = \xi + c_1 h + c_2 h^2 + c_3 h^3 + \cdots$$

for some constants $\{c_j\}_{j=1}^{\infty}$. (For example, $c_1 = \Phi'(0)$.)

This expansion implies that taking $\Phi(h)$ as an approximation for ξ incurs an $\mathcal{O}(h)$ error. Halving the parameter h should roughly halve the error, according to the expansion

$$\Phi(h/2) = \xi + c_1 \tfrac{1}{2} h + c_2 \tfrac{1}{4} h^2 + c_3 \tfrac{1}{8} h^3 + \cdots .$$

Here comes the trick that is key to the whole lecture: Combine the expansions for $\Phi(h)$ and $\Phi(h/2)$ in such a way that eliminates the $\mathcal{O}(h)$ term. In particular, define

$$\begin{aligned}
\Psi(h) &:= 2\Phi(h/2) - \Phi(h) \\
&= 2\Big(\xi + c_1 \tfrac{1}{2} h + c_2 \tfrac{1}{4} h^2 + c_3 \tfrac{1}{8} h^3 + \cdots\Big) \\
&\qquad - \Big(\xi + c_1 h + c_2 h^2 + c_3 h^3 + \cdots\Big) \\
&= \xi - c_2 \tfrac{1}{2} h^2 - c_3 \tfrac{3}{4} h^3 + \cdots .
\end{aligned}$$

Thus, $\Psi(h)$ also approximates $\xi = \Psi(0) = \Phi(0)$, but with an $\mathcal{O}(h^2)$ error, rather than the $\mathcal{O}(h)$ error that pollutes $\Phi(h)$. For small h, this $\mathcal{O}(h^2)$ approximation will be considerably more accurate.

Why stop with $\Psi(h)$? Repeat the procedure, combining $\Psi(h)$ and $\Psi(h/2)$ to eliminate the $\mathcal{O}(h^2)$ term. Since

$$\Psi(h/2) = \xi - c_2 \tfrac{1}{8} h^2 - c_3 \tfrac{3}{32} h^3 + \cdots ,$$

we have

$$\Theta(h) := \frac{4\Psi(h/2) - \Psi(h)}{3} = \xi + c_3 \tfrac{1}{8} h^3 + \cdots .$$

To compute $\Theta(h)$, we must have access to both $\Psi(h)$ and $\Psi(h/2)$. These, in turn, require $\Phi(h)$, $\Phi(h/2)$, and $\Phi(h/4)$. In many cases, Φ becomes increasingly expensive to compute as the parameter h is reduced. Thus there is some practical limit to how small we can take h when evaluating $\Phi(h)$.

One could continue this procedure repeatedly, each time improving the accuracy by one order, at the cost of one additional Φ computation with a smaller h. To facilitate generalization and to avoid a further tangle of Greek characters, we adopt a new notation: Define

$$\begin{aligned}
R(j,0) &:= \Phi(h/2^j), && j \geq 0; \\
R(j,k) &:= \frac{2^k R(j,k-1) - R(j-1,k-1)}{2^k - 1}, && j \geq k > 0.
\end{aligned}$$

For the sake of clarity let us discuss a concrete case, elaborated upon in Example 3.2 below. Suppose we wish to compute $\xi = f'(\alpha)$ using the finite difference formula

$$\Phi(h) = \frac{f(\alpha + h) - f(\alpha)}{h}.$$

The quotient rule gives

$$\Phi'(h) = \frac{f(\alpha) - f(\alpha + h)}{h^2} + \frac{f'(\alpha + h)}{h},$$

which will depend smoothly on h provided f is smooth near α. In particular, a Taylor expansion for f gives

$$f(\alpha + h) = f(\alpha) + h f'(\alpha) + \tfrac{1}{2} h^2 f''(\alpha) + \tfrac{1}{6} h^3 f'''(\eta)$$

for some $\eta \in [\alpha, \alpha + h]$. Substitute this formula into the equation for $\Phi'(h)$ and simplify to get

$$\Phi'(h) = \frac{f'(\alpha + h) - f'(\alpha)}{h} - \tfrac{1}{2} f''(\alpha) - \tfrac{1}{6} h f'''(\eta).$$

Now this expression leads to a clean formula for the first coefficient of the Taylor series for $\Phi(h)$:

$$c_1 = \Phi'(0) = \lim_{h \to 0} \Phi'(h) = \tfrac{1}{2} f''(\alpha).$$

The moral of the example: while it might seem strange to take a Taylor series of the "algorithm" Φ, the quantities involved often have a very natural interpretation in terms of the underlying problem at hand.

