Lecture 13

Definite Integrals: Newton Cotes

T. Gambill

Department of Computer Science
University of Illinois at Urbana-Champaign

April 14, 2011

Theorem

The Fundamental Theorem of Calculus Given a continuous function $f(x) : [a, b] \to \mathbb{R}$ then a function $F(x)$ satisfies,

$$F(x) = F(a) + \int_a^x f(x)dx$$

if and only if

$$F'(x) = f(x) \text{ for } x \in [a, b]$$

Next...

- Can we integrate $f(x)$?
- What about $f(x) = e^{-x^2}$?
- What if $f(x)$ is only known implicitly (known at a certain number of points)?

Integration

What is the integral $\int_a^b$?

- Let P be a partition of $[a, b]$ of $n+1$ distinct and ordered points with $x_0 = a$ and $x_n = b$.
- For interval $[x_i, x_{i+1}]$ let m_i be a lower bound on $f(x)$
- For interval $[x_i, x_{i+1}]$ let M_i be an upper bound on $f(x)$
- Lower Sum:

$$L(f;P) = \sum_{i=0}^{n-1} m_i(x_{i+1} - x_i)$$

- Upper Sum:

$$U(f;P) = \sum_{i=0}^{n-1} M_i(x_{i+1} - x_i)$$

Integration

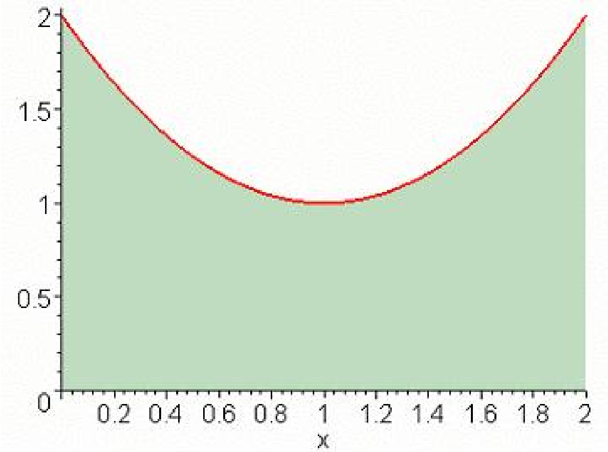
- The lower sum always under-approximates the integral
- The upper sum always over-approximates the integral

$$L(f;P) \leqslant \int_a^b f(x)\,dx \leqslant U(f;P)$$

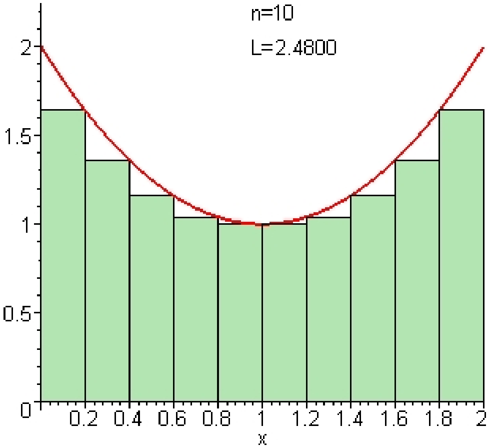
- In the limit, they are equal

$$\lim_{n\to\infty} L(f;P) = \int_a^b f(x)\,dx = \lim_{n\to\infty} U(f;P)$$

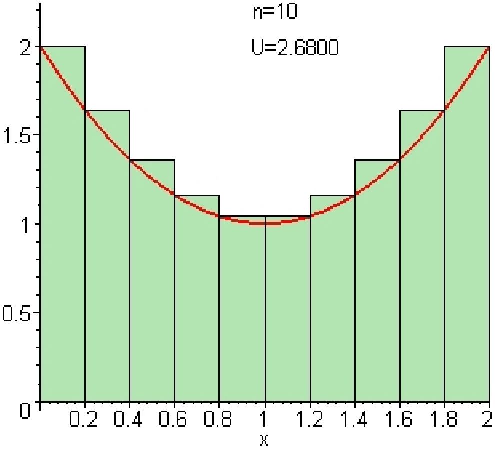
Graphically: Integral



Graphically: Lower sum



Graphically: Upper sum



Left-Riemann, Right-Riemann, Mid-Point

- The upper and lower bounds are often difficult to identify
- Use Left-Riemann, Right-Riemann, and Middle Riemann Sums
- Generally the Riemann sum is

