MATH 5520
Numerical Integration 2.

Dmitriy Leykekhman

Spring 2009

- Gauss Quadrature.
- Composite Quadrature Formulas.
- MATLAB's Functions.



Gauss Quadrature.

- Idea of the Gauss Quadrature is to choose nodes $x_0, \ldots, x_n$ and the weights $w_0, \ldots, w_n$ such that the formula

$$\int_a^b p(x)\ dx \approx \sum_{i=0}^{n} w_i p(x_i).$$

  is exact for a polynomial of maximum degree.

- **Lemma**

  There is no choice of nodes $x_0, \ldots, xn$ and weights $w_0, \ldots, w_n$ such that

$$\int_a^b p_N(x)\ dx \approx \sum_{i=0}^{n} w_i p_N(x_i).$$

  for all polynomials p_N of degree less or equal to N if $N > 2n + 1$.

  - The above lemma give an upper bound on the maximum degree.



Gauss Quadrature.

Example

Let's determine the weights w_0 and w_1 and the nodes x_0 and x_1 such that

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

holds for polynomials of degree 3 or less.

This seems possible since we have 4 parameters to choose w_0, w_1, x_0, x_1 and exactly 4 numbers are needed in order to define uniquely a polynomial of degree 3.



Gauss Quadrature.

Example

Let's force the formula to be exact for 1, x, x^2, and x^3. This gives us

$$w_1 + w_2 = \int_{-1}^{1} 1 \, dx = 2$$

$$w_1 x_1 + w_2 x_2 = \int_{-1}^{1} x \, dx = 0$$

$$w_1 x_1^2 + w_2 x_2^2 = \int_{-1}^{1} x^2 \, dx = \frac{2}{3}$$

$$w_1 x_1^3 + w_2 x_2^3 = \int_{-1}^{1} x^3 \, dx = 0.$$

a nonlinear system of 4 equations with for unknowns.
Usually we need a nonlinear solver to solve nonlinear systems, but in this example we can solve it analytically to obtain

$$w_1 = w_2 = 1, \quad x_1 = -\frac{1}{\sqrt{3}}, \; x_2 = \frac{1}{\sqrt{3}}.$$



Table of Gauss-Legendre Quadrature Formulas.

The weights and nodes for the first 3 Gauss-Legendre formulas on $[-1, 1]$.

x_i	w_i	exact for p_N,
$-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}$	1,1	$N = 3$
$-\sqrt{\frac{3}{5}}, 0, \sqrt{\frac{3}{5}}$	$\frac{5}{9}, \frac{8}{9}, \frac{5}{9}$	$N = 5$
-0.8611363115940525 -0.3399810435848562 0.3399810435848562 0.8611363115940525	0.3478548451374538 0.6521451548625461 0.6521451548625461 0.3478548451374538	$N = 7$



Composite Quadrature Formulas.

Let $a = x_0 < x_1 < \cdots < x_n = b$ be a partition of $[a, b]$ Then

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

Now we can approximate $\int_a^b f(x)dx$ by approximating each integral $\int_{x_i}^{x_{i+1}} f(x)dx$ by a (low degree) quadrature formula,

$$\int_{x_i}^{x_{i+1}} f(x)dx \approx \sum_{j=0}^{m} w_{ji} f(x_{ji})$$

and

$$\int_a^b f(x)dx = \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(x)dx \approx \sum_{i=0}^{n-1} \sum_{j=0}^{m} w_{ji} f(x_{ji}).$$



Composite Midpoint Rule.

Example

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



Composite Trapezoidal Rule.

Example

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

The function values $f(x_1), f(x_2), \ldots, f(x_{n-1})$ appear twice in the summation. This has to be utilized in the implementation of the composite Trapezoidal rule:

$$\begin{aligned}\int_a^b f(x)dx \approx & \frac{x_1 - x_0}{2} f(x_0) \\ & + \sum_{i=1}^{n-1} \left(\frac{x_i - x_{i-1}}{2} + \frac{x_{i+1} - x_i}{2} \right) f(x_i) \\ & + \frac{x_n - x_{n-1}}{2} f(x_n)\end{aligned}$$



Composite Simpsons Rule.

Example

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n-1} \frac{x_{i+1}-x_i}{6}\left(f(x_i)+4f(\frac{x_{i+1}+x_i}{2})+f(x_{i+1})\right).$$

Notice that the function values $f(x_1), f(x_2), \ldots, f(x_{n-1})$ appear twice in the summation. This has to be utilized in the implementation of the composite Simpson rule.



MATLAB's quad, trapz

MATLAB has several build in functions for numerical integration. We will mention a couple quad and trapz. You can get more information by typing

```
>>help quad
>> help trapz
```



MATLAB's quad

The syntax for quad

```
QUAD   Numerically evaluate integral, adaptive Simpson
quadrature.

Q = QUAD(FUN,A,B) tries to approximate the integral of
scalar-valued function FUN from A to B to within an error
of 1.e-6 using recursive adaptive Simpson quadrature.

FUN is a function handle.

The function Y=FUN(X) should accept a vector argument X
and return a vector result Y, the integrand evaluated at
each element of X.
```



Example.

For example we want to approximate

$$\int_0^{10} e^{-x^2}\, dx.$$

Then

```
>> quad('exp(-x.^2)',0,10)
```

produces

```
>> 0.886226046613606
```

If we need 10 digits of accuracy then

```
>> quad('exp(-x.^2)',0,10,1e-10)
```

produces more accurate answer

```
>> 0.886226925457492
```

