LAB -8

MA202 Numerical Techniques

B. Tech. II year CSE

Name - Hritik Kumar

Roll No - 202051088

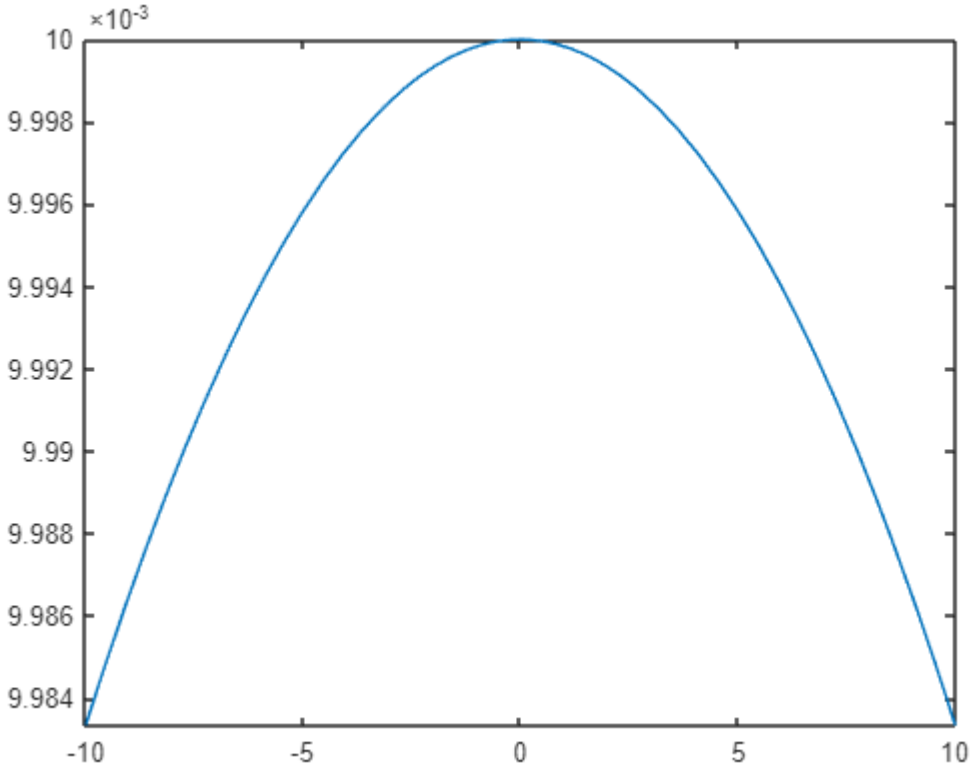
Question 1:

1. For lambda (here represented as c) = 0.01

In the case of sinx/x we used taylor form to get the answer not the normal expression as sinx/x because during integrating the function will misbehave at the point x = 0 as we have x in the denominator in this expression.

```
clear variables
close all
clc

syms p
c = 0.01;

f_taylor = @(x) symsum(((-1)^p)*(c^(2*p+1))*x^(2*p)/factorial(2*p +1),p ,0,100);
f_original = @(x) sin(c*x)/x;

a = -10;
b = 10;

n = 300;

fplot(f_original, [-10 10]);
```

```
Warning: Function behaves unexpectedly on array inputs. To improve performance, properly vectorize
your function to return an output with the same size and shape as the input arguments.
```




```
h = (b-a)/n;
sum = 0;
sum2 = 0;
```

1(i). Trapezoidal Rule

```
for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f_taylor(x(k));
    sum = sum + y(k);
    sum2 = sum2 + f_original(x(k));
end

answer1 = h/2* (f_original(a) + f_original(b) + 2*sum2);
answer2 = h/2* (f_taylor(a) + f_taylor(b) + 2*sum);

fprintf("The value of the integration from Trapezoidal Rule and lambda = 0.01 (using Or
```

```
The value of the integration from Trapezoidal Rule and lambda = 0.01 (using Original Expression) is NaN.
```

```
fprintf("The value of the integration from Trapezoidal Rule and lambda = 0.01 (using Ta
```

```
The value of the integration from Trapezoidal Rule and lambda = 0.01 (using Taylor Form) is 0.199889.
```

1(ii). Simpson 1/3 Rule

```
if (rem(n,2) == 1)
    fprintf("\n please Enter invalid n for Simpson 1/3 Rule!!!");
    n = input('\n Enter n as even number');
end
s0 = 0; s02 = 0;
se = 0; se2 = 0;

for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f_taylor(x(k));
    y2(k) = f_original(x(k));
    if(rem(k,2) == 1)
        s0 = s0 + y(k); %sum of odd terms
        s02 = s02 + y2(k);
    else
        se = se + y(k); %sum of even terms
        se2 = se2 + y2(k);
    end
end

answer_from_taylor = (h/3)* (f_taylor(a)+f_taylor(b)+4*s0+2*se);
answer_from_original = (h/3)* (f_original(a)+f_original(b)+4*s02+2*se2);

fprintf("The value of the integration from Simpson 1/3 Rule and lambda = 0.01 (using Or
```

```
The value of the integration from Simpson 1/3 Rule and lambda = 0.01 (using Original Expression) is NaN.
```

```
fprintf("The value of the integration from Simpson 1/3 Rule and lambda = 0.01 (using Ta
```

```
The value of the integration from Simpson 1/3 Rule and lambda = 0.01 (using Taylor Form) is 0.199889.
```

1(iii). Simpson 3/8 Rule

```
if (rem(n,3) ~= 0)
    fprintf("\n Enter invalid n for Simpson 3/8 Rule!!!");
    n = input('\n Enter n as even multiple of 3: ');
    %n = 12;
end
s0 = 0; s02 = 0;
sm3 = 0; sm32 = 0;

for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f_taylor(x(k));
    y2(k) = f_original(x(k));
    if(rem(k,3) == 0)
        sm3 = sm3 + y(k); %sum of terms that are multiple of 3
        sm32 = sm32 + y2(k);
    else
        s0 = s0 + y(k); %sum of other terms
```

```
        s02 = s02 + y2(k);
    end
end

answer1 = (3*h/8) * (f_original(a)+f_original(b)+3*s02+2*sm32);
answer2 = (3*h/8) * (f_taylor(a)+f_taylor(b)+3*s0+2*sm3);

fprintf("The value of the integration from Simpson 3/8 Rule and lambda = 0.01 (using Or
```

```
The value of the integration from Simpson 3/8 Rule and lambda = 0.01 (using Original Expression) is NaN.
```