Thus: $R(0,0) = \Phi(h)$, $R(1,0) = \Phi(h/2)$, and $R(1,1) = \Psi(h)$. This procedure is called *Richardson extrapolation* after the British applied mathematician Lewis Fry Richardson, a pioneer of the numerical solution of partial differential equations, weather modeling, and mathematical models in political science. The numbers $R(j,k)$ are arranged in a triangular *extrapolation table*:

$$\begin{array}{llllll}
R(0,0) & & & & & \\
R(1,0) & R(1,1) & & & & \\
R(2,0) & R(2,1) & R(2,2) & & & \\
R(3,0) & R(3,1) & R(3,2) & R(3,3) & & \\
\vdots & \vdots & \vdots & \vdots & \ddots & \\
\cdots & \cdots & \cdots & \cdots & \cdots & . \\
\uparrow & \uparrow & \uparrow & \uparrow & & \\
\mathcal{O}(h) & \mathcal{O}(h^2) & \mathcal{O}(h^3) & \mathcal{O}(h^4) & &
\end{array}$$

To compute any given element in the table, one must first determine entries above and to the left. Note that only the first column will require significant work; the subsequent columns follow from easy arithmetic. The theory suggests that the bottom-right element in the table will be the most accurate approximation to ξ. Indeed this bottom-right entry will generally be the most accurate, provided the assumption that Φ is infinitely continuously differentiable holds. When floating point roundoff errors spoil what otherwise would have been an infinitely continuously differentiable procedure, the bottom-right entry will suffer acutely from this pollution. Such errors will be apparent in the forthcoming example.

Example 3.2 (Finite difference approximation of the first derivative)**.** We seek $\xi = f'(\alpha)$ for some function continuously differentiable function f. Recall from Section 1.7 the simple finite difference approximation to the first derivative that follows from differentiating the linear interpolant to f through the points $x = \alpha$ and $x = \alpha + h$:

$$f'(\alpha) \approx \frac{f(\alpha + h) - f(\alpha)}{h}.$$

In fact, in Theorem 1.6 we quantified the error to be $\mathcal{O}(h)$ as $h \to 0$:

$$f'(\alpha) = \frac{f(\alpha + h) - f(\alpha)}{h} + \mathcal{O}(h).$$

Thus we define

$$\Phi(h) = \frac{f(\alpha + h) - f(\alpha)}{h}.$$

As a simple test problem, take $f(x) = \mathrm{e}^x$. We will use Φ and Richardson extrapolation to approximate $f'(1) = \mathrm{e} = 2.7182818284\ldots$.

The simple finite difference method produces crude answers:

h	$\Phi(h)$	error
1	4.670774270	1.95249×10^{0}
1/2	3.526814484	8.08533×10^{-1}
1/4	3.088244516	3.69963×10^{-1}
1/8	2.895480164	1.77198×10^{-1}
1/16	2.805025851	8.67440×10^{-2}
1/32	2.761200889	4.29191×10^{-2}
1/64	2.739629446	2.13476×10^{-2}
1/128	2.728927823	1.06460×10^{-2}
1/256	2.723597892	5.31606×10^{-3}
1/512	2.720938130	2.65630×10^{-3}

Even with $h = 1/512 = 0.00195\ldots$ we fail to approximate $f'(1)$ to even three correct digits. As we take h smaller and smaller, finite precision arithmetic eventually causes unacceptable errors; Figure 3.9 shows the error in $\Phi(h)$ as $h \to 0$. (The red line shows what perfect $\mathcal{O}(h)$ convergence would look like.)