$$S = \sum_{i=0}^{n-1} f(z_i)(x_{i+1} - x_i)$$

  for $x_i \leqslant z_i \leqslant x_{i+1}$
- $z_i = x_i$ is a Left Riemann Sum
- $z_i = x_{i+1}$ is a Right Riemann Sum
- $z_i = \frac{x_{i+1}+x_i}{2}$ is a Middle Riemann Sum

Left-Riemann, Right-Riemann, Mid-Point

We have a way to compute integrals. Why aren't we done?

What is the cost?
How accurate are the results?

Left Riemann Error Bound

If we assume that $f'(x)$ is continuous on the interval $[a, b]$ then we can apply the Taylor Series to our error analysis. For equally spaced intervals $[x_k, x_{k+1}]$ ($h = x_{k+1} - x_k$) the Taylor series can be written as,

$$f(x) = f(x_k) + f'(\xi_x)(x - x_k)$$

$$\begin{aligned}
error &= \left| \sum_{k=0}^{n-1} f(x_k) * h - \int_b^a f(x)dx \right| \\
&= \left| \sum_{k=0}^{n-1} \int_{x_k}^{x_{k+1}} f(x_k) - (f(x_k) + f'(\xi_x)(x - x_k))dx \right| \\
&\leqslant M \sum_{k=0}^{n-1} h^2/2 \text{ where } |f'(x)| \leqslant M \text{ for } x \in [a, b] \\
&= Mnh^2/2 = M(b-a)h/2
\end{aligned}$$

So the error is $O(h)$. Can we do better?

Goals

Methods:

- Newton-Cotes in general
- Trapezoid Rule
- Composite Trapezoid Rule
- Simpson Rule
- Composite Simpson Rule
- Sections 7.1-7.3

Newton-Cotes, using an interpolating polynomial

Approximate $f(x)$ on the entire interval $[a, b]$ using the Lagrange form of the interpolating polynomial of degree n at equidistant points x_k.

$$f(x) \approx p_n(x) = \sum_{k=0}^{n} f(x_k)\ell_k(x)$$

then we have

$$\int_a^b f(x)dx \approx \sum_{k=0}^{n} f(x_k)w_k$$

where the w_k are determined by

$$w_k = \int_a^b \ell_k(x)dx$$

Newton-Cotes, using an interpolating polynomial

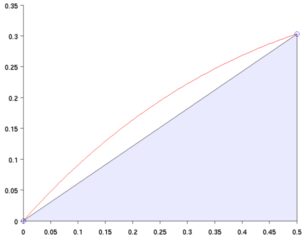
(basic) Newton-Cotes rules:

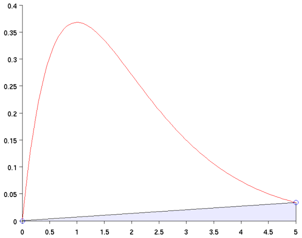
name	n	formula
Trapezoid	1	$\frac{(b-a)}{2}\left[f(a)+f(b)\right]$
Simpson's $1/3$	2	$\frac{(b-a)}{6}\left[f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right]$
Simpson's $3/8$	3	$\frac{(b-a)}{8}\left[f(a)+3f(a+h)+3f(b-h)+f(b)\right]$
Boole's	4	$\frac{(b-a)}{90}\left[7f(a)+32f(a+h)+12f\left(\frac{a+b}{2}\right)+32f(b-h)+7f(b)\right]$

Basic Trapezoid

Use endpoints $[a, b]$ to obtain a linear approximation to $f(x)$. The area under this function is the area of a trapezoid:

$$\int_a^b f(x)\,dx \approx \frac{1}{2}(b-a)(f(a)+f(b))$$





Basic Trapezoid

- Trapezoid Rule:

$$\int_{x_0}^{x_1} f(x)\,dx \approx \int_{x_0}^{x_1} P_1(x)\,dx = \frac{1}{2}(f(x_0)+f(x_1))h$$

$$\int_{x_0}^{x_1} f(x)\,dx \approx \frac{1}{2}(f(x_0)+f(x_1))h \text{ , where } f(x) = 15\,x^2$$

