4.3 Estimating Derivatives and Richardson Extrapolation

A numerical experiment outlined in Chapter 1 (at the end of Section 1.1, p. 10) showed that determining the derivative of a function f at a point x is not a trivial numerical problem. Specifically, if $f(x)$ can be computed with only n digits of precision, it is difficult to calculate $f'(x)$ numerically with n digits of precision. This difficulty can be traced to the subtraction between quantities that are nearly equal. In this section, several alternatives are offered for the numerical computation of $f'(x)$ and $f''(x)$.

First-Derivative Formulas via Taylor Series

First, consider again the obvious method based on the definition of $f'(x)$. It consists of selecting one or more small values of h and writing

$$f'(x) \approx \frac{1}{h}[f(x+h) - f(x)] \tag{1}$$

What error is involved in this formula? To find out, use Taylor's Theorem from Section 1.2:

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2 f''(\xi)$$

Rearranging this equation gives

$$f'(x) = \frac{1}{h}[f(x+h) - f(x)] - \frac{1}{2}hf''(\xi) \tag{2}$$

Hence, we see that approximation (1) has error term $-\frac{1}{2}hf''(\xi) = \mathcal{O}(h)$, where ξ is in the interval having endpoints x and $x+h$.

Equation (2) shows that in general, as $h \to 0$, the difference between $f'(x)$ and the estimate $h^{-1}[f(x+h) - f(x)]$ approaches zero at the same rate that h does—that is, $\mathcal{O}(h)$. Of course, if $f''(x) = 0$, then the error term will be $\frac{1}{6}h^2 f'''(\gamma)$, which converges to zero somewhat faster at $\mathcal{O}(h^2)$. But usually, $f''(x)$ is not zero.

Equation (2) gives the **truncation error** for this numerical procedure, namely, $-\frac{1}{2}hf''(\xi)$. This error is present even if the calculations are performed with *infinite* precision; it is due to our imitating the mathematical limit process by means of an approximation formula. Additional (and worse) errors must be expected when calculations are performed on a computer with finite word length.

EXAMPLE 1 In Section 1.1, the program named *First* used the one-sided rule (1) to approximate the first derivative of the function $f(x) = \sin x$ at $x = 0.5$. Explain what happens when a large number of iterations are performed, say $n = 50$.

Solution There is a total loss of all significant digits! When we examine the computer output closely, we find that, in fact, a good approximation $f'(0.5) \approx 0.87758$ was found, but it deteriorated as the process continued. This was caused by the subtraction of two nearly equal quantities $f(x+h)$ and $f(x)$, resulting in a loss of significant digits as well as a magnification of this effect from dividing by a small value of h. We need to stop the iterations sooner! When to stop an iterative process is a common question in numerical algorithms. In this case, one can monitor the iterations to determine when they settle down, namely, when two successive ones are within a prescribed tolerance. Alternatively, we can use the truncation error term. If we want six significant digits of accuracy in the results, we set

$$\left| -\frac{1}{2}hf''(\xi) \right| \leqq \frac{1}{2}4^{-n} < \frac{1}{2}10^{-6}$$

since $|f''(x)| < 1$ and $h = 1/4^n$. We find $n > 6/\log 4 \approx 9.97$. So we should stop after about ten steps in the process. (The least error of 3.1×10^{-9} was found at iteration 14.) ■

As we saw in Newton's method (Chapter 3) and will see in the Romberg method (Chapter 5), it is advantageous to have the convergence of numerical processes occur with higher powers of some quantity approaching zero. In the present situation, we want an approximation to $f'(x)$ in which the error behaves like $\mathcal{O}(h^2)$. One such method is easily obtained with the aid of the following two Taylor series:

$$\begin{cases} f(x+h) = f(x) + hf'(x) + \frac{1}{2!}h^2 f''(x) + \frac{1}{3!}h^3 f'''(x) + \frac{1}{4!}h^4 f^{(4)}(x) + \cdots \\ f(x-h) = f(x) - hf'(x) + \frac{1}{2!}h^2 f''(x) - \frac{1}{3!}h^3 f'''(x) + \frac{1}{4!}h^4 f^{(4)}(x) - \cdots \end{cases} \tag{3}$$

By subtraction, we obtain

$$f(x+h) - f(x-h) = 2hf'(x) + \frac{2}{3!}h^3 f'''(x) + \frac{2}{5!}h^5 f^{(5)}(x) + \cdots$$

This leads to a very important formula for approximating $f'(x)$:

$$f'(x) = \frac{1}{2h}[f(x+h) - f(x-h)] - \frac{h^2}{3!}f'''(x) - \frac{h^4}{5!}f^{(5)}(x) - \cdots \tag{4}$$

Expressed otherwise,

$$f'(x) \approx \frac{1}{2h}[f(x+h) - f(x-h)] \tag{5}$$

with an error whose leading term is $-\frac{1}{6}h^2 f'''(x)$, which makes it $\mathcal{O}(h^2)$.