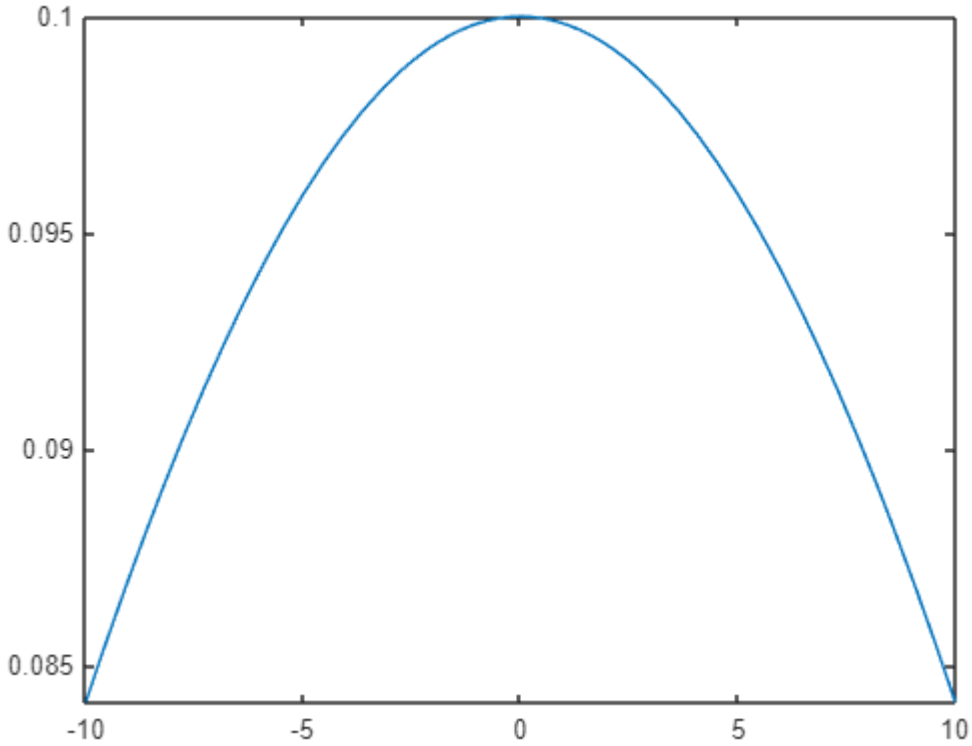
```
fprintf("The value of the integration from Simpson 3/8 Rule and lambda = 0.01 (using Ta
```

```
The value of the integration from Simpson 3/8 Rule and lambda = 0.01 (using Taylor Form) is 0.199889.
```

2. For lambda (here represented as c) = 0.1

```
clear variables
close all
clc

syms p
c = 0.1;

f_taylor = @(x) symsum(((-1)^p)*(c^(2*p+1))*x^(2*p)/factorial(2*p +1),p ,0,100);
f_original = @(x) sin(c*x)/x;

a = -10;
b = 10;

n = 300;

fplot(f_original, [-10 10]);
```

```
Warning: Function behaves unexpectedly on array inputs. To improve performance, properly vectorize
your function to return an output with the same size and shape as the input arguments.
```




```
h = (b-a)/n;
sum = 0;
sum2 = 0;
```

2(i). Trapezoidal Rule

```
for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f_taylor(x(k));
    sum = sum + y(k);
    sum2 = sum2 + f_original(x(k));
end

answer1 = h/2* (f_original(a) + f_original(b) + 2*sum2);
answer2 = h/2* (f_taylor(a) + f_taylor(b) + 2*sum);

fprintf("The value of the integration from Trapezoidal Rule and lambda = 0.1 (using Ori
```

```
The value of the integration from Trapezoidal Rule and lambda = 0.1 (using Original Expression) is NaN.
```

```
fprintf("The value of the integration from Trapezoidal Rule and lambda = 0.1 (using Tay
```

```
The value of the integration from Trapezoidal Rule and lambda = 0.1 (using Taylor Form) is 1.892164.
```

2(ii). Simpson 1/3 Rule

```
if (rem(n,2) == 1)
    fprintf("\n please Enter invalid n for Simpson 1/3 Rule!!!");
    n = input('\n Enter n as even number');
end
s0 = 0; s02 = 0;
se = 0; se2 = 0;

for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f_taylor(x(k));
    y2(k) = f_original(x(k));
    if(rem(k,2) == 1)
        s0 = s0 + y(k); %sum of odd terms
        s02 = s02 + y2(k);
    else
        se = se + y(k); %sum of even terms
        se2 = se2 + y2(k);
    end
end

answer_from_taylor = (h/3)* (f_taylor(a)+f_taylor(b)+4*s0+2*se);
answer_from_original = (h/3)* (f_original(a)+f_original(b)+4*s02+2*se2);

fprintf("The value of the integration from Simpson 1/3 Rule and lambda = 0.1 (using Ori
```

```
The value of the integration from Simpson 1/3 Rule and lambda = 0.1 (using Original Expression) is NaN.
```

```
fprintf("The value of the integration from Simpson 1/3 Rule and lambda = 0.1 (using Tay
```

```
The value of the integration from Simpson 1/3 Rule and lambda = 0.1 (using Taylor Form) is 1.892166.
```

2(iii). Simpson 3/8 Rule

```
if (rem(n,3) ~= 0)
    fprintf("\n Enter invalid n for Simpson 3/8 Rule!!!");
    n = input('\n Enter n as even multiple of 3: ');
    %n = 12;
end
s0 = 0; s02 = 0;
sm3 = 0; sm32 = 0;

for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f_taylor(x(k));
    y2(k) = f_original(x(k));
    if(rem(k,3) == 0)
        sm3 = sm3 + y(k); %sum of terms that are multiple of 3
        sm32 = sm32 + y2(k);
    else
        s0 = s0 + y(k); %sum of other terms
```

```
        s02 = s02 + y2(k);
    end
end

answer1 = (3*h/8) * (f_original(a)+f_original(b)+3*s02+2*sm32);
answer2 = (3*h/8) * (f_taylor(a)+f_taylor(b)+3*s0+2*sm3);

fprintf("The value of the integration from Simpson 3/8 Rule and lambda = 0.1 (using Ori
```

```
The value of the integration from Simpson 3/8 Rule and lambda = 0.1 (using Original Expression) is NaN.
```

