

Gaussian Quadrature

Tuesday, October 13, 2020 5:42 PM

Gaussian quadrature is a family of quadrature rules (Gauss-Legendre,

Chebyshev-Gauss, Gauss-Laguerre, Gauss-Hermite) that choose n nodes

such that the degree of exactness/accuracy/precision is as large as possible ($2n-1$).

Closely related to Gauss-Lobatto and Clenshaw-Curtis

Gauss-Legendre to compute $\int_a^b f(x) dx$

This will be an open formula (nodes do not include endpoints a, b) so (following Burden and Faires) we're going to enumerate the nodes

as $\{x_1, x_2, \dots, x_n\}$

Δ Our previous convention was $n+1$ nodes,
 $\{x_0, x_1, x_2, \dots, x_n\}$

Just like Newton-Cotes, this will be an interpolating formula, meaning the high-level idea is

- (1) polynomial interpolation of f on the nodes
- (2) integrate the polynomial

The big difference is that now we won't require uniformly spaced nodes, instead we'll pick the nodes to optimize a criterion

we'll use degree of exactness

For Newton-Cotes, recall

Name	# nodes	degree of exactness	error
midpoint rule (open),	1 node ($n=1$),	1 (lucky)	$h^3/3 f''(\xi)$
trapezoidal rule (closed),	2 nodes ($n=2$)	1	$-h^3/12 f''(\xi)$
Simpson's rule (closed),	3 nodes ($n=3$)	3 (lucky)	$-h^5/90 f^{(4)}(\xi)$
Simpson's 3/8 rule (closed),	4 nodes ($n=4$)	3	$-3h^5/80 f^{(4)}(\xi)$

Δ we used to call this " $n=1$ " since we were 0-based

How do we know degree of exactness?

Method 1 (general purpose but only a lower bound)

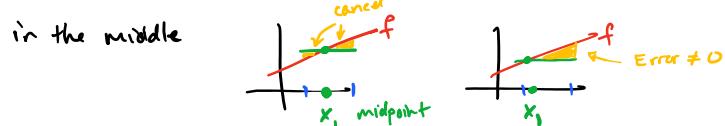
CW

For n nodes, we've seen the $(n-1)$ degree interpolating polynomial is unique
So if f itself is a $(n-1)$ degree polynomial, since it interpolates
itself, it must be p . Hence we integrate it exactly

\Rightarrow For any n nodes, interpolating quadrature has **degree of exactness**
at least $n-1$

But... doesn't explain why midpoint rule has degree of exactness 1

Here, we got lucky. It was because we used a node exactly
in the middle



Method 2 (more work to derive)

Derive an error estimate like $h^5 f^{(4)}(\xi)$

hence if $f^{(4)}(x) = 0 \ (\forall x) \Rightarrow$ no error.

If f is a n -degree polynomial, its $(n+1)$ derivative is 0
i.e., $f(x) = ax^2 + bx + c, f'''(x) = 0$

So, Newton-Cotes w/ n nodes gives **order of exactness** $n-1$ (if n even)
or n (if n odd)

Can we do better? We've been thinking of $\{x_1, \dots, x_n\} \subseteq [a, b]$
as given, and then find weights w_i s.t. our formula is

$$\sum_{i=1}^n w_i f(x_i)$$

What if we choose nodes $\{x_i\}$? This gives n more parameters,
so might hope we can get **order of exactness** $2n-1$.
In fact, we can!

Remember!

