13. Numerical integration (quadrature)

Last time

- Finite differences for numerical derivatives
- Taylor series
- Interpolation

Goals for today

- Numerical integration (quadrature)
- Error analysis

Need for numerical integration

- Recall: (definite) integrals are **more difficult** than derivatives
- No analytical solution in general

Need for numerical integration

- Recall: (definite) integrals are **more difficult** than derivatives
- No analytical solution in general
- e.g. **error function**

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} \, dt$$

Need for numerical integration

- Recall: (definite) integrals are **more difficult** than derivatives
- No analytical solution in general
- e.g. **error function**

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2}\, dt$$

- Hence numerical integration is of paramount importance

Numerical integration (quadrature) problem

- Numerical integration problem:
- Given a *function* f
  Calculate $I(f) := \int_a^b f(x)\,dx$

Numerical integration (quadrature) problem

- Numerical integration problem:
- Given a *function* f
  Calculate $I(f) := \int_a^b f(x)\,dx$
- Want approximation

$$\int_a^b f(x) \simeq \sum_i w_k f(t_k)$$

- **Quadrature nodes (points)** t_k
- **Quadrature weights** w_k

Numerical integration II

- Approximation

$$\int_a^b f(x) \simeq \sum_i w_k f(t_k)$$

- Weights w_k should be **independent** of f
- Should work for “any” f

Numerical integration II

- Approximation

$$\int_a^b f(x) \simeq \sum_i w_k f(t_k)$$

- Weights w_k should be **independent** of f
- Should work for “any” f

- If given a function, can choose nodes and weights

Numerical integration II

- Approximation

$$\int_a^b f(x) \simeq \sum_i w_k f(t_k)$$

- Weights w_k should be **independent** of f
- Should work for “any” f

- If given a function, can choose nodes and weights
- Note: Integral and approximation are both **linear** operations

Simplest case

- What is simplest approximation of f?

Simplest case

- What is simplest approximation of f?
- Approximate f using **rectangles** – rectangular rule
- As in Riemann integration

Simplest case

- What is simplest approximation of f?
- Approximate f using **rectangles** – rectangular rule
- As in Riemann integration

- Split $[a, b]$ into N intervals (or **panels**) of length $h = \frac{b-a}{N}$
- $t_k := a + k\,h$ (with $t_0 = a$ and $t_N = b$)

Rectangular rule II

- Approximate f by piecewise-constant function p
- Choose value of p for each subinterval X_k

Rectangular rule II

- Approximate f by piecewise-constant function p
- Choose value of p for each subinterval X_k

- e.g. $p(x) = f(t_k)$ for $x \in X_k := [t_k, t_{k+1})$
- So $p(x) = \sum_k f(t_k) \mathbb{1}_{X_k}(x)$
- Where $\mathbb{1}_{X_k}$ is **indicator function** of set

  = 1 if $x \in X_k$ and 0 if not

Rectangular rule III

- Area A_k of kth rectangle is $hf(t_k)$
- So $I(f) \simeq A(f,h) := h\sum_k f(t_k)$

Rectangular rule III

- Area A_k of kth rectangle is $hf(t_k)$
- So $I(f) \simeq A(f,h) := h\sum_k f(t_k)$

- Weights $w_k = h$ except $w_N = 0$

Error of rectangular rule

- How good is rectangular rule?
- Calculate **error** $E(h) := |A(f, h) - I(f)|$

Error of rectangular rule

- How good is rectangular rule?
- Calculate **error** $E(h) := |A(f,h) - I(f)|$

- Taylor: $f(x) = f(t_k) + (x - t_k) f'(\xi(x))$ in kth interval

Error of rectangular rule

- How good is rectangular rule?
- Calculate **error** $E(h) := |A(f, h) - I(f)|$

- Taylor: $f(x) = f(t_k) + (x - t_k)f'(\xi(x))$ in kth interval

- Suppose $|f'|$ is bounded in X_K by M_k
- Then $|f(x) - p(x)| \leq M_k h$ in X_k

Error of rectangular rule II

- $|f(x) - p(x)| \leq Mh$ for x in X_k
- So $E_k := |\int_{X_k}(f-p)| \leq Mh^2$

Error of rectangular rule II

- $|f(x) - p(x)| \leq Mh$ for x in X_k
- So $E_k := |\int_{X_k}(f-p)| \leq Mh^2$

- We have $N \sim 1/h$ subintervals
- So global error in integral is $E(fh) = \sum_k E_k$

$$E(h) = \int_a^b [f(x) - p(x)] = \mathcal{O}(h)$$

Improving the approximation

- How can we improve this, i.e. reduce error?

