Ch 4.4: Composite quadrature

Monday, November 17, 2025 6:22 PM

Recap

Goal: estimate $\overbrace{\int_a^b f(x)\,dx}^{I}$, usually via $\overbrace{\int_a^b p_n(x)\,dx}^{I_n}$ where p_n is n degree polynomial interpolating f on the nodes $\{x_0, x_1, \ldots, x_n\}$. We discussed the case of equispaced nodes and the corresponding open/closed Newton Cotes formulae.

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

Our metrics for accuracy:

1) show error $E = |I - I_n| = O(h^k)$

$$k = \begin{cases} n+2 & n \text{ odd} \\ n+3 & n \text{ even} \end{cases}$$

2) order of exactness (aka degree of accuracy)

highest degree polynomial that has no error via the scheme

$$= \begin{cases} n & n \text{ odd} \\ n+1 & n \text{ even} \end{cases}$$

So to improve accuracy,

we want to decrease h, i.e., increase n

but w/ large n:

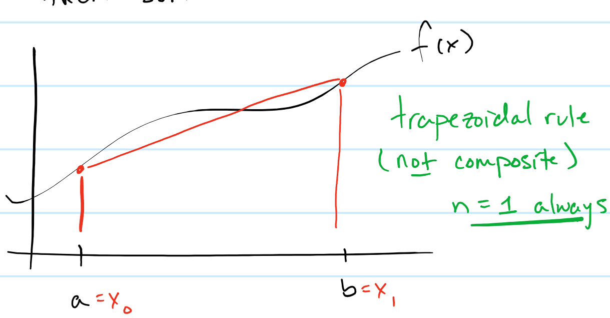
1) need to lookup formula every time we change n
or derive
(also might get negative weights $\Rightarrow$ less stable)

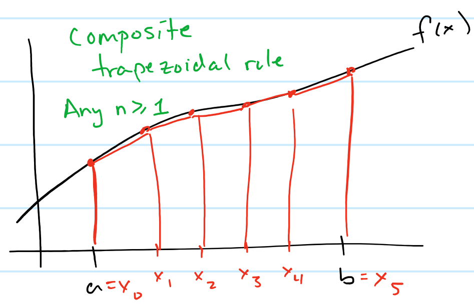
2) interpolating polynomial p_n has Runge phenomenon

3) we decrease h (i.e. increase n) to reduce the error bound, but this bound only holds if $f \in C^{n+1}$ or C^{n+2}

... thus simple alternative is composite integration

i.e., break $[a,b]$ into pieces, apply our old quadrature rules to each piece, then sum.





That's the idea. Now, details and analysis.

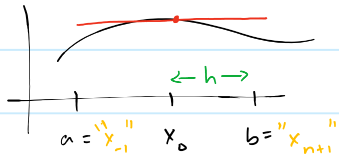
p. 2 (composite midpoint rule)

Monday, November 17, 2025 6:22 PM

For notation, there are several ways you could do it. We'll follow Burden + Faires

Composite midpoint rule
$n=0$, open

Recall (non-composite) midpoint rule

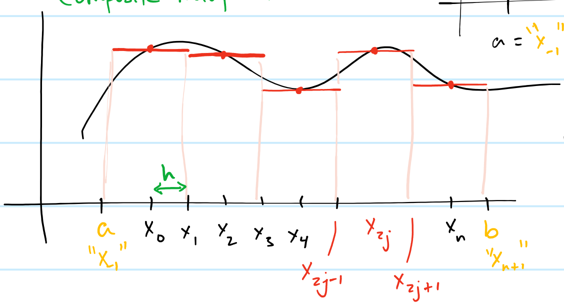


$n=0$

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

weight is $2h$

Composite midpoint



with our book's convention, we don't actually evaluate f at all the "nodes", only even nodes

(somewhat confusing, but we can still keep our definition $h = \frac{b-a}{n+2}$)