By using Taylor's Theorem with its error term, we could have obtained the following two expressions:

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2 f''(x) + \frac{1}{6}h^3 f'''(\xi_1)$$
$$f(x-h) = f(x) - hf'(x) + \frac{1}{2}h^2 f''(x) - \frac{1}{6}h^3 f'''(\xi_2)$$

Then the subtraction would lead to

$$f'(x) = \frac{1}{2h}[f(x+h) - f(x-h)] - \frac{1}{6}h^2\left[\frac{f'''(\xi_1) + f'''(\xi_2)}{2}\right]$$

The error term here can be simplified by the following reasoning: The expression $\frac{1}{2}[f'''(\xi_1)+f'''(\xi_2)]$ is the average of two values of f''' on the interval $[x-h, x+h]$. It therefore lies between the least and greatest values of f''' on this interval. If f''' is continuous on this interval, then this average value is assumed at some point ξ. Hence, the formula with its error term can be written as

$$f'(x) = \frac{1}{2h}[f(x+h) - f(x-h)] - \frac{1}{6}h^2 f'''(\xi)$$

This is based on the sole assumption that f''' is continuous on $[x-h, x+h]$. This formula for numerical differentiation turns out to be very useful in the numerical solution of certain differential equations, as we shall see in Chapter 14 (on boundary value problems) and Chapter 15 (on partial differential equations).

EXAMPLE 2 Modify program *First* in Section 1.1 so that it uses the central difference formula (5) to approximate the first derivative of the function $f(x) = \sin x$ at $x = 0.5$.

Solution Using the truncation error term for the central difference formula (5), we set

$$\left|-\frac{1}{6}h^2 f'''(\xi)\right| \leqq \frac{1}{6}4^{-2n} < \frac{1}{2}10^{-6}$$

or $n > (6 - \log 3)/\log 16 \approx 4.59$. We obtain a good approximation after about five iterations with this higher-order formula. (The least error of 3.6×10^{-12} was at step 9.) ■

Richardson Extrapolation

Returning now to Equation (4), we write it in a simpler form:

$$f'(x) = \frac{1}{2h}[f(x+h) - f(x-h)] + a_2h^2 + a_4h^4 + a_6h^6 + \cdots \tag{6}$$

in which the constants $a_2, a_4, \ldots$ depend on f and x. When such information is available about a numerical process, it is possible to use a powerful technique known as *Richardson extrapolation* to wring more accuracy out of the method. This procedure is now explained, using Equation (6) as our model.

Holding f and x fixed, we define a function of h by the formula

$$\varphi(h) = \frac{1}{2h}[f(x+h) - f(x-h)] \tag{7}$$

From Equation (6), we see that $\varphi(h)$ is an approximation to $f'(x)$ with error of order $\mathcal{O}(h^2)$. Our objective is to compute $\lim_{h\to 0}\varphi(h)$ because this is the quantity $f'(x)$ that we wanted

in the first place. If we select a function f and plot $\varphi(h)$ for $h = 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots$, then we get a graph (Computer Problem 4.3.5). Near zero, where we cannot actually calculate the value of φ from Equation (7), φ is approximately a quadratic function of h, since the higher-order terms from Equation (6) are negligible. Richardson extrapolation seeks to estimate the limiting value at 0 from some computed values of $\varphi(h)$ near 0. Obviously, we can take any convenient sequence h_n that converges to zero, calculate $\varphi(h_n)$ from Equation (7), and use these as approximations to $f'(x)$.

But something much more clever can be done. Suppose we compute $\varphi(h)$ for some h and then compute $\varphi(h/2)$. By Equation (6), we have

$$\varphi(h) = f'(x) - a_2h^2 - a_4h^4 - a_6h^6 - \cdots$$
$$\varphi\left(\frac{h}{2}\right) = f'(x) - a_2\left(\frac{h}{2}\right)^2 - a_4\left(\frac{h}{2}\right)^4 - a_6\left(\frac{h}{2}\right)^6 - \cdots$$

We can eliminate the dominant term in the error series by simple algebra. To do so, multiply the second equation by 4 and subtract it from the first equation. The result is

$$\varphi(h) - 4\varphi\left(\frac{h}{2}\right) = -3f'(x) - \frac{3}{4}a_4h^4 - \frac{15}{16}a_6h^6 - \cdots$$

We divide by -3 and rearrange this to get

$$\varphi\left(\frac{h}{2}\right) + \frac{1}{3}\left[\varphi\left(\frac{h}{2}\right) - \varphi(h)\right] = f'(x) + \frac{1}{4}a_4h^4 + \frac{5}{16}a_6h^6 + \cdots$$

This is a marvelous discovery. Simply by adding $\frac{1}{3}[\varphi(h/2) - \varphi(h)]$ to $\varphi(h/2)$, we have apparently improved the precision to $\mathcal{O}(h^4)$ because the error series that accompanies this new combination begins with $\frac{1}{4}a_4h^4$. Since h will be small, this is a dramatic improvement.