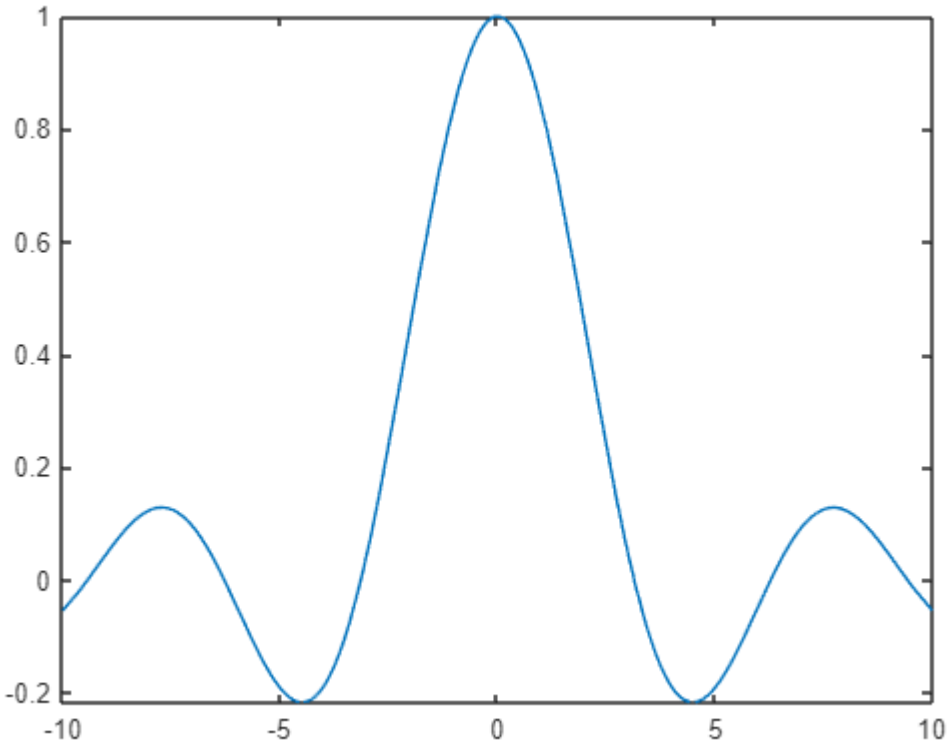
```
fprintf("The value of the integration from Simpson 3/8 Rule and lambda = 0.1 (using Tay
```

```
The value of the integration from Simpson 3/8 Rule and lambda = 0.1 (using Taylor Form) is 1.892166.
```

3. For lambda (here represented as c) = 1

```
clear variables
close all
clc

syms p
c = 1;

f_taylor = @(x) symsum(((-1)^p)*(c^(2*p+1))*x^(2*p)/factorial(2*p +1),p ,0,100);
f_original = @(x) sin(c*x)/x;

a = -10;
b = 10;

n = 300;

fplot(f_original, [-10 10]);
```

```
Warning: Function behaves unexpectedly on array inputs. To improve performance, properly vectorize
your function to return an output with the same size and shape as the input arguments.
```




```
h = (b-a)/n;
sum = 0;
sum2 = 0;
```

3(i). Trapezoidal Rule

```
for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f_taylor(x(k));
    sum = sum + y(k);
    sum2 = sum2 + f_original(x(k));
end

answer1 = h/2* (f_original(a) + f_original(b) + 2*sum2);
answer2 = h/2* (f_taylor(a) + f_taylor(b) + 2*sum);

fprintf("The value of the integration from Trapezoidal Rule and lambda = 1 (using Origi
```

```
The value of the integration from Trapezoidal Rule and lambda = 1 (using Original Expression) is NaN.
```

```
fprintf("The value of the integration from Trapezoidal Rule and lambda = 1 (using Taylo
```

```
The value of the integration from Trapezoidal Rule and lambda = 1 (using Taylor Form) is 3.316637.
```

3(ii). Simpson 1/3 Rule

```
if (rem(n,2) == 1)
    fprintf("\n please Enter invalid n for Simpson 1/3 Rule!!!");
    n = input('\n Enter n as even number');
end
s0 = 0; s02 = 0;
se = 0; se2 = 0;

for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f_taylor(x(k));
    y2(k) = f_original(x(k));
    if(rem(k,2) == 1)
        s0 = s0 + y(k); %sum of odd terms
        s02 = s02 + y2(k);
    else
        se = se + y(k); %sum of even terms
        se2 = se2 + y2(k);
    end
end

answer_from_taylor = (h/3)* (f_taylor(a)+f_taylor(b)+4*s0+2*se);
answer_from_original = (h/3)* (f_original(a)+f_original(b)+4*s02+2*se2);

fprintf("The value of the integration from Simpson 1/3 Rule and lambda = 1 (using Origi
```

```
The value of the integration from Simpson 1/3 Rule and lambda = 1 (using Original Expression) is NaN.
```

```
fprintf("The value of the integration from Simpson 1/3 Rule and lambda = 1 (using Taylo
```

```
The value of the integration from Simpson 1/3 Rule and lambda = 1 (using Taylor Form) is 3.316695.
```

3(iii). Simpson 3/8 Rule

```
if (rem(n,3) ~= 0)
    fprintf("\n Enter invalid n for Simpson 3/8 Rule!!!");
    n = input('\n Enter n as even multiple of 3: ');
    %n = 12;
end
s0 = 0; s02 = 0;
sm3 = 0; sm32 = 0;

for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f_taylor(x(k));
    y2(k) = f_original(x(k));
    if(rem(k,3) == 0)
        sm3 = sm3 + y(k); %sum of terms that are multiple of 3
        sm32 = sm32 + y2(k);
    else
        s0 = s0 + y(k); %sum of other terms
```

```
        s02 = s02 + y2(k);
    end
end

answer1 = (3*h/8) * (f_original(a)+f_original(b)+3*s02+2*sm32);
answer2 = (3*h/8) * (f_taylor(a)+f_taylor(b)+3*s0+2*sm3);

fprintf("The value of the integration from Simpson 3/8 Rule and lambda = 1 (using Origi
```

```
The value of the integration from Simpson 3/8 Rule and lambda = 1 (using Original Expression) is NaN.
```

```
fprintf("The value of the integration from Simpson 3/8 Rule and lambda = 1 (using Taylo
```

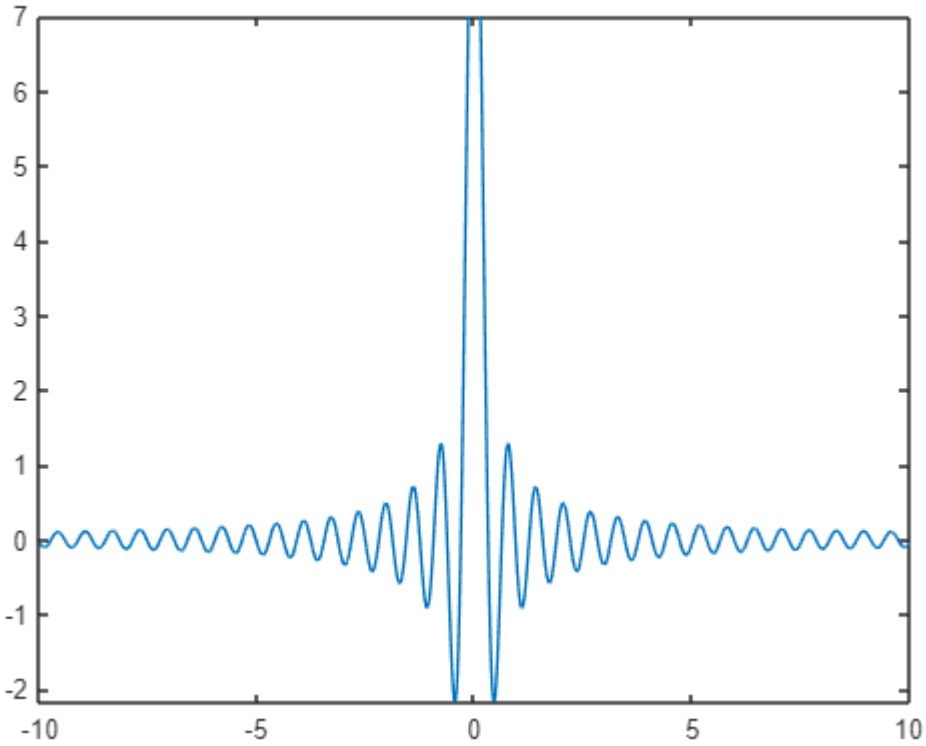
```
The value of the integration from Simpson 3/8 Rule and lambda = 1 (using Taylor Form) is 3.316695.
```

```

