Lecture 8: Numerical integration

CSC 338: Numerical Methods

Ray Wu

University of Toronto

March 8, 2023

Numerical integration

- We consider computing definite integrals in one dimension, that is, approximate

$$I = \int_a^b f(x)\,dx \tag{1}$$

- Specifically, we approximate I with a finite sum, that is

$$I \approx \sum a_j f(x_j) \tag{2}$$

  - x_j are the **abscissae**.
  - a_j are the **weights**.

Basic rules

- Basic rules are defined on only the interval of integration $[a, b]$.
- When the interval is partitioned, then we have **composite numerical integration** (next subsection).
- Basic rules are defined based on polynomial interpolation: we choose $x_0, x_1, \ldots, x_n$, interpolate a polynomial through these points, and integrate the polynomial exactly.

Deriving Basic rules

- Assume that the interpolating polynomial is in Lagrange form:

$$p_n(x) = \sum f(x_j)L_j(x) \tag{3}$$

- Then,

$$\begin{aligned}\int_a^b f(x)\ dx &\approx \int_a^b p_n(x)\ dx = \int_a^b \sum f(x_j)L_j(x)\ dx \\ &= \sum f(x_j) \int_a^b L_j(x)\ dx\end{aligned}$$

- In other words,

$$a_j = \int_a^b L_j(x)\ dx \tag{4}$$

Basic rules – Trapezoid rule

- We select $n = 1$ (linear interpolant).
- This gives us $x_0 = a, x_1 = b$, and $f(x_0), f(x_1)$.
- We have

$$L_0(x) = \frac{x-b}{a-b} \quad L_1(x) = \frac{x-a}{b-a} \tag{5}$$

- Integrating,

$$a_0 = \int_a^b \frac{x-b}{a-b}\, dx = \frac{b-a}{2} \tag{6}$$

$$a_1 = \int_a^b \frac{x-a}{b-a}\, dx = \frac{b-a}{2} \tag{7}$$

- Resulting trapezoid rule:

$$I_{\text{trap}} = \frac{b-a}{2}(f(a) + f(b)) \tag{8}$$

Simpson's rule

- Instead of interpolating with a line, consider interpolating with a quadratic, so we have three points x_0, x_1, x_2.
- This gives rise to **Simpson's rule**, which is given by

$$I_{simp} = \frac{b-a}{6}[f(a) + 4f(\frac{a+b}{2}) + f(b)]. \tag{9}$$

Newton-Cotes formulas

- Trapezoidal and Simpson's rules are examples of Newton-Cotes formulas.
- Based on polynomial interpolation at equidistant abscissae
- If we include the endpoints, the formula is closed. Otherwise, it is open.

Error in basic rules

- What is the error in these basic rules?
- Recall that the error of polynomial interpolation is given by

$$f(x) - p_n(x) = f[x_0, x_1, \ldots, x_n, x] \prod_{i=0}^{n} (x - x_i) \tag{10}$$

- To compute the quadrature error, integrate the error over the entire domain:

$$E = \int_a^b f[x_0, x_1, \ldots, x_n, x] \prod_{i=0}^{n} (x - x_i) \, dx \tag{11}$$

Error in Trapezoid

- From the trapezoid rule:

$$E = \int_a^b f[a,b,x](x-a)(x-b)\ dx \tag{12}$$

- By IVT and nonpositivity of $(x-a)(x-b)$, there is some value ξ such that

$$E = f[a,b,\xi] \int_a^b (x-a)(x-b)\ dx \tag{13}$$

- Additionally, there exists some value η such that $f[a,b,\xi] = f''(\eta)/2$, the integral evaluates to $-\frac{1}{6}(b-a)^3$, so the basic trapezoid rule has error

$$E = \frac{f''(\eta)}{12}(b-a)^3 \tag{14}$$

Error in Simpson's rule

- Using a similar derivation, the error in Simpson's rule can be shown to be

$$-\frac{f''''(\zeta)}{90}\left(\frac{b-a}{2}\right)^5 \tag{15}$$

- For the derivation, see p. 445 of Ascher & Greif.
- How do we reduce the error?
  - We can change the x_i to nonuniform gridpoints. However, if we need to sample many data points, this again goes back to high-degree polynomial interpolation – which is not guaranteed to have good results.
  - Far more simple and stable is **composite integration** - just reduce the interval of integration $[a, b]$.

