Student Name: Vibhanshu Jain
Student ID: CS19B1027

ICME MS5523 (2022)

Homework-3

---

1. Question 1

The word quadrature means any method for approximating the value of a definite integral. The quadrature schemes in numerical analysis is very useful in finding the approximate value of integrals with very less errors.
There are three types of quadrature schemes in numerical analysis.

(a) **Trapezoidal Rule**
The rule is based on approximating the region under the graph of the function f(x) as a trapezoid and calculating the area. The function can be written as:
$I_t \simeq \left(\frac{f(a)+f(b)}{2}\right) * (b-a) + O(h^2)$
Where, h = b - a. Here the error will be $Error \simeq O(h^2)$

Let's take the example of the function $f(x) = x^2$ integrating from 1 to 2. Therefore

$I_t \simeq \left(\frac{2^2+1^2}{2}\right) * (2-1) = 2.5$
There the error is
$Error = \frac{2^3-1^3}{3} - 2.5 = -0.166667$

(b) **Simpson's rule**
Simpson's rules are several approximations for definite integrals , named after Thomas Simpson. The most basic of these rules called Simpson's 1/3 rule. The rule can be presented in the form of the equation as:
$I_t \simeq \left(\frac{f(a)+4f(\frac{a+b}{4})+f(b)}{6}\right) * (b-a)$
Let's take the example of the function $f(x) = x^2$ integrating from 1 to 2. Therefore
$I_t \simeq \left(\frac{2^2+4(\frac{1+2}{4})^2+1^2}{6}\right) * (2-1) = \frac{29}{24}$
There the error is
$Error = \frac{2^3-1^3}{3} - \frac{29}{24} = 1.124966$

(c) **Generalization**
This is the most general rule for the quadrature schemes. This rule can be equated as:
$I_x = \int_a^b f(x)dx \simeq (w_1 * f(x_1) + w_2 * f(x_2) + w_3 * f(x_3) + ... + w_n * f(x_n))$
Let's define this equation as, where x_i represents the points and w_i represents the weights corresponding to these points.
$I^t = (w_1 * f(x_1) + w_2 * f(x_2) + w_3 * f(x_3) + ... + w_n * f(x_n))$

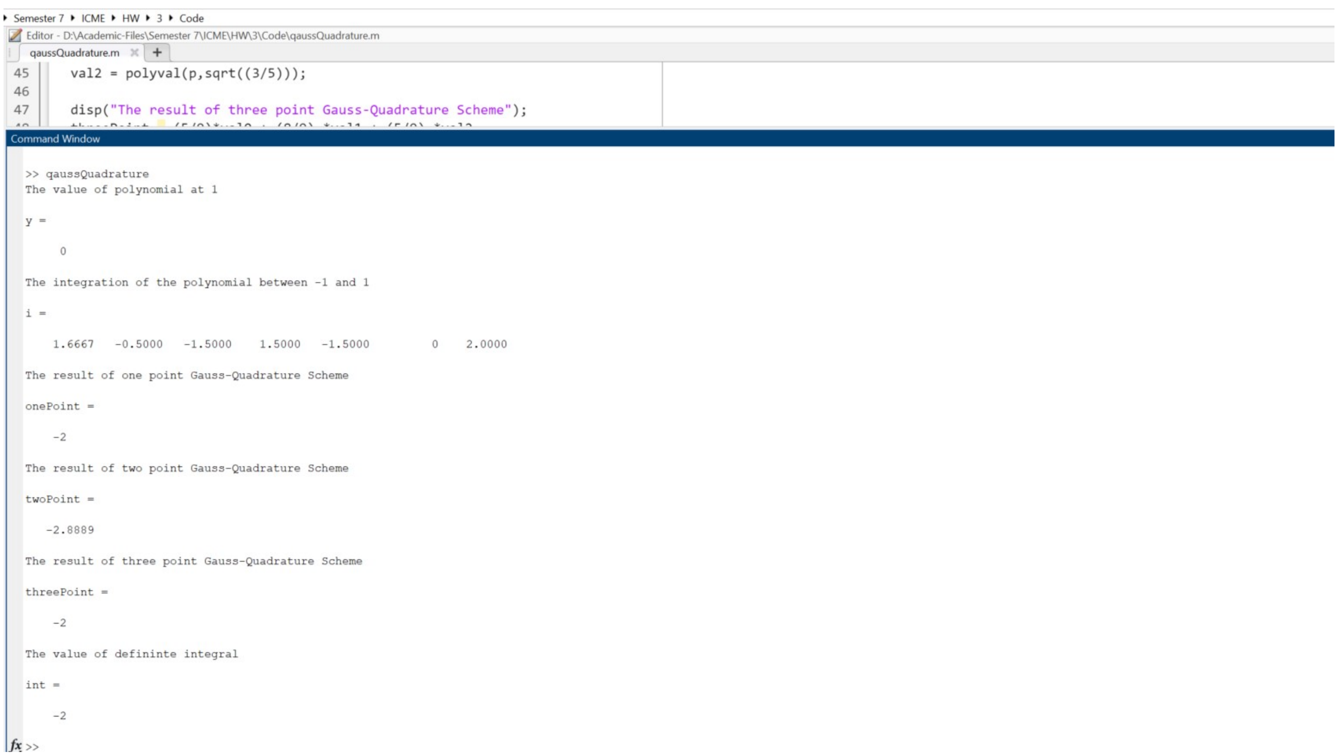
The error in the solution can be written as
$Error = \int_a^b f(x)dx - I^t$

Let's take the example of the function $f(x) = x^2$ integrating from 1 to 2 taking the $x_1 = 1$ and $x_2 = 2$. Therefore
$I^t = \left(w_1 * 2^2 + w_2 * 1^2\right) = 4 * w_1 + w_2$

2. **Question 2**

The solution of the One Point Gauss-Quadrature Scheme is -2, two point Gauss-Quadrature Scheme is -2.8889, three point Gauss-Quadrature Scheme is -2 and the value of definite integral is -2. As we can see that the value for the three point Gauss-Quadrature Scheme is exactly same as the correct which confirms the fact that it gives correct values till the polynomial of degree 5.