Example

$$\int_1^2 15\,x^2 \approx \frac{1}{2}(15*1^2+15*2^2)*1$$

$$= \frac{1}{2}(15+60) = 37.5$$

- Analytical answer is $\int_1^2 15\,x^2 = 5\,x^3\big|_1^2 = 40-5 = 35.$

Trapezoid, Error Bound

From a previous lecture we stated:

Theorem

Given function f with $n+1$ continuous derivatives in the interval formed by I = $[min(\{x, x_0, \ldots, x_n\}), max(\{x, x_0, \ldots, x_n\})]$. If $p(x)$ is the unique interpolating polynomial of degree $\leqslant n$ with,

$$p(x_i) = f(x_i), \quad i = 0, 1, \ldots, n$$

then the error is computed by the formula,

$$p(x) - f(x) = \frac{f^{(n+1)}(\xi(x))}{(n+1)!}(x - x_0)(x - x_1)\ldots(x - x_n), \quad \text{for some } \xi(x) \in I$$

Trapezoid, Error Bound

For the Trapezoidal Rule we have,

$$
\begin{aligned}
error &= \left| \int_a^b p_1(x) - f(x)dx \right| \\
&= \left| \int_a^b \frac{f^{(2)}(\xi(x))}{2!}(x-a)(x-b)dx \right| \\
&\leqslant \frac{M}{2} \int_a^b |(x-a)(x-b)|\, dx \text{ where } |f''(x)| \leqslant M \text{ for } x \in [a,b] \\
&= \frac{M}{12}(b-a)^3
\end{aligned}
$$

If $b - a \ll 1$ we denote $h = b - a$ then our error bound is $O(h^3)$.
Note: If $f(x)$ is a linear function then $f''(x) = 0$ for all $x \in [a,b]$ and then $M = 0$ and our error bound is exact.
What if $h = b - a$ is large? Use a higher degree interpolating polynomial? Is there an alternative?

Newton-Cotes,Exact Error Bounds

The error,

$$error = \int_a^b f(x)dx - \text{ approximate formula}$$

for the various rules is given by the following table

(basic) Newton-Cotes rules:

name of formula	n	error
Trapezoid	1	$-\frac{(b-a)^3}{12} f^{(2)}(\xi)$
Simpson's $1/3$	2	$-\frac{(b-a)^5}{2880} f^{(4)}(\xi)$
Simpson's $3/8$	3	$-\frac{(b-a)^5}{6480} f^{(4)}(\xi)$
Boole's	4	$-\frac{(b-a)^7}{1935360} f^{(6)}(\xi)$

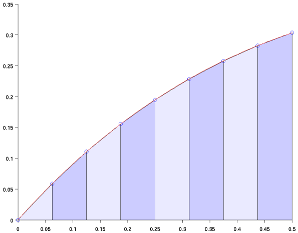
Composite Trapezoid

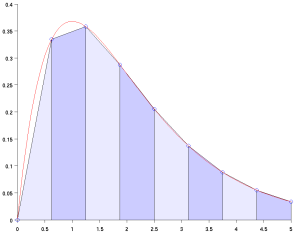
Obviously a naive linear approximation won't cut it.

Consider a partition $P = \{x_0 = a < \dots x_n = b\}$ of $[a, b]$.

In each interval $[x_i, x_{i+1}]$ use the basic Trapezoid:

$$\int_a^b f(x)\,dx \approx \sum_{i=0}^{n-1} \frac{1}{2}(x_{i+1} - x_i)(f(x_i) + f(x_{i+1}))$$





Composite Trapezoid

- With uniform spacing of P, $h_i = x_{i+1} - x_i = h$ is constant

$$T(f;P) = \int_a^b f(x)\,dx \approx \frac{h}{2}\sum_{i=0}^{n-1} f(x_i) + f(x_{i+1})$$

- This becomes

$$T(f;P) = \int_a^b f(x)\,dx \approx \frac{h}{2}\left(f(x_0) + 2f(x_1) + 2f(x_2) + \cdots + 2f(x_{n-1}) + f(x_n)\right)$$

```
h = (b - a)/n
sum = (f(a) + f(b))/2
for i = 1 to n - 1
   sum = sum + f(x_i)
end
sum = sum · h
```

Example

Test composite trapezoid for

$$\int_0^5 xe^{-x}$$

Question: What is the order of accuracy (the p in $O(h^p)$)?

