LECTURE 9: ADAPTIVITY

Two approaches

- Order adaptive - increase n (p-refinement)
  - Expensive? (Need to compute nodes and weights for each n)
  - Will struggle for non-smooth functions
- Interval adaptive (h-refinement)
  - Apply same rule on subintervals

$$I_n^m[a,b] = \sum_{j=0}^{m-1} I_n[a_j, a_{j+1}] \qquad : a_0 = a, a_m = b.$$

  - Sometimes points can be recycled when doubling M (→ better efficiency).
  - Subintervals need not be uniform - put them where needed!
  - Better for nonsmooth functions.

Computable error estimates

Aim: A black box quadrature routine which computes $I(f)$ to a given tolerance.

```
function I = int(f, a, b, tol)
% INT(F, A, B, TOL) integrates F on [A, B] to a tolerance TOL.
   n = 17;
   [x, w] = quadNodesAndWeights(n); % nodes and weights on [-1, 1]
   y = (b*(x+1)+a*(1-x))/2;          % nodes on [a, b]
   w = w*(b-a)/2;                    % weights on [a, b]
   I = w'*f(y);
   err ≈ |I-∫_a^b f(x)dx|;           % local error estimate.
   if ( err > tol )
      I = int(f, a, (a+b)/2, tol/2) + int(f, (a+b)/2, b, tol/2);
   end
end
```

Question: How to compute err?

Error estimates

The million dollar question in quadrature!

- Difference between two estimates (say, $n_2 = 2n_1$).
  - $I_{n_1} - I_{n_2} \approx I_{n_1} - \int_a^b f(x)dx$
- Gauss-Kronrod schemes.
  - Used by Matlab in `quadl()` and `quadgk()`.
  - Pairs of quadrature schemes where nodes of one are subset of the other.
- Richardson extrapolation.

Successive differences

The error in Simpson's rule:

$$\begin{aligned} \int_a^b f(x)dx &= S(f;a,b) - \frac{h_1^5}{90} f^{(4)}(\mu_1), \quad \mu_1 \in (a,b) \\ &= S(f;a,b) - E(f;h_1,\mu_1). \end{aligned}$$

Consider the first step of Composite Simpson's rule:

$$\int_a^b f(x)dx = S(f;a,\frac{a+b}{2}) + S(f;\frac{a+b}{2},b) - E(f;h_2,\mu_2).$$

Applying the same error formula twice:

$$E(f;h_2,\mu_2) = \frac{1}{16}\left(\frac{h_1^5}{90} f^{(4)}(\mu_2)\right) = \frac{1}{16} E(f;h_1,\mu_2).$$

The error in Simpson's rule:

$$\begin{aligned}\int_a^b f(x)dx &= S(f;a,b) - \frac{h_1^5}{90} f^{(4)}(\mu_1), \quad \mu_1 \in (a,b) \\ &= S(f;a,b) - E(f;h_1,\mu_1).\end{aligned}$$

Consider the first step of Composite Simpson's rule:

$$\int_a^b f(x)dx = S(f;a,\frac{a+b}{2}) + S(f;\frac{a+b}{2},b) - E(f;h_2,\mu_2).$$

Applying the same error formula twice:

$$E(f;h_2,\mu_2) = \frac{1}{16}\left(\frac{h_1^5}{90} f^{(4)}(\mu_2)\right) = \frac{1}{16} E(f;h_1,\mu_2).$$

Successive differences

$$E(f;h_2,\mu_2) = \frac{1}{32}\left(\frac{h_1^5}{90} f^{(4)}(\mu_2^1)\right) + \frac{1}{32}\left(\frac{h_1^5}{90} f^{(4)}(\mu_2^2)\right).$$

If $f \in C^4[a,b]$, apply the intermediate value theorem:

$$\exists \mu \in [\mu_2^1, \mu_2^2] \subset [a,b] : f^{(4)} = \frac{f^{(4)}(\mu_2^1) + f^{(4)}(\mu_2^2)}{2}.$$

Hence

$$E(f;h_2,\mu_2) = \frac{1}{16}\left(\frac{h_1^5}{90} f^{(4)}(\mu_2)\right).$$

Now if we assume

$$f^{(4)}(\mu_1) \approx f^{(4)}(\mu_2),$$

then

$$E(f; h_1, \mu) \approx 16E(f; h_2, \mu).$$

So

$$I = S(f; a, b) - 16E(f; h_2, \mu) = S(f; a, \frac{a+b}{2}) + S(f; \frac{a+b}{2}, b) - E(f; h_2, \mu)$$

Hence,

$$E(f; h_2, \mu) \approx \frac{1}{15}\left[S(f; a, b) - S(f; a, \frac{a+b}{2}) - S(f; \frac{a+b}{2}, b)\right].$$

Nice!