$\Rightarrow$ n must be even

Formula:

$$\int_{x_{-1}}^{x_{n+1}} f(x)\,dx = \underbrace{\int_{x_{-1}}^{x_1} f(x)\,dx}_{2h f(x_0) + \frac{h^3}{3} f''(\xi)} + \underbrace{\int_{x_1}^{x_3} f(x)\,dx}_{2h f(x_2) + \frac{h^3}{3} f''(\tilde{\xi})} + \dots + \underbrace{\int_{x_{n-1}}^{x_{n+1}} f(x)\,dx}_{2h f(x_n) + \frac{h^3}{3} f''(\tilde{\tilde{\xi}})}$$

$\underbrace{2h f(x_0)}_{\text{non-composite midpoint}}$, $\underbrace{\frac{h^3}{3} f''(\xi)}_{\text{error}}$

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

assume $|f''(\xi)| \le M$

$$= \underbrace{2h \sum_{j=0}^{n/2} f(x_{2j})}_{\text{composite midpoint}} + \underbrace{\underbrace{\frac{n}{2}\left(\frac{h^3}{3} M\right)}_{O(h^2)}}_{\text{error}}$$

FORMULA

a bit loose

note $n = O(\frac{1}{h})$

a more precise characterization: Thm. 4.6

$$\frac{b-a}{6} h^2 f''(\eta) \quad \text{for some } \eta \in (a,b)$$

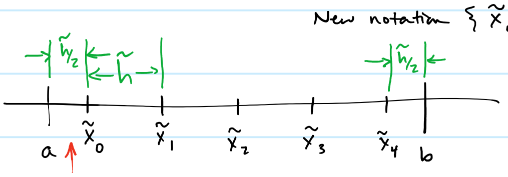
In implementation, it might be simpler to relabel nodes, counting just the even ones.

Old notation $\{x_0, x_1, x_2, \dots, x_n\}$ (n even) book's notation

New notation $\{\tilde{x}_0, \tilde{x}_1, \dots, \tilde{x}_m\}$ simpler notation

$m = n/2$

$\tilde{x}_j = x_{2j}$



⚠ In this new notation, while node spacing is $\tilde{h}$, the first node is offset by $\tilde{h}/2$

Old formula: $I \approx 2h \sum_{j=0}^{n/2} f(x_{2j})$

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

($m = n/2$ i.e. $n = 2m$)

New formula: $\tilde{h} = \frac{b-a}{m+1} = \frac{2\cdot(b-a)}{2(m+1)} = 2h$

Simplest quadrature formula ever!

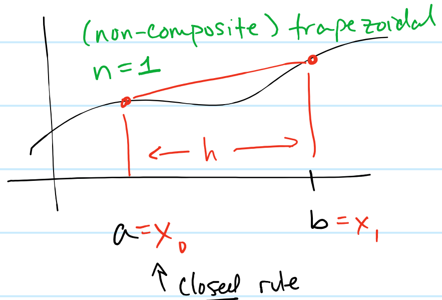
so $\boxed{I \approx \tilde{h} \sum_{j=0}^{m} f(\tilde{x}_j)}$

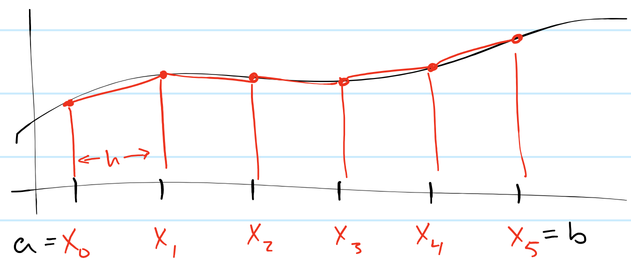
p. 3 (composite trapezoidal rule)

Monday, November 17, 2025 6:22 PM

Composite Trapezoidal Rule

In this case, notation is simpler, no need for "$\tilde{x}$" version, and no restriction that n is even