We can repeat this process by letting

$$\Phi(h) = \frac{4}{3}\varphi\left(\frac{h}{2}\right) - \frac{1}{3}\varphi(h)$$

Then we have from the previous derivation that

$$\Phi(h) = f'(x) + b_4h^4 + b_6h^6 + \cdots$$
$$\Phi\left(\frac{h}{2}\right) = f'(x) + b_4\left(\frac{h}{2}\right)^4 + b_6\left(\frac{h}{2}\right)^6 + \cdots$$

We can combine these equations to eliminate the first term in the error series

$$\Phi(h) - 16\Phi\left(\frac{h}{2}\right) = -15f'(x) + \frac{3}{4}b_6h^6 + \cdots$$

Hence, we have

$$\Phi\left(\frac{h}{2}\right) + \frac{1}{15}\left[\Phi\left(\frac{h}{2}\right) - \Phi(h)\right] = f'(x) - \frac{1}{20}b_6h^5 + \cdots$$

This is yet another apparent improvement in the precision to $\mathcal{O}(h^6)$. And now, to top it off, note that the same procedure can be repeated over and over again to *kill* higher and higher terms in the error. This is **Richardson extrapolation**.

Essentially the same situation arises in the derivation of Romberg's algorithm in Chapter 5. Therefore, it is desirable to have a general discussion of the procedure here. We start with an equation that includes both situations. Let φ be a function such that

$$\varphi(h) = L - \sum_{k=1}^{\infty} a_{2k} h^{2k} \tag{8}$$

where the coefficients a_{2k} are not known. Equation (8) is not interpreted as the *definition* of φ but rather as a *property* that φ possesses. It is assumed that $\varphi(h)$ can be computed for any $h > 0$ and that our objective is to approximate L accurately using φ.

Select a convenient h, and compute the numbers

$$D(n, 0) = \varphi\left(\frac{h}{2^n}\right) \qquad (n \geq 0) \tag{9}$$

Because of Equation (8), we have

$$D(n, 0) = L + \sum_{k=1}^{\infty} A(k, 0)\left(\frac{h}{2^n}\right)^{2k}$$

where $A(k, 0) = -a_{2k}$. These quantities $D(n, 0)$ give a crude estimate of the unknown number $L = \lim_{x \to 0} \varphi(x)$. More accurate estimates are obtained via Richardson extrapolation. The extrapolation formula is

$$D(n, m) = \frac{4^m}{4^m - 1} D(n, m-1) - \frac{1}{4^m - 1} D(n-1, m-1) \qquad (1 \leq m \leq n) \tag{10}$$

THEOREM 1 **RICHARDSON EXTRAPOLATION THEOREM**

The quantities $D(n, m)$ defined in the Richardson extrapolation process (10) obey the equation

$$D(n, m) = L + \sum_{k=m+1}^{\infty} A(k, m)\left(\frac{h}{2^n}\right)^{2k} \qquad (0 \leq m \leq n) \tag{11}$$

Proof Equation (11) is true by hypothesis if $m = 0$. For the purpose of an inductive proof, we *assume* that Equation (11) is valid for an arbitrary value of $m-1$, and we prove that Equation (11) is then valid for m. Now from Equations (10) and (11) for a fixed value m, we have

$$\begin{aligned} D(n, m) = {} & \frac{4^m}{4^m - 1}\left[L + \sum_{k=m}^{\infty} A(k, m-1)\left(\frac{h}{2^n}\right)^{2k}\right] \\ & - \frac{1}{4^m - 1}\left[L + \sum_{k=m}^{\infty} A(k, m-1)\left(\frac{h}{2^{n-1}}\right)^{2k}\right] \end{aligned}$$

After simplification, this becomes

$$D(n, m) = L + \sum_{k=m}^{\infty} A(k, m-1)\left(\frac{4^m - 4^k}{4^m - 1}\right)\left(\frac{h}{2^n}\right)^{2k} \tag{12}$$

Thus, we are led to define

$$A(k,m) = A(k,m-1)\left(\frac{4^m - 4^k}{4^m - 1}\right)$$

At the same time, we notice that $A(m,m) = 0$. Hence, Equation (12) can be written as

$$D(n,m) = L + \sum_{k=m+1}^{\infty} A(k,m)\left(\frac{h}{2^n}\right)^{2k}$$

Equation (11) is true for m, and the induction is complete. ■

The significance of Equation (11) is that the summation *begins* with the term $(h/2^n)^{2m+2}$. Since $h/2^n$ is small, this indicates that the numbers $D(n,m)$ are approaching L very rapidly, namely,