Numerical example: we compute

$$I = \int_0^{\pi/2} \sin(x)dx = 1.$$

Simpson's rule:

$$S(\sin(x), 0, \pi/2) = 1.00227987749221.$$

First composite Simpson's rule:

$$S(\sin(x), 0, \pi/4) + S(\sin(x), \pi/4, \pi/2) = 1.00013458497419.$$

Error estimate:

$$E = \frac{1}{15}[1.00227987749221 - 1.00013458497419] = 0.00014301950120.$$

which compares very well with the actual error, 0.00013458497419.

Gauss–Kronrod

- Different idea: build an interpolation scheme where a subset of the nodes form a different interpolation scheme.
- Problem: Gaussian quadrature rules don't share nodes $\Rightarrow$ no chance to reuse computations.
- Solution: Gauss-Kronrod rules: augment a Gaussian rule with additional points in a clever way.

Gaussian quadrature again:

$$\int_{-1}^{1} f(x)dx \approx \sum w_i f(x_i),$$

with x_i the roots of a Legendre polynomial.

Kronrod's idea: augment the approximation with m other points y_j.

$$\int_{-1}^{1} f(x)dx \approx \sum a_i f(x_i) + \sum b_j f(y_j),$$

with y_j the roots of a Stieltjes polynomial E_n:

$$\langle E_n, P_{n-1} x^k \rangle = 0 \quad \forall k = 0 \ldots n-1.$$

For example: 15-point rule of order 22, with 7-point rule of order 13 embedded!

Richardson extrapolation

- Suppose we have an approximation scheme for a quantity A with known error formula:

$$A = A_H + a_1 H^1 + a_2 H^2 + \cdots .$$

- Suppose we also compute the approximation for a different h:

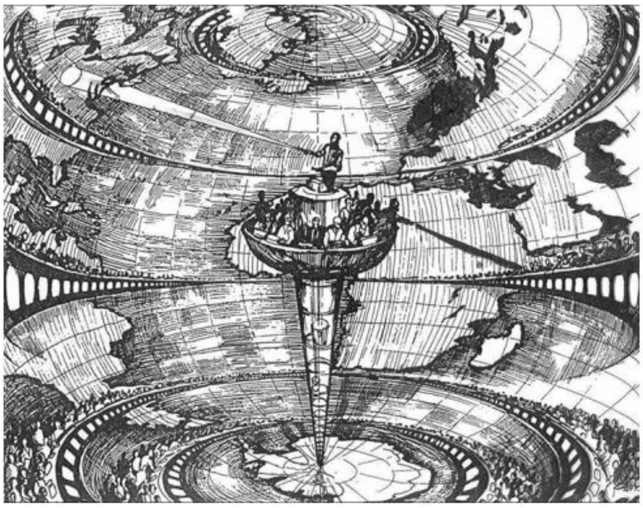
$$A = A_h + a_1 h^1 + a_2 h^2 + \cdots .$$

- Multiplying and subtracting:

$$\begin{aligned} (h - H)A &= hA_H - HA_h + a_2(hH^2 - Hh^2) + \cdots \\ \Rightarrow A &= \frac{hA_H - HA_h}{h - H} + O(H^2) \\ \Rightarrow A &= A_h + \frac{h(A_H - A_h)}{h - H} + O(H^2) \end{aligned}$$

- The key to Richardson extrapolation is knowing the error formula.
- "*... its usefulness for practical computations can hardly be overestimated.*"

- Lewis Fry Richardson (1881 - 1953)
- Richardson extrapolation
- Richardson iteration (for solving linear systems)
- Mathematical analysis of the fractal nature of coastlines, flow through peat, turbulence, . . .
- *Statistics of Deadly Quarrels*
- *Weather Prediction by Numerical Process*
- Computed the first numerical weather forecast (blew up)



