BRAC UNIVERSITY
CSE 330: Numerical Methods (LAB)

Lab 8: Numerical Integration

Introduction:

There are two cases in which engineers and scientists may require the help of numerical integration technique (1) Where experimental data is obtained whose integral may be required and (2) where a closed form formula for integrating a function using calculus is difficult or so complicated as to be almost useless. For example the integral $F(x) = \int_0^A \frac{x^3}{e^x - 1} dx$ has no analytic expression, numerical integration technique must be used to obtain approximate values of F(A).

Formula for numerical integration called quadrature are based on fitting a polynomial through a specified set of points (experimental data or function values of the complicated function) and integrating (finding the area under the fitted polynomial.) this approximate function(fitted polynomial). Lagrange or Newton interpolating polynomial may be used for this purpose.

Methods of Numerical Integration:

The Newton-Cotes.formulas are the most common numerical integration schemes. They are based on the strategy of replacing n complicated function or tabulated data with a polynomial that is easy to integrate:

$$I = \int_a^b f(x)dx \cong \int_a^b \phi(x)dx$$

In this case we replace $f(x)$ by an interpolating polynomial $\phi(x)$ and obtain an approximate value of the definite integral by integrating $\phi(x)$.

The integral can also be approximated using a series of polynomials applied piecewise to the function or data over segments of constant length

The First Newton-Cotes Formula: The Trapezoidal Rule:

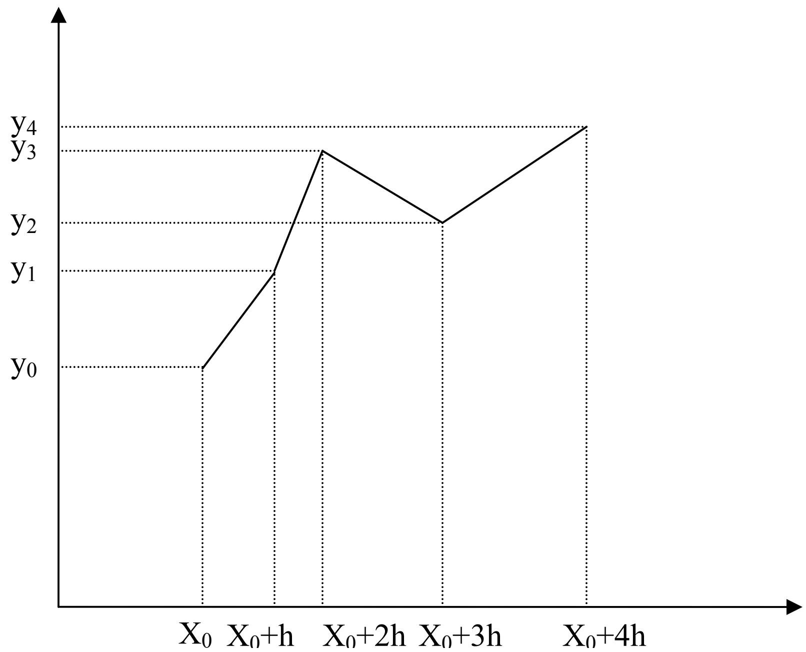


Figure 1: The trapezoidal method

Assume that the values of function f(x) are given at x_0, x_0+h, x_0+2h….x_0+nh and it is required to find the integral of f(x) between x1 and x1+nh. The simplest technique is to fit piece-wise straight lines through f(x_0), f(x_0+h)……f(x_0+nh) and determine the area under this approximating function as shown in figure 2.

For the first two points we can write: $$\int_{x_0}^{x_1} y\,dx = \frac{h}{2}\left[y_0 + y_1\right] \quad \text{......... ....(1)}$$

From the above figure it is evident that the result of integration between x_0 and x_0+nh is nothing but the sum of areas of some trapezoids. In equation form this can be written as

$$\int_{x_0}^{x_n} y\,dx = \sum_{i=1}^{n} \frac{(f_i + f_{i+1})}{2} h \quad \text{.......... ....(2)}$$

The above integration formula is known as Composite Trapezoidal rule.

The composite trapezoidal rule can be explicitly written as:

$$\int_{x_0}^{x_n} y\,dx = \frac{h}{2}\left[y_0 + 2\left(y_1 + y_2 + \cdots + y_{n-1}\right) + y_n\right] \quad \text{......... ....(3)}$$

Example 1: Evaluate $I = \int_0^{0.5} [10\, e^{-x} \sin(2\,\pi x)]^2\, dx,$ using trapezoidal formula. Use step size h = 0.125.
First write a code and then compare your result with that obtained from built in function.
Ans: 15.401

Editor - L:\MATLAB71\work\trapppp.m*

```
clc;
clear all;

f=@(x) (10*exp(-x)*sin(2*pi*x))^2; % declaring the function
x_low=0;
x_up=0.5;
h=0.125;
x=x_low:h:x_up;

for k=1:length (x)
   y(k)=f(x(k));          % evaluating the function at each x point
end

s=0;
for i=1:length(x)-1
   s=s+(y(i)+y(i+1))*h/2;
end
fprintf('Result from manual calculation = %d\n',s);

r=trapz(x,y);   % using the "trapz ()" built in function.
fprintf('Result from MATLAB built-in function= %d\n',s);
```

Figure 2: Code for trapezoidal method

<u>Lab task 1:</u> There is no closed form solution for the error function $\operatorname{erf}(a) = \frac{2}{\sqrt{\pi}} \int_0^a e^{-x^2}\, dx$. Using trapezoidal rule evaluate erf (0.5). Use h = 0.1. Compare your result with that obtained by built-in MATLAB function *"erf (0.5)"*

