9. Approximation of Integrals – Continued

9.1. **Iterative Approaches to the Approximation of Integrals.** One of the simplest iterative procedures to use is interval doubling. For example, using the composite trapezoidal rule on N subintervals (call this approximation T_N), we could calculate T_2, T_4. T_8, etc. doubling the number of subintervals at each step and stop when $|T_N - T_{2N}| \leq \epsilon$, the prescribed error tolerance. To do such a procedure efficiently, we would not want to recalculate any function values. Observe that

$$T_N = h\sum_{i=1}^{N-1} f(a+ih) + \frac{h}{2}[f(a)+f(b)], \qquad h = (b-a)/N,$$
$$T_{2N} = \frac{h}{2}\sum_{i=1}^{2N-1} f(a+ih/2) + \frac{h}{4}[f(a)+f(b)].$$

Hence,

$$T_{2N} = \frac{h}{2}\left[\sum_{i=1}^{N-1} f(a+ih) + \sum_{i=1}^{N} f(a+[i-1/2]h)\right] + \frac{h}{4}[f(a)+f(b)]$$
$$= \frac{1}{2}T_N + \frac{h}{2}\sum_{i=1}^{N} f(a+[i-1/2]h) = \frac{1}{2}(T_N + M_N).$$

So only the midpoint rule M_N must be computed at each step.

9.2. **Richardson Extrapolation and Romberg Integration.** One can greatly improve the rate of convergence of the composite trapezoidal rule by applying Richardson extrapolation. The resulting method is called Romberg integration. The basic idea is described as follows.

Suppose for every $h > 0$, we let $L_h(f)$ denote an approximation to the number $L(f)$. For example, $L(f) = \int_a^b f(x)\,dx$ and $L_h(f) = T_N(f)$, where $h = (b-a)/N$, the composite trapezoidal rule on N subintervals. Suppose $L(f) = L_h(f) + Ch^r + o(h^r)$, where c is a constant independent of h and r is a positive number. We say a quantity E is $o(h^r)$ if $|E/h^r| \to 0$ as $h \to 0$. We showed that

$$L(f) = T_N(f) - (b-a)h^2 f''(\xi)/12.$$

Unfortunately, this ξ may depend on h so we are not exactly of the above form. Recall that

$$f''(\xi) = \frac{1}{N}\sum_{i=1}^{N} f''(\eta_i) = \frac{h}{b-a}\sum_{i=1}^{N} f''(\eta_i), \qquad \eta_i \in [x_{i-1}, x_i],$$

which depends on h.

However, we can also derive another form of the error, known as the Euler-Maclaurin summation formula (refr: Dahlquist, pg. 297). If $f \in C^{2r+2}$, then

$$T_N = \int_a^b f(x)\,dx + \frac{h^2}{12}[f'(b) - f'(a)] - \frac{h^4}{720}[f'''(b) - f'''(a)] + \cdots + c_{2r}h^{2r}[f^{(2r-1)}(b) - f^{(2r-1)}(a)] + O(h^{2r+2}).$$

Hence the error does have the desired form with $r = 2$, i.e.,

$$L(f) = T_N + c_2h^2 + O(h^4), \qquad c_2 = [f'(a) - f'(b)]/12.$$

Now pick an h and a number $q > 1$ (e.g., $q = 2$) and calculate

$$L_h^{(1)}(f) = L_h(f) + \frac{L_h(f) - L_{qh}(f)}{q^r - 1}$$

from the two numbers $L_h(f)$ and $L_{qh}(f)$.

Example: $q = 2$:

$$T_N^{(1)}(f) = T_N(f) + \frac{T_N(f) - T_{N/2}(f)}{4 - 1},$$

i.e., $L_h \equiv T_N$ and $L_{2h} \equiv T_{N/2}$.

Claim: $L(f) = L_h^{(1)}(f) + o(h^r)$.

Example:

$$L(f) = T_N + c_2h^2 + O(h^4), \qquad L(f) = T_{N/2} + c_2(2h)^2 + O([2h]^4).$$

So

$$\begin{aligned} L(f) - T_N^{(1)}(f) &= L(f) - T_N(f) - [T_N(f) - T_{N/2}(f)]/3 \\ &= L(f) - T_N(f) - [T_N(f) - L(f) + L(f) - T_{N/2}(f)]/3 \\ &= c_2h^2 + O(h^4) - [-c_2h^2 - O(h^4) + 4c_2h^2 + O(h^4)]/3 = O(h^4) = o(h^2). \end{aligned}$$

Hence $T_N^{(1)}(f)$ is an $O(h^4)$ approximation to $L(f)$ while $T_N(f)$ is only an $O(h^2)$ approximation.