Def Gaussian quadrature means picking n nodes such that
order of exactness is $2n-1$

Example

$n=2$, find weights and nodes so that $w_1 f(x_1) + w_2 f(x_2)$ approximates
 $\int_{-1}^1 f(x) dx$ with **order of exactness** 3 ($= 2n-1$).

i.e., integrate exactly any $f(x) = a_3 x^3 + a_2 x^2 + a_1 x + a_0$, i.e., $\int_{-1}^1 f(x) dx = I_2(f)$
since $\int f(x) + g(x) dx = \int f(x) dx + \int g(x) dx$

$$\text{and } I_2(f+g) = I_2(f) + I_2(g) \quad \text{for generic } f, g$$

$$\text{and } \int a \cdot f(x) dx = a \cdot \int f(x) dx$$

$$I_2(a \cdot f) = a \cdot I_2(f),$$

ie., both $\int dx$ and I_2
are linear operators

it suffices to show:

$$(1) \int_{-1}^1 x^3 dx = I_2(x^3)$$

$$(2) \int_{-1}^1 x^2 dx = I_2(x^2)$$

$$(3) \int_{-1}^1 x dx = I_2(x)$$

$$(4) \int_{-1}^1 1 dx = I_2(1)$$

4 equations, 4 unknowns
 x_1, x_2, w_1, w_2

so... compute

$$(1) \int_{-1}^1 x^3 dx = 0 \quad (\text{by symmetry}) = w_1 \cdot x_1^3 + w_2 \cdot x_2^3$$

$$(2) \int_{-1}^1 x^2 dx = \frac{1}{3} x^3 \Big|_{-1}^1 = \frac{2}{3} = w_1 \cdot x_1^2 + w_2 \cdot x_2^2$$

$$(3) \int_{-1}^1 x dx = 0 \quad (\text{by symmetry}) = w_1 \cdot x_1 + w_2 \cdot x_2$$

$$(4) \int_{-1}^1 1 dx = 2 = w_1 + w_2$$

you can solve this (to make it more bearable, note
we have a symmetry, so expect $w_1 = w_2$ and $x_1 = -x_2$,

$$\text{so } (4) \quad 2 = w_1 + w_2 \quad \text{and symmetry } (w_1 = w_2) \Rightarrow w_1 = w_2 = 1$$

then (3) $\Rightarrow x_1 = -x_2$ (more symmetry) (and (4) is redundant)

$$\text{so } (2) \Rightarrow 2/3 = 2 \cdot x_1^2 \Rightarrow x_1 = -\sqrt{1/3} = -\sqrt{3}/3$$

$$x_2 = +\sqrt{1/3} = +\sqrt{3}/3$$

Making it more systematic

(*) **Theorem:** The nodes of the n -point Gauss-Legendre quadrature rule
are given by the roots of the n th Legendre polynomial P_n .

Before we prove this, we better say what P_n is!

Notation: let $P_n = \{ \text{all polynomials with degree } n \text{ or less} \}$
and recall from linear algebra that this is a vector space
of dimension $n+1$

Def Legendre polynomials $\{ P_0, P_1, P_2, \dots \}$ are polynomials such that

(1) $\forall p \in P_{n-1} \quad \int_{-1}^1 p(x) P_n(x) dx = 0 \quad \text{"orthogonality"}$

(2a) P_n is a polynomial of degree n

(2b) P_n is monic (= leading coefficient is 1)

(not too important, just a way to ensure uniqueness)

$$\text{i.e., } \text{Span}(\{P_0, \dots, P_n\}) = P_n$$

$\Rightarrow \{P_0, P_1, \dots, P_n\}$ is a basis for P_n

and in fact it's orthogonal by ①
(not orthonormal though)

$\{1, x, x^2, \dots, x^n\}$ ^{"monomial basis"} is also a basis for P_n , but not orthogonal and not as nice to work with

If we orthonormalize the monomial basis (e.g., Gram-Schmidt) then (up to scaling, ②b) we get the Legendre polynomials. So these are pretty fundamental.