Improving the approximation

- How can we improve this, i.e. reduce error?
- Use better approximation: piecewise linear

Improving the approximation

- How can we improve this, i.e. reduce error?
- Use better approximation: piecewise linear

- $p_1(x) = \ell_0(x) f(a) + \ell_1(x) f(b)$ using Lagrange cardinal polynomials
- So $p_1(x) = \frac{x-b}{a-b} f(a) + \frac{x-a}{b-a} f(b)$

Trapezium rule

- Need to integrate p_1:
- $\int_a^b p_1(x)\,dx = f(a)\int \ell_0 + f(b)\int \ell_1$
- Obtain $\int_a^b p_1(x)\,dx = \frac{1}{2}(b-a)[f(a)+f(b)]$

Trapezium rule

- Need to integrate p_1:
- $\int_a^b p_1(x)\,dx = f(a)\int \ell_0 + f(b)\int \ell_1$
- Obtain $\int_a^b p_1(x)\,dx = \frac{1}{2}(b-a)[f(a)+f(b)]$

- A_k is now area of **trapezium**, $A_k = \frac{h}{2}[f(t_k)+f(t_{k+1})]$
- Total area $A(h) = \frac{h}{2}[f(a)+2f(t_1)+\cdots+2f(t_{k-1})+f(b)]$

Interpolation: Newton–Cotes

- Rectangular and trapezium rules: *interpolate* then *integrate*
- **Newton–Cotes** rules (equally-spaced nodes)

Interpolation: Newton–Cotes

- Rectangular and trapezium rules: *interpolate* then *integrate*
- **Newton–Cotes** rules (equally-spaced nodes)

- Can show: interpolation error by degree-n polynomial p_n is

$$f(x) = p_n(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!}\pi_n(x)$$

where $\pi_n(x) = \prod_{k=0}^{n}(x - t_k)$

Interpolation: Newton–Cotes

- Rectangular and trapezium rules: *interpolate* then *integrate*
- **Newton–Cotes** rules (equally-spaced nodes)

- Can show: interpolation error by degree-n polynomial p_n is

$$f(x) = p_n(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!}\pi_n(x)$$

where $\pi_n(x) = \prod_{k=0}^{n}(x - t_k)$

- Note that interpolation error $= 0$ at nodes!

Error for Newton–Cotes rules

- Leads to estimates for error when integrating interpolant
- $|\int f - \int p_n| \leq \frac{M_{n+1}}{(n+1)!} |\int \pi_n|$

  where M_{n+1} is a bound for $|f^{(n+1)}|$ on $[a, b]$

Error for Newton–Cotes rules

- Leads to estimates for error when integrating interpolant
- $|\int f - \int p_n| \leq \frac{M_{n+1}}{(n+1)!} |\int \pi_n|$

  where M_{n+1} is a bound for $|f^{(n+1)}|$ on $[a, b]$
- E.g. trapezium rule has error $\mathcal{O}(h^2)$

Alternative method: Integration by parts

- Recall integration by parts formula:

$$\int_a^b u(x)v'(x)\,dx = [u(x)v(x)]_a^b - \int_a^b u'(x)v(x)\,dx$$

- Error in one interval for rectangular rule: Take $v(x) = x$

$$\int_0^h [f(x) - p(x)]\,dx = [(f(x) - p(x))\,x]_a^b - \int_a^b f'(x)\,x\,dx$$

- If $|f'| \leq M$ then get bound $Mh^2/2$ on error

Conditioning of numerical integration

- Is quadraturcondition number of the quadrature problem?
- Input: f; output: $I(f) = \int_a^b f$

Conditioning of numerical integration

- Is quadraturcondition number of the quadrature problem?
- Input: f; output: $I(f) = \int_a^b f$
- Perturb input function f by function Δf
- Output perturbation ΔI is

$$\Delta I = I(f + \Delta f) - I(f)$$

Conditioning of numerical integration

- Is quadraturcondition number of the quadrature problem?
- Input: f; output: $I(f) = \int_a^b f$
- Perturb input function f by function Δf
- Output perturbation ΔI is

$$\Delta I = I(f + \Delta f) - I(f)$$

- So $\Delta I = I(\Delta f)$

Conditioning II

- $\Delta I = I(\Delta f)$
- So

$$|\Delta I| = \left| \int \Delta f \right| \leq \int |\Delta f| =: \|\Delta f\|_1$$

- Relative error:

$$\left| \frac{\Delta I}{I} \right| \leq \frac{\|\Delta f\|_1}{|I|}$$

Conditioning III

- So relative condition number

$$\kappa = \frac{|\Delta I\ |/|I|}{\|\Delta f\|/\|f\|} = \frac{\|f\|_1}{|I|}$$

- So

$$\kappa = \frac{\int_a^b |f(x)|\, dx}{\left|\int_a^b f(x)\, dx\right|}$$

- Ill conditioned when $|f|$ is large but $\int f$ is small
- I.e. when integral of **highly oscillatory** function

Summary

- Numerical integration approximates definite integral
- Interpolate then integrate
- Degree n polynomial leads to $\mathcal{O}(h^{n+1})$ error