```

4. For lambda (here represented as c) = 10

```
clear variables
close all
clc

syms p
c = 10;

f_taylor = @(x) symsum(((-1)^p)*(c^(2*p+1))*x^(2*p)/factorial(2*p +1),p ,0,100);
f_original = @(x) sin(c*x)/x;

a = -10;
b = 10;

n = 300;

fplot(f_original, [-10 10]);
```

```
Warning: Function behaves unexpectedly on array inputs. To improve performance, properly vectorize
your function to return an output with the same size and shape as the input arguments.
```




```
h = (b-a)/n;
sum = 0;
sum2 = 0;
```

4(i). Trapezoidal Rule

```
for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f_taylor(x(k));
    sum = sum + y(k);
    sum2 = sum2 + f_original(x(k));
end

answer1 = h/2* (f_original(a) + f_original(b) + 2*sum2);
answer2 = h/2* (f_taylor(a) + f_taylor(b) + 2*sum);

fprintf("The value of the integration from Trapezoidal Rule and lambda = 10 (using Orig
```

```
The value of the integration from Trapezoidal Rule and lambda = 10 (using Original Expression) is NaN.
```

```
fprintf("The value of the integration from Trapezoidal Rule and lambda = 10 (using Tayl
```

```
The value of the integration from Trapezoidal Rule and lambda = 10 (using Taylor Form) is 1404611032481818
```

4(ii). Simpson 1/3 Rule

```
if (rem(n,2) == 1)
    fprintf("\n please Enter invalid n for Simpson 1/3 Rule!!!");
    n = input('\n Enter n as even number');
end
s0 = 0; s02 = 0;
se = 0; se2 = 0;

for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f_taylor(x(k));
    y2(k) = f_original(x(k));
    if(rem(k,2) == 1)
        s0 = s0 + y(k); %sum of odd terms
        s02 = s02 + y2(k);
    else
        se = se + y(k); %sum of even terms
        se2 = se2 + y2(k);
    end
end

answer_from_taylor = (h/3)* (f_taylor(a)+f_taylor(b)+4*s0+2*se);
answer_from_original = (h/3)* (f_original(a)+f_original(b)+4*s02+2*se2);

fprintf("The value of the integration from Simpson 1/3 Rule and lambda = 10 (using Orig
```

```
The value of the integration from Simpson 1/3 Rule and lambda = 10 (using Original Expression) is NaN.
```

```
fprintf("The value of the integration from Simpson 1/3 Rule and lambda = 10 (using Tayl
```

```
The value of the integration from Simpson 1/3 Rule and lambda = 10 (using Taylor Form) is 1242727566557987
```

4(iii). Simpson 3/8 Rule

```
if (rem(n,3) ~= 0)
    fprintf("\n Enter invalid n for Simpson 3/8 Rule!!!");
    n = input('\n Enter n as even multiple of 3: ');
    %n = 12;
end
s0 = 0; s02 = 0;
sm3 = 0; sm32 = 0;

for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f_taylor(x(k));
    y2(k) = f_original(x(k));
    if(rem(k,3) == 0)
        sm3 = sm3 + y(k); %sum of terms that are multiple of 3
        sm32 = sm32 + y2(k);
    else
        s0 = s0 + y(k); %sum of other terms
```

```
        s02 = s02 + y2(k);
    end
end

answer1 = (3*h/8) * (f_original(a)+f_original(b)+3*s02+2*sm32);
answer2 = (3*h/8) * (f_taylor(a)+f_taylor(b)+3*s0+2*sm3);

fprintf("The value of the integration from Simpson 3/8 Rule and lambda = 10 (using Orig
```

```
The value of the integration from Simpson 3/8 Rule and lambda = 10 (using Original Expression) is NaN.
```

```
fprintf("The value of the integration from Simpson 3/8 Rule and lambda = 10 (using Tayl
```

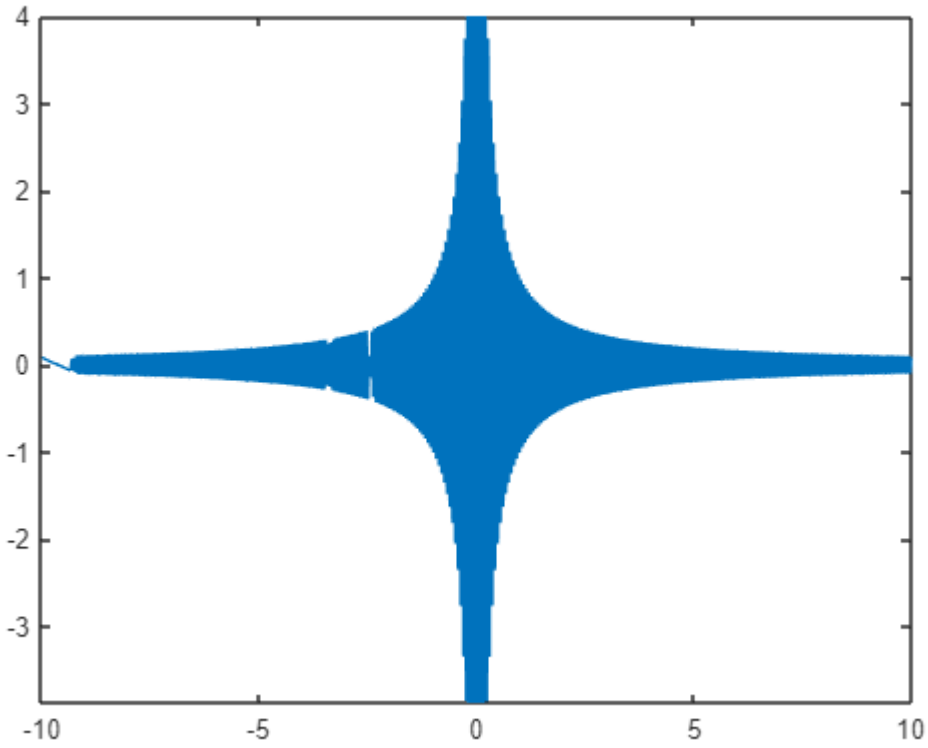
```
The value of the integration from Simpson 3/8 Rule and lambda = 10 (using Taylor Form) is 1259305340316092
```

```
```

