Review of integration using Lagrange interpolating polynomials:

Goal: Approximate

$$\int_a^b f(x)\,dx.$$

Select nodes $x_0, x_1, \ldots, x_n \in [a,b]$.

Lagrange interpolating polynomials of degree n:

$$l_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}, \qquad 0 \leq i \leq n.$$

Cool idea: The polynomial of degree n that "best approximates" f is

$$p(x) = \sum_{i=1}^{n} f(x_i)\, l_i(x),$$

so $p(x_j) = f(x_j)$ for all $j = 0, 1, \ldots, n$. Then

$$\int_a^b f(x)\,dx \approx \int_a^b p(x)\,dx = \sum_{i=1}^{n} f(x_i) \underbrace{\int_a^b l_i(x)\,dx}_{A_i}.$$

So,

$$\int_a^b f(x)\,dx = \sum_{i=1}^{n} A_i f(x_i) \qquad (*)$$

with

$$A_i = \int_a^b l_i(x)\,dx.$$

So given the nodes x_i, the weights A_i are determined.

If f is a polynomial of degree n then

$$p(x) = f(x), \quad x \in [a,b]$$

and so ⊛ is exact. To improve the accuracy of the integral, all you can do, assuming equally spaced nodes, is to increase the number of nodes.

Section 7.3: Gaussian quadrature

Idea: To improve the accuracy of numerically approximating

$$\int_a^b f(x)\,dx$$

we could both increase the degree of the polynomials used in interpolation and carefully choose both nodes x_i and weights A_i in the formula.

In fact, we compute

$$\int_a^b w(x) f(x)\,dx$$

where w is a weight function with

- $w \geq 0$ on $[a,b]$
- w is integrable on $[a,b]$.

Notice the A_i's have nothing to do a priori with ω even though both are called weights.

We want to approximate

$$\int_a^b \omega(x) f(x)\,dx \approx \sum_{i=1}^{n} \omega_i f(x_i)$$

and we want the formula to be exact as high a degree polynomial as possible.

<u>Intuition</u>: Since x_i and A_i will be chosen independently, (we have 2^n #) we should be able to get the formula to be exact for degree $2n-1$ polynomials.

Let

Only need linearity of $\int$ here

$$\mathcal{E}_n(f) = \int_a^b f(x)\omega(x)\,dx - \sum_{i=0}^{n} \omega_i(x) f(x_i)$$

be the error in our approximation. Since $\mathcal{E}_n$ is linear

$$\mathcal{E}_n\left(\sum_{k=0}^{m} a_k x^k\right) = \sum_{k=0}^{m} a_k \mathcal{E}(x^k).$$

To make $\mathcal{E}_n(f)=0$ for all polynomial f of degree m we just need

$$\mathcal{E}_n(x^k)=0, \quad k \le m.$$

Use these equations to determine nodes and weights.

Solve for ω_1, x_1 in

$$\int_a^b f(x)\,w(x)\,dx \approx \omega_1 f(x_1).$$

From now on, choose $[a,b] = [-1,1]$.
(But we can always rescale any $[a,b]$ into $[-1,1]$).
We want (**) exact for degree 1 polynomials.

So need to use

$$\varepsilon_1(x^0) = 0, \quad \text{and} \quad \varepsilon_1(x^1) = 0$$

to get x_1, ω_1.

ⓐ $\underline{f(x) = x^0 = 1:}$

$$\int_{-1}^{1} f(x) = \int_{-1}^{1} 1\,dx = 2$$

$$\omega_1 f(x_1) = \omega_1 \cdot 1 = \omega_1$$

So,

$$0 = \varepsilon_1(x^0) \Rightarrow \omega_1 = 2.$$

ⓑ $\underline{f(x) = x^1:}$

$$\int_{-1}^{1} f(x)\,dx = \int_{-1}^{1} x\,dx = 0$$

$$\omega_1 f(x_1) = \omega_1 x_1^1 = 2x_1.$$

So

$$\varepsilon_1(x^1) = 0 \Rightarrow 2x_1 = 0$$

$$x_1 = 0.$$

Thus our quadrature formula is

$$\int_{-1}^{1} f(x)\,dx \approx 2f(0)$$

$\Delta x = 2$ midpoint rule with one subinterval

Case $n=2$, $w \equiv 1$:

4 parameters: w_1, w_2, x_1, x_2.

Need 4 equations

$$\varepsilon_2(x^i) = 0 = \int_{-1}^{1} f(x)\,dx - (w_1 x_1^i + w_2 x_2^i), \quad i = 0, 1, 2, 3.$$

$i=0$:

$$\int_{-1}^{1} 1\,dx - (w_1 + w_2) = 0 \qquad \Big| \qquad w_1 + w_2 = 2$$

$i=1$:

$$\int_{-1}^{1} x\,dx - (w_1 x_1 + w_2 x_2) = 0 \qquad \Big| \qquad w_1 x_1 + w_2 x_2 = 0$$

$i=2$:

$$\int_{-1}^{1} x^2\,dx - (w_1 x_1^2 + w_2 x_2^2) = 0 \qquad \Big| \qquad w_1 x_1^2 + w_2 x_2^2 = \frac{2}{3}$$

$i=3$:

$$\int_{-1}^{1} x^3\,dx - (w_1 x_1^3 + w_2 x_2^3) = 0 \qquad \Big| \qquad w_1 x_1^3 + w_2 x_2^3 = 0$$

Particular choice: $w_1 = w_2 = 1$.