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

In practice, one can arrange the quantities in a two-dimensional triangular array as follows:

$$\begin{array}{llll} D(0,0) & & & \\ D(1,0) & D(1,1) & & \\ D(2,0) & D(2,1) & D(2,2) & \\ \vdots & \vdots & \vdots & \ddots \\ D(N,0) & D(N,1) & D(N,2) & \cdots \quad D(N,N) \end{array} \tag{13}$$

The main tasks to generate such an array are as follows:

■ **ALGORITHM 2** ***Richardson Extrapolation***

1. Write a function for φ.
2. Decide on suitable values for N and h.
3. For $i = 0, 1, \ldots, N$, compute $D(i,0) = \varphi(h/2^i)$.
4. For $0 \leqq i \leqq j \leqq N$, compute
$$D(i,j) = D(i,j-1) + (4^j - 1)^{-1}[D(i,j-1) - D(i-1,j-1)]$$

Notice that in this algorithm, the computation of $D(i,j)$ follows Equation (10) but has been rearranged slightly to improve its numerical properties.

EXAMPLE 3 Write a procedure to compute the derivative of a function at a point by using Equation (5) and Richardson extrapolation.

Solution The input to the procedure will be a function f, a specific point x, a value of h, and a number n signifying how many rows in the array (13) are to be computed. The output will

be the array (13). Here is a suitable pseudocode:

```
procedure Derivative(f, x, n, h, (d_ij))
integer i, j, n;   real h, x;   real array (d_ij)_{0:n×0:n}
external function f
for i = 0 to n do
    d_i0 ← [f(x + h) − f(x − h)]/(2h)
    for j = 1 to i do
        d_{i,j} ← d_{i,j−1} + (d_{i,j−1} − d_{i−1,j−1})/(4^j − 1)
    end for
    h ← h/2
end for
end procedure Derivative
```

To test the procedure, choose $f(x) = \sin x$, where $x_0 = 1.23095\,94154$ and $h = 1$. Then $f'(x) = \cos x$ and $f'(x_0) = \frac{1}{3}$. A pseudocode is written as follows:

```
program Test_Derivative
real array (d_ij)_{0:n×0:n};   external function f
integer n ← 10;   real h ← 1;   x ← 1.23095 94154
call Derivative(f, x, n, h, (d_ij))
output (d_ij)
end program Test_Derivative

real function f(x)
real x
f ← sin(x)
end function f
```

We invite the reader to program the pseudocode and execute it on a computer. The computer output is the triangular array (d_{ij}) with indices $0 \leqq j \leqq i \leqq 10$. The most accurate value is $(d_{4,1}) = 0.33333\,33433$. The values d_{i0}, which are obtained solely by Equations (7) and (9) without any extrapolation, are not as accurate, having no more than four correct digits. ■

Mathematical software is now available with algebraic manipulation capabilities. Using them, we could write a computer program to find derivatives symbolically for a rather large class of functions—probably all those you would encounter in a calculus course. For example, we could verify the numerical results above by first finding the derivative exactly and then evaluating the numerical answer $\cos(1.23095\,94154) \approx 0.33333\,33355$ since $\arccos\left(\frac{1}{3}\right) \approx 1.23095\,941543$. Of course, the procedures discussed in this section are for approximating derivatives that cannot be determined exactly.

5.3 Romberg Algorithm

Description

The *Romberg algorithm* produces a triangular array of numbers, all of which are numerical estimates of the definite integral $\int_a^b f(x)\,dx$. The array is denoted here by the notation

$$\begin{array}{llllll} R(0,0) & & & & & \\ R(1,0) & R(1,1) & & & & \\ R(2,0) & R(2,1) & R(2,2) & & & \\ R(3,0) & R(3,1) & R(3,2) & R(3,3) & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ R(n,0) & R(n,1) & R(n,2) & R(n,3) & \cdots & R(n,n) \end{array}$$

The first column of this table contains estimates of the integral obtained by the recursive trapezoid formula with decreasing values of the step size. Explicitly, $R(n, 0)$ is the result of applying the trapezoid rule with 2^n equal subintervals. The first of them, $R(0, 0)$, is obtained with just one trapezoid:

$$R(0,0) = \tfrac{1}{2}(b-a)[f(a) + f(b)]$$

Similarly, $R(1, 0)$ is obtained with two trapezoids:

$$\begin{aligned} R(1,0) &= \frac{1}{4}(b-a)\left[f(a)+f\left(\frac{a+b}{2}\right)\right]+\frac{1}{4}(b-a)\left[f\left(\frac{a+b}{2}\right)+f(b)\right] \\ &= \frac{1}{4}(b-a)[f(a)+f(b)]+\frac{1}{2}(b-a)\ f\left(\frac{a+b}{2}\right) \\ &= \frac{1}{2}R(0,0)+\frac{1}{2}(b-a)\ f\left(\frac{a+b}{2}\right) \end{aligned}$$

These formulas agree with those developed in the preceding section. In particular, note that $R(n, 0)$ is obtained easily from $R(n-1, 0)$ if Equation (10) in Section 5.2 is used; that is,

$$R(n,0)=\frac{1}{2}R(n-1,0)+h\sum_{k=1}^{2^{n-1}} f[a+(2k-1)h] \tag{1}$$

where $h = (b-a)/2^n$ and $n \geqq 1$.