Ex of Legendre polynomials

$$P_0(x) = 1$$

$$P_3(x) = x^3 - \frac{3}{5}x$$

$$P_1(x) = x$$

$$P_4(x) = x^4 - \frac{6}{7}x^2 + \frac{3}{35}$$

$$P_2(x) = x^2 - \frac{1}{3}$$

have many known facts, e.g.,

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n} (x^2 - 1)^n$$

$$P_n(x) = \frac{2n-1}{n} x \cdot P_{n-1}(x) - \frac{n-1}{n} P_{n-2}(x) \quad \text{3-term recurrence}$$

The roots of $P_n(x)$ can be efficiently found in $O(n^2)$ time by the 1969 Golub and Welsch algorithm (efficiently finds eigenvalues of a tridiagonal matrix) — before then it was hard to use in practice. (but known since 1814 by Gauss) NOT ON TEST

The weights of the Gauss-Legendre formula are easier:

form the interpolating polynomial p (Lagrange interpolating polynomial) and integrate that

⚠ Lagrange \neq Legendre

So...

Def. Gauss-Legendre quadrature for $\int_{-1}^1 f(x) dx$:

we'll do \int_a^b later

① pick nodes $\{x_1, \dots, x_n\}$ to be the roots of P_n Legendre polynomial

Remember

② pick weights $w_i = \int_{-1}^1 \prod_{j=1, j \neq i}^n \frac{x - x_j}{x_i - x_j} dx$

Fact: all roots are simple, real, and lie in $(-1, 1)$

Since our interpolating polynomial is $\sum_{i=1}^n f(x_i) L_i(x)$

and define $I_n^{GL}(f) = \sum_{i=1}^n w_i f(x_i)$

now back to theorem

Thm if f is a polynomial of degree $2n-1$ or less ($f \in P_{2n-1}$),

then $\int_{-1}^1 f(x) dx = I_n^{GL}(f)$, i.e., Gauss-Legendre quadrature with n nodes has degree of exactness $2n-1$

Proof

(1) first, suppose $f \in P_{n-1}$. Then since this is an interpolating quadrature on n nodes, f is its own interpolating polynomial (by uniqueness) hence it is integrated exactly

(2) now, let f have degree between $[n, 2n-1]$.

Divide f by the Legendre polynomial P_n :

$$f(x) = Q(x)P_n(x) + R(x) \quad \text{remainder}$$

where $\deg(R) < \deg(P_n) = n$.

see wikipedia
"polynomial long division"
or "Euclidean division"

$$\begin{array}{r} Q(x) \\ \hline P_n(x) \overline{) f(x)} \\ \hline R(x) \end{array}$$

Also, $\deg(f) \in [n, 2n-1]$

$$\begin{aligned} \deg(f) &= \deg(Q \cdot P_n) = \deg(Q) + \deg(P_n) \\ &= \deg(Q) + n \end{aligned}$$

so, $\deg(Q) \in [0, n-1]$

$\Rightarrow \deg(Q) \leq n-1$, i.e., $Q \in P_{n-1}$

Thus

$$\int_{-1}^1 f(x) dx = \int_{-1}^1 Q(x)P_n(x) dx + \int_{-1}^1 R(x) dx \quad \begin{array}{l} \text{by orthogonality} \\ \text{equals G-L quadrature by part (1)} \end{array}$$

$$= 0 + I_n^{GL}(R)$$

$$= I_n^{GL}(f) \quad \text{since } R(x_i) = f(x_i). \text{ Why?}$$

$$f(x_i) = Q(x_i)P_n(x_i) + R(x_i)$$

x_i is a root of P_n

□

Extending from $\int_{-1}^1 f(x) dx$ to $\int_a^b f(x) dx$ ⚠️ Important detail!

Just do a change-of-variables

$$\begin{aligned}
 x &= a & x &= b \\
 t &= -1 & t &= 1 \\
 \text{Define } t &= \frac{x - \frac{a+b}{2}}{\frac{b-a}{2}} & & \begin{array}{l} \text{center it} \\ \text{normalize it} \end{array} \\
 \text{so } x &= \frac{1}{2}[(b-a)t + a+b], \text{ so } dx = \frac{b-a}{2} dt
 \end{aligned}$$

$$\int_a^b f(x) dx = \frac{b-a}{2} \int_{-1}^1 f\left(\frac{(b-a)t + a+b}{2}\right) dt$$

i.e., define $g(t) = f\left(\frac{(b-a)t + a+b}{2}\right)$

so the Gauss-Legendre quadrature estimate of $\int_a^b f(x) dx$ is $\frac{b-a}{2} I_n^{GL}(g)$.

