Gaussian Quadrature

So far: nodes chosen from outside.
Can we gain something if we let the quadrature rule choose the nodes, too? **Hope:** More design freedom $\rightarrow$ Exact to higher degree.

**Demo:** Gaussian quadrature weight finder [cleared]
Composite Quadrature

High-order polynomial interpolation requires a high degree of smoothness of the function.

Idea: Stitch together multiple lower-order quadrature rules to alleviate smoothness requirement.

e.g. trapezoidal
Error in Composite Quadrature

What can we say about the error in the case of composite quadrature?

Error for one panel:

$$\left| \int_a^b f(x) \, dx - \sum_{j=1}^{m} \sum_{i=1}^{n_j} w_j f(x_{ij}) \right| \leq C \| f^{(r)} \|_{\infty} \cdot h^{n+1}$$

$$\leq C \| f^{(r)} \|_{\infty} \cdot \sum_{j=1}^{m} (a_j - a_{j-1})^{n+1}$$

$$\leq C \cdot \| f^{(r)} \|_{\infty} \cdot \sum_{j=1}^{m} (a_j - a_{j-1})^n (a_j - a_{j-1}) \leq h^n$$

$$\leq C \cdot \| f^{(r)} \|_{\infty} \cdot (b-a) \cdot \frac{1}{n^{n+1}}$$
Composite Quadrature: Notes

Observation: Composite quadrature loses an order compared to non-composite.

Idea: If we can estimate errors on each subinterval, we can shrink (e.g. by splitting in half) only those contributing the most to the error. (adaptivity, → hw)
Taking Derivatives Numerically

Why *shouldn’t* you take derivatives numerically?

**Demo:** Taking Derivatives with Vandermonde Matrices [cleared]
Taking Derivatives Numerically

Why *shouldn’t* you take derivatives numerically?

- "unbounded"

\[ \| [\sin(kx)]' \|_\infty = k \| \cos(kx) \|_\infty = k \]

\[ \| \sin(kx) \|_\infty = 1 \]

- no *ise amplification* 

\[ \hat{f} = f + e \quad \| \hat{f}' \| \geq \| e \| \]

- cancellation error

\[ f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \]

- less accurate

interp: \( \sim h^n \)  
quad: \( \sim h^{n+1} \)

diff \( \sim h^{n-1} \)

**Demo:** Taking Derivatives with Vandermonde Matrices [cleared]
Finite Differences

\[ f'(x) \approx \frac{f(x+h) - f(x)}{h} \]

forward difference.

first order.

\[ f(x+h) = f(x) + h \cdot f'(x) + O(h^2) \]

\[ \Rightarrow \text{err} = O(h) \]
More Finite Difference Rules

Similarly:

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

(Centered differences)

Can also take higher order derivatives:

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

Can find these by trying to match Taylor terms.
Alternative: Use linear algebra with interpolate-then-differentiate to find FD formulas.

**Demo:** Finite Differences vs Noise [cleared]
**Demo:** Floating point vs Finite Differences [cleared]