Lecture 17

Integration: Gauss Quadrature

David Semeraro

University of Illinois at Urbana-Champaign

March 20, 2014

Today:

Objectives

- identify the most widely used quadrature method
- is it cheap?
- is it effective?
- how does it compare to Newton-Cotes (Trapezoid, Simpson, etc)?

Material

- Section 5.4

Quadrature

- up until now, our quadrature methods were of the form

$$\int_a^b f(x)\,dx \approx \sum_{j=0}^{n} w_j f(x_j)$$

  where x_j are equally spaced nodes
- Trapezoid:

$$\int_a^b f(x)\,dx \approx \frac{b-a}{2} f(a) + \frac{b-a}{2} f(b)$$

- Simpson:

$$\int_a^b f(x)\,dx \approx \frac{b-a}{6} f(a) + \frac{2(b-a)}{3} f\left(\frac{a+b}{2}\right) + \frac{b-a}{6} f(b)$$

- Similar for higher order polynomial Newton-Cotes rules

Quadrature with Freedom

!

These quadrature rules have one thing in common: they're restrictive

- e.g. Simpson:

$$\int_a^b f(x)\,dx \approx \frac{b-a}{6}f(a) + \frac{2(b-a)}{3}f\left(\frac{a+b}{2}\right) + \frac{b-a}{6}f(b)$$

- Trapezoid, Simpson, etc (Newton-Cotes) are based on equally spaced nodes
- We know one thing already from interpolation: equally spaced nodes result in *wiggle*.
- What other choice do we have? (...recall how we fixed wiggle in interpolation: by moving the location of the nodes)

Gaussian Quadrature

- free ourselves from equally spaced nodes
- combine selection of the nodes and selection of the weights into one quadrature rule

Gaussian Quadrature

Choose the nodes and coefficients optimally to maximize the degree of precision of the quadrature rule:

$$\int_a^b f(x)\,dx \approx \sum_{j=0}^{n} w_j f(x_j)$$

Goal

Seek w_j and x_j so that the quadrature rule is exact for really high polynomials

Gaussian Quadrature

$$\int_a^b f(x)\,dx \approx \sum_{j=0}^{n} w_j f(x_j)$$

- we have $n+1$ points $x_j \in [a,b]$, $a \leqslant x_0 < x_1 < \cdots < x_{n-1} < x_n \leqslant b$.
- we have $n+1$ real coefficients w_j

- so there are $2n+2$ total unknowns to take care of

- there were only 2 unknowns in the case of trapezoid (2 weights)
- there were only 3 unknowns in the case of Simpson (3 weights)
- there were only $n+1$ unknowns in the case of general Newton-Cotes ($n+1$ weights)

Gaussian Quadrature

$$\int_a^b f(x)\,dx \approx \sum_{j=0}^{n} w_j f(x_j)$$

- we have $n+1$ points $x_j \in [a,b]$, $a \leqslant x_0 < x_1 < \cdots < x_{n-1} < x_n \leqslant b$.
- we have $n+1$ real coefficients w_j

- so there are $2n+2$ total unknowns to take care of

- there were only 2 unknowns in the case of trapezoid (2 weights)
- there were only 3 unknowns in the case of Simpson (3 weights)
- there were only $n+1$ unknowns in the case of general Newton-Cotes ($n+1$ weights)

$2n+2$ unknowns (using $n+1$ nodes) can be used to exactly interpolate and integrate polynomials of degree up to $2n+1$

Better Nodes Example

The first thing we do is SIMPLIFY

- consider the case of $n = 1$ (2-point)
- consider $[a, b] = [-1, 1]$ for simplicity
- we *know* how the trapezoid rule works
- Question: can we possibly do better using only 2 function evaluations?
- Goal: Find w_0, w_1, x_0, x_1 so that

