8 Numerical Integration and Differentiation

8.1 Numerical Integration ('Quadrature')

Given a, b, f: Compute (an approximation of) $\int_a^b f(x)\,dx$

Existence, uniqueness: f "integrable"
(Riemann, Lebesgue)
pw. continuous, bounded $\Rightarrow$ exists, unique

Conditioning:

$\hat{f}(x) := f(x) + e(x)$

$$\left| \int_a^b f(x)\,dx - \int \hat{f}(x)\,dx \right|$$

$$= \left| \int_a^b e(x)\,dx \right|$$

$$\leq \int_a^b |e(x)|\,dx$$

$$\leq (b-a) \underbrace{\max_{x\in[a,b]} |e(x)|}_{\|e\|_\infty}$$

abs. condition number?
(upper bound? achieved?)

LEFT OUT

Relative cond #?

$$\frac{|\Delta \text{Result}| / |\text{Result}|}{\|\Delta \text{Input}\| / \|\text{Input}\|} \overset{\text{here}}{=} \frac{\left|\int_a^b f(x) - \hat{f}(x)\,dx\right| / \left|\int_a^b f(x)\,dx\right|}{\|e\|_\infty / \|f\|_\infty}$$

$$\leq \frac{(b-a)\cancel{\|e\|_\infty} / \left|\int_a^b f(x)\,dx\right|}{\cancel{\|e\|_\infty} / \|f\|_\infty} = \frac{(b-a)\|f\|_\infty}{\left|\int_a^b f(x)\,dx\right|}$$

8.2 Quadrature methods

Idea: Result ought to be a linear combination of a few function values

$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i) \quad \leftarrow (x_i) \text{ and } (w_i)$$

make a quadrature rule

0 Why linear combination?

Any interpolation method (nodes + basis) gives rise to an interpolatory quadrature

Example: Fix (x_i).

Lagrange polynomial

$$f(x) \approx \sum_i f(x_i)\, \ell_i(x)$$

Then $$\int_a^b f(x)\,dx \approx \int_a^b \sum_i f(x_i)\,\ell_i(x)\,dx$$

$$= \sum_i f(x_i) \underbrace{\int_a^b \ell_i(x)\,dx}_{w_i :=}$$

ugh too hard 3

1 With polynomials and (often) equispaced nodes, called Newton-Cotes quadrature.

2 With Chebyshev nodes and Chebyshev weights, called Clenshaw-Curtis quadrature.

0 [Want easier way to find weights (w_i).

<u>Idea:</u> Knowing the answer for a few functions is enough!

$$1 \longleftarrow \qquad \longrightarrow 2$$

$$b-a = \int_a^b 1\,dx = w_1 \cdot 1 + \dots + w_n \cdot 1$$

$$\frac{1}{k+1}\left(b^{k+1} - a^{k+1}\right) = \int_a^b x^k\,dx = w_1 \cdot x_1^k + \dots + w_n \cdot x_n^k$$

$\rightarrow$ Linear system we can solve!

"Method of undetermined coefficients"

Demo

Examples

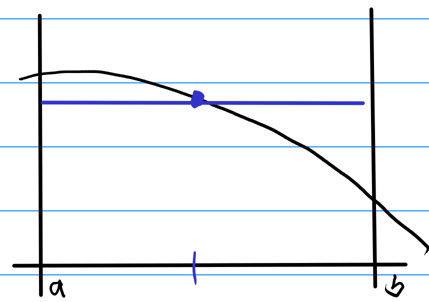
1

2
(ask)

Exact up to polynomial degree

Midpoint rule

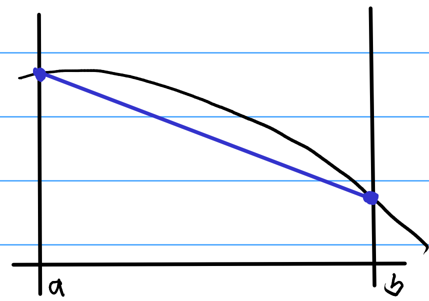
$$(b-a)\,f\left(\frac{a+b}{2}\right)$$



$\cancel{0}$ 1

Trapezoidal rule

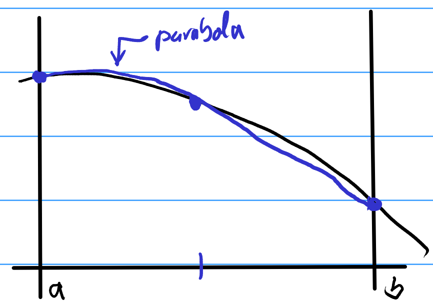
$$\frac{(b-a)}{2}\left(f(a)+f(b)\right)$$



