Numerical Integration using MATLAB

Gauss-Legendre quadrature

Q1: Solve: $\int_{0.2}^{1.4} \sin x + e^x$.

Sol^n :

```
clear all;

% function and inputs
f = @(x) sin(x) + exp(x);
ll = input('Enter the lower limit: ');
ul = input('Enter the upper limit: ');
n = input('enter the gauss point: ');

% We know for Gauss quadrature the value of x and w came from the legendre
% polynomial
h = (ul-ll)/2;
k = (ul+ll)/2;
s= 0 ;

% for 2 point formula
if n == 2
    w1 =1;
    w2 =1;
    x1 = 1/sqrt(3);
    x2 = -1/sqrt(3);
    G_2 = h *(w1*f(h*x1+k)+w2*f(h*x2+k))
    disp(G_2)

% for 3 point formula
elseif n == 3
    w1 = 8/9;
    w2 = 5/9;
    w3 = 5/9;
    x1 = 0;
    x2 = sqrt(3/5);
    x3 = - sqrt(3/5);
    G_3 = h *(w1*f(h*x1+k)+w2*f(h*x2+k)+w3*f(h*x3+k))
    disp(G_3)
end
```

```
G_3 = 3.6439
    3.6439
```

Trapizoidal Rule:

Q2: Solve: $\int_{0.2}^{1.4} \left(\sin x - \log_e x + e^x\right)$ **using Trapizoidal rule.**

Sol^n :

```
clear all;

% function and inputs
f = @(x) sin(x) - log(x) + exp(x);
ll = input('Enter the lower limit: ');
ul = input('Enter the upper limit: ');
n = input('Enter the number of sub-division: ');

h = (ul - ll) / n; % width of sub-intervals

% values of x0 and functional values of x0
x0 = ll : h : ul;
y0 = f(x0);

% performing Trapizoidal rule of integration
 I = (h / 2) * ((y0(1) + y0(end)) + 2 * sum(y0(2 : end - 1)));

 fprintf('The numerical integral value of the function is: ');
```

```
The numerical integral value of the function is:
```

```
 disp(I);
```

```
    4.0562
```

Simpson's $\frac{1}{3}$ rule:

Q3: Solve: $\int_{0.2}^{1.4} \left(\sin x - \log_e x + e^x\right)$ **using Simpson's** $\frac{1}{3}$ **rule.**

Sol^n :

```
clear all;

% function and inputs
f = @(x) sin(x) - log(x) + exp(x);
ll = input('Enter the lower limit: ');
ul = input('Enter the upper limit: ');
n = input('Enter the number of sub-division: ');

h = (ul - ll) / n; % width of sub-intervals

% values of x0 and functional values of x0
x0 = ll : h : ul;
y0 = f(x0);

% performing Simpson 1/3 rule of integration
I = (h / 3) * ((y0(1) + y0(end)) + 4 * sum(y0(2 : 2 : end - 1)) + 2 * sum(y0(3 : 2 : end - 2)))
```

```
fprintf('The numerical integral value of the function is: ');
```

```
The numerical integral value of the function is:
```

```
disp(I);
```

```
    4.0511
```

Simpson's $\frac{3}{8}$ rule:

Q4: Solve: $\int_{0.2}^{1.4} \left(\sin x - \log_e x + e^x\right)$ using Simpson's $\frac{3}{8}$ rule.

Sol^n :

```
clear all;

% function and inputs
f = @(x) sin(x) - log(x) + exp(x);
ll = input('Enter the lower limit: ');
ul = input('Enter the upper limit: ');
n = input('Enter the number of sub-division: ');

h = (ul - ll) / n; % width of sub-intervals

% values of x0 and functional values of x0
x0 = ll : h : ul;
y0 = f(x0);

% performing Simpson 3/8 rule of integration
I = y0(1) + y0(end);
for i = 2 : n
    if mod(i - 1, 3) == 0
        I = I + 2 * y0(i);
    else
        I = I + 3 * y0(i);
    end
end
I = (3 / 8) * I  * h;

fprintf('The numerical integral value of the function is: ');
```

```
The numerical integral value of the function is:
```

```
disp(I);
```

```
    4.0512
```

Weddle's rule:

Q5: Solve: $\int_{0.2}^{1.4} \left(\sin x - \log_e x + e^x\right)$ **using Weddle's rule.**

Sol^n :

```
clear all;

% function and inputs
f = @(x) sin(x) - log(x) + exp(x);
ll = input('Enter the lower limit: ');
ul = input('Enter the upper limit: ');
n = input('Enter the number of sub-division: ');

h = (ul - ll) / n; % width of sub-intervals

% values of x0 and functional values of x0
x0 = ll : h : ul;
y0 = f(x0);

% performing Weddle's rule of integration
I = y0(1) + y0(end);
for i = 2 : n
    if mod(i - 1, 6) == 1 || mod(i - 1, 6) == 5
        I = I + 5 * y0(i);
    elseif mod(i - 1, 6) == 3
        I = I + 6 * y0(i);
    elseif mod(i - 1, 6) == 0
        I = I + 2 * y0(i);
    else
        I = I + y0(i);
    end
end
I = (3 / 10) * I  * h;

fprintf('The numerical integral value of the function is: ');
```

```
The numerical integral value of the function is:
```

```
disp(I);
```

```
    4.0510
```