5. For lambda (here represented as c) = 100

```
clear variables
close all
clc

syms p
c = 100;

f_taylor = @(x) symsum(((-1)^p)*(c^(2*p+1))*x^(2*p)/factorial(2*p +1),p ,0,100);
f_original = @(x) sin(c*x)/x;

a = -10;
b = 10;

n = 300;

fplot(f_original, [-10 10]);
```

```
Warning: Function behaves unexpectedly on array inputs. To improve performance, properly vectorize
your function to return an output with the same size and shape as the input arguments.
```




```
h = (b-a)/n;
sum = 0;
sum2 = 0;
```

5(i). Trapezoidal Rule

```
for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f_taylor(x(k));
    sum = sum + y(k);
    sum2 = sum2 + f_original(x(k));
end

answer1 = h/2* (f_original(a) + f_original(b) + 2*sum2);
answer2 = h/2* (f_taylor(a) + f_taylor(b) + 2*sum);

fprintf("The value of the integration from Trapezoidal Rule and lambda = 100 (using Ori
```

```
The value of the integration from Trapezoidal Rule and lambda = 100 (using Original Expression) is NaN.
```

```
fprintf("The value of the integration from Trapezoidal Rule and lambda = 100 (using Tay
```

```
The value of the integration from Trapezoidal Rule and lambda = 100 (using Taylor Form) is 690540211757540
```

5(ii). Simpson 1/3 Rule

```
if (rem(n,2) == 1)
    fprintf("\n please Enter invalid n for Simpson 1/3 Rule!!!");
    n = input('\n Enter n as even number');
end
s0 = 0; s02 = 0;
se = 0; se2 = 0;

for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f_taylor(x(k));
    y2(k) = f_original(x(k));
    if(rem(k,2) == 1)
        s0 = s0 + y(k); %sum of odd terms
        s02 = s02 + y2(k);
    else
        se = se + y(k); %sum of even terms
        se2 = se2 + y2(k);
    end
end

answer_from_taylor = (h/3)* (f_taylor(a)+f_taylor(b)+4*s0+2*se);
answer_from_original = (h/3)* (f_original(a)+f_original(b)+4*s02+2*se2);

fprintf("The value of the integration from Simpson 1/3 Rule and lambda = 100 (using Ori
```

```
The value of the integration from Simpson 1/3 Rule and lambda = 100 (using Original Expression) is NaN.
```

```
fprintf("The value of the integration from Simpson 1/3 Rule and lambda = 100 (using Tay
```

```
The value of the integration from Simpson 1/3 Rule and lambda = 100 (using Taylor Form) is 611863603454217
```

5(iii). Simpson 3/8 Rule

```
if (rem(n,3) ~= 0)
    fprintf("\n Enter invalid n for Simpson 3/8 Rule!!!");
    n = input('\n Enter n as even multiple of 3: ');
    %n = 12;
end
s0 = 0; s02 = 0;
sm3 = 0; sm32 = 0;

for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f_taylor(x(k));
    y2(k) = f_original(x(k));
    if(rem(k,3) == 0)
        sm3 = sm3 + y(k); %sum of terms that are multiple of 3
        sm32 = sm32 + y2(k);
    else
        s0 = s0 + y(k); %sum of other terms
```

```
        s02 = s02 + y2(k);
    end
end

answer1 = (3*h/8) * (f_original(a)+f_original(b)+3*s02+2*sm32);
answer2 = (3*h/8) * (f_taylor(a)+f_taylor(b)+3*s0+2*sm3);

fprintf("The value of the integration from Simpson 3/8 Rule and lambda = 100 (using Ori
```

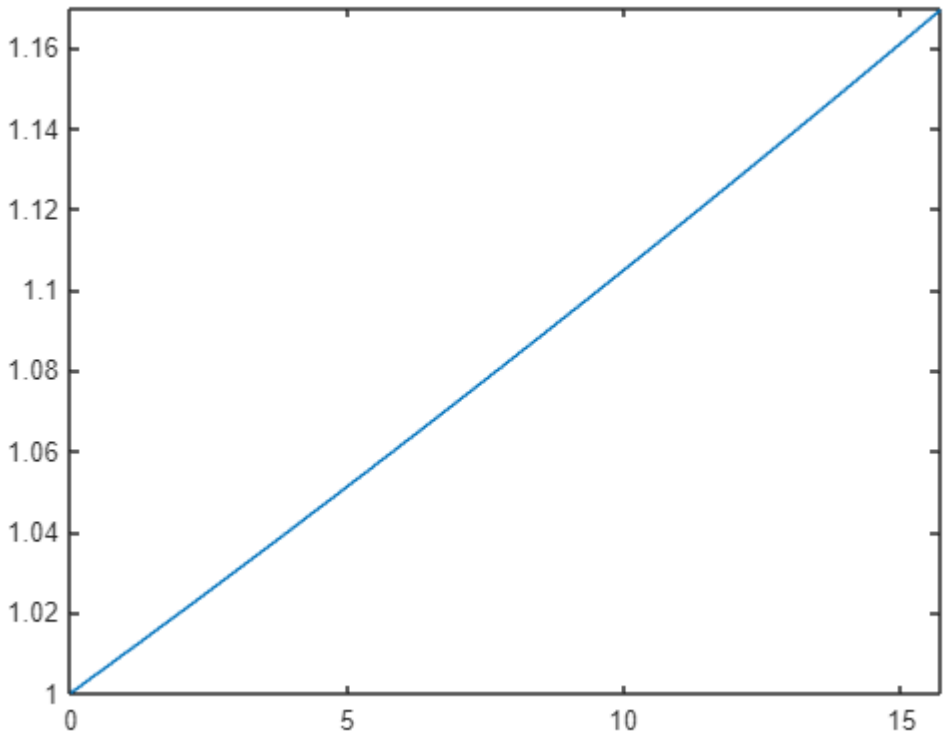
```
The value of the integration from Simpson 3/8 Rule and lambda = 100 (using Original Expression) is NaN.
```

```
fprintf("The value of the integration from Simpson 3/8 Rule and lambda = 100 (using Tay
```

```
The value of the integration from Simpson 3/8 Rule and lambda = 100 (using Taylor Form) is 619836738070499
```