1

Simpson's rule

$$\frac{(b-a)}{6}\left(f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right)$$



$\cancel{2}$ 3

Idea: Difference between trapezoidal rule and midpoint rule is a measure of the error

8.2 Accuracy and stability (for interpolatory Q)

Let p_n be interpolant of f at nodes $x_1, \dots, x_n$. (deg $n-1$) ask

Recall: $\sum_i w_i f(x_i) = \int_a^b p_n(x)\,dx$

Accuracy: $\left|\int_a^b f(x)\,dx - \int_a^b p_n(x)\,dx\right|$

$$\leq \int_a^b |f(x) - p_n(x)|\,dx$$

$$\leq (b-a)\,\|f - p_n\|_\infty$$

$$\underset{\text{Intp. error}}{\leq} (b-a)\frac{h^n}{4n}\|f^{(n)}\|_\infty$$

$$\leq \frac{1}{4} h^{n+1}\|f^{(n)}\|_\infty \longleftarrow$$

Quadrature gets one order "higher" than interpolation

0

Stability $f(x)$ $\quad \hat{f}(x) = f(x) + e(x)$

$$\left|\sum_i w_i f(x_i) - \sum_i w_i \hat{f}(x_i)\right|$$

$$= \left|\sum_i w_i e(x_i)\right|$$

$$\leq \sum_i |w_i e(x_i)|$$

$$\leq \underbrace{\left(\sum_i |w_i|\right)}_{\text{abs. condition number}} \|e\|_\infty$$

WS26p1

0 So when are quadrature weights bad?

Demo: N-C with many nodes

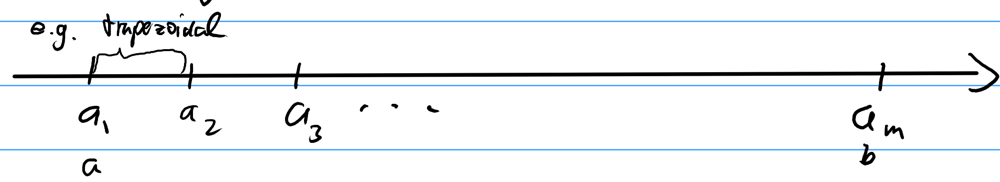
[In fact, N-C must have neg. weight as soon as $n \geq 11$.

So, what is not to like about Newton-Cotes?

- all the fun of high-order interpolation w/ monomials & equispaced (convergence not guaranteed)
- weights wiggle
- coefficients determined using Vdm
- Hard to extend to arbitrary number of points

(8.3) Composite quadrature

Idea: String together a few quadrature rules



Error: single-interval: $\left| \int f - p_n \right| \leq C\, h^n \, \|f^{(n)}\|_\infty$

$$\left| \int_a^b f(x)\,dx - \sum_{j=1}^{m} \sum_{i=1}^{n} w_{ji}\, f(x_{ji}) \right|$$

$$\leq C \, \|f^{(n)}\|_\infty \sum_{j=1}^{m} \left(\frac{a_{j+1} - a_j}{n} \right)^n$$

$$\approx C \, \|f^{(n)}\|_\infty \sum_{j=1}^{m} \frac{a_{j+1} - a_j}{n} \left(\frac{a_{j+1} - a_j}{n} \right)^{n-1}$$

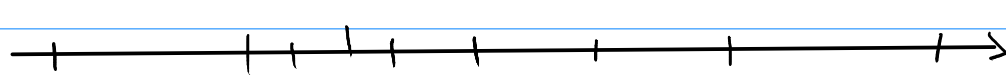
$$\approx C \, \|f^{(n)}\|_\infty \, (b-a) \left(\frac{1}{n}\right) \left(\frac{1}{nm} \right)^{n-1}$$

meh.

I.e. composite quadrature loses an order, compared to un-composite one.

Recall: Reducing interval size reduces error.

Idea: If we can estimate errors on each subinterval (how?), only subdivide intervals with insufficient error.



"adaptive quadrature"

8.4 Gaussian quadrature

So far: nodes prescribed

now: let quad. rule determine nodes

hope: more design freedom → exact to higher degree

Could use method of undet. coefficients

but: system would be nonlinear

too hard.