$\boxed{i=1}$ gives

$x_2 = -x_1$.

$\boxed{i=2}$ gives

$2x_1^2 = \frac{2}{3}$

$x_1 = \pm\frac{1}{\sqrt{3}}$

Pick

$x_1 = \frac{1}{\sqrt{3}}$, $x_2 = -\frac{1}{\sqrt{3}}$.

Formula is

$$\int_{-1}^{1} f(x)\,dx \simeq f\left(\frac{1}{\sqrt{3}}\right) + f\left(-\frac{1}{\sqrt{3}}\right).$$

So need 2 function evaluations to get integral exact for a cubic polynumial.
Whereas for Simpson's rule, we need 3 function evalutions to get exactness for cubic polynomial.

<u>Notation</u>: Π_n is the space of degree $\leq n$.

<u>Theorem</u>: Let w be a positive weight function on $[a,b]$. Let q be a non-zero polynomial of degree $n+1$ that is w orthogonal to Π_n, i.e.

$$\int_a^b p(x)\, q(x)\,dx = 0, \quad p \in \Pi_n.$$

Let $x_0, \dots, x_n$ be the $n+1$ zeroes of q. Then the quadrature formula

$$\int_a^b f(x) w(x)\,dx \approx \sum_{i=0}^{n} A_i f(x_i) = \int_a^b w(x) \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x-x_j}{x_i-x_j}$$

is exact for all $f \in \Pi_{2n+1}$.

In words, gaussian quadrature allows us to exactly integrate polynomials of degree $\leq 2n+1$ using $n+1$ node points. Moreover if f is well approximated by a degree $\leq 2n+1$ polynomial and we can calculate f at any x, then we can approximate

$$\int_a^b f(x)\,dx$$

using $n+1$ function evaluations.

For previous methods, this only works when f is well approximated by a degree $\leq n$ polynomial.

<u>Proof</u>: Let $f \in \Pi_{2n+1}$. Write

$$f = pq + r, \qquad p, r \in \Pi_n .$$

Since the nodes $x_0, \dots, x_n$ satisfy

$$q(x_i), \qquad i = 0, \dots, n$$

we have

$$\underbrace{f(x_i)}_{\text{degree } 2n+1} = q(x_i)\,p(x_i) + r(x_i) = \underbrace{r(x_i)}_{\text{degree } n}. \quad (1)$$

So

$$\int_a^b f(x)\,dx = \underbrace{\int_a^b q(x)\,p(x)\,w(x)\,dx}_{0} + \int_a^b r(x)\,w(x)\,dx$$

$$\int_a^b f(x)\,dx \underset{\perp}{=} \int_a^b r(x)\,w(x)\,dx.$$

q is $w \perp \Pi_n$

We also know from "last time" that

$$\int_a^b r(x)\,w(x)\,dx = \sum_{i=0}^{n} A_i\, r(x_i), \quad r \in \Pi_n$$

is exact with

$$A_i = \int_a^b w(x)\, l_i(x)\, dx.$$

So by (1)

$$\int_a^b f(x)\,w(x)\,dx = \sum_{i=0}^{n} A_i f(x_i)$$

holds. $f = qp + r \in \Pi_{2n+1}$. $\square$

Main remaining issues:

(1) How to find q?

(2) How to calculate zeroes of q?

Notice these tasks are independent of f. (They do depend on $w(x)$).

Next time: See that x_i are the roots of the Legendre polynomials.

Example:

$$I = \int_0^{\pi} e^{x} \cos x \, dx$$

True answer is $I \approx -12.0703463164$.

n = # nodes	I_n^{Trap}	error	I_n^{Simp}	error
2	-17.89259	5.32	-11.592840	$-4.78 \cdot 10^{-1}$
4	-13.336023	1.27		$-8.54 \cdot 10^{-2}$
8	-12.382162	$3.12 \cdot 10^{-1}$		$-6.14 \cdot 10^{-3}$
16	-12.148004	$7.77 \cdot 10^{-2}$		$-3.95 \cdot 10^{-4}$
32	-12.089742	$1.94 \cdot 10^{-2}$		$-2.49 \cdot 10^{-5}$
64	-12.075194	$4.85 \cdot 10^{-3}$		$-1.56 \cdot 10^{-6}$
128	-12.071558	$1.21 \cdot 10^{-3}$		$-9.73 \cdot 10^{-8}$

---

n = # nodes	I_n^{GQ}	Error
2	-12.336210	$2.66 \cdot 10^{-1}$
3	-12.127420	$5.71 \cdot 10^{-2}$
4	-12.070189	$-1.57 \cdot 10^{-4}$
5	-12.0703285	$-1.78 \cdot 10^{-5}$
6	-12.070346	$1.47 \cdot 10^{-8}$

So GQ is superior in this example.