# Department of Computing

# MATH 333: Numerical Analysis

# Class: BSCS-9ABC

# Lab 11: Numerical Integration

# Date: April 20, 2022

# Time: 10:00 - 1:00

# Lab Engineer: Anum Asif

Fatima Seemab

291310

BSCS 9B

# Lab 11: Numerical Integration

**Introduction**

Matlab represents polynomials with a vector of coefficients. The length of the vector will always be one more than the order of the polynomial

**Objectives**

The purpose of this lab is to get familiar with Numerical Integration.

**Tools/Software Requirement**

Matlab

**Description**

Need of Numerical Integration: For evaluating the definite integral of a function that has no explicit anti-derivatives or whose anti-derivatives is not easy to obtain.

**Help (integration function):**

For help, the following integration function has been provided. Either understand it and implement the two tasks or you can write your own code, based on your understanding.

function [integral\_value] = numInt(x,y)

n = length(x);

for i = 1 : n-1

d(i) = x(i+1) - x(i);

end

intervalCount=1;

for j = 1:length(d)-1

if d(j+1)~=d(j)

intervalCount=intervalCount+1;

boundary(j) = j + 1;

end

end

boundary(j+1)=j+1;

boundary=nonzeros(boundary);

boundary = unique(boundary);

if boundary(1) - 1 == 1

integral(1) = (boundary(1)-x(1))/2 \* (y(1) + y(boundary(1)));

elseif boundary(1) - 1 == 2

integral(1) = (boundary(1) - x(1))/6 \* (y(1) + 4\*y((boundary(1)+1)/2) + y(boundary(1)));

elseif boundary(1) - 1 == 3

integral(1) = 3\*(boundary(1) - x(1))/8 \* (y(1) + 3\*y((boundary(1)+1)/3 - 1) + 3\*y((boundary(1)+1)/3 + 1) + y(boundary(1)));

end

for k = 2:intervalCount-1

if boundary(k) - boundary(k-1) == 1

integral(k) = (x(boundary(k))-x(boundary(k-1)))/2 \* (y(boundary(k-1)) + y(boundary(k)));

elseif boundary(k) - boundary(k-1) == 2

integral(k) = (x(boundary(k)) - x(boundary(k-1)))/6 \* (y(boundary(k-1)) + 4\*y((boundary(k)+boundary(k-1))/2) + y(boundary(k)));

elseif boundary(k) - boundary(k-1) == 3

integral(k) = 3\*(x(boundary(k)) - x(boundary(k-1)))/24 \* (y(boundary(k-1)) + 3\*y(fix((boundary(k)+boundary(k-1))/2)) + 3\*y(fix((boundary(k)+boundary(k-1))/2) + 1) + y(boundary(k)));

end

end

if n - boundary(length(boundary)) == 1

integral(k+1) = (x(n)-x(boundary(length(boundary))))/2 \* (y(n) + y(boundary(length(boundary))));

elseif n - boundary(length(boundary)) == 2

integral(1) = (x(n)-boundary(length(boundary)))/6 \* (y(n) + 4\*y((n+boundary(length(boundary)))/2) + y(boundary(length(boundary))));

elseif n - boundary(length(boundary)) == 3

integral(1) = 3\*(x(n)-boundary(length(boundary)))/24 \* (y(n) + 3\*y((fix((n + boundary(length(boundary)))/2))) +3\*y((fix((n + boundary(length(boundary)))/2))+1) + y(boundary(length(boundary))));

end

integral\_value = sum(integral);

**Lab Task:**

1. The work produced by a constant temperature, pressure-volume thermodynamic process can be computed as

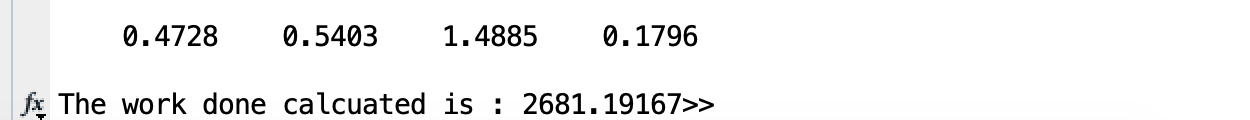
Where is work, is pressure and is volume. Using above mentioned Integral function, compute the work in kN.m for following data:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Pressure | 336 | 294.4 | 266.4 | 260.8 | 2260.5 | 249.6 | 193.6 | 165.6 |
| , | 0.5 | 2 | 3 | 4 | 6 | 8 | 10 | 11 |

**CODE**

|  |
| --- |
| x=[336 294.4 266.4 260.8 260.5 249.6 193.6 165.6];  y=[0.5000 2.0000 3.0000 4.0000 6.0000 8.0000 10.0000 11.0000];  result = numInt(y,x);  disp(result) |

**OUTPUT**



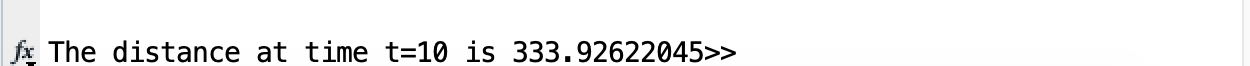
1. Suppose that the upward force of air resistance on a falling object is proportional to the square of the velocity. For this case velocity can be computed as

Where a second-order drag coefficient. If , and , use trapezoidal rule to determine how far the object falls in . Use a sufficiently high that you get eight significant digits of accuracy.

**CODE**

|  |
| --- |
| clear  t = 0:0.001:10;  i=length(t);  m=68.1;  cd=0.25;  g=9.8;  for f=1:i  v(f)=sqrt((g\*m)/cd)\*tanh(sqrt((g\*cd)/m)\*t(f));  end  disp(t)  disp(v)  result = trapz(t,v);  fprintf("The distance at time t=10 is %0.8f",result) |

**OUTPUT**



**Deliverables**

Submit single word file with matlab code and screen shot of Output.