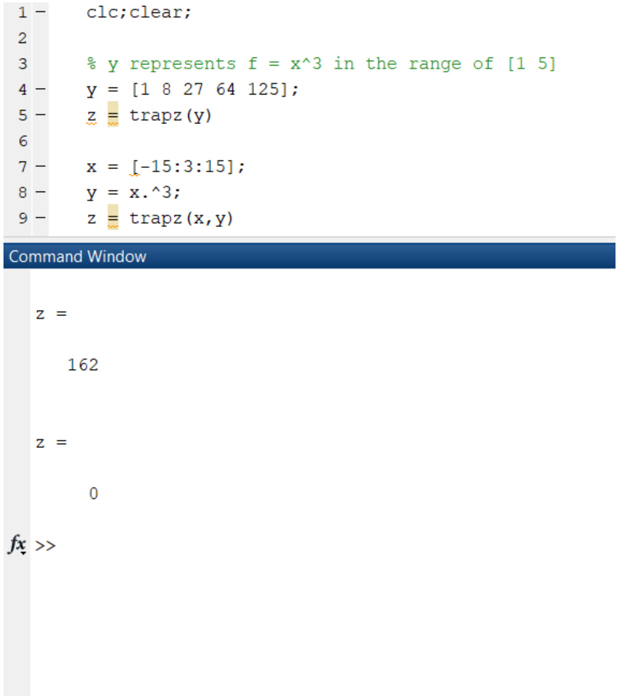
MATLAB – HW5 Upload Date: 29 Nov. 2020

Dorreen Rostami – 97243034
Zahra Hashemi – 97243072

5-1)

	Syntax	Description
trapz	trapz(Y)	trapz evaluates the approximate integral of Y via a special method called *trapezoidal method*. If Y is a vector, this function calculates the approximate integral of Y, if Y is a matrix, it integrates over each column and returns a row vector of integration values.
	trapz(X,Y)	This function integrates Y with respect to the coordinates of X. If X is a vector of coordinates, then `length(X)` must be equal to the size of the first dimension of Y whose size does not equal 1.

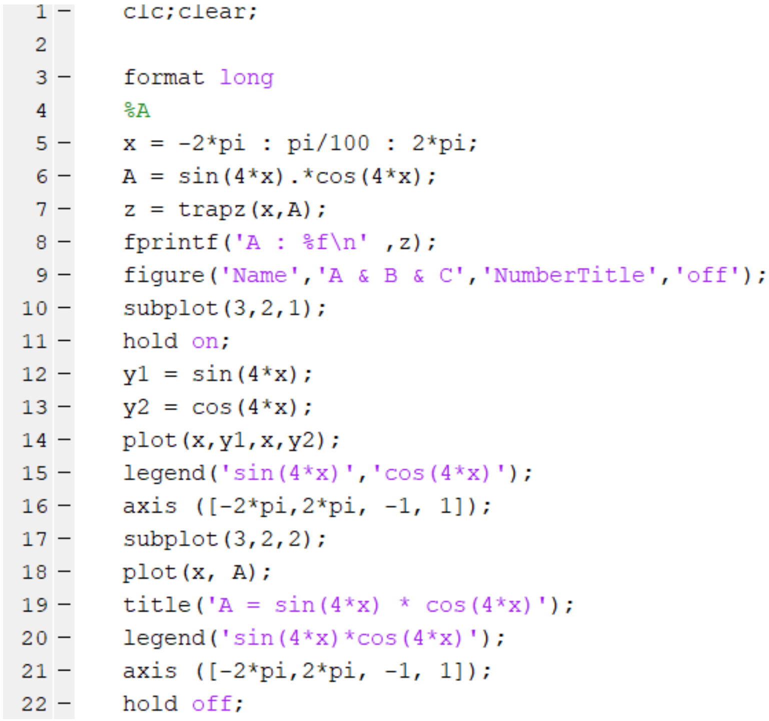


```
clc;clear;

% y represents f = x^3 in the range of [1 5]
y = [1 8 27 64 125];
z = trapz(y)

x = [-15:3:15];
y = x.^3;
z = trapz(x,y)
```