The second and successive columns in the Romberg array are generated by the extrapolation formula

$$R(n,m)=R(n,m-1)+\frac{1}{4^m-1}[R(n,m-1)-R(n-1,m-1)] \tag{2}$$

with $n \geqq 1$ and $m \geqq 1$. This formula will be derived later using the theory of Richardson extrapolation from Section 4.3.

EXAMPLE 1 If $R(4, 2) = 8$ and $R(3, 2) = 1$, what is $R(4, 3)$?

Solution From Equation (2), we have

$$\begin{aligned} R(4,3) &= R(4,2)+\tfrac{1}{63}[R(4,2)-R(3,2)] \\ &= 8+\tfrac{1}{63}(8-1)=\tfrac{73}{9} \end{aligned}$$

■

Pseudocode

The objective now is to develop computational formulas for the **Romberg algorithm**. By replacing n with i and m with j in Equation (2), we obtain, for $i \geqq 1$ and $j \geqq 1$,

$$R(i,j)=R(i,j-1)+\frac{1}{4^j-1}[R(i,j-1)-R(i-1,j-1)]$$

and

$$R(i,0)=\frac{1}{2}R(i-1,0)+h\sum_{k=1}^{2^{i-1}} f[a+(2k-1)h]$$

The range of the summation is $1 \leqq k \leqq 2^{i-1}$, so that $1 \leqq 2k-1 \leqq 2^i-1$.

One way to generate the Romberg array is to compute a reasonable number of terms in the first column, $R(0, 0)$ up to $R(n, 0)$, and then use the extrapolation Formula (2) to construct columns $1, 2, \ldots, n$ in order. Another way is to compute the array row by row. Observe, for example, that $R(1, 1)$ can be computed by the extrapolation formula as soon as $R(1, 0)$ and $R(0, 0)$ are available. The procedure *Romberg* computes, row by row, n rows

and columns of the Romberg array for a function f and a specified interval $[a, b]$:

```
procedure Romberg(f, a, b, n, (r_ij))
integer i, j, k, n;  real a, b, h, sum;  real array (r_ij)_{0:n×0:n}
external function f
h ← b − a
r_00 ← (h/2)[f(a) + f(b)]
for i = 1 to n do
    h ← h/2
    sum ← 0
    for k = 1 to 2^i − 1 step 2 do
        sum ← sum + f(a + kh)
    end for
    r_i0 ← ½ r_{i−1,0} + (sum)h
    for j = 1 to i do
        r_ij ← r_{i,j−1} + (r_{i,j−1} − r_{i−1,j−1})/(4^j − 1)
    end for
end for
end procedure Romberg
```

This procedure is used with a main program and a function procedure (for computing values of the function f). In the main program and perhaps in the procedure *Romberg*, some language-specific interface must be included to indicate that the first argument is an external function. Remember that in the Romberg algorithm as described, the number of subintervals is 2^n. Thus, a modest value of n should be chosen—for example, $n = 5$. A more sophisticated program would include automatic tests to terminate the calculation as soon as the error reaches a preassigned tolerance.

As an example, one can approximate π by using the procedure *Romberg* with $n = 5$ to obtain a numerical approximation for the integral

$$\int_0^1 \frac{4}{1+x^2}\,dx$$

We obtain the following results:

3.00000 00000 000
3.09999 99046 326 3.13333 32061 768
3.13117 64717 102 3.14156 86607 361 3.14211 77387 238
3.13898 84948 730 3.14159 25025 940 3.14159 41715 240 3.14158 58268 738
3.14094 16198 730 3.14159 27410 126 3.14159 27410 126 3.14159 27410 126 3.14159 27410 126

Euler-Maclaurin Formula

Here we explain the source of Equation (2), which is used for constructing the successive columns of the Romberg array. We begin with a formula that expresses the error in the trapezoid rule over 2^{n-1} subintervals:

$$\int_a^b f(x)\,dx = R(n-1, 0) + a_2h^2 + a_4h^4 + a_6h^6 + \cdots \tag{3}$$

Here, $h = (b-a)/2^{n-1}$ and the coefficients a_i depend on f but not on h. This equation is one form of the **Euler-Maclaurin formula** and is given here without proof. (See Young and Gregory [1972].) In this equation, $R(n-1, 0)$ denotes a typical element of the first column in the Romberg array; hence, it is one of the trapezoidal estimates of the integral. Notice particularly that the error is expressed in powers of h^2, and the error series is $\mathcal{O}(h^2)$. For our purposes, it is not necessary to know the coefficients, but, in fact, they have definite expressions in terms of f and its derivatives. For the theory to work smoothly, it is assumed that f possesses derivatives of all orders on the interval $[a, b]$.