Alternative: can use orth. polynomials!

$p(x)$: polynomial of degree n with

$$\int_a^b p(x) x^k = 0 \qquad k=0,\dots,n-1$$

i.e. orthogonal to monomials of these orders

Then (→ hw5)

- p has n simple, real roots in (a,b).
- using these roots as interpolation nodes yields an interpolatory quadrature that is exact up to degree $2n-1$.

Demo WS27p1

<lec29>

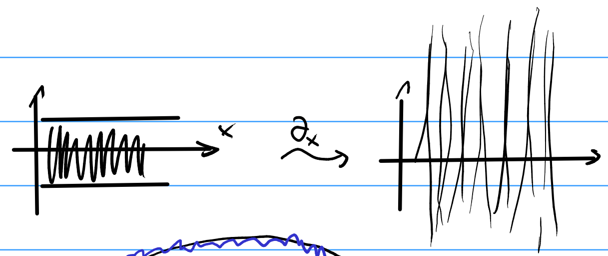
<u>Double integrals:</u>

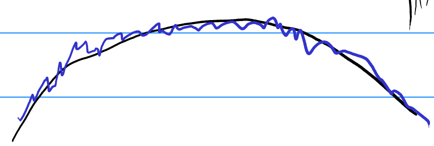
$$\int_0^1\int_0^1 f(x,y)\,dx\,dy \approx \int_0^1 \sum_i w_i f(x_i, y)\,dy$$

$$\approx \sum_j w_j \sum_i w_i f(x_i, x_j)$$

8.5 Numerical Differentiation

⚠ Don't do it. (if you can)

- Unbounded: 

- Amplifies noise: 

- Subject to cancellation error

- Inherently less accurate than integration

For polynomial interpolant of degree p:

- interpolation error $\sim h^{p+1}$
- quadrature error $\sim h^{p+2}$
- differentiation error $\sim h^{p}$

!

[If necessary, how?

<u>Ideas:</u>

- Compute interpolation coefficients, diff. basis

- "Finite differences"

Finite differences

$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$ ("forward difference(s)")

(bw differences?)

Taylor: $f(x+h) = f(x) + f'(x)h + f''(x)\frac{h^2}{2} + \dots$

$$= f(x) + f'(x)h + O(h^2)$$

plug in: $\frac{f(x+h) - f(x)}{h}$

$$= \frac{[\cancel{f(x)} + f'(x)h + O(h^2)] - \cancel{f(x)}}{h}$$

$$= f'(x) + O(h)$$

→ first-order accurate

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2)$$ ("centered difference(s)")

→ second-order accurate

Can derive by trying to match Taylor terms.
But: Interpolate-then-diff. usually easier.

Demo

8.6 Richardson extrapolation

Suppose we have p-th order approximation $\tilde{F}$ to F

$$F = \tilde{F}(h) + O(h^p)$$

Pull out one more term in error:

$$F = \tilde{F}(h) + a\,h^p + O(h^q) \qquad (q > p, \text{ usually } q = p+1)$$

Compute for two values of h: $\tilde{F}(h_1)$ $\tilde{F}(h_2)$

Find numbers α, β s.t.

$$F = \alpha \tilde{F}(h_1) + \beta \tilde{F}(h_2) + O(h^q)$$

$$\alpha a h_1^p + \beta a h_2^p \overset{!}{=} 0$$

Also require: $\alpha + \beta \overset{!}{=} 1$

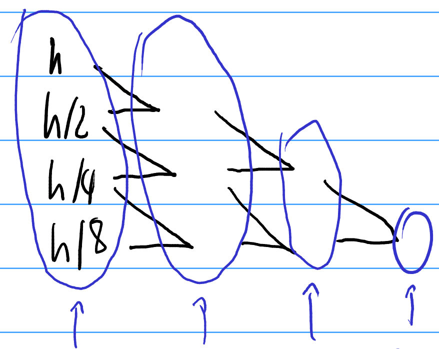
solve: $\alpha = \dfrac{h_2^p}{h_2^p - h_1^p}$ $\qquad \beta = 1 - \alpha$

Do not need to know a!

WS28p1 Demo

Can repeat for even higher accuracy.

Example:



e.g. 1st 2nd 3rd 4th order

← This for quadrature:

Romberg integration