**Mathematical Methods of Engineering**

**Lecture Note 8**

**Numerical Integration**

**8.1 Introduction**

Numerical integration is an essential tool used by scientists and engineers to obtain approximate values for definite integrals that cannot be solved analytically. For example, the integral



has no closed form solution. The function  is a continuous bounded function over the interval [0, 0.5] and hence the integral exists. But it is not possible to evaluate it analytically. Numerical technique is also required when the data for variables are available in the form of table, but no mathematical relationship between them is known, as is often the case of experimental data.

The purpose of this chapter is to develop the basic methods of numerical integration.

**8.2 Method of Numerical Integration**

Recall the definition of definite integral



where  and , .

For numerical integration we shall take the formula as



where  are called the nodes which are distributed within the limits of integration [*a*, *b*], are called the weighting factors and *E* is the error.

**8.3 Gaussian Quadrature Rule**

In Gaussian quadrature formulas, the nodes and weights are determined by using a polynomial of as high degree as possible for  If there are *n* nodes and *n* weighting factors, we need to find 2*n* equations to determine 2*n* unknowns. This may be achieved by taking the formula exact i.e. for a polynomial of degree 2*n*. Using the fact that integration is additive, it will suffice to require that the integration formula is exact for 2*n* functions 

**Degree of Precision:** A quadrature formula is said to be of precision *m*, if it produces exact results when *f*(*x*) is a polynomial  but .

**Example 8.1: Derive the quadrature rule of the form and using this quadrature rule find the following integral to 3 d.p :**

**Solution:** Here there are two unknowns *a* and *b.* So we will use and then **.[If there are three unknowns , we will use ].**

Now for we can write

* ………………………(1)

Now for we can write

* …………………….(2)

Solving (1) and (2) we get

Thus we get

Here we have limit 0 & *h* but if the limits are *xo* & *x1*other then 0 & *h* then the formula changes as below

Now

Here

**8.4 Newton-Cotes Quadrature Rule**

If the nodes ’s are uniformly distributed in  with ,  and the spacing , the method is known as Newton-Cotes integration method and has the order *n*. When both the end points of the interval are included as nodes, the methods are called closed type methods, otherwise they are called open type methods.

**Closed Newton Cotes Quadrature Rule**

Assume that  are equally spaced nodes and  the first few Newton-Cotes quadrature formulas are listed below:

**8.4.1 Trapezoidal Rule**

For *n* = 1, we obtain the Trapezoidal rule.

In this case, the quadrature rule is of the form



where..

To make the simplification short and simple, the axis is translated to make  as the origin. Thus the formula we are looking for is of the form



For two unknown parameters we may assume that the method is exact for  and .

Thus

,  (1)

, 

or  (2)

From (1), we have



The Trapezoidal rule becomes



To find the precision and an estimate of error, let us take the rule is of the form



Taking , we have+



Here . Hence the degree of precision is .

Note that 

Assuming that  for  we can write the error term as

.

The Trapezoidal rule for arbitrary points  with step size *h* can be obtained by identifying the points 0 by , *h* by . In this case  and. Thus





where the notation  is used.

**8.4.2 Simpson’s Rule (Simpson 1/3 Rule)**

For *n* = 2, we obtain the Simpson’s rule (Simpson’s 1/3 rule)



where .

Simpson’s rule can easily be proved by considering the integral



or 

and then by translating the axis. This is left as an exercise for the reader.

The Simpson’s rule has degree of precision three.

The error in the formula is

, 

**8.4.3 Simpson’s 3/8-Rule**

For *n* = 3, we obtain the Simpson’s 3/8-rule



where .

The precision of the rule is 3.

The error in the formula is

, 

Higher order formula can be derived in a similar way.

**Example 8.2:** Evaluate numerically using the values given below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *x* | 0.4 | 0.5 | 0.7 | 1.0 |
| *f*(*x*) | 1.083 | 1.133 | 1.287 | 1.649 |

**Solution:** Here subinterval sizes are unequal. Using the Trapezoidal rule in each subinterval separately, we have

**Example 8.3:** The table below shows the values of  at different values of *x*:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *x* | 0.4 | 0.5 | 0.6 | 0.8 | 1.0 |
| *f(x)* | 1.083 | 1.133 | 1.197 | 1.377 | 1.649 |

Evaluate using Simpson’s rule.

**Solution :** Simpson’s rule is applied to two consecutive subintervals of equal length. Thus

for the given data we may divide the subintervls as follows:

**8.5 Composite Quadrature Rules**

To avoid the use of higher order methods and still obtain accurate results, we use the composite integration methods. We divide the interval [*a*, *b*] into a number of subintervals and evaluate the integral in each required number of subintervals by a particular method.