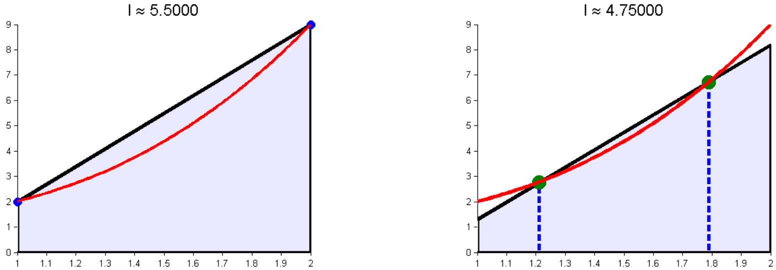
$$\int_{-1}^{1} f(x)\,dx \approx w_0 f(x_0) + w_1 f(x_1)$$

is as accurate as possible...

Graphical View

Consider

$$\int_1^2 x^3 + 1\,dx = 4.75$$



Derive...

Again, we are considering $[a, b] = [-1, 1]$ for simplicity:

$$\int_{-1}^{1} f(x)\, dx \approx w_0 f(x_0) + w_1 f(x_1)$$

Goal: find w_0, w_1, x_0, x_1 so that the approximation is exact up to cubics. So try any cubic:

$$f(x) = c_0 + c_1 x + c_2 x^2 + c_3 x^3$$

This implies that:

$$\begin{aligned} \int_{-1}^{1} f(x)\, dx &= \int_{-1}^{1} \left(c_0 + c_1 x + c_2 x^2 + c_3 x^3\right)\, dx \\ &= w_0 \left(c_0 + c_1 x_0 + c_2 x_0^2 + c_3 x_0^3\right) + \\ &\quad\; w_1 \left(c_0 + c_1 x_1 + c_2 x_1^2 + c_3 x_1^3\right) \end{aligned}$$

Derive...

$$\begin{aligned}\int_{-1}^{1} f(x)\,dx &= \int_{-1}^{1}\left(c_0 + c_1x + c_2x^2 + c_3x^3\right)\,dx \\ &= w_0\left(c_0 + c_1x_0 + c_2x_0^2 + c_3x_0^3\right) + \\ &\quad\; w_1\left(c_0 + c_1x_1 + c_2x_1^2 + c_3x_1^3\right)\end{aligned}$$

Rearrange into constant, linear, quadratic, and cubic terms:

$$c_0\left(w_0 + w_1 - \int_{-1}^{1} dx\right) \quad + \quad c_1\left(w_0x_0 + w_1x_1 - \int_{-1}^{1} x\,dx\right) +$$

$$c_2\left(w_0x_0^2 + w_1x_1^2 - \int_{-1}^{1} x^2\,dx\right) \quad + \quad c_3\left(w_0x_0^3 + w_1x_1^3 - \int_{-1}^{1} x^3\,dx\right) = 0$$

Since c_0, c_1, c_2 and c_3 are arbitrary, then their coefficients must all be zero.

Derive...

This implies:

$$w_0 + w_1 = \int_{-1}^{1} dx = 2 \qquad w_0 x_0 + w_1 x_1 = \int_{-1}^{1} x\, dx = 0$$

$$w_0 x_0^2 + w_1 x_1^2 = \int_{-1}^{1} x^2\, dx = \frac{2}{3} \qquad w_0 x_0^3 + w_1 x_1^3 = \int_{-1}^{1} x^3\, dx = 0$$

Some algebra leads to:

$$w_0 = 1 \quad w_1 = 1 \quad x_0 = -\frac{\sqrt{3}}{3} \quad x_1 = \frac{\sqrt{3}}{3}$$

Therefore:

$$\int_{-1}^{1} f(x)\, dx \approx f\left(-\frac{\sqrt{3}}{3}\right) + f\left(\frac{\sqrt{3}}{3}\right)$$

Over another interval?