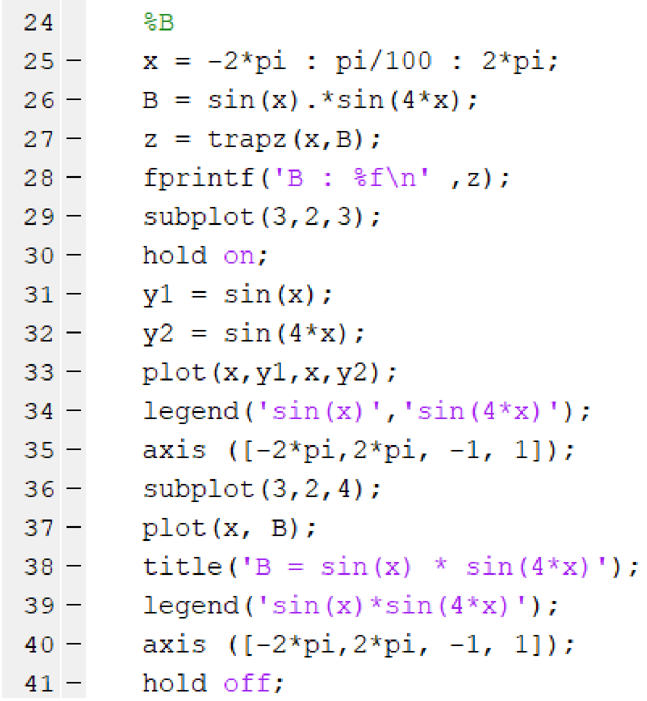
```
z =

   162


z =

     0

fx >>
```

In the second example, because x^3 is a symmetric function, with respect to the y axis, and the range is also symmetric, the value would be 0.

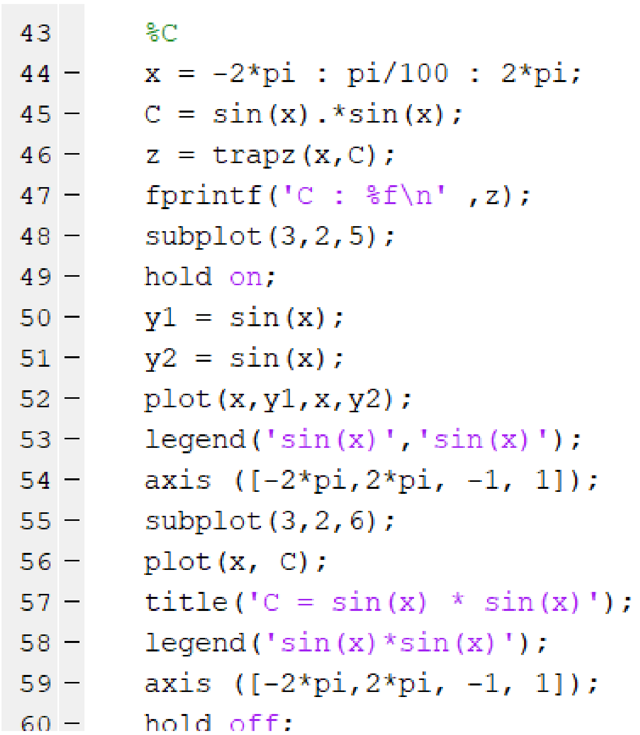
5-2)