Composite Trapezoid Error Bound

The error in computing the integral is,

$$\begin{aligned}
error &= \left| \int_a^b f(x)\,dx - \frac{h}{2}\sum_{i=0}^{n-1} f(x_i) + f(x_{i+1}) \right| \\
&= \left| \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} \left(f(x) - \frac{h}{2}(f(x_i) + f(x_{i+1})) \right) dx \right| \\
&\leqslant \sum_{i=0}^{n-1} \left| \int_{x_i}^{x_{i+1}} \left(f(x) - \frac{h}{2}(f(x_i) + f(x_{i+1})) \right) dx \right| \\
&= \sum_{i=0}^{n-1} E_i
\end{aligned}$$

where the E_i are the error bounds in each interval, $[x_i, x_{i+1}]$,

$$E_i = \frac{M_i}{12} h^3 \text{ where } |f''(x)| \leqslant M_i \text{ for } x \in [x_i, x_{i+1}]$$

Composite Trapezoid Error Bound

So the total error is

$$\begin{aligned}
\sum_{i=0}^{n-1} E_i &= \sum_{i=0}^{n-1} \frac{M_i}{12} h^3 \\
&\leqslant \frac{M}{12} \sum_{i=0}^{n-1} h^3 \text{ where } |f''(x)| \leqslant M \text{ for } x \in [a, b] \\
&= \frac{M}{12} n h^3 \\
&= \frac{M}{12} (b - a) h^2
\end{aligned}$$

Example

How many points should be used to ensure the composite Trapezoid rule is accurate to 10^{-6} for $\int_0^1 e^{-x^2}\,dx$? Need

$$\frac{|f''(\eta)|}{12}(b-a)h^2 \leqslant 10^{-6}$$

How big is $f''(x)$?

$$\begin{aligned} f(x) &= e^{-x^2} \\ f'(x) &= -2xe^{-x^2} \\ f''(x) &= -2e^{-x^2} + 4x^2e^{-x^2} \\ f'''(x) &= 12xe^{-x^2} - 8x^3e^{-x^2} \end{aligned}$$

So f''' is always positive for $x > 0$. So f'' is monotone increasing and thus $|f''|$ takes on a maximum at an endpoint: $|f''(0)| = 2$ and $|f''(1)| = \frac{2}{e}$. Then bound

$$\frac{(b-a)2h^2}{12} \leqslant 10^{-6}$$

Or

$$h^2 \leqslant 6 \times 10^{-6} \quad \Rightarrow \quad \sqrt{(1/6)}10^3 \leqslant n$$

or $n + 1 >= 410$.

How do we improve Composite Trapezoid?

- instead of a linear approximation, use a quadratic approximation
- $\Rightarrow$ Composite Simpson's Rule

Composite Simpson

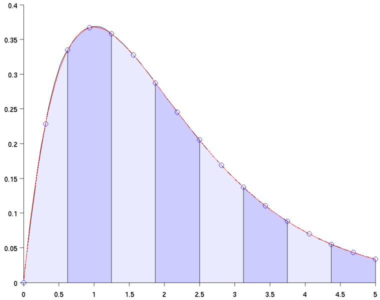
Over a uniform partition $P = x_0, x_1, \ldots, x_n$, use Basic Simpson's Rule over each subinterval $[x_{2i}, x_{2i+2}]$ where n is even and $h = \frac{b-a}{n}$.

$$\begin{aligned}
\int_a^b f(x)\,dx &= \sum_{i=0}^{n/2-1} \int_{x_{2i}}^{x_{2i+2}} f(x)\,dx \\
&\approx \sum_{i=0}^{n/2-1} \frac{2h}{6} \left[f(x_{2i}) + 4f(x_{2i+1}) + f(x_{2i+2})\right] \\
&\approx \frac{h}{3} \left[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \cdots + 4f(x_{n-1}) + f(x_n)\right]
\end{aligned}$$