Question 2:

```
clear variables
close all
clc

c = 0.01;

f = @(x) exp(sin(c*x));

a = 0;
b = 5*pi;
n = 300;

fplot(f, [a b]);
```




```
h = (b-a)/n;
sum = 0;
```

1(i). Trapezoidal Rule

```
for k = 1:1:n-1
    x(k) = a + k*h;
    sum = sum + f(x(k));
end
answer = (h/2)* (f(a) + f(b) + 2*sum);

fprintf("The value of the integration from Trapezoidal Rule and lambda = 0.01 is %f.\n'
```

```
The value of the integration from Trapezoidal Rule and lambda = 0.01 is 17.006005.
```

1(ii). Simpson 1/3 Rule

```
if (rem(n,2) == 1)
    fprintf("\n please Enter invalid n for Simpson 1/3 Rule!!!");
    n = input('\n Enter n as even number');
end
s0 = 0;
se = 0;

for k = 1:1:n-1
    x(k) = a +k*h;
```

```
    y(k) = f(x(k));
    if(rem(k,2) == 1)
        s0 = s0 + y(k); %sum of odd terms
    else
        se = se + y(k); %sum of even terms
    end
end

answer = (h/3)* (f(a)+f(b)+4*s0+2*se);
fprintf("The value of the integration from Simpson 1/3 Rule and lambda = 0.01 is %f.\n'
```

```
The value of the integration from Simpson 1/3 Rule and lambda = 0.01 is 17.006004.
```

1(iii). Simpson 3/8 Rule

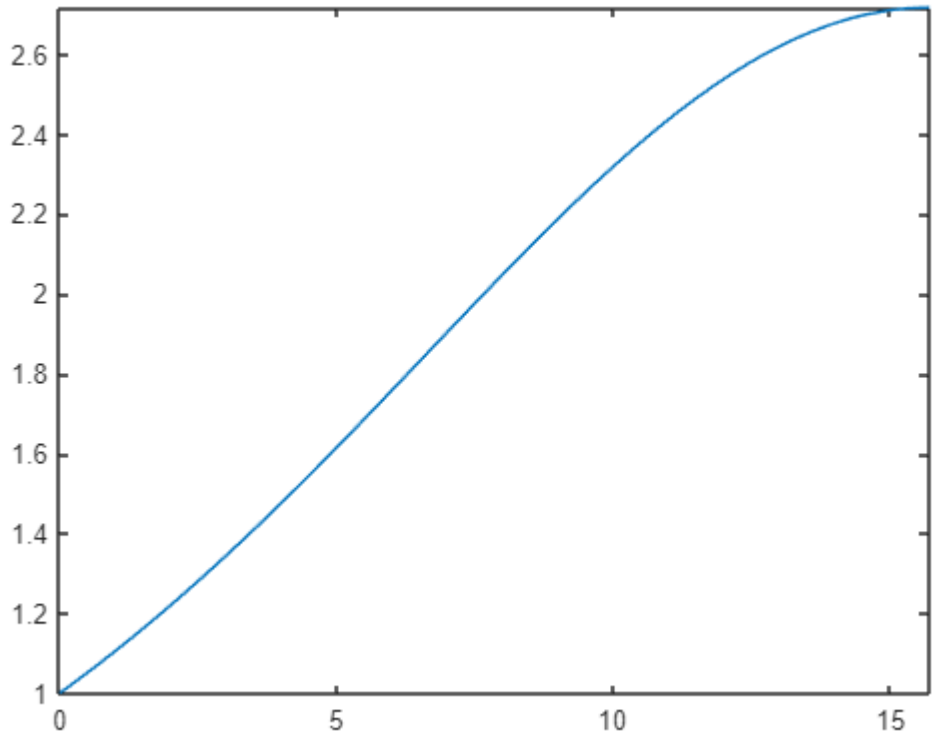
```
if (rem(n,3) ~= 0)
    fprintf("\n Enter invalid n for Simpson 3/8 Rule!!!");
    n = input('\n Enter n as even multiple of 3: ');
    %n = 12;
end
s0 = 0;
sm3 = 0;

for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f(x(k));

    if(rem(k,3) == 0)
        sm3 = sm3 + y(k); %sum of terms that are multiple of 3
    else
        s0 = s0 + y(k); %sum of other terms
    end
end

answer = (3*h/8) * (f(a)+f(b)+3*s0+2*sm3);

fprintf("The value of the integration from Simpson 3/8 Rule and lambda = 0.01 is %f.\n'
```

```
The value of the integration from Simpson 3/8 Rule and lambda = 0.01 is 17.006004.
```

2. For lambda (here represented as c) = 0.1

```


clear variables
close all
clc

c = 0.1;

f = @(x) exp(sin(c*x));
```

```
a = 0;
b = 5*pi;
n = 300;

fplot(f, [a b]);
```




```
h = (b-a)/n;
sum = 0;
```

2(i). Trapezoidal Rule

```
for k = 1:1:n-1
    x(k) = a + k*h;
    sum = sum + f(x(k));
end
answer = (h/2)* (f(a) + f(b) + 2*sum);

fprintf("The value of the integration from Trapezoidal Rule and lambda = 0.1 is %f.\n",
```

```
The value of the integration from Trapezoidal Rule and lambda = 0.1 is 31.043767.
```

2(ii). Simpson 1/3 Rule

```
if (rem(n,2) == 1)
    fprintf("\n please Enter invalid n for Simpson 1/3 Rule!!!");
    n = input('\n Enter n as even number');
```

```
end
s0 = 0;
se = 0;

for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f(x(k));
    if(rem(k,2) == 1)
        s0 = s0 + y(k); %sum of odd terms
    else
        se = se + y(k); %sum of even terms
    end
end

answer = (h/3)* (f(a)+f(b)+4*s0+2*se);
fprintf("The value of the integration from Simpson 1/3 Rule and lambda = 0.1 is %f.\n",
```

```
The value of the integration from Simpson 1/3 Rule and lambda = 0.1 is 31.043790.
```