The reader should now recall the theory of Richardson extrapolation as outlined in Section 4.3. That theory is applicable because of Equation (3). In Equation (8) of Section 4.3, $L = \phi(h) + \sum_{k=1}^{\infty} a_{2k} h^{2k}$. Here, L is the value of the integral and $\phi(h)$ is $R(n-1, 0)$, the trapezoidal estimate of L using subintervals of size h. Equation (10) of Section 4.3 gives the approximate extrapolation formula, which in this situation is Equation (2).

We briefly review this procedure. Replacing n with $n+1$ and h with $h/2$ in Equation (3), we have

$$\int_a^b f(x)\,dx = R(n, 0) + \frac{1}{4}a_2h^2 + \frac{1}{16}a_4h^4 + \frac{1}{64}a_6h^6 + \cdots \tag{4}$$

Subtract Equation (3) from 4 times Equation (4) to obtain

$$\int_a^b f(x)\,dx = R(n, 1) - \frac{1}{4}a_4h^4 - \frac{5}{16}a_6h^6 - \cdots \tag{5}$$

where

$$R(n, 1) = R(n, 0) + \tfrac{1}{3}[R(n, 0) - R(n-1, 0)] \qquad (n \geqq 1)$$

Note that this is the first case ($m = 1$) of the extrapolation Formula (2). Now $R(n, 1)$ should be considerably more accurate than $R(n, 0)$ or $R(n-1, 0)$ because its error formula begins with an h^4 term. Hence, the error series is now $\mathcal{O}(h^4)$. This process can be repeated using Equation (5) slightly modified as the starting point—that is, with n replaced by $n-1$ and with h replaced by $2h$. Then combine the two equations appropriately to eliminate the h^4 term. The result is a new combination of elements from column 2 in the Romberg array:

$$\int_a^b f(x)\,dx = R(n, 2) + \frac{1}{4^3}a_6h^6 + \frac{21}{4^5}a_8h^8 + \cdots \tag{6}$$

where

$$R(n, 2) = R(n, 1) + \tfrac{1}{15}[R(n, 1) - R(n-1, 1)] \qquad (n \geqq 2)$$

which agrees with Equation (2) when $m = 2$. Thus, $R(n, 2)$ is an even more accurate approximation to the integral because its error series is $\mathcal{O}(h^6)$.

The basic assumption on which of all this analysis depends is that Equation (3) is valid for the function f being integrated. Of course, in practice, we will use a modest number of rows in the Romberg algorithm, and only this number of terms in Equation (3) is needed.

Here is the theorem that governs the situation:

THEOREM 1 **EULER-MACLAURIN FORMULA AND ERROR TERM**

If $f^{(2m)}$ exists and is continuous on the interval $[a, b]$, then

$$\int_a^b f(x)\,dx = \frac{h}{2}\sum_{i=0}^{n-1}[f(x_i) + f(x_{i+1})] + E$$

where $h = (b-a)/n$, $x_i = a + ih$ for $0 \leqq i \leqq n$, and

$$E = \sum_{k=1}^{m-1} A_{2k} h^{2k}[f^{(2k-1)}(a) - f^{(2k-1)}(b)] - A_{2m}(b-a)h^{2m} f^{(2m)}(\xi)$$

for some ξ in the interval (a, b).

In this theorem, the A_k's are constants (related to the **Bernoulli numbers**) and ξ is some point in the interval (a, b). The interested reader should refer to Young and Gregory [1972, vol. 1, p. 374]. It turns out that the A_k's can be defined by the equation

$$\frac{x}{e^x - 1} = \sum_{k=0}^{\infty} A_k x^k \tag{7}$$

Observe that in the Euler-Maclaurin formula, the right-hand side contains the trapezoid rule and an error term, E. Furthermore, E can be expressed as a finite sum in ascending powers of h^2. This theorem gives the formal justification (and the details) of Equation (3).

If the integrand f does not possess a large number of derivatives but is at least Riemann-integrable, then the Romberg algorithm still converges in the following sense: The limit of each *column* in the array equals the integral:

$$\lim_{n\to\infty} R(n, m) = \int_a^b f(x)\,dx \qquad (m \geqq 0)$$

The convergence of the first column is easily justified by referring to the upper and lower sums. (See Problem 5.2.23.) After the convergence of the first column has been established, the convergence of the remaining columns can be proved by using Equation (2). (See Problems 5.3.24 and 5.3.25.)