Simpson

Composite Simpson's Rule

$$\int_a^b f(x)\,dx \approx \frac{h}{3}\left[f(a) + f(b) + 4\sum_{i=1}^{n/2} f(a + (2i-1)h) + 2\sum_{i=1}^{n/2-1} f(a + 2ih)\right]$$



Error Bound for Composite Simpson Method

Taylor Series:

$$f(a+h) = f + hf' + \frac{1}{2!}h^2f'' + \frac{1}{3!}h^3f''' + \frac{1}{4!}h^4f^{(4)} + \frac{1}{5!}h^5f^{(5)} + \dots$$
$$f(a+2h) = f + 2hf' + 2h^2f'' + \frac{4}{3}h^3f''' + \frac{2}{3}h^4f^{(4)} + \frac{4}{15}h^5f^{(5)} + \dots$$

This gives

$$\frac{h}{3}\left[f(a) + 4f(a+h) + f(b)\right] = 2hf + 2h^2f' + \frac{4}{3}h^3f'' + \frac{2}{3}h^4f''' + \frac{5}{18}h^5f^{(4)}$$

Integrating the Taylor Series expansion of $f(x)$ exactly gives

$$\int_a^b f(x)\,dx = 2hf + 2h^2f' + \frac{4}{3}h^3f'' + \frac{2}{3}h^4f''' + \frac{4}{15}h^5f^{(4)}$$

So basic Simpson's Rule gives an error of

$$-\frac{1}{90}\left(\frac{b-a}{2}\right)^5 f^{(4)}(\xi)$$

Why is composite Simpson $\mathcal{O}(h^4)$?

basic Simpson's Rule:

$$-\frac{1}{90}\left(\frac{b-a}{2}\right)^5 f^{(4)}(\xi)$$

Over $n/2$ subintervals $[x_{2i}, x_{2i+2}]$ becomes:

$$\begin{aligned} err &= \sum_{i=1}^{n/2} -\frac{1}{90}\left(\frac{x_{2i+2}-x_{2i}}{2}\right)^5 f^{(4)}(\xi_i) = -\frac{1}{90}\sum_{i=1}^{n/2}\left(\frac{2h}{2}\right)^5 f^{(4)}(\xi_i) \\ &= -\frac{1}{90}\frac{n}{2}h^5 f^{(4)}(\xi) = -\frac{1}{180}\frac{(b-a)}{h}h^5 f^{(4)}(\xi) \\ &= -\frac{b-a}{180}h^4 f^{(4)}(\xi) \end{aligned}$$

Composite Simpson's Rule

$$-\frac{b-a}{180}h^4 f^{(4)}(\xi)$$

We "gain" two orders over Trapezoid

Can we generalize?

Summary:

- left/right Riemann: approximate $f(x)$ by 0-degree $p(x)$ and integrate
- Trapezoid: approximate $f(x)$ by 1-degree $p(x)$ and integrate
- Simpson: approximate $f(x)$ by 2-degree $p(x)$ and integrate

Degree of Precision

If the integration rule has zero error when integrating any polynomial of degree $\leqslant r$ and if the error is nonzero for some polynomial of degree $r+1$, then the rule has *degree of precision* equal to r.

Exact Error bounds for composite Newton-Cotes

The exact error,

$$error = \int_a^b f(x)\, dx - \text{ approximate formula}$$

for the various rules is given by the following table,

name of formula	error
Trapezoid	$-\frac{(b-a)h^2}{12} f''(\xi)$
Simpson's $1/3$	$-\frac{(b-a)h^4}{180} f^{(4)}(\xi)$
Simpson's $3/8$	$-\frac{(b-a)h^4}{80} f^{(4)}(\xi)$
Boole's	$-\frac{2(b-a)h^6}{945} f^{(6)}(\xi)$

where $h = \frac{(b-a)}{n}$ and n is the number of intervals of the partition of $[a, b]$.