$$\int_{-1}^{1} f(x)\,dx \approx f\left(-\frac{\sqrt{3}}{3}\right) + f\left(\frac{\sqrt{3}}{3}\right)$$

$$\int_{a}^{b} f(x)\,dx \approx ?$$

- integrating over $[a, b]$ instead of $[-1, 1]$ needs a transformation: a change of variables
- want $t = c_1 x + c_0$ with $t = -1$ at $x = a$ and $t = 1$ at $x = b$
- let $t = \frac{2}{b-a}x - \frac{b+a}{b-a}$
- (verify)
- let $x = \frac{b-a}{2}t + \frac{b+a}{2}$
- then $dx = \frac{b-a}{2}dt$

Over another interval?

$$\int_a^b f(x)\,dx \approx ?$$

- let $x = \frac{b-a}{2}t + \frac{b+a}{2}$
- then $dx = \frac{b-a}{2}dt$

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

- now use the quadrature formula over $[-1, 1]$
- note: using two points, $n = 1$, gave us exact integration for polynomials of degree less 2*1+1 = 3 and less.

Over another interval?

Previous example...

$$\int_1^2 x^3 + 1\,dx = 4.75$$

$$\int_1^2 f(x)dx = \frac{1}{2}\int_{-1}^{1} f\left(\frac{t+3}{2}\right)dt$$

$$\approx \frac{1}{2}\left[f\left(\frac{3+\frac{\sqrt{3}}{3}}{2}\right) + f\left(\frac{3-\frac{\sqrt{3}}{3}}{2}\right)\right]$$

where $x = \frac{3+t}{2}$

Over another interval?

Evaluating...

$$\int_1^2 x^3 + 1\,dx$$

$$\begin{aligned}
\int_1^2 x^3 + 1\,dx &\approx \frac{1}{2}\left[f\left(\frac{3+\frac{\sqrt{3}}{3}}{2}\right) + f\left(\frac{3-\frac{\sqrt{3}}{3}}{2}\right)\right] \\
&\approx \frac{1}{2}\left[f\left(\frac{9+\sqrt{3}}{6}\right) + f\left(\frac{9-\sqrt{3}}{6}\right)\right] \\
&\approx \frac{1}{2}\left[f\left(1.788651\right) + f\left(1.211325\right)\right] \\
&\approx \frac{1}{2}\left[6.722382 + 2.777387\right] \\
&= 4.749885
\end{aligned}$$

where $f(x) = x^3 + 1$

Extending Gauss Quadrature

- we need more to make this work for more than two points
- A sensible quadrature rule for the interval $[-1, 1]$ based on 1 node would use the node $x = 0$. This is a root of $\phi(x) = x$
- Notice: $\pm\frac{1}{\sqrt{3}}$ are the roots of $\phi(x) = 3x^2 - 1$
- general $\phi(x)$?

Gauss Quadrature Theorem

Karl Friedrich Gauss proved the following result:
Let $q(x)$ be a nontrivial polynomial of degree $n+1$ such that

$$\int_a^b x^k q(x)dx = 0 \qquad (0 \leqslant k \leqslant n)$$

and let $x_0, x_1, \ldots, x_n$ be the zeros of $q(x)$. Then

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n} A_i f(x_i), A_i = \int_a^b \ell_i(x)dx$$

will be exact for all polynomials of degree at most $2n+1$.

Sketch of Proof

Let $f(x)$ be a polynomial of degree $2n+1$. Then we can write $f(x) = p(x)q(x) + r(x)$, where $p(x)$ and $r(x)$ are of degree at most n (This is basically dividing f by q with remainder r).

Then by the hypothesis, $\int_a^b p(x)q(x)dx = 0$. Further, $f(x_i) = p(x_i)q(x_i) + r(x_i) = r(x_i)$. Thus,

$$\int_a^b f(x)dx = \int_a^b r(x)dx \approx \sum_{i=0}^{n} f(x_i) \int_a^b \ell_i(x)dx$$

But this is exact because $r(x)$ is (at most) a degree n polynomial. Thus, we need to find the polynomials $q(x)$.