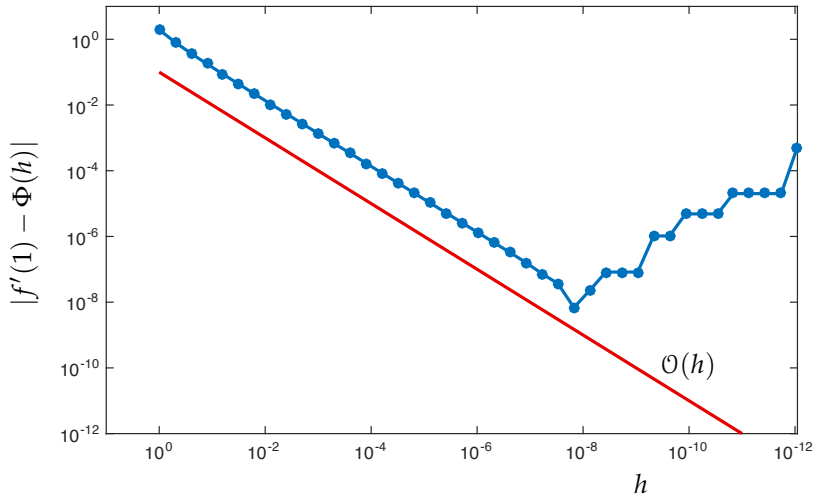

Figure 3.9: Linear convergence of the estimate $\Phi(h)$ to $f'(1)$ (blue line). As h gets small, rounding errors spoil the $\mathcal{O}(h)$ convergence (red line). An accuracy of about 10^{-8} seems to be the best we can do for this method and this problem.

A few steps of Richardson extrapolation on the data in the table above reveals greatly improved solutions, five correct digits in $R(4,4)$:

j	$R(j,0)$	$R(j,1)$	$R(j,2)$	$R(j,3)$	$R(j,4)$
0	4.67077427047160				
1	3.52681448375804	2.38285469704447			
2	3.08824451601118	2.64967454826433	2.73861449867095		
3	2.89548016367188	2.70271581133258	2.72039623235534	2.71779362288168	
4	2.80502585140344	2.71457153913500	2.71852344840247	2.71825590783778	2.71828672683485

The good performance of this method depends on f having sufficiently many smooth derivatives. If higher derivatives are not

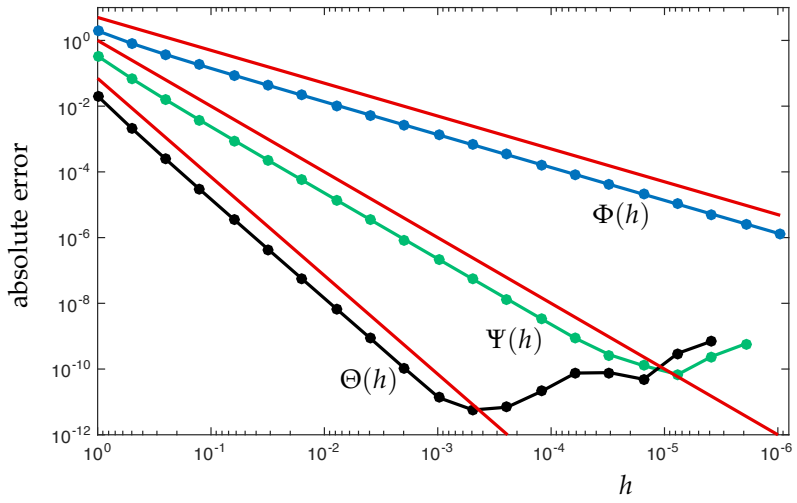

Figure 3.10: The convergence of $\Phi(h)$ (blue) along with its first two Richardson refinements, $\Psi(h)$ (green) and $\Theta(h)$ (black). The red lines show $\mathcal{O}(h)$, $\mathcal{O}(h^2)$, and $\mathcal{O}(h^3)$ convergence. For these values of h, rounding errors are not apparent in the $\Phi(h)$ plot; however, they lurk in the later digits of $\Phi(h)$, enough to interfere with the $\Psi(h)$ and $\Theta(h)$ approximations. Before these errors take hold, $\Theta(h)$ gives several additional orders of magnitude accuracy than was obtained by $\Phi(h)$ with much smaller h, in Figure 3.9.

smooth, then $\Phi(h)$ will not have smooth derivatives, and the accuracy breaks down. The accuracy also eventually degrades because of rounding errors that subtly pollute the initial column of data, as shown in the Figure 3.10.