Matlab trapz

The Matlab *trapz* function is based on the composite trapezoidal rule. From the previous slide we see that the error for the composite trapezoid rule is proportional to $f''(\xi)$ and thus exact for linear functions.

$$>> x = linspace(-1, 1, 200)$$
$$>> y = 3 * x - 2$$
$$>> trapz(x, y)$$
$$ans =$$
$$-4$$
$$>> syms \quad x$$
$$>> int(3 * x - 2, -1, 1)$$
$$ans =$$
$$-4$$

Adaptive Simpson's Method

Why use a fixed length interval h?
Use an interval that varies in proportion to the error!

Algorithm

Compute the approximate area using Simpson's rule.

$$S(a,b) = \frac{(b-a)}{6}\left[f(a) + 4f(\frac{a+b}{2}) + f(b)\right]$$

Halve the interval and compute $S(a, \frac{a+b}{2})$ and $S(\frac{a+b}{2}, b)$
Estimate the error,

$$error \approx \frac{1}{15}\left|\left(S(a, \frac{a+b}{2}) + S(\frac{a+b}{2}, b)\right) - S(a,b)\right|$$

If the error is less than some specified tolerance $= tol$, we are done, otherwise recursively compute each of $S(a, \frac{a+b}{2})$ and $S(\frac{a+b}{2}, b)$ with tolerance $= \frac{tol}{2}$.

Adaptive Simpson's Method - Why does this method work?

Denote $I(a,b) = \int_a^b f(x)\,dx$ then we can write, using (basic) Simpson's rule denoted by $S(a,b)$, and the error is defined as $E(a,b)$,

$$E(a,b) = I(a,b) - S(a,b)$$

Integration over the interval $[a,b]$ can be broken into halves,

$$I(a,b) = I(a, \frac{a+b}{2}) + I(\frac{a+b}{2}, b)$$

thus we can write these integrals as,

$$\begin{aligned} E(a,b) + S(a,b) &= E(a,(a+b)/2) + S(a,(a+b)/2) + \\ &\quad E((a+b)/2,b) + S((a+b)/2,b) \end{aligned}$$

and collecting terms gives,

$$E(a,b) - (E(a,(a+b)/2) + E((a+b)/2,b)) =$$
$$(S(a,(a+b)/2) + S((a+b)/2,b)) - S(a,b)$$

Adaptive Simpson's Method- Why does this method work?

and since the error for Simpson's rule is

$$\begin{aligned} E(a,b) &= -\frac{h^5}{2880} f^{(4)}(\xi_{[a,b]}) \\ E(a,(a+b)/2) + E((a+b)/2,b) &= -\frac{1}{32} * \frac{h^5}{2880} f^{(4)}(\xi_{[a,(a+b)/2]}) + \\ & \quad -\frac{1}{32} * \frac{h^5}{2880} f^{(4)}(\xi_{[(a+b)/2,b]}) \end{aligned}$$

As we recursively compute the integral the widths of the intervals $b - a$ will become smaller, and sufficiently small so that $f^{(4)}(x)$ is constant on that interval and therefore,

$$E(a,b) \approx 16 * (E(a,(a+b)/2) + E((a+b)/2,b))$$

Adaptive Simpson's Method - Why does this method work?

Thus

$$\frac{E(a,b) - (E(a,(a+b)/2) + E((a+b)/2,b))}{(S(a,(a+b)/2) + S((a+b)/2,b)) - S(a,b)}$$

becomes

$$\frac{-15 * E(a,b)}{(S(a,(a+b)/2) + S((a+b)/2,b)) - S(a,b)}$$

Matlab quad

The Matlab *quad* function is based on the adaptive Simpson's rule.