More info

- Not only is integration more accurate w/ Gauss-Legendre nodes, but so is interpolation. The Runge phenomenon goes away.
- Nodes are precomputed (see Table 4.2 in book) or see `scipy.special.roots_legendre` or `glint` (Driscoll, Braun text)
- A similar method that works almost as well is **Clenshaw-Curtis** see "Is Gaussian Quadrature Better than Clenshaw-Curtis?" Lloyd Trefethen, SIAM Review 2008 (it's faster: $O(n \log n)$ vs $O(n^2)$)
- The **Legendre polynomials** are orthogonal in the sense

$$\underbrace{\int_{-1}^1 P_m(x) P_n(x) dx}_{\langle P_m, P_n \rangle} = 0 \text{ if } m \neq n$$

$\langle P_m, P_n \rangle$ an inner-product (just as $\vec{x}^T \vec{y}$ is an inner product)

If we make a weighted inner product,

$$\int_{-1}^1 f(x) g(x) w(x) dx = \langle f, g \rangle \text{ for } w(x) > 0$$

we get different "orthogonal" polynomials, (below is not an any test)

e.g.: $w(x) = \frac{1}{\sqrt{1-x^2}}$ on $(-1, 1)$ gives rise to T_n Chebyshev polynomials of the 1st kind

$$\begin{aligned}
 T_0(x) &= 1 & T_n(x) &= 2x T_{n-1}(x) - T_{n-2}(x) \\
 T_1(x) &= x & T_n(x) &= \cos(n \cdot \arccos(x)) \quad (*)
 \end{aligned}$$

all roots simple, real, lie in $(-1, 1)$.

In fact, $(\rightarrow) \Rightarrow$ roots x_i are $x_i = \cos\left(\frac{2i-1}{2n}\pi\right)$ $i=1, \dots, n$
"Chebyshev nodes of the 1st kind"

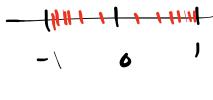
ex $w(x) = \sqrt{1-x^2}$ on $[-1, 1]$, gives rise to U_n Chebyshev polynomials
of the 2nd kind

ω_1 roots

$$x_i = -\cos\left(\frac{i\pi}{n}\right), i=0, 1, \dots, n$$

"Chebyshev nodes/points of the 2nd kind"

These are x-projections of equispaced pts on unit circle

 and beats Runge phenomenon,
so good for interpolation

Thm (9.3.1 Driscoll + Braun) "Spectral Convergence"

* real analytic:

Taylor series converges everywhere

let p_n be the unique degree n polynomial interpolant
of f using $n+1$ Chebyshev nodes of the 2nd kind

If f is real analytic \rightarrow then $\exists C > 0, \exists K > 1$ such that

$$\max_{x \in [-1, 1]} |f(x) - p_n(x)| \leq C \cdot K^{-n}$$

way better than $n^{-\alpha}$

ex. weight $w(x) = e^{-x}$ (or $e^{-x/2}$) on domain $[0, \infty)$

gives Laguerre polynomials

ex weight $w(x) = e^{-x^2}$ (or $e^{-x^2/2}$) on domain $(-\infty, \infty)$

gives Hermite polynomials

For all these choices of weights, we have a corresponding Gaussian quadrature rule

Name	Interval	Weight
Gauss-Legendre	$[-1, 1]$	1
Chebyshev-Gauss	$(-1, 1)$	$(1-x^2)^{-1/2}$
	$[-1, 1]$	$(1-x^2)^{1/2}$
Gauss-Laguerre	$[0, \infty)$	e^{-x}
Gauss-Hermite	$(-\infty, \infty)$	e^{-x^2}