From the Euler-Maclaurin summation formula,

$$L(f) = T_N(f) + c_2h^2 + c_4h^4 + \cdots,$$

with c_r independent of h. Hence if f is sufficiently smooth, we can extrapolate again to get a higher order approximation. Each extrapolation amounts to eliminating the next lowest order term in the error $L(f) - T_N(f)$. The second extrapolation $T_N^{(2)}(f)$ is defined by:

$$T_N^{(2)}(f) = T_N^{(1)}(f) + [T_N^{(1)}(f) - T_{N/2}^{(1)}(f)]/15.$$

In general, we define the mth extrapolate by:

$$T_N^{(m)}(f) = T_N^{(m-1)}(f) + [T_N^{(m-1)}(f) - T_{N/2}^{(m-1)}(f)]//(4^m - 1).$$

This will be a $O(h^{2m+2})$ approximation to $L(f)$.

What quantities do we need to compute these extrapolations?

$T_N^{(1)}$ requires T_N and $T_{N/2}$. $T_N^{(2)}$ requires $T_N^{(1)}$ and $T_{N/2}^{(1)}$. But $T_{N/2}^{(1)}$ requires $T_{N/2}$ and $T_{N/4}$, so $T_N^{(2)}$ requires T_N, $T_{N/2}$, and $T_{N/4}$. $T_N^{(m)}$ requires $T_N^{(m-1)}$ and $T_{N/2}^{(m-1)}$ which requires $T_N^{(m-2)}$, $T_{N/2}^{(m-2)}$, and $T_{N/4}^{(m-2)}$. Continuing in this way, $T_N^{(m)}$ will require T_N, $T_{N/2}$, $\cdots$, $T_{N/2^m}$.

Suppose $N/2^m = I$, an integer. The Romberg entries form the following table, where we denote by $T_J^{(0)}$ the composite trapezoidal rule with J subintervals.

TABLE 3

Romberg Integration Table

T_I^0				
T_{2I}^0	T_{2I}^1			
T_{4I}^0	T_{4I}^1	T_{4I}^2		
$\cdots$	$\cdots$	$\cdots$		
$T_{2^mI}^0$	$T_{2^mI}^1$	$T_{2^mI}^2$	$\cdots$	$T_{2^mI}^m$

The table is constructed one row at a time. When $|T_{2^mI}^m - T_{2^mI}^{(m-1)}| <$ the given tolerance, the iteration is stopped. We thus have the following algorithm for computing an approximation to $\int_a^b f(x)\,dx$ by Romberg Integration.

Romberg Integration

Given a function f defined on $[a,b]$ and a positive integer I (say $I = 2$), set $h = (b-a)/I$.

$$\text{Calculate} \qquad T_I^0 = h\sum_{i=1}^{I-1} f(a+ih) + \frac{h}{2}[f(a)+f(b)].$$

`For` $k = 1, 2, \cdots,$ `do:`
`Calculate` $M_{2^{k-1}I} = h\sum_{i-1}^{2^{k-1}I} f(a+[i-1/2]h)$.
`Calculate` $T_{2^kI}^0 = [T_{2^{k-1}I}^0 + M_{2^{k-1}I}]/2$.
`Set` $h = h/2$ `and for` $m = 1, \cdots, k,$ `do`
`Calculate` $T_{2^kI}^m = T_{2^kI}^{m-1} + [T_{2^kI}^{m-1} - T_{2^{k-1}I}^{m-1}]/(4^m - 1)$.
`Stop when` $|T_{2^kI}^m - T_{2^kI}^{(m-1)}| <$ `tolerance` ϵ.

$$\text{If } f \in C^{2m+2}, \text{ then } L(f) = \int_a^b f(x)\,dx = T_{2^kI}^m + O\left(\left[\frac{b-a}{2^kI}\right]^{2m+2}\right).$$

Note: In these procedures, we are not taking advantage of the fact that the algorithm may have already stabilized on some subintervals. Hence, we are doing function evaluations where we don't need to. Most modern adaptive codes take into account this information and only decrease the subinterval size where the function appears to be rapidly changing. We shall discuss this in more detail later.