Example: $\int_0^1 x^5\, dx$

$>> quad(@(x)x.\hat{}5, -1, 1, 1.0e-3)$ (tolerance $= 1.0e-3$)
$ans =$
$-2.775557561562891e-017$
$>> quad(@(x)x.\hat{}5, -1, 1, 1.0e-7)$ (tolerance $= 1.0e-7$)
$ans =$
0

Monte Carlo integration

We compute the integral of $f : \mathbb{R}^d \to \mathbb{R}, \;\; d \geqslant 1$ by generating n random points in $\Omega \subset \mathbb{R}^d$ and use the approximation,

$$\int\int \ldots \int_\Omega f(x_1, x_2, \ldots, x_d) \;\; dx_1 dx_2 \ldots dx_d \approx volume(\Omega) * \frac{\sum_{i=1}^n f(\mathbf{z_i})}{n}$$

where $\mathbf{z_i}$ are randomly chosen values from $\mathbb{R}^d$. We can also use this technique to compute volumes (areas) in $\mathbb{R}^d$. Define the characteristic function χ_Ω of a region Ω as,

$$\begin{aligned} \chi_\Omega(x) &= 1 \text{ if } x \in \Omega \\ &= 0 \text{ if } x \notin \Omega \end{aligned}$$

then for a rectangular region that bounds Ω we have,

$$volume(\Omega) = \int_{a_1}^{b_1} \int_{a_2}^{b_2} \ldots \int_{a_d}^{b_d} \chi_\Omega(x) \;\; dx_1 dx_2 \ldots dx_d \approx \prod_{i=1}^{n} (b_i - a_i) * \frac{\sum_{i=1}^n \chi(\mathbf{z_i})}{n}$$

Monte Carlo integration Error

The error in computing the integral of $f : \mathbb{R}^d \to \mathbb{R}, \quad d \geqslant 1$ by generating n random points in $\mathbb{R}^d$ and using the Monte Carlo Method is,

$$O(\frac{1}{\sqrt{n}}) = \left| \int \int \ldots \int_{\Omega} f(x_1, x_2, \ldots, x_d) \;\; dx_1 dx_2 \ldots dx_d - volume(\Omega) * \frac{\sum_{i=1}^{n} f(\mathbf{z_i})}{n} \right|$$

where $\mathbf{z_i}$ are randomly chosen values from $\Omega \subset \mathbb{R}^d$. Thus, to increase the accuracy of your approximation by one decimal digit using a Monte Carlo method you must increase the number of sample points by a factor of 100.

Stochastic Simulation

From M. Heath, *Scientific Computing, 2nd ed.*, CS450

- Two requirements for MC:
  - knowing which probability distributions are needed
  - generating sufficient random numbers
- The probability distribution depends on the problem (theoretical or empirical evidence)
- The probability distribution can be approximated well by simulating a large number of trials

`http://www.cse.uiuc.edu/iem/random/bfnneedl/`

Randomness

From M. Heath, *Scientific Computing, 2nd ed.*, CS450

- Randomness $\approx$ unpredictability
- One view: a sequence is random if it has no shorter description
- Physical processes, such as flipping a coin or tossing dice, are deterministic with enough information about the governing equations and initial conditions.
- But even for deterministic systems, sensitivity to the initial conditions can render the behavior practically unpredictable.
- we need random simulation methods

Repeatability

From M. Heath, *Scientific Computing, 2nd ed.*, CS450

- With unpredictability, true randomness is not repeatable
- ...but lack of repeatability makes testing/debugging difficult
- So we want repeatability, but also independence of the trials

Use the 'twister' method for Monte Carlo methods.

```
>> rand('twister',1234) % rand('method',seed)
>> rand(10,1)
```

Pseudorandom Numbers

From M. Heath, *Scientific Computing, 2nd ed.*, CS450

Computer algorithms for random number generations are deterministic

- ...but may have long periodicity (a long time until an apparent pattern emerges)
- These sequences are labeled *pseudorandom*
- Pseudorandom sequences are predictable and reproducible (this is mostly good)

Random Number Generators

From M. Heath, *Scientific Computing, 2nd ed.*, CS450

Properties of a good random number generator:

Random pattern: passes statistical tests of randomness

Long period: long time before repeating

Efficiency: executes rapidly and with low storage

Repeatability: same sequence is generated using same initial states

Portability: same sequences are generated on different architectures

Random Number Generators

From M. Heath, *Scientific Computing, 2nd ed.*, CS450

- Early attempts relied on complexity to ensure randomness
- “midsquare” method: square each member of a sequence and take the middle portion of the results as the next member of the sequence
- ...simple methods with a statistical basis are preferable

