INDIAN INSTITUTE OF TECHNOLOGY GANDHINAGAR

MA 202: MATHEMATICS - IV
Semester–II, Academic Year 2022-23

Tutorial Set -3
Question - 4

By

Kush Patel
[20110131]

➔ In question 4, we are supposed to find the integration of the given function by taking the value of x and y by ourselves. I have used Trapezoidal method and Simpson's $\frac{1}{3}$ rule. We also need to ensure that error should be less than the tolerance which is 0.001. The equations are given below.

$$\text{(a)} \qquad I(y) = \int_{-\infty}^{y} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx$$

$$\text{(b)} \qquad Ei(x) = \int_{x}^{\infty} \frac{e^{-t}}{t} dx$$

$$\text{(c)} \qquad Si(z) = \int_{0}^{z} \frac{\sin(t)}{t} dt$$

➔ We only need to find the integral of equation (b) and (c). So, for equation (b), my function takes the values a, b and n as an input. As we can see above, 'b' should be big enough as compared to 'a' so it looks like infinity for 'a'. The value of n should be large. I made a function that takes input and returns the value of f(x) and its derivative values.

➔ Then, to find the error, I ran a loop that checked all the possib;e values of its derivative and returned a maximum value. We need that maximum value to compute the error because we need to make sure that our error should be less than tolerance. Then, I have applied both the Trapezoidal method and Simpson's method for each of these equations.

```
% Trapezoidal Method
function answer = T4_20110131(a,b,n)
format('long')

% % Part-a
% function [f,ddf] = MyFunc(x)
%     f = (1/(sqrt(2*pi))) * exp((-x^2)/2);
%     ddf = diff(diff(f));
% end

% % Part-b
% function [f,ddf] = MyFunc(x)
%     f = exp(-x)/x;
%     ddf = diff(diff(f));
% end

% Part-c
function [f,ddf] = MyFunc(x)
    f = sin(x)/x;
    ddf = diff(diff(f));
end

max = 0;
h = (b-a)/n;
for j =0:n
    k = a + j*h;
    [~,ddf] = MyFunc(k);
    if abs(ddf)>max
```

```
max = 0;
h = (b-a)/n;
for j =0:n
    k = a + j*h;
    [~,ddf] = MyFunc(k);
    if abs(ddf)>max
        max = abs(ddf);
    else
        continue
    end
end
err = (((b-a)^3)/(12*n*n))*max;
tol = 0.001;
if err < tol
    x = a;
    sum = 0;
    for i = 1:(n-1)
        x = x+h;
        [f1,~] = MyFunc(x);
        sum = sum + f1;
    end
    [fa,~] = MyFunc(a);
    [fb,~] = MyFunc(b);
    answer = (h/2) * (fa + fb + 2*(sum));
else
    disp('Please change the value of n')
end
end
```

Code of Trapezoidal Method

```
>> answer = T4_20110131(0.0001,1000,1000000)

answer =

   1.570133122033466

>> answer = T4_20110131(0.001,1000,1000000)

answer =

   1.569233122089593
```

Simulated output by Gauss Elimination

➔ Here, I took a = 0.001, 0.0001 and b = 1000. Here, b is high enough as compared to a so we can assume that b is like infinity for a. Here the value of n is 1000000 which is good enough to make sure that error is less than the tolerance.

➔ For equation (c), the value of a should be nearly equal to zero and the value of b could be anything. Here also I apply the same concept to encounter the error and tolerance. Then, I have applied both the Trapezoidal method and Simpson's method for them.

```
% Simpson's 1/3 Integration method
function answer = T4_20110131(a,b,n)
format('long')

% % Part-a
% function [f,ddf] = MyFunc(x)
%     f = (1/(sqrt(2*pi))) * exp((-x^2)/2);
%     ddf = diff(f,4);
% end

% % Part-b
% function [f,ddf] = MyFunc(x)
%     f = exp(-x)/x;
%     ddf = diff(f,4);
% end

% Part-c
function [f,ddf] = MyFunc(x)
    f = sin(x)/x;
    ddf = diff(f,4);
end

max = 0;
h = (b-a)/n;

if rem(n,2)==1
    disp('Enter a valid n!! (Even Number)')
else
```

```
for j =0:n
    k = a + j*h;
    [~,ddf] = MyFunc(k);
    if abs(ddf)>max
        max = abs(ddf);
    else
        continue
    end
end
err = (((b-a)^5)/(180*n^4))*max;
tol = 0.001;
if err < tol
    x = a;
    sumo = 0;
    sume = 0;
    for i = 1:1:(n-1)
        x = x+h;
        [f1,~] = MyFunc(x);
        if rem(i,2) == 1
            sumo = sumo + f1;
        else
            sume = sume + f1;
        end
    end
    [fa,~] = MyFunc(a);
    [fb,~] = MyFunc(b);
    answer = (h/3) * (fa + fb + 4*(sumo) + 2*(sume));
else
```

Code of Simpson's $\frac{1}{3}$ Integration Rule

```
>> answer = T4_20110131(0.0001,1000,1000000)

answer =

   1.570133121983929

>> answer = T4_20110131(0.001,1000,1000000)

answer =

   1.569233122015102
```

Simulated output by Gauss-Seidel Method

➔ As we can see that both values from the both methods are the same.