

Elements of Numerical Integration (Quadrature)

Sunday, October 4, 2020 4:31 PM

What? Why?

We saw that numerical differentiation via finite differences is unstable (BAD)
but analytical differentiation is easy (GOOD): just use
chain rule, product rule, sum rule

In contrast, analytical integration is not easy! "Most" integrals do not
have a closed form. Fortunately, numerical integration is stable.

Main idea

In Calc I you saw the "trapezoid", "midpoint" formulas.

These are examples of "quadrature" rules, i.e., numerical integration.

They are Riemann sums and converge to the integral as $h \rightarrow 0$.

We'll use $h > 0$

For nodes $\{x_0, x_1, \dots, x_n\}$, often equispaced ($x_{i+1} - x_i = h$),

our strategy is to fit an interpolant to these nodes, then

integrate the interpolant "by hand" (since it's usually a polynomial)

This lecture:

interpolate w/ a polynomial (n is small)

Next lecture:

interpolate w/ a piecewise polynomial (n is large),
"composite" formulas

(You could interpolate w/ a Spline, but historically this wasn't
done since finding splines by hand is hard)

(non-composite) interpolatory quadrature

Goal: Find $I = \int_a^b f(x) dx$, we can evaluate f but
don't know its antiderivative

We'll evaluate f on the $n+1$ nodes $\{x_0, x_1, \dots, x_n\}$ and

interpolate, forming the polynomial $p_n(x) := \sum_{i=0}^n f(x_i) L_{n,i}(x)$
 $\text{degree } \leq n$

where

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

is the Lagrange polynomial

Then (key conceptual idea!)

$$\begin{aligned} I_n &:= \int_a^b p_n(x) dx = \int_a^b \sum_{i=0}^n f(x_i) L_{n,i}(x) dx \\ &= \sum_{i=0}^n f(x_i) \underbrace{\int_a^b L_{n,i}(x) dx}_{w_i} \\ &= \sum_{i=0}^n w_i f(x_i) \end{aligned}$$

is our estimate of I .

The weights do not depend on $f(x_i)$, only on x_i .

In the common case of equispaced nodes, they really
 only depend on n (or h),

so we can precompute them!

| Key Fact!

Any estimate of I in the form $\sum_{i=0}^n w_i f(x_i)$ is a quadrature rule

Aside: an alternative approach, which is a bad idea in 1D but
 can be a good idea in high dimensions, is

to estimate $I \approx \sum_{i=0}^n \frac{1}{n+1} f(x_i)$, $x_i \in [a, b]$ chosen randomly
 ↪ ie. equi-weights

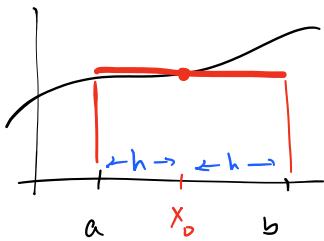
This is Monte Carlo (and quasi-Monte Carlo) integration.

First formula: $n=0$ "midpoint" aka "rectangle" formula

$n=0$? $\int_a^b f(x) dx$? The professor's gone crazy!

⚠ we always have
 $n+1$ nodes

No, we still want $\int_a^b f(x) dx$ $b > a$ but we're not going to
 have $a = x_0, \dots$ Instead, pick $x_0 = \frac{b-a}{2}$

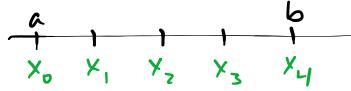


In general, we have 2 types of equispaced nodes:

① **Closed**, meaning $\{x_0 = a, x_1, \dots, x_{n-1}, x_n = b\}$

$$w_i, h = \frac{b-a}{n}$$

and

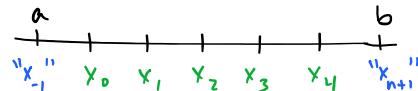


② **Open**, meaning $\{x_0 > a, x_1, \dots, x_{n-1}, x_n < b\}$

both
 $x_{i+1} = x_i + h$

but closed: $x_0 = a$

open: $x_0 = a + h$



$$h = \frac{b-a}{n+2}$$

So, back to $n=0$: nodes $\{x_0\}$, must do "open", $h = \frac{b-a}{n+2} = \frac{b-a}{2}$

The unique interpolant to $f(x_0)$ of degree 0 is the constant function,

$$p(x) = f(x_0), \text{ and } \int_a^b p(x) dx = f(x_0) \cdot (b-a)$$

$$\text{So } w_0 = (b-a), I_0 = \sum_{i=0}^0 w_i f(x_i) = f(x_0) (b-a).$$

Simple enough... what about analyzing the error though?