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

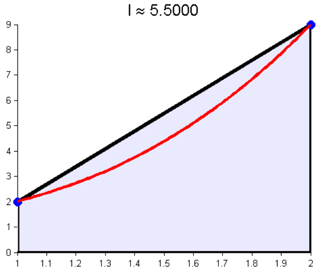
$$\int_{-1}^{1} f(x)\,dx \approx w_0 f(x_0) + w_1 f(x_1)$$

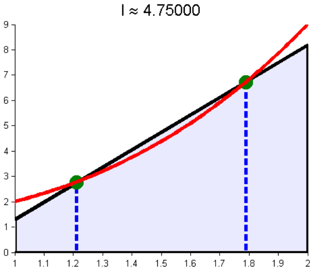
  is as accurate as possible...

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...

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

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

$$c_0\left(w_0 + w_1 - \int_{-1}^{1} dx\right) + c_1\left(w_0 x_0 + w_1 x_1 - \int_{-1}^{1} x\,dx\right) +$$
$$c_2\left(w_0 x_0^2 + w_1 x_1^2 - \int_{-1}^{1} x^2\,dx\right) + c_3\left(w_0 x_0^3 + w_1 x_1^3 - \int_{-1}^{1} x^3\,dx\right) = 0$$

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.

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 $q(x)$. If $\ell_i(x)$ is the i-th Lagrange basis function based on the nodes $x_0, x_1, \ldots, x_n$ then,

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

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

Sketch of Proof

Let $f(x)$ be a polynomial of degree $2n+1$. Assuming that we can find the function $q(x)$ as mentioned in the previous slide 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 $[-1, 1]$ if

$$\int_{-1}^{1} g(x)h(x)\, dx = 0$$

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

Legendre Polynomials

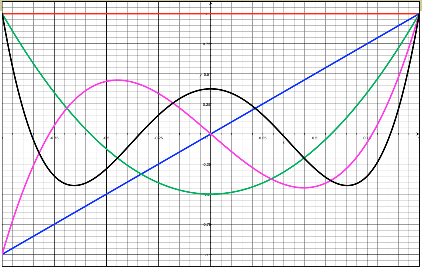
given on the exam

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

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



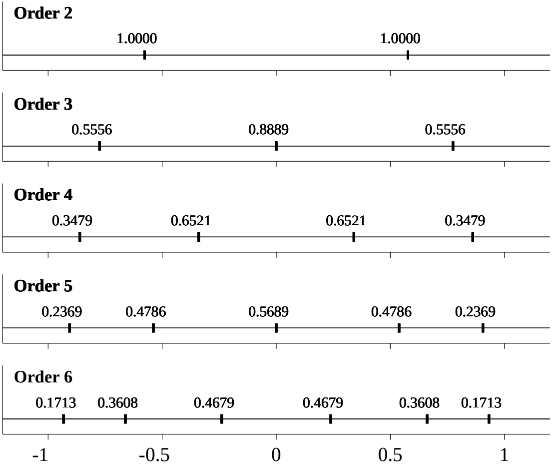
- The Legendre Polynomials are orthogonal (nice!)
- The Legendre Polynomials increase in polynomials order (like monomials)
- The Legendre Polynomials don't suffer from poor conditioning (unlike monomials...more in the linear algebra section)
- The Legendre Polynomials don't have a closed form expression (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 $\phi_{n+1}(x)$ 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 $\int_1^{1.5} x^2 \ln x \, dx = 0.192259357732796$ 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 $\int_0^1 x^2 e^{-x}\,dx = 0.160602794142788$ using Gaussian quadrature with $n = 1$.

<u>Solution</u> 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:

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

Matlab quadl

The Matlab *quadl* function is based on the adaptive Gauass-Lobatto's rule.

Gauss-Lobatto integration is similar to Gaussian quadrature except that,

- The end points of the interval are included in the nodes
- Accurate with polynomials up to degree $2n - 1$.

Example: $\int_{-1}^{1} x^5 \, dx$

$>> quadl(@(x)x.\hat{}5, -1, 1, 1.0e-20)$ (tolerance $= 1.0e-20$)
$ans =$
0