In practice, we may not know whether the function f whose integral we seek satisfies the smoothness criterion upon which the theory depends. Then it would not be known whether Equation (3) is valid for f. One way of testing this in the course of the Romberg algorithm is to compute the ratios

$$\frac{R(n, m) - R(n-1, m)}{R(n+1, m) - R(n, m)}$$

and to note whether they are close to 4^{m+1}. Let us verify, at least for the case $m = 0$, that this ratio is near 4 for a function that obeys Equation (3).

If we subtract Equation (4) from (3), the result is

$$R(n, 0) - R(n-1, 0) = \tfrac{3}{4}a_2h^2 + \tfrac{15}{16}a_4h^4 + \tfrac{63}{64}a_6h^6 + \cdots \tag{8}$$

If we write down the same equation for the *next* value of n, then the h of that equation is half the value of h used in Equation (8). Hence,

$$R(n+1,0) - R(n,0) = \frac{3}{4^2}a_2h^2 + \frac{15}{16^2}a_4h^4 + \frac{63}{64^2}a_6h^6 + \cdots \tag{9}$$

Equations (8) and (9) are now used to express the ratio mentioned previously:

$$\frac{R(n,0) - R(n-1,0)}{R(n+1,0) - R(n,0)} = 4\left[\frac{1 + \frac{5}{4}\left(\frac{a_4}{a_2}\right)h^2 + \frac{21}{16}\left(\frac{a_6}{a_2}\right)h^4 + \cdots}{1 + \frac{5}{4^2}\left(\frac{a_4}{a_2}\right)h^2 + \frac{21}{16^2}\left(\frac{a_6}{a_2}\right)h^4 + \cdots}\right]$$

$$= 4\left[1 + \frac{15}{4^2}\left(\frac{a_4}{a_2}\right)h^2 + \cdots\right]$$

For small values of h, this expression is close to 4.

General Extrapolation

In closing, we return to the extrapolation process that is the heart of the Romberg algorithm. The process is Richardson extrapolation, which was discussed in Section 4.3. It is an example of a general dictum in numerical mathematics that if anything is known about the errors in a process, then that knowledge can be exploited to improve the process.

The only type of extrapolation illustrated so far (in this section and Section 4.3) has been the so-called h^2 extrapolation. It applies to a numerical process in which the error series is of the form

$$E = a_2h^2 + a_4h^4 + a_6h^6 + \cdots$$

In this case, the errors behave like $\mathcal{O}(h^2)$ as $h \to 0$, but the basic idea of Richardson extrapolation has much wider applicability. We could apply extrapolation if we knew, for example, that

$$E = ah^\alpha + bh^\beta + ch^\gamma + \cdots$$

provided that $0 < \alpha < \beta < \gamma < \cdots$. It is sufficient to see how to annihilate the first term of the error expansion because the succeeding steps would be similar.

Suppose therefore that

$$L = \varphi(h) + ah^\alpha + bh^\beta + ch^\gamma + \cdots \tag{10}$$

Here, L is a mathematical entity that is approximated by a formula $\varphi(h)$ depending on h with the error series $ah^\alpha + bh^\beta + \cdots$. It follows that

$$L = \varphi\left(\frac{h}{2}\right) + a\left(\frac{h}{2}\right)^\alpha + b\left(\frac{h}{2}\right)^\beta + c\left(\frac{h}{2}\right)^\gamma + \cdots$$

Hence, if we multiply this by 2^α, we get

$$2^\alpha L = 2^\alpha\varphi\left(\frac{h}{2}\right) + ah^\alpha + 2^\alpha b\left(\frac{h}{2}\right)^\beta + 2^\alpha c\left(\frac{h}{2}\right)^\gamma + \cdots$$

By subtracting Equation (10) from this equation, we rid ourselves of the h^α term:

$$(2^\alpha - 1)L = 2^\alpha\varphi\left(\frac{h}{2}\right) - \varphi(h) + (2^{\alpha-\beta} - 1)bh^\beta + (2^{\alpha-\gamma} - 1)ch^\gamma + \cdots$$

We rewrite this as

$$L = \frac{2^\alpha}{2^\alpha - 1}\varphi\left(\frac{h}{2}\right) - \frac{1}{2^\alpha - 1}\varphi(h) + \tilde{b}h^\beta + \tilde{c}h^\gamma + \cdots \tag{11}$$

Thus, the special linear combination

$$\frac{2^\alpha}{2^\alpha - 1}\varphi\left(\frac{h}{2}\right) - \frac{1}{2^\alpha - 1}\varphi(h) = \varphi\left(\frac{h}{2}\right) + \frac{1}{2^\alpha - 1}\left[\varphi\left(\frac{h}{2}\right) - \varphi(h)\right] \tag{12}$$

should be a more accurate approximation to L than either $\varphi(h)$ or $\varphi(h/2)$ because their error series, in Equations (10) and (11), improve from $\mathcal{O}(h^\alpha)$ to $\mathcal{O}(h^\beta)$ as $h \to 0$ and $\beta > \alpha > 0$. Notice that when $\alpha = 2$, the combination in Equation (12) is the one we have already used for the second column in the Romberg array.