**Composite Trapezoidal Rule**

We divide the interval [*a*, *b*] into N subintervals , each of length , ,  and . We write



Evaluating each integral on the right hand side by the trapezoidal rule





The error in the formula is

, 

**Composite Simpson’s Rule**

We divide the interval [*a*, *b*] into N, an even number of subintervals , each of length , ,  by using. We write



Evaluating each integral on the right hand side by the Simpson’s rule





The error in the formula is

, 

**Example 8.4 :** The values of  are given for different values of *x* below.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *x* | 0 | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 | 1.2 |
| *f*(*x*) | 1.000 | 1.780 | 1.954 | 2.000 | 1.976 | 1.909 | 1.814 |

Evaluate I = using extrapolation and (i) trapezoidal rule (ii) Simpson’s rule

**Solution :** (i) Consider the integral with trapezoidal rule.





 = 

Using Richardson extrapolation



(ii) Consider the integral with Simpson;s rule



With step size *h* = 0.4, the number of subintervals =  and Simpson’s rule cannot be used.

Taking *h* = 0.6, we get 

Using Richardson’s extrapolation 

**8.6 Romberg Integration**

Romberg integration is an extrapolation formula of the Trapezoidal Rule for integration. It provides a better approximation of the integration by repeated applications of the Richardson’s Extrapolation formula. It is known that the Trapezoial rule approximation to an integral  has error behavior



Suppose that is a Trapezoidal estimation of the integral  with *n* subintervals and step size *h.* By doubling the step size the corresponding estimate is .

The first Richarson extrapolated value is

 has error of order 

The Simpson’s rule has an error of order . In fact,  is exactly the Simpson’s rule estimate. The second improved estimate  is



has error of order and so on.

**Example 8.5 :** Evaluateusing Trapezoidal rule with 1, 2 and 4 subintervals.

Improve the results using **Romberg** integration.

**Solution:** Here .

Using the trapezoidal rule we have

*n* = 1, ; 

*n* = 2, ; 

*n* = 4, 

First order extrapolated values are 



And 



Second extrapolated value is 



Results are summarized below in a Table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *N* | *H* |  |  |  |
| 1 | 1 | 0.8244 |  |  |
| 2 | 0.5 | 0.6955 | 0.6528 |  |
| 4 | 0.25 | 0.6606 | 0.6490 | 0.6488 |

**8.7 Numerical Evaluation of Double Integrals**

Multiple integrals are evaluated by expressing them as iterated integrals. For example a double integral can be expressed as follows;

or

In computing the iterated integral of the first form, we hold *x* constant while integrating with respect to *y* and then integrate with respect to *x*(vice versa for second form).

Numerical evaluation of integrals we need to to choose interval length and nodal points where the functional values to be calculated. For fixed limits, selection of interval length and nodal points are straight forward but for variable limits interval length and nodal points should be calculated for a fixed value of other variable. For example in firtst form the interval at is

**Example 8.6**

Using Simpson’s rule with 2-subintervals evaluate the double integral

The integral is over the rectangular region bounded by

Using 2-subintervals in each direction, we have

The integrand is

Integration using fixed values of are as follows:

Finally combining the integral for *x*we have

I

[MATLAB result is I = 0.38846]

Calculations are summarized in the following table:



**Example 8.7** Using Simpson’s rule with 2-subinervals evaluate the double integral

**Solution**: The region of integration is

The integrand is

With 2 equal subintervals for *x*, the interval length is =0.2. The end points of subintervals are

The interval length for *y* is variable depending on the values of *x*.

The integral over y is

The integral over y is

The integral over y is

Finally combining the integrals for *x*, we have

Summary of calculations are shown below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| kr | xr | yr | f(xr,yr) | I(xr,kr) | I(Simp) |
|  | 1 | 1 | 2 |  |  |
| 0.5 | 1 | 1.5 | 2.5 | 2.5 |  |
|  | 1 | 2 | 3 |  |  |
|  | 1.2 | 1.2 | 2.44 |  |  |
| 0.62 | 1.2 | 1.82 | 3.184 | 3.94816 | 1.640917 |
|  | 1.2 | 2.44 | 3.928 |  |  |
|  | 1.4 | 1.4 | 2.96 |  |  |
| 0.78 | 1.4 | 2.18 | 4.052 | 6.32112 |  |
|  | 1.4 | 2.96 | 5.144 |  |  |

**8.8 Numerical evaluation of Integrals using MATLAB**

Syntax of MATLAB commands for different types of integrals are as follows:

integral(fun, xmin, xmax)

integral2(fun, xmin, xmax, ymin, ymax)

integral3(fun, xmin, xmax, ymin, ymax,zmin,zmax)

Command for required number of digits

vpa(value, n) % to display value to n digits

**Note:** Define the function using “anonymous handle”. Format is shown below:

**fname=@(argument) <space> formula**

operation / (division) and ^ (for power) must preceded with . (dot).