2(iii). Simpson 3/8 Rule

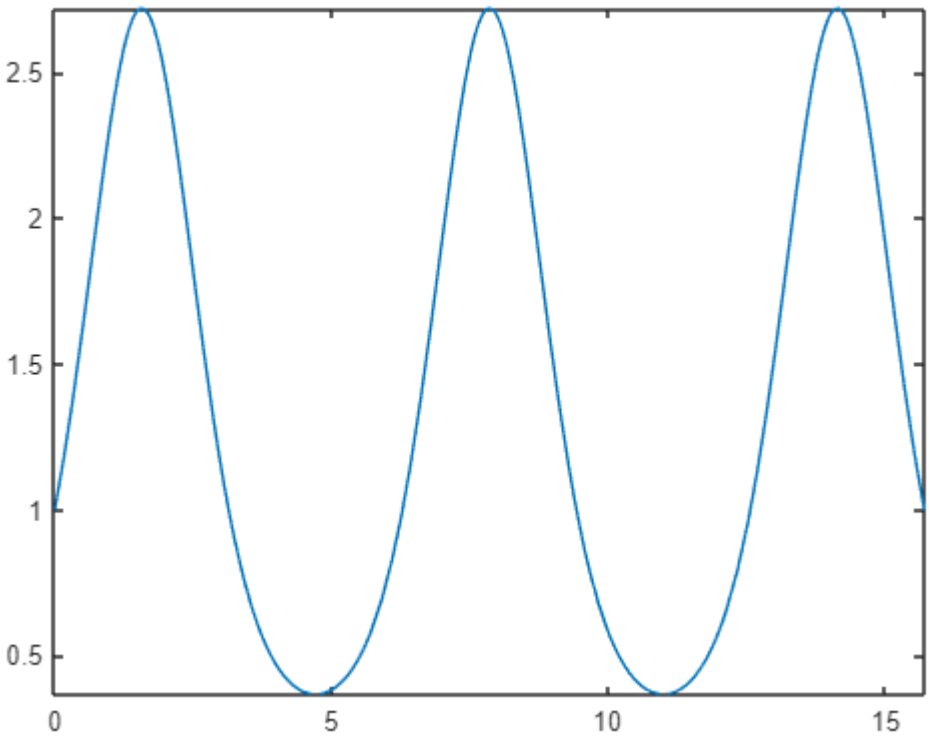
```
if (rem(n,3) ~= 0)
    fprintf("\n Enter invalid n for Simpson 3/8 Rule!!!");
    n = input('\n Enter n as even multiple of 3: ');
    %n = 12;
end
s0 = 0;
sm3 = 0;

for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f(x(k));

    if(rem(k,3) == 0)
        sm3 = sm3 + y(k); %sum of terms that are multiple of 3
    else
        s0 = s0 + y(k); %sum of other terms
    end
end

answer = (3*h/8) * (f(a)+f(b)+3*s0+2*sm3);

fprintf("The value of the integration from Simpson 3/8 Rule and lambda = 0.1 is %f.\n",
```

```
The value of the integration from Simpson 3/8 Rule and lambda = 0.1 is 31.043790.
```

```
```

3. For lambda (here represented as c) = 1

```
clear variables
close all
clc

c = 1;

f = @(x) exp(sin(c*x));

a = 0;
b = 5*pi;
n = 300;

fplot(f, [a b]);
```




```
h = (b-a)/n;
sum = 0;
```

3(i). Trapezoidal Rule

```
for k = 1:1:n-1
    x(k) = a + k*h;
    sum = sum + f(x(k));
end
answer = (h/2)* (f(a) + f(b) + 2*sum);

fprintf("The value of the integration from Trapezoidal Rule and lambda = 1 is %f.\n",ar
```

```
The value of the integration from Trapezoidal Rule and lambda = 1 is 22.118154.
```

3(ii). Simpson 1/3 Rule

```
if (rem(n,2) == 1)
    fprintf("\n please Enter invalid n for Simpson 1/3 Rule!!!");
    n = input('\n Enter n as even number');
end
s0 = 0;
se = 0;

for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f(x(k));
    if(rem(k,2) == 1)
        s0 = s0 + y(k); %sum of odd terms
    else
        se = se + y(k); %sum of even terms
    end
end

answer = (h/3)* (f(a)+f(b)+4*s0+2*se);
fprintf("The value of the integration from Simpson 1/3 Rule and lambda = 1 is %f.\n",an
```

```
The value of the integration from Simpson 1/3 Rule and lambda = 1 is 22.118611.
```

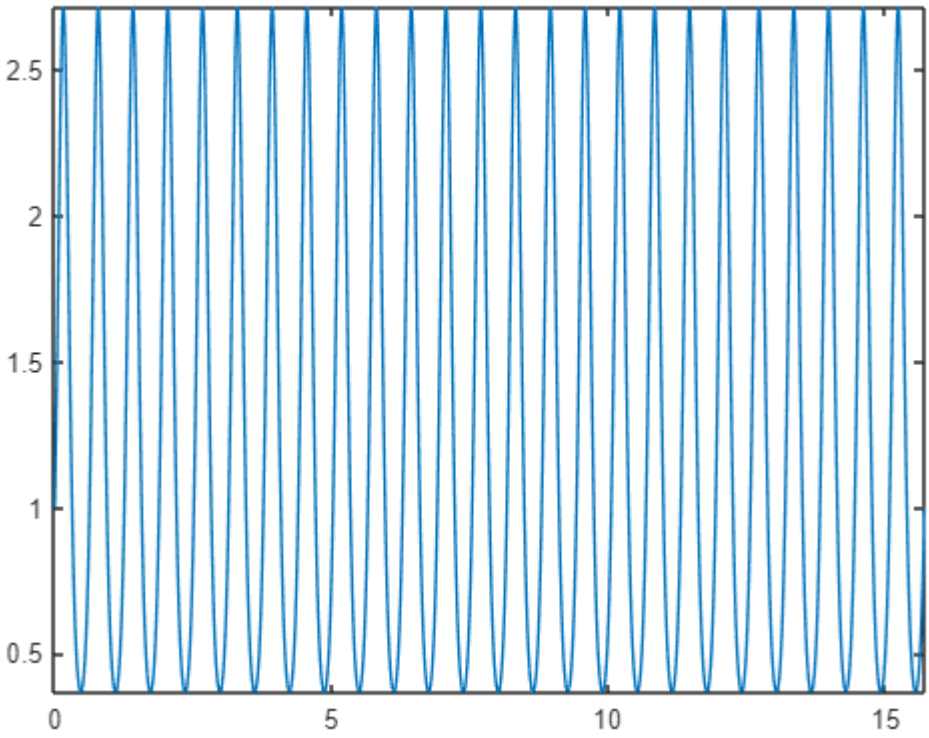
3(iii). Simpson 3/8 Rule

```
if (rem(n,3) ~= 0)
    fprintf("\n Enter invalid n for Simpson 3/8 Rule!!!");
    n = input('\n Enter n as even multiple of 3: ');
    %n = 12;
end
s0 = 0;
sm3 = 0;

for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f(x(k));

    if(rem(k,3) == 0)
        sm3 = sm3 + y(k); %sum of terms that are multiple of 3
    else
        s0 = s0 + y(k); %sum of other terms
    end
end

answer = (3*h/8) * (f(a)+f(b)+3*s0+2*sm3);

fprintf("The value of the integration from Simpson 3/8 Rule and lambda = 1 is %f.\n",an
```

```
The value of the integration from Simpson 3/8 Rule and lambda = 1 is 22.118611.
```