(non-composite) formula (assuming $f\in C^2[a,b]$)

$$I=\int_a^b f(x)\,dx = \frac{h}{2}\left(f(x_0)+f(x_1)\right)-\frac{h^3}{12}f''(\xi)$$

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

So composite:

$$I=\int_{x_0}^{x_1}f(x)\,dx+\int_{x_1}^{x_2}f(x)\,dx+\dots+\int_{x_{n-1}}^{x_n}f(x)\,dx$$

$$\text{"}h\text{"}=\frac{\overset{x_1}{\text{"}b\text{"}}-\overset{x_0}{\text{"}a\text{"}}}{\text{"}n\text{"}\to 1}=x_1-x_0=h$$

$$=\underbrace{\frac{h}{2}\left(f(x_0)+f(x_1)\right)-\frac{h^3}{12}f''(\xi)}+\frac{h}{2}\left(f(x_1)+f(x_2)\right)-\frac{h^3}{12}f''(\tilde{\xi})+\dots+\frac{h}{2}\left(f(x_{n-1})+f(x_n)\right)-\frac{h^3}{12}f''(\tilde{\tilde{\xi}})$$

once appears twice twice... twice... once

$$=\frac{h}{2}\left(f(x_0)+2\cdot\sum_{j=1}^{n-1}f(x_j)+f(x_n)\right)\underbrace{-\frac{b-a}{12}h^2f''(\eta)}_{\text{this is Thm. 4.5}}\qquad \eta\in(a,b)$$

(✱) FORMULA

interior nodes treated differently

Very similar to composite midpoint rule, except

1) nodes are closed, not open (for midpoint, $x_0=a+h/2$
trapezoid, $x_0=a$)

2) adjust weights on 1st and last node

Note: if $f(a)=f(b)$, the formula simplifies to $h\sum_{j=0}^{n-1}f(x_j)$

If also $f'(a)=f'(b)$
$f''(a)=f''(b)$, etc. then we'll see the error decays very quickly!
"periodic"

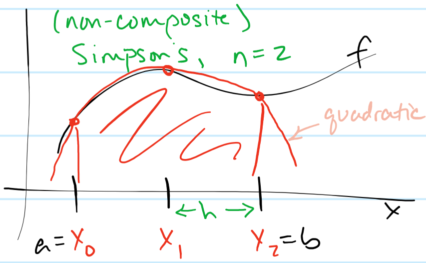
p. 4 (composite Simpson's rule)

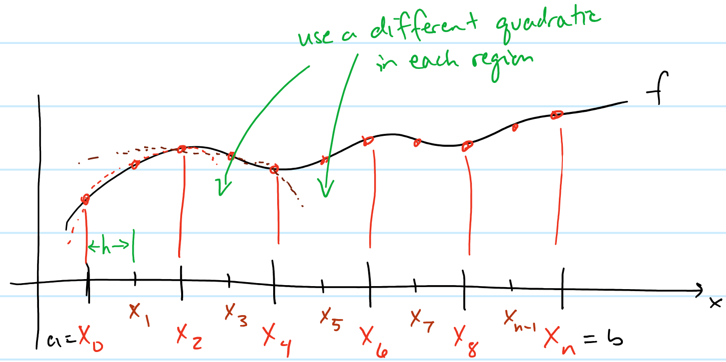
Monday, November 17, 2025 6:22 PM

Composite Simpson's Rule

The "industry standard", ie., your goto method for all but very tricky (highly oscillatory) functions

We'll use book's notation
(non-composite) Simpson's, $n=2$





(closed, $h = \frac{b-a}{n} = \frac{b-a}{2}$)