Composite integration

- Choose a partition of $[a, b]$ and apply a basic rule to each subinterval.
- For simplicity, choose a uniform partition: divide $[a, b]$ into r subintervals of size $h = (b - a)/r$ each.
- Then, we apply the integration rules to each of the r subintervals directly and add them up

$$\int_a^b f(x)\,dx = \sum_{i=1}^{r} \int_{t_{i-1}=a+(i-1)h}^{t_i=a+ih} f(x)\,dx \tag{16}$$

- The associated error is the sum of the errors on each interval.
  - Suppose we use some basic rule that has an error term $K(b-a)^{q+1}$
  - Then, each subinterval has an error contribution $K_i h^{q+1}$.
  - Since there are $r = (b-a)/h$ of these subintervals, then the total error is given by Kh^q

Composite trapezoidal integration

- Recall that the trapezoidal rule gives

$$\int_{t_{i-1}}^{t_i} f(x)\,dx \approx \frac{h}{2}(f(t_{i-1}) + f(t_i)). \tag{17}$$

- Hence, the composite trapezoidal method is

$$\int_a^b f(x)\,dx \approx \frac{h}{2}[f(a) + 2f(t_1) + \cdots + 2f(t_{r-1}) + f(b)]. \tag{18}$$

- The error on each subinterval is $\mathcal{O}(h^3)$, hence, the total error is $\mathcal{O}(h^2)$
- In other words, if you double the number of subintervals, you reduce the error to a quarter of the previous size.

Composite Simpson

- Again, we partition the interval $[a, b]$ into subintervals of equal size. However, this time, we denote the length of each subinterval by $2h$ instead of h.
  - This is because we need to evaluate also the midpoints of each subinterval in Simpson's method.
- On each of the subintervals, apply Simpson's rule. Then we have

$$\int_a^b f(x)\,dx \approx \frac{h}{3}\left[f(a) + 4\sum_{i \text{ odd}} f(t_i) + 2\sum_{i \text{ even}} f(t_i) + f(b)\right] \tag{19}$$

- Each subinterval has an $\mathcal{O}(h^5)$ contribution to the error, hence, the total error of composite Simpson is $\mathcal{O}(h^4)$ (assuming the fourth derivative is bounded).
- This means if you double the number of subintervals, the error is reduced to $1/16$ of the previous size.

Example of composite integration

- Function to integrate: $y = \sin(x)$.
- Interval: $[0, \pi]$.
- Rest of demo on blackboard

Second example of composite integration

- Function to integrate: $y = \sqrt{x}$.
- Interval: $[0, 1]$.
- Rest of demo on blackboard

Gaussian Quadrature

- Gaussian Quadrature is one way to intelligently choose nonuniform points of integration
- Precision of a method: the highest degree polynomial that can be integrated exactly.
- Another closely related method is to use the Chebyshev points – leading to Clenshaw-Curtis rules
- How can we intelligently choose the points of integration?
- **Orthogonal polynomials** will help us.

Vector spaces

- What is a vector space?
- Set of vectors, must satisfy two properties:
    1. If u and v are elements of a vector space V, then so must $u + v$.
    2. If u is an element in a vector space V and α is a real number, so must αu.
- Do vector spaces have to be comprised of vectors?

Function spaces

- Suppose we define a set of function F.
- As long as our elements f and g in F satisfy the two vector space properties, it's still a vector space.
- Example: The space of linear splines.
  1. If you scale a linear spline by a constant, it's still a linear spline.
  2. If you add two linear splines, it's still a linear spline.

Norms, Inner products, and orthogonality of functions

- Norms for functions are similar to norms of vectors. If g is a function, then

$$\|g\| = \|g\|_2 = \left(\int_a^b (g(x))^2 \; dx \right)^{1/2} \tag{20}$$

$$\|g\|_1 = \int_a^b |g(x)| \; dx \tag{21}$$

$$\|g\|_\infty = \max_{x \in [a,b]} |g(x)| \tag{22}$$

- Inner product of two functions f and g is defined as

$$\langle f, g \rangle = \int_a^b f(x) g(x) \; dx \tag{23}$$

- Two functions f and g are orthogonal to each other if their inner product is zero (just like vectors in $\mathbb{R}^n$).

Intuition of Gaussian quadrature

- Suppoes $f(x)$ is a polynomial of degree m. If we use $n \geq m$ points, then

$$f[x_0, x_1, x_2, \ldots, x_n, x] = \frac{f^{(n+1)(\zeta)}}{(n+1)!} = 0 \tag{24}$$

- If we are allowed to choose the $n+1$ points, then intuitively, we can increase the precision by $n+1$ to $2n+1$.
- For orthogonal polynomials $\phi_0(x), \phi_1(x), \ldots, \phi_{n+1}(x)$, we have

$$\int_a^b g(x)\phi_{n+1}(x)\ dx = 0 \tag{25}$$

  if g has degree $\leq n$.
- g can be written as a linear combination of basis functions $\phi_j(x)$.
- Orthogonality directly follows and so does the integral being zero.

Legendre Polynomials and Gaussian Quadrature

- We choose the **canoical interval** $[-1, 1]$.
- Other intervals can be obtained by scaling and shifting.
- Legendre polynomials are defined by the relation

$$\phi_0(x) = 1 \tag{26}$$

$$\phi_1(x) = x \tag{27}$$

$$\phi_{j+1}(x) = \frac{2j+1}{j+1} x\phi_j(x) - \frac{j}{j+1}\phi_{j-1}(x) \tag{28}$$

- These functions are orthogonal to each other.
- We pick the abscissae as the roots of these polynomials.
- The weights are obtained with integration, and are given by

$$a_j = \frac{2(1-x_j^2)}{[(n+1)\phi_n(x_j)]^2} \tag{29}$$

- The precision is $2n+1$.