**For example**  may be typed as

ff=@(x,y) sin(x\*y)./(x.^2+y.^2))

This will be treated as ff(x,y).

**Example 8.6**: Evaluate numerically to 7 digits using MATLAB command.

>> fun=@(x) exp(x).\*sin(x.^2) % enter function as @function

fun = @(x)exp(x).\*sin(x.^2)

>> int=integral(fun,0,1); % command for integration

>> int7=vpa(int,7) % used to increase precision

int7 =0.662701

**Example 8.8:** Evaluate numerically using MATLAB command.

>> fun2=@(x,y) 1./sqrt(x.^2+y.^2) % define as @-function

fun2 = @(x,y)1./sqrt(x.^2+y.^2)

>> ymin=@(x) –x % lower limit as function of *x*

ymin = @(x)-x

>> ymax=@(x) sqrt(x) % upper limit as a function of *x*

ymax = @(x)sqrt(x)

>> int2=integral2(fun2, 0, 2, ymin,ymax);

>> int22=vpa(int2,7) % output to 7 digits

int22 =3.811758

**Example 8.9:** Evaluate , where *R* is the region bounded by the coordinate planes and the plane , numerically using MATLAB command.

The integral can be written as an iterated integral of the form

**MATLAB commands and output**

>> clear

>> fun3=@(x,y,z) 1./(x+y+z+1).^2 % define the integrand as a @-function

fun3 = @(x,y,z)1./(x+y+z+1).^2

>> ymax=@(x) 1-x

ymax = @(x)1-x

>>zmax=@(x,y) 1-x-y

zmax = @(x,y)1-x-y

>> int3=integral3(fun3,0,1,0,ymax,0,zmax); % command for integration

>> int31=vpa(int3,7) % result using 7 digits.

int31 =0.05685282

**Exercise 8**

1. The table shows the power P supplied to the driving wheels of a car as a function of the speed . If the mass of the car is m=2000 kg, determine the time it takes for the car to accelerate from 1 to 6 . Use the trapezoidal rule for integration.

Hint:

which can be derived from Newton’s law and the definition of power .

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| v(m/s) | 0 | 1.0 | 1.8 | 2.4 | 3.5 | 4.4 | 5.1 | 6 |
| P(kW) | 0 | 4.7 | 12.2 | 19.0 | 31.8 | 40.1 | 43.8 | 43.2 |

2. The table below shows the values of  at different values of *x*:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *x* | 1.2 | 1.4 | 1.6 | 1.8 | 2.0 |
| *f*(*x*) | 3.728 | 4.124 | 4.525 | 5.123 | 5.626 |

(i) Use Trapezoidal rule and Richardson’s extrapolation to estimate .

(ii) Use Simpson’s rule and Richardson’s extrapolation to estimate .

3. The table below shows the values of  at different values of *x*:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *x* | 1.2 | 1.35 | 1.5 | 1.65 | 1.8 |
| *f*(*x*) | 3.32 | 3.86 | 4.48 | 5.21 | 6.05 |

(i) Use Trapezoidal rule and Richardson’s extrapolation to estimate .

(ii) Use Simpson’s rule and Richardson’s extrapolation to estimate .

4. A river is 50 meters wide. The depth ‘d’ in meters at distance *x* meters from one back is given by the following table. Calculate the arc of cross-section of the river using Simpson’s rule.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *x* | 0 | 10 | 20 | 30 | 40 | 50 |
| *d* | 0 | 4 | 7 | 9 | 12 | 15 |

5. The car gives the velocity of a moving particle at time t seconds. Find the distance covered by the particle in 8 seconds.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *t* | 0 | 2 | 4 | 6 | 8 | 9 |
| *v* | 4 | 6 | 16 | 34 | 60 | 75 |

6. Derive the following quadrature rules :

(a)

(b)

In each case find its degree of precision.

Use the above quadrature rule to estimate the following integrals to 3 decimal places.

(i) , (ii) .

7. The table below shows the values of  at different values of *x*:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *x* | 1.0 | 1.2 | 1.4 | 1.6 | 1.8 |
| *f(x)* | 1.831 | 2.592 | 3.515 | 4.643 | 5.926 |

Evaluate using Tapezoidal rule with 1, 2 and 4 subintervals.

Improve your results using Romberg integration.

8. Compute following integrals using Trapezoidal rule with subintervals 1, 2 and 4.

Improve your results using Romberg integration. And write MATLAB code to evaluate the integrals. Also find exact result ( if possible ) & find percentage error.

1. (b) (c) (d)

9. Using Simpson’s rule with two subintervals evaluate the following double integrals and also write MATLAB code to evaluate.

(a) , (b) ,

(c) , (d) .

(e) (f)

10. Evaluate the following integrals using MATLAB commands.

(a)

(b)