letting $h = \frac{b-a}{2}$, $x_0 = \frac{a+b}{2}$, Taylor expand around x_0 (assume f'' exists)

$$f(x) = f(x_0) + f'(x_0)(x-x_0) + f''(\xi)(x-x_0)^2/2! \quad \text{for some } \xi \in (a,b)$$

$$\begin{aligned} I_0 &= \int_a^b f(x) dx = \underbrace{f(x_0)(b-a)}_{I_0} + f'(x_0) \int_a^b (x-x_0) dx + \underbrace{\int_a^b f''(\xi) \frac{(x-x_0)^2}{2!} dx}_{\text{error}} \\ &\quad \downarrow \\ &\quad \frac{1}{2}(x-x_0)^2 \Big|_a^b = \frac{1}{2}h^2 - \frac{1}{2}h^2 = 0 \end{aligned}$$

$$\text{since } (b-x_0) = (x_0-a) = h$$

NOT quite as simple as

$$f''(\xi) \int_a^b \frac{(x-x_0)^2}{2!} dx = f''(\xi) \frac{(x-x_0)^3}{3!} \Big|_a^b$$

$$= f''(\xi) h^3 / 3$$

Why not?

we can't pull this out of the integral since $\xi = \xi(x)$ (depends on x)

2 ways to fix:

$$\textcircled{1} f \in C^2[a,b] \Rightarrow f'' \in C[a,b] \stackrel{\text{EVT}}{\Rightarrow} \exists M \text{ st. } \forall \xi \in [a,b], |f''(\xi)| \leq M$$

and our error is

$$\begin{aligned} E &= \left| \int_a^b f''(\xi) \frac{(x-x_0)^2}{2!} dx \right| \leq \int_a^b |f''(\xi)| \cdot \frac{(x-x_0)^2}{2!} dx \\ &\leq M \cdot \int_a^b \frac{(x-x_0)^2}{2!} dx \\ &= M/3 h^3 \end{aligned}$$

(2) Recall MVT etc:

Aside:

MVT: $F \in C[a, b]$ then $\exists \eta \in (a, b)$ s.t. $F'(\eta) = \frac{F(b) - F(a)}{b-a}$

e.g. $F(a) = F(b)$ 

MVT, integral form, let $F(x) = \int_0^x f(t) dt$, $F' = f$
 so via MVT $\exists \eta$ s.t. $f(\eta) = \frac{1}{b-a} \int_a^b f(t) dt \stackrel{F(b)-F(a)}{=}$

MVT, weighted version (Thm 1.13)

let $g(x)$ not change sign over $[a, b]$, $f \in C[a, b]$,

then $\exists \eta$ s.t. $f(\eta) = \frac{\int_a^b f(t) g(t) dt}{\int_a^b g(t) dt} \stackrel{g=1}{=} b-a$

... so, $\int_a^b f''(g(x)) (x-x_0)^2 dx$, apply wtd. MVT

$$\exists \eta \text{ s.t. } f''(\xi(\eta)) = \frac{\int_a^b f''(\xi(x)) (x-x_0)^2 dx}{\int_a^b (x-x_0)^2 dx} \stackrel{\text{some } \xi \in (a, b)}{=} \frac{1}{3} h^3$$

so $E = f''(\xi) \cdot \frac{1}{3} h^3$

so, conclusion, error due to midpoint rule is at most

$$E = \boxed{M/3 h^3} \quad \text{where } M = \max_{x \in [a, b]} |f''(x)|$$

Right now, a, b, n are given,

so we can't choose h

We can either increase n , or use a composite rule.

Implication

Suppose f is a constant function (degree 0 polynomial)

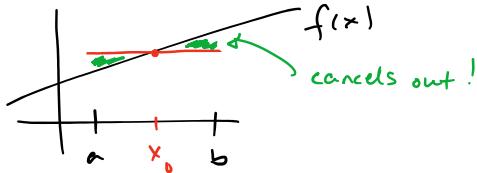
then by looking at picture, clearly error in midpt. rule is 0. Or, via our derivation,

$$M = \max_x |f''(x)| = 0, \text{ so another way to see it}$$

And actually, let f be linear (degree 1 polynomial)

then $M = 0$ also! So we make no error!

(What's happening? Our positive and negative errors exactly cancel)



If f is quadratic, then $M \neq 0$, so we do make an error.

Definition

The "degree of accuracy" or "degree of precision"] our book or "degree of exactness"] other books

of a quadrature rule is the largest integer K such that polynomials of degree K have 0 integration error.

Ex. Our midpt. rule has degree of exactness 1

Next, more rules (different values of n , closed nodes)