Extrapolation of the same type can be used in still more general situations, as is illustrated next (and in the problems).

EXAMPLE 2 If φ is a function with the property

$$\varphi(x) = L + a_1x^{-1} + a_2x^{-2} + a_3x^{-3} + \cdots$$

how can L be estimated using Richardson extrapolation?

Solution Obviously, $L = \lim_{x\to\infty}\varphi(x)$; thus, L can be estimated by evaluating $\varphi(x)$ for a succession of ever-larger values of x. To use extrapolation, we write

$$\begin{aligned}
\varphi(x) &= L + a_1x^{-1} + a_2x^{-2} + a_3x^{-3} + \cdots \\
\varphi(2x) &= L + 2^{-1}a_1x^{-1} + 2^{-2}a_2x^{-2} + 2^{-3}a_3x^{-3} + \cdots \\
2\varphi(2x) &= 2L + a_1x^{-1} + 2^{-1}a_2x^{-2} + 2^{-2}a_3x^{-3} + \cdots \\
2\varphi(2x) - \varphi(x) &= L - 2^{-1}a_2x^{-2} - 3\cdot 2^{-2}a_3x^{-3} - \cdots
\end{aligned}$$

Thus, having computed $\varphi(x)$ and $\varphi(2x)$, we can compute a new function $\psi(x) = 2\varphi(2x) - \varphi(x)$. It should be a better approximation to L because its error series begins with x^{-2} and is $\mathcal{O}(x^{-2})$ as $x \to \infty$. This process can be repeated, as in the Romberg algorithm. ■

Here is a concrete illustration of the preceding example. We want to estimate $\lim_{x\to\infty}\varphi(x)$ from the following table of numerical values:

x	1	2	4	8	16	32	64	128
$\phi(x)$	21.1100	16.4425	14.3394	13.3455	12.8629	12.6253	12.5073	12.4486

A tentative hypothesis is that φ has the form in the preceding example. When we compute the values of the function $\psi(x) = 2\varphi(2x) - \varphi(x)$, we get a new table of values:

x	1	2	4	8	16	32	64
$\psi(x)$	11.7750	12.2363	12.3516	12.3803	12.3877	12.3893	12.3899

It therefore seems reasonable to believe that the value of $\lim_{x\to\infty}\varphi(x)$ is approximately 12.3899. If we do another extrapolation, we should compute $\theta(x) = [4\psi(2x) - \psi(x)]/3$;

values for this table are

x	1	2	4	8	16	32
$\theta(x)$	12.3901	12.3900	12.3899	12.3902	12.3898	12.3901

For the precision of the given data, we conclude that $\lim_{x\to\infty}\varphi(x) = 12.3900$ to within roundoff error.

Summary

(1) By using the Recursive Trapezoid Rule, we find that the first column of the **Romberg algorithm** is

$$R(n,0) = \frac{1}{2}R(n-1,0) + h\sum_{k=1}^{2^{n-1}} f[a+(2k-1)h]$$

where $h = (b-a)/2^n$ and $n \geqq 1$. The second and successive columns in the Romberg array are generated by the Richardson extrapolation formula and are

$$R(n,m) = R(n,m-1) + \frac{1}{4^m-1}[R(n,m-1) - R(n-1,m-1)]$$

with $n \geqq 1$ and $m \geqq 1$. The error is $\mathcal{O}(h^2)$ for the first column, $\mathcal{O}(h^4)$ for the second column, $\mathcal{O}(h^6)$ for the third column, and so on. Check the ratios

$$\frac{R(n,m) - R(n-1,m)}{R(n+1,m) - R(n,m)} \approx 4^{m+1}$$

to test whether the algorithm is working.

(2) If the expression L is approximated by $\varphi(h)$ and if these entities are related by the error series

$$L = \varphi(h) + ah^\alpha + bh^\beta + ch^\gamma + \cdots$$

then a more accurate approximation is

$$L \approx \varphi\left(\frac{h}{2}\right) + \frac{1}{2^\alpha - 1}\left[\varphi\left(\frac{h}{2}\right) - \varphi(h)\right]$$

with error $\mathcal{O}(h^\beta)$.

Additional References

For additional study, see Abramowitz and Stegun [1964], Clenshaw and Curtis [1960], Davis and Rabinowitz [1984], de Boor [1971], Dixon [1974], Fraser and Wilson [1966], Gentleman [1972], Ghizetti and Ossiccini [1970], Havie [1969], Kahaner [1971], Krylov [1962], O'Hara and Smith [1968], Stroud [1974], and Stroud and Secrest [1966].