Examples of Gaussian Quadrature

- Show derivation of 2 and 4 point Gaussian on blackboard.
- The rules for the canonical interval can be found at `https://en.wikipedia.org/wiki/Gaussian_quadrature`

Richardson Extrapolation

- It can be shown that the error term of composite trapezoidal rule is a sum of the **even powers** of h:

$$E = K_1 h^2 + K_2 h^4 + K_3 h^6 + \dots \tag{30}$$

- As a result, if we compute the integral twice with h and $h/2$ resulting in R_1 and R_2, then

$$E_1 = K_1 h^2 + K_2 h^4 + \dots$$
$$E_2 = (1/4) K_1 h^2 + (1/16) K_2 h^4 + \dots.$$

- Then, **Richardson Extrapolation** is the process of cancelling out the principle error term Kh^2: consider $(4R_2 - R_1)/3$, the associated error is

$$\frac{4E_2 - E_1}{3} = \frac{1}{3}(4((1/4)K_1 h^2 + (1/16)K_2 h^4) - (K_1 h^2 + K_2 h^4)) = 4K_2 h^4 \tag{31}$$

Romberg Integration

- Romberg integration is an iterative process where we repeatedly apply Richardson Extrapolation to cancel out lower and lower power terms
- we construct a triangular table of values:

$\mathcal{O}(h^2)$	$\mathcal{O}(h^4)$	$\mathcal{O}(h^6)$	$\dots$	$\mathcal{O}(h^{2s})$
$R_{1,1}$				
$R_{2,1}$	$R_{2,2}$			
$R_{3,1}$	$R_{3,2}$	$R_{3,3}$		
$\vdots$	$\vdots$	$\vdots$	$\ddots$	
$R_{s,1}$	$R_{s,2}$	$R_{s,3}$	$\dots$	$R_{s,s}$

- First row: computed with composite trapezoidal rule, double the gridpoints each time.
- Subsequent rows: use Richardson Extrapolation:

$$R_{j+1,k} = R_{j+1,k-1} + \frac{R_{j+1,k-1} - R_{j,k-1}}{4^{k-1} - 1} \tag{32}$$

Adaptive integration

- Suppose we are writing general-purpose software, and the user is not interested in technical questions such as
  - Which rule we are using
  - How many subintervals is appropriate
- Only obtain the necessary information: the function f to integrate, the interval $[a, b]$, and the accuracy ϵ required.
- Our function's job is to produce a number Q such that

$$|Q - I| \leq \epsilon \tag{33}$$

- For convenience, let's pick trapezoidal rule.

Error estimates

- We must be able to estimate the error. Without that, there is no guidance for how many subintervals we need.
- Recall that for composite rules, we have error given as

$$E = Kh^q + \mathcal{O}(h^{q+1}) \tag{34}$$

- The first term Kh^q is called the **principle error term**, and with two approximations we can estimate it:
  - Compute R_1 and R_2 with h and $h/2$ respectively.
  - Error in R_1 is approximately Kh^2 (using trapezoid rule)
  - Error in R_2 is approximately $\frac{1}{4}Kh^2$
  - Then we have

$$I - R_1 = (I - R_2) + (R_2 - R_1) \approx \frac{1}{4}(I - R_1) + (R_2 - R_1) \tag{35}$$

  - and we get the immediate error estimate

$$I - R_1 \approx \frac{4}{3}(R_2 - R_1), \quad I - R_2 \approx \frac{1}{3}(R_2 - R_1) \tag{36}$$

Adaptive subdivision

- Let I_i be the value of the integral on the i-th partition.
- Then, if we require that

$$|Q_i - I_i| < \frac{h_i}{b-a}\epsilon \tag{37}$$

  then summing over every subinterval, the left side becomes at most $|Q - I|$, and the right side becomes simply ϵ since $\sum h_i = b - a$.
- So the idea of adaptive integration is
  - Evaluate R_1 and R_2 for the partitions $[a, b]$ and $[a, \frac{a+b}{2}, b]$;
  - Estimate the error on each subinterval
  - If the error is small enough, then end the computation for the subinterval, otherwise, double the number of gridpoints, but **only** on the subintervals where the error is not small enough.

More on adaptive integration

- Generally, we want to use adaptive integration when we know that the function is not uniformly varying on the domain of integration.
  - One example would be functions that look like $\sin(1/x)$.
- If the error estimate fails, then the adaptive integration also fails. For example,

$$\int_0^1 f(x)\ dx = \int_0^1 \exp(-x)\sin(2\pi x)\ dx \tag{38}$$

  would fail, due to the fact that $f(0), f(1/2), f(1)$ are all zero.
- Iterative refinement of a grid locally is difficult to parallelize/vectorize, which may be a significant drawback in certain applications.