Orthogonal Polynomials

Orthogonality of Functions

Two functions $g(x)$ and $h(x)$ are *orthogonal* on $[a, b]$ if

$$\int_a^b g(x)h(x)\,dx = 0$$

- so the nodes we're using are roots of orthogonal polynomials
- these are the *Legendre* Polynomials

Legendre Polynomials

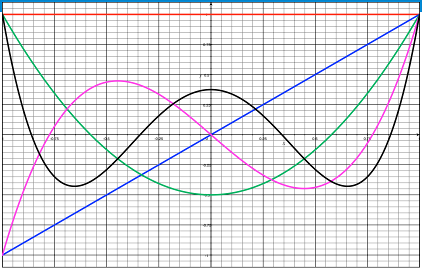
given on the exam

$$\begin{aligned}
\phi_0 &= 1 \\
\phi_1 &= x \\
\phi_2 &= \frac{3x^2 - 1}{2} \\
\phi_3 &= \frac{5x^3 - 3x}{2} \\
&\vdots
\end{aligned}$$

In general:

$$\phi_n(x) = \frac{2n-1}{n} x \phi_{n-1}(x) - \frac{n-1}{n} \phi_{n-2}(x)$$

Notes on Legendre Roots



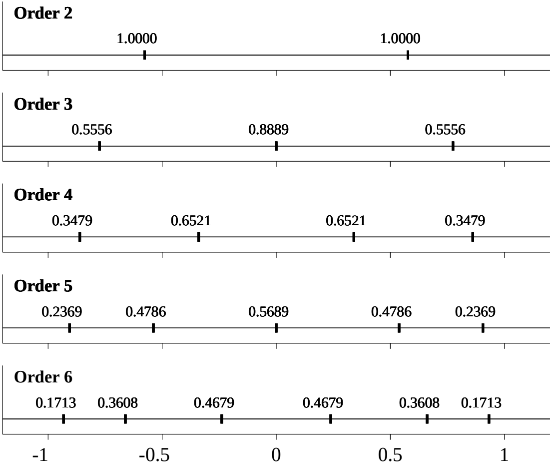
- The Legendre Polynomials are orthogonal (nice!)
- The Legendre Polynomials increase in polynomials order (like monomials)
- The Legendre Polynomials don't suffer from poor conditioning (unlike monomials)
- The Legendre Polynomials don't have a closed form expression (recursion relation is needed)
- The roots of the Legendre Polynomials are the nodes for Gaussian Quadrature (GL nodes)

Quadrature Nodes (see)

- Often listed in tables
- Weights determined by extension of above
- Roots are symmetric in $[-1,1]$
- Example:

```
if(n==0)
  x = 0;   w = 2;
if(n==1)
  x(1) = -1/sqrt(3);    x(2) = -x(1);
  w(1) =  1;            w(2) =  w(1);
if(n==2)
  x(1) = -sqrt(3/5);   x(2) =  0;      x(3) = -x(1);
  w(1) =  5/9;         w(2) =  8/9;    w(3) =  w(1);
if(n==3)
  x(1) = -0.861136311594053;    x(4) = -x(1);
  x(2) = -0.339981043584856;    x(3) = -x(2);
  w(1) =  0.347854845137454;    w(4) =  w(1);
  w(2) =  0.652145154862546;    w(3) =  w(2);
if(n==4)
  x(1) = -0.906179845938664;    x(5) = -x(1);
  x(2) = -0.538469310105683;    x(4) = -x(2);
  x(3) =  0;
  w(1) =  0.236926885056189;    w(5) =  w(1);
  w(2) =  0.478628670499366;    w(4) =  w(2);
  w(3) =  0.568888888888889;
if(n==5)
  x(1) = -0.932469514203152;    x(6) = -x(1);
  x(2) = -0.661209386466265;    x(5) = -x(2);
  x(3) = -0.238619186083197;    x(4) = -x(3);
  w(1) =  0.171324492379170;    w(6) =  w(1);
  w(2) =  0.360761573048139;    w(5) =  w(2);
  w(3) =  0.467913934572691;    w(4) =  w(3);
```

View of Nodes



Theory

The connection between the roots of the Legendre polynomials and exact integration of polynomials is established by the following theorem.