Here is the code and the screen shots:




```
%% HW3 Q2
%% Vibhanshu Jain - CS19B1027

%% Polynomial functions

function qaussQuadrature

%% The polynomial equation
%% y(x) = (-4)*x^5 + (5)*x^4 + (2)*x^3 + (-3)*x^2 + (1)*x^1 + (-1)*x^0

%% the polynomial representation of matlab
p = [-4 5 2 -3 1 -1];
```

```
%% Printing the value at x = 1
disp("The value of polynomial at 1");
y = polyval(p,1)

%% The intergration of the polynomial
disp("The integration of the polynomial between -1 and 1");
i = diff(polyint(p,[-1 1]))

%% One Point Gauss-Quadrature Scheme
%% x1 = 0 and w1 = 2
%% I = w1*f(x1)

disp("The result of one point Gauss-Quadrature Scheme");
val0 = polyval(p,0);
onePoint = 2.*val0

%% Two Point Gauss-Quadrature Scheme
%% x1 = 1/3 & x2 = -1/3 and w1 = 1 & w2 = 1
%% I = w1*f(x1) + w2+f(x2)

val0 = polyval(p,1/sqrt(3));
val1 = polyval(p,-(1/sqrt(3)));
disp("The result of two point Gauss-Quadrature Scheme");
twoPoint = 1.*val0 + 1.*val1

%% Three Point Gauss-Quadrature Scheme
%% x1 = -(3/5) & x2 = 0 & x3 = (3/5) and w1 = 8/9 & w2 = 5/9 & w3 = 5/9
%% I = w1*f(x1) + w2+f(x2)

val0 = polyval(p,-sqrt((3/5)));
val1 = polyval(p,0);
val2 = polyval(p,sqrt((3/5)));

disp("The result of three point Gauss-Quadrature Scheme");
threePoint = (5/9)*val0 + (8/9).*val1 + (5/9).*val2

%% The value of the definite intergral
disp("The value of defininte integral");
intfun = polyint(p);
int = polyval(intfun,1) - polyval(intfun,-1)

end
```

Question 3
Here attached the code snippet and screenshots.

```
%% HW3 Q3
%% Vibhanshu Jain - CS19B1027
```

```
%% Polynomial functions

function qaussQuadratureCosh


%% One Point Gauss-Quadrature Scheme
%% x1 = 0 and w1
%% I = w1*f(x1)

disp("The result of one point Gauss-Quadrature Scheme");
val0 = cosh(0);
onePoint = 2.*val0

%% Two Point Gauss-Quadrature Scheme
%% x1 = 1/3 & x2 = -1/3 and w1 = 2 & w2 = 1
%% I = w1*f(x1) + w2+f(x2)

val0 = cosh(1/sqrt(3));
val1 = cosh(-(1/sqrt(3)));
disp("The result of two point Gauss-Quadrature Scheme");
twoPoint = 1.*val0 + 1.*val1

%% Three Point Gauss-Quadrature Scheme
%% x1 = 0 & x2 = (3/5) & x3 = -(3/5) and w1 = 2 & w2 = 1
%% I = w1*f(x1) + w2+f(x2)

val0 = cosh(-sqrt((3/5)));
val1 = cosh(0);
val2 = cosh(sqrt((3/5)));
disp("The result of three point Gauss-Quadrature Scheme");
threePoint = (5/9)*val0 + (8/9).*val1 + (5/9).*val2

end
```

Semester 7 ▸ ICME ▸ HW ▸ 3 ▸ Code

Editor - D:\Academic-Files\Semester 7\ICME\HW\3\Code\qaussQuadratureCosh.m

qaussQuadratureCosh.m

```matlab


%% One Point Gauss-Quadrature Scheme
%% x1 = 0 and w1
%% I = w1*f(x1)

disp("The result of one point Gauss-Quadrature Scheme");
val0 = cosh(0);
onePoint = 2.*val0

%% Two Point Gauss-Quadrature Scheme
%% x1 = 1/√3 & x2 = -1/√3 and w1 = 2 & w2 = 1
%% I = w1*f(x1) + w2+f(x2)

val0 = cosh(1/sqrt(3));
val1 = cosh(-(1/sqrt(3)));
disp("The result of two point Gauss-Quadrature Scheme");
twoPoint = 1.*val0 + 1.*val1

%% Three Point Gauss-Quadrature Scheme
```

Command Window

```
>> qaussQuadratureCosh
The result of one point Gauss-Quadrature Scheme

onePoint =

     2

The result of two point Gauss-Quadrature Scheme

twoPoint =

    2.3427

The result of three point Gauss-Quadrature Scheme

threePoint =

    2.3503

fx >>
```