Ans : 0.5205

Simpson's 1/3 rule:

This is based on approximating the function f(x) by fitting quadratics through sets of three points. For only three points it can be written as:

$$\int_{x_0}^{x_0+2h} y\, dx = \frac{h}{3}\left[y_0 + 4y_1 + y_2\right]$$

It is evident that the result of integration between x_0 and x_0+nh can be written as

$$\int_{x_0}^{x_0+nh} y\ dx = \sum_{i=0,,2,4\ldots n-2} \frac{h}{3}\left[y_i + 4y_{i+1} + y_{i+2}\right]$$

$$\Rightarrow \int_{x_0}^{x_n} y\ dx = \frac{h}{3}\left[y_0 + 4\left(y_1 + y_3 + y_5 + \cdots + y_{n-1}\right) + 2\left(y_2 + y_4 + y_6 + \cdots + y_{n-2}\right) + y_n\right]$$

In the above formula it is implied that y is known at an **odd number** of points. [even number of sub intervals].

Simpson's 3/8 rule:

This is based on approximating the function f(x) by fitting *cubic interpolating* polynomial through sets of *four* points. For only four points it can be written as:

$$\int_{x_0}^{x_0+3h} y\ dx = \frac{3h}{8}\left[y_0 + 3y_1 + 3y_2 + y_3\right]$$

It is evident that the result of integration between x_0 and x_0+nh can be written as

$$\int_{x_0}^{x_0+nh} y\ dx = \sum_{i=0,3,6\ldots n-3} \frac{h}{3}\left[y_i + 3y_{i+1} + 3y_{i+2} + y_{i+3}\right]$$

$$\Rightarrow \int_{x_0}^{x_n} y\ dx = \frac{3h}{8}\left[y_0 + 3y_1 + 3y_2 + 2y_3 + 3y_4 + 3y_5 + 2y_6 + \ldots + 2y_{n-3} + 3y_{n-2} + 3y_{n-1} + y_n\right]$$

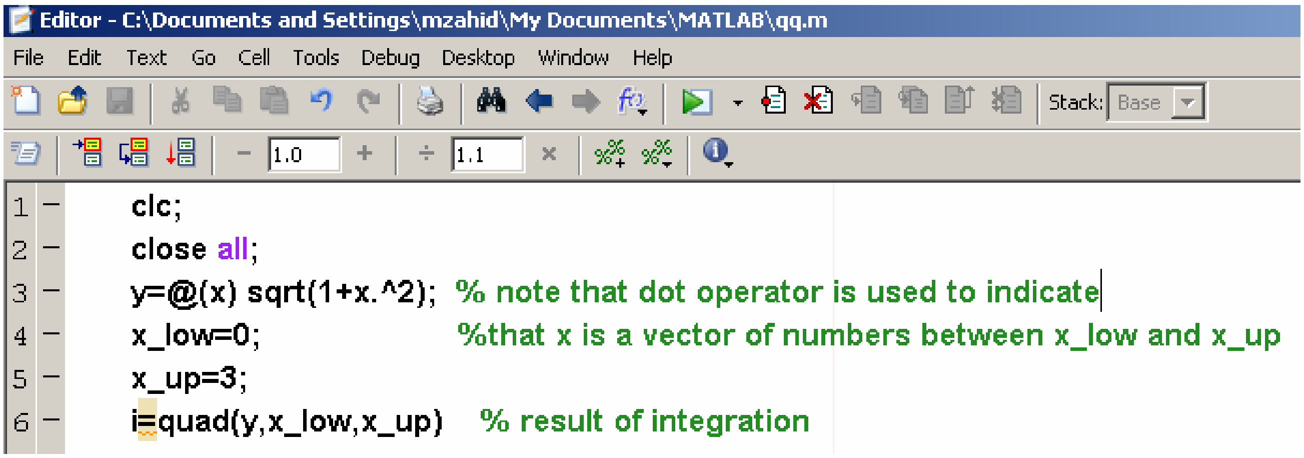
In the above formula it is implied that y is known at (n+1) number of points where n is divisible by 3.

Use of MATLAB`s Integration Features:

In MATLAB the simplest function for integration is **trapz ( ).** A more accurate function based on adaptive Simpson quadrature is **quad ( ).**

The syntax for **quad ( )** is: **q = quad(fun,a,b)**
Where,
fun =the function to be integrated.
a=lower limit of integration
b= upper limit of integration

Example : Evaluate $I = \int_0^3 \sqrt{1+x^2}\,dx$



Lab task 2: Evaluate $I = \int_{-1}^{2} \frac{x^3}{e^x - 1}\,dx$ [Don't forget to use the dot operator in power and division]

Ans: 1.6511

Home Work:

[1]. Write a code to simulate Simpson's 1/3 rule. i.e implement the expression

$$\int_{x_0}^{x_n} y\,dx = \frac{h}{3}\left[y_0 + 4(y_1 + y_3 + y_5 + \cdots + y_{n-1}) + 2(y_2 + y_4 + y_6 + \cdots + y_{n-2}) + y_n\right]$$

In your code you will input step size h [or the number of steps n] , the x values. First use your function to compute y values, then find out the integration based on Simpson's 1/3 rule. [See figure 2 for coding style.]

[2]. Evaluate $I = \int_2^5 \frac{Sin(x)}{x^2 - 1}\,dx$ use **quad ( )** and **trapz ( )** functions. For trapz use 30 intervals (h = 0.1034). [to generate the x values you can use, *x = **linspace** (2,5,30)]* .Remember for trapz ( ) you need to input x and y vectors whereas for quad ( ) you need to input the function and limits.

Ans: For trapz ( ), I=0.0542 , for quad( ), I=0.0537