⚠ Need n even
like midpt. rule

$$I = \int_a^b f(x)dx = \int_{x_0}^{x_2} f(x)dx + \int_{x_2}^{x_4} f(x)dx + \cdots + \int_{x_{n-2}}^{x_n} f(x)dx$$

$$= \frac{h}{3}\left(f(x_0)+4f(x_1)+f(x_2)\right) + \frac{h}{3}\left(f(x_2)+4f(x_3)+f(x_4)\right) + \ldots + \frac{h}{3}\left(f(x_{n-2})+4f(x_{n-1})+f(x_n)\right)$$
$$- \frac{h^5}{90} f^{(4)}(\xi) \qquad - \frac{h^5}{90} f^{(4)}(\tilde{\xi}) \qquad - \frac{h^5}{90} f^{(4)}(\tilde{\tilde{\xi}})$$

appears twice

$$= \frac{h}{3}\left(f(x_0) + 2\sum_{j=1}^{n/2-1} f(x_{2j}) + 4\sum_{j=1}^{n/2} f(x_{2j-1}) + f(x_n)\right) - \frac{b-a}{180} h^4 f^{(4)}(\eta)$$

(*) FORMULA
Composite Simpson's

for some $\eta \in (a,b)$

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

Thm 4.4

Btw, observe that the weights for non-composite Simpson's add up to 2

... which makes sense since it integrates constant functions exactly, and

$$2 = \int_0^2 1\, dx = \frac{h^{(h=1)}}{3}(1+4+1)$$

since exact

wts

p. 5 (stability)

Monday, November 17, 2025 6:22 PM

Stability

We saw that for differentiation, finite $\underbrace{\text{difference schemes}}_{\text{algorithms}}$ were unstable

i.e., total error was like

$$\underbrace{\frac{e}{h}}_{\substack{\text{roundoff}\\ \text{(computer)}}} + \underbrace{h^k}_{\substack{\text{approximation}\\ \text{(math)}}}$$

k = order of method (1, 2, 3, etc)

error $f(x) - \underbrace{\tilde{f}(x)}_{\text{floating pt.}}$

So we couldn't take $h \to 0$ because roundoff error $\to \infty$ then.

For integration, things are better: it is stable in general.

Ex: let each computation have error e_i, $\tilde{f}(x_i) = f(x_i) + e_i$
with each $|e_i| \le \varepsilon$ for some small ε

Consider Composite Simpson's Rule

$$\frac{h}{3}\left(f(x_0) + e_0 + 2 \sum_{j=1}^{n/2-1} f(x_{2j}) + e_{2j} + 4 \sum_{j=1}^{n/2} f(x_{2j-1}) + e_{2j-1} + f(x_n) + e_n \right)$$

So error is $\frac{h}{3}\left(e_0 + 2 \sum_{j=1}^{n/2-1} e_{2j} + 4 \sum_{j=1}^{n/2} e_{2j-1} + e_n \right)$

Use triangle inequality, $|a+b| \le |a| + |b|$, and $\left|\sum a_i\right| \le \sum |a_i|$

So $|\text{error}| \le \frac{h}{3}\left(\varepsilon + \underbrace{2\left(\frac{n}{2}-1\right)}_{n}\varepsilon + \underbrace{4\left(\frac{n}{2}\right)}_{2n}\varepsilon + \varepsilon \right)$

cancel

$= \frac{h}{3}(3n\varepsilon) = nh\varepsilon$ and $h = \frac{b-a}{n}$

$= (b-a)\varepsilon$

or... $\int_a^b \varepsilon \, dx = \varepsilon \cdot (b-a)$ since we know it integrates constant functions exactly

So we make roundoff error, but it's not too large (if f is well-conditioned and x, $f(x)$ near 1, then $\varepsilon \approx \varepsilon_{\text{machine}} \approx 10^{-16}$)

and most importantly, roundoff error does not increase as $h \to 0$ or $n \to \infty$.

So we can just pick h very small (n very large) to make the "mathematical" error as small as we like. (...if we use composite rules)

(For non-composite rules, roundoff gets worse if h is too small)