"A myriad computers are at work upon the weather of the part of the map where each sits, but each computer attends only to one equation or part of an equation. Numerous little "night signs" display the instantaneous values so that neighbouring computers can read them." (1992)

- Suppose only even terms occur in the error expansion:

$$A = A_h + a_2 h^2 + a_4 h^4 + \cdots$$

- Apply Richardson extrapolation once with $h/2$:

$$A = A_{h/2} + a_2 \frac{h^2}{4} + a_4 \frac{h^4}{16} + \cdots$$

$$A = \frac{1}{3}(4A_{h/2} - A_h) - \frac{a_4}{4} h^4 + \cdots .$$

Define this more accurate formula to be $A_h^{(2)}$.

- Now recurse!

$$A = A_h^{(2)} + a_4^{(2)} h^4 + \cdots$$

$$A = A_{h/2}^{(2)} + \frac{a_4^{(2)}}{16} h^4 + \cdots$$

$$A = \frac{1}{15}(16A_{h/2}^{(2)} - A_h^{(2)}) + a_6^{(3)} h^6 + \cdots$$

Romberg integration is composite trapezoidal + Richardson extrapolation.
Let $R_{k,1}$ be the composite trapezoidal rule with 2^{k-1} subintervals:

$$\begin{aligned} R_{1,1} &= \frac{h_1}{2}[f(a)+f(b)] \\ R_{2,1} &= \frac{h_2}{2}[f(a)+2f(a+h_2)+f(b)] \\ &= \frac{1}{2}[R_{1,1}+h_1 f(a+h_2)] \\ &\cdot \\ &\cdot \\ &\cdot \\ R_{k,1} &= \frac{1}{2}\left[R_{k-1,1}+h_{k-1}\sum_{i=1}^{2^{k-2}} f(a+(2i-1)h_k)\right] \end{aligned}$$

We know that the error term is $O(h^2)$, so apply Richardson extrapolation:

$$R_{k,2} = R_{k,1} + \frac{R_{k,1}-R_{k-1,1}}{2^2-1}$$

This will be $O(h^4)$. (Actually Composite Simpson again.)

To get $O(h^6)$:

$$R_{k,3} = R_{k,2} + \frac{R_{k,2}-R_{k-1,2}}{4^2-1}$$

This is the same error estimator as before!

A numerical example:

$$I = \int_0^{\pi} \sin(x)dx.$$

$R_{k,1}$	$R_{k,2}$	$R_{k,3}$	$R_{k,4}$
1.570796326794897	2.094395102393195		
1.896118897937040	2.004559754984421	1.998570731823836	
1.974231601945551	2.000269169948388	1.999983130945986	2.000005549979671
1.993570343772340	2.000016591047935	1.999999752454572	2.000000016288042
1.998393360970145	2.000001033369413	1.999999996190845	2.000000000059674
1.999598388640037	2.000000064530001	1.999999999940707	2.000000000000229

Aitken's Δ^2–method

Aside: other sequence acceleration methods are possible: Aitken's Δ^2–method defines

$$\hat{s}_i = s_i - \frac{(s_{i+1} - s_i)^2}{s_{i+2} - 2s_{i+1} + s_i}.$$

Let us apply this to the calculation of π via

$$\pi = 4\sum_{n=0}^{N} \frac{(-1)^n}{2n+1}.$$

For $N = 1000$, the unaccelerated error is $\sim 10^{-3}$ but the accelerated error is $\sim 10^{-10}$.

SUMMARY

- Quadrature has close ties with interpolation (integrate the 'proxy'!)
- Good / bad interpolation points → good / bad quadrature points
- Two kinds of adaptivity:
  - Interval adaptive (split the intervals: *h*-refinement)
  - Order adaptive (increase degree: *p*-refinement)
- Adaptivity needs a good error estimate.

  (See Gonnet, "*A review of error estimation in adaptive quadrature*", ACM CSUR, 2014)
- These ideas are crucial for robustness and automation generally.

  (See Trefethen, "*Predictions for scientific computing 50 years from now*", Mathematics Today, 2000)