Theorem

Suppose that $x_0, x_1, \ldots, x_n$ are roots of the nth Legendre polynomial $P_n(x)$ and that for each $i = 0, 1, \ldots, n$ the numbers w_i are defined by

$$w_i = \int_{-1}^{1} \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} dx = \int_{-1}^{1} \ell_i(x)\, dx$$

Then

$$\int_{-1}^{1} f(x) dx = \sum_{i=0}^{n} w_i f(x_i),$$

where $f(x)$ is any polynomial of degree less or equal to $2n + 1$.

Do not!

!!!

When evaluating a quadrature rule

$$\int_{-1}^{1} f(x)dx = \sum_{i=0}^{n} w_i f(x_i).$$

do not generate the nodes and weights each time. Use a lookup table...

Example

Approximate $\int_1^{1.5} x^2 \ln x \, dx$ using Gaussian quadrature with $n = 1$.

<u>Solution</u> As derived earlier we want to use $\int_{-1}^{1} f(x)\,dx \approx f\left(-\frac{\sqrt{3}}{3}\right) + f\left(\frac{\sqrt{3}}{3}\right)$

From earlier we know that we are interested in

$$\int_1^{1.5} f(x)\,dx = \int_{-1}^{1} f\left(\frac{(1.5-1)t + (1.5+1)}{2}\right) \frac{1.5-1}{2}\,dt$$

Therefore, we are looking for the integral of

$$\frac{1}{4}\int_{-1}^{1} f\left(\frac{x+5}{4}\right) dx = \frac{1}{4}\int_{-1}^{1} \left(\frac{x+5}{4}\right)^2 \ln\left(\frac{x+5}{4}\right) dx$$

Using Gaussian quadrature, our numerical integration becomes:

$$\frac{1}{4}\left[\left(\frac{-\frac{\sqrt{3}}{3}+5}{4}\right)^2 \ln\left(\frac{-\frac{\sqrt{3}}{3}+5}{4}\right) + \left(\frac{\frac{\sqrt{3}}{3}+5}{4}\right)^2 \ln\left(\frac{\frac{\sqrt{3}}{3}+5}{4}\right)\right] = 0.1922687$$

Example

Approximate $\int_0^1 x^2 e^{-x}\, dx$ using Gaussian quadrature with $n = 1$.

SOLUTION We again want to convert our limits of integration to -1 to 1. Using the same process as the earlier example, we get:

$$\int_0^1 x^2 e^{-x} dx = \frac{1}{2}\int_{-1}^{1}\left(\frac{t+1}{2}\right)^2 e^{(t+1)/2} dt.$$

Using the Gaussian roots we get:

$$\int_0^1 x^2 e^{-x} dx \approx \frac{1}{2}\left[\left(\frac{-\frac{\sqrt{3}}{3}+1}{2}\right)^2 e^{(-\frac{\sqrt{3}}{3}+1)/2} + \left(\frac{\frac{\sqrt{3}}{3}+1}{2}\right)^2 e^{(\frac{\sqrt{3}}{3}+1)/2}\right] = 0.1594104$$

Numerical Question

How does n point Gauss quadrature compare with n point Newton-Cotes...

Examples

with Python...

Example

int_gauss.py: base routine for Gauss quadrature

Example

int_gauss_test.py: integrate $\int_0^5 xe^{-x}\,dx$ with

1. 1 subinterval, increasing number of nodes
2. 3 nodes, increases number of intervals (panels)

Result: fewer total evaluations in GL quadrature with 1 subinterval and many nodes versus 3 nodes and many subpanels. Also more accurate.