4. For lambda (here represented as c) = 10

```
clear variables
close all
clc

c = 10;

f = @(x) exp(sin(c*x));

a = 0;
b = 5*pi;
n = 300;

fplot(f, [a b]);
```




```
h = (b-a)/n;
sum = 0;
```

4(i). Trapezoidal Rule

```
for k = 1:1:n-1
    x(k) = a + k*h;
    sum = sum + f(x(k));
end
answer = (h/2)* (f(a) + f(b) + 2*sum);

fprintf("The value of the integration from Trapezoidal Rule and lambda = 10 is %f.\n",a
```

```
The value of the integration from Trapezoidal Rule and lambda = 10 is 19.887316.
```

4(ii). Simpson 1/3 Rule

```
if (rem(n,2) == 1)
    fprintf("\n please Enter invalid n for Simpson 1/3 Rule!!!");
    n = input('\n Enter n as even number');
end
s0 = 0;
se = 0;

for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f(x(k));
    if(rem(k,2) == 1)
        s0 = s0 + y(k); %sum of odd terms
    else
        se = se + y(k); %sum of even terms
    end
end

answer = (h/3)* (f(a)+f(b)+4*s0+2*se);
fprintf("The value of the integration from Simpson 1/3 Rule and lambda = 10 is %f.\n",a
```

```
The value of the integration from Simpson 1/3 Rule and lambda = 10 is 19.887552.
```

4(iii). Simpson 3/8 Rule

```
if (rem(n,3) ~= 0)
    fprintf("\n Enter invalid n for Simpson 3/8 Rule!!!");
    n = input('\n Enter n as even multiple of 3: ');
    %n = 12;
end
s0 = 0;
sm3 = 0;

for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f(x(k));

    if(rem(k,3) == 0)
```

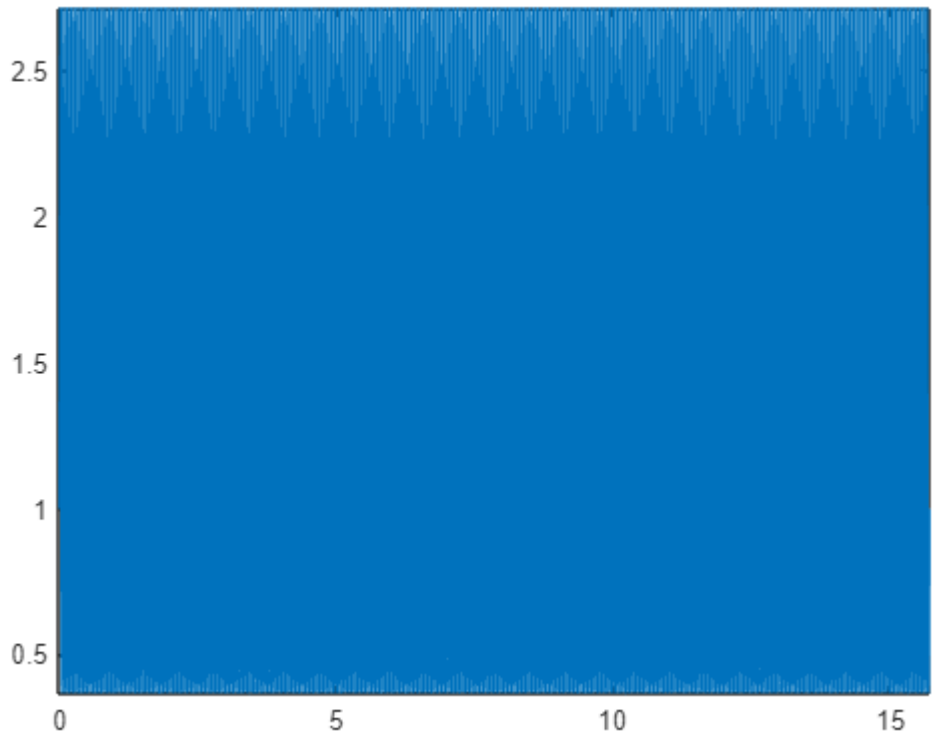
```
        sm3 = sm3 + y(k); %sum of terms that are multiple of 3
    else
        s0 = s0 + y(k); %sum of other terms
    end
end

answer = (3*h/8) * (f(a)+f(b)+3*s0+2*sm3);

fprintf("The value of the integration from Simpson 3/8 Rule and lambda = 10 is %f.\n",a
```

```
The value of the integration from Simpson 3/8 Rule and lambda = 10 is 19.876567.
```

```

```

5. For lambda (here represented as c) = 100

```
clear variables
close all
clc

c = 100;

f = @(x) exp(sin(c*x));

a = 0;
b = 5*pi;
n = 300;

fplot(f, [a b]);
```




```
h = (b-a)/n;
sum = 0;
```

5(i). Trapezoidal Rule

```
for k = 1:1:n-1
    x(k) = a + k*h;
    sum = sum + f(x(k));
end
answer = (h/2)* (f(a) + f(b) + 2*sum);

fprintf("The value of the integration from Trapezoidal Rule and lambda = 100 is %f.\n",
```

```
The value of the integration from Trapezoidal Rule and lambda = 100 is 19.886610.
```

5(ii). Simpson 1/3 Rule

```
if (rem(n,2) == 1)
    fprintf("\n please Enter invalid n for Simpson 1/3 Rule!!!");
    n = input('\n Enter n as even number');
end
s0 = 0;
se = 0;

for k = 1:1:n-1
    x(k) = a +k*h;
```

```
    y(k) = f(x(k));
    if(rem(k,2) == 1)
        s0 = s0 + y(k); %sum of odd terms
    else
        se = se + y(k); %sum of even terms
    end
end

answer = (h/3)* (f(a)+f(b)+4*s0+2*se);
fprintf("The value of the integration from Simpson 1/3 Rule and lambda = 100 is %f.\n",
```

```
The value of the integration from Simpson 1/3 Rule and lambda = 100 is 19.886610.
```

5(iii). Simpson 3/8 Rule

```
if (rem(n,3) ~= 0)
    fprintf("\n Enter invalid n for Simpson 3/8 Rule!!!");
    n = input('\n Enter n as even multiple of 3: ');
    %n = 12;
end
s0 = 0;
sm3 = 0;

for k = 1:1:n-1
    x(k) = a +k*h;
    y(k) = f(x(k));

    if(rem(k,3) == 0)
        sm3 = sm3 + y(k); %sum of terms that are multiple of 3
    else
        s0 = s0 + y(k); %sum of other terms
    end
end

answer = (3*h/8) * (f(a)+f(b)+3*s0+2*sm3);

fprintf("The value of the integration from Simpson 3/8 Rule and lambda = 100 is %f.\n",
```

```
The value of the integration from Simpson 3/8 Rule and lambda = 100 is 20.408941.
```