3.5.1 *Extrapolation for higher order approximations*

In many cases, the initial algorithm $\Phi(h)$ is better than $\mathcal{O}(h)$ accurate, and in this case the formula for $R(j,k)$ should be adjusted to take advantage. Suppose that

$$\Phi(h) = \xi + c_1 h^r + c_2 h^{2r} + c_3 h^{3r} + \cdots$$

Notice that this structure is rather special: for example, if $r = 2$, then the Taylor series for $\Phi(h)$ must avoid all *odd-order* terms.

for some integer $r \geq 1$. Then define

$$R(j,0) := \Phi(h/2^j) \qquad \text{for } j \geq 0$$

$$R(j,k) := \frac{2^{rk} R(j,k-1) - R(j-1,k-1)}{2^{rk} - 1} \qquad \text{for } j \geq k > 0. \tag{3.3}$$

In this case, the $R(:,k)$ column will be $\mathcal{O}(h^{(k+1)r})$ accurate.

3.5.2 *Extrapolating the composite trapezoid rule: Romberg integration*

Suppose $f \in C^\infty[a,b]$, and we wish to approximate $\int_a^b f(x)\,\mathrm{d}x$ with the composite trapezoid rule,

$$T(h) = \frac{h}{2}\Big[f(a) + 2\sum_{j=1}^{n-1} f(a+jh) + f(b)\Big].$$

Notice that $T(h)$ only makes sense (as the composite trapezoid rule) when $h = (b-a)/n$ for some integer n. Notice that $T((b-a)/n)$

If you find this restriction on h distracting, just define $T(h)$ to be a sufficiently smooth interpolation between the values of $T((b-a)/n)$ for $n = 1, 2, \ldots$.

requires $n+1$ evaluations of the function f, and so increasing n (decreasing h) increases the expense.

One can show that for any $f \in C^\infty[a,b]$,

$$T(h) = \int_a^b f(x)\,\mathrm{d}x + c_1 h^2 + c_2 h^4 + c_3 h^6 + \cdots .$$

Now perform the generalized Richardson extrapolation (3.3) on $T(h)$ with $r = 2$:

$$R(j,0) = T(h/2^j) \qquad \text{for } j \geq 0$$

$$R(j,k) = \frac{4^k R(j,k-1) - R(j-1,k-1)}{4^k - 1} \qquad \text{for } j \geq k > 0.$$

This procedure is called *Romberg integration*.

In cases where $f \in C^\infty[a,b]$ (or if f has many continuous derivatives), the Romberg table will converge to high accuracy, though it may be necessary to take h to be relatively small before this is observed. When f does not have many continuous derivatives, each column of the Romberg table will still converge to the true integral, but not at the ever-improving clip we expect for smoother functions.

This procedure's utility is best appreciated through an example.

Example 3.3. For purposes of demonstration, we should use an integral we know exactly, say

$$\int_0^\pi \sin(x)\,\mathrm{d}x = 2.$$

Start the table with $h = \pi$ to generate $R(0,0)$, requiring 2 evaluations of $f(x)$. To build out the table, compute the composite trapezoid approximation based on an increasing number of function evaluations at each step. The final entry in the first column requires 129 function evaluations, and has four digits correct. This may not seem particularly impressive, but after refining these computations through a few steps of Romberg integration, we have an approximation that is accurate to full precision.

Ideally, one would exploit the fact that some grid points used to compute $T(h)$ are also required for $T(h/2)$, etc., thus limiting the number of new function evaluations required at each step.

j	$R(j,0)$	$R(j,1)$	$R(j,2)$	$R(j,3)$	$R(j,4)$	$R(j,5)$	$R(j,6)$
0	0.000000000000						
1	1.570796326795	2.094395102393					
2	1.896118897937	2.004559754984	1.998570731824				
3	1.974231601946	2.000269169948	1.999983130946	2.000005549980			
4	1.993570343772	2.000016591048	1.999999752455	2.000000016288	1.999999994587		
5	1.998393360970	2.000001033369	1.999999996191	2.000000000060	1.999999999996	2.000000000001	
6	1.999598388640	2.000000064530	1.999999999941	2.000000000000	2.000000000000	2.000000000000	2.000000000000

Be warned that Romberg results are not always as clean as this example, but this procedure is important tool to have at hand when high precision integrals are required. The general strategy of Richardson extrapolation can be applied to great effect in a wide variety of numerical settings.