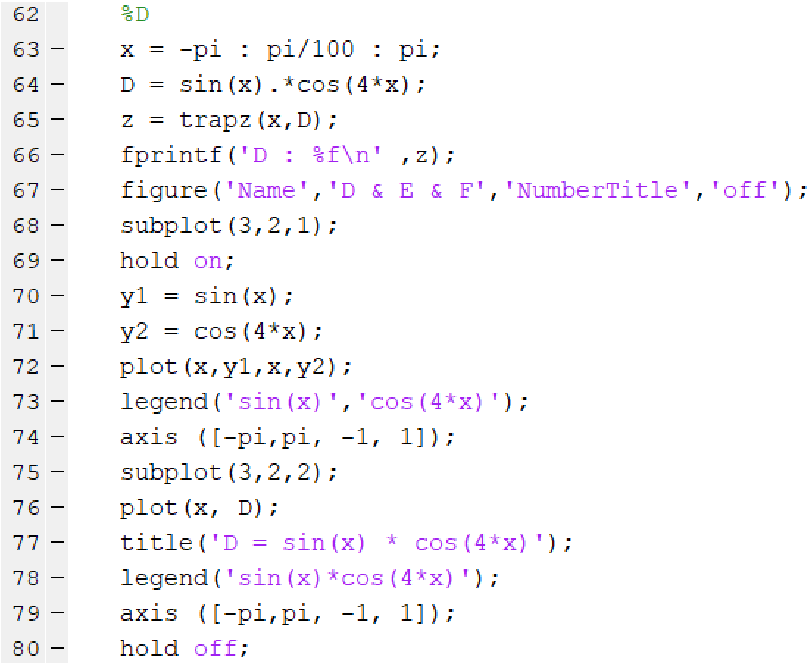
```
clc;clear;

format long
%A
x = -2*pi : pi/100 : 2*pi;
A = sin(4*x).*cos(4*x);
z = trapz(x,A);
fprintf('A : %f\n' ,z);
figure('Name','A & B & C','NumberTitle','off');
subplot(3,2,1);
hold on;
y1 = sin(4*x);
y2 = cos(4*x);
plot(x,y1,x,y2);
legend('sin(4*x)','cos(4*x)');
axis ([-2*pi,2*pi, -1, 1]);
subplot(3,2,2);
plot(x, A);
title('A = sin(4*x)  * cos(4*x)');
legend('sin(4*x)*cos(4*x)');
axis ([-2*pi,2*pi, -1, 1]);
hold off;
```



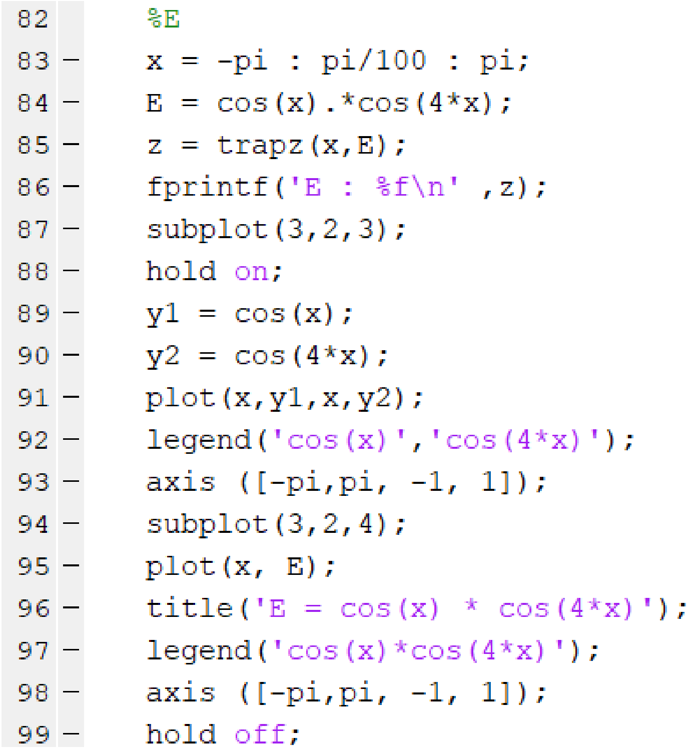
```
%B
x = -2*pi : pi/100 : 2*pi;
B = sin(x).*sin(4*x);
z = trapz(x,B);
fprintf('B : %f\n' ,z);
subplot(3,2,3);
hold on;
y1 = sin(x);
y2 = sin(4*x);
plot(x,y1,x,y2);
legend('sin(x)','sin(4*x)');
axis ([-2*pi,2*pi, -1, 1]);
subplot(3,2,4);
plot(x, B);
title('B = sin(x)  * sin(4*x)');
legend('sin(x)*sin(4*x)');
axis ([-2*pi,2*pi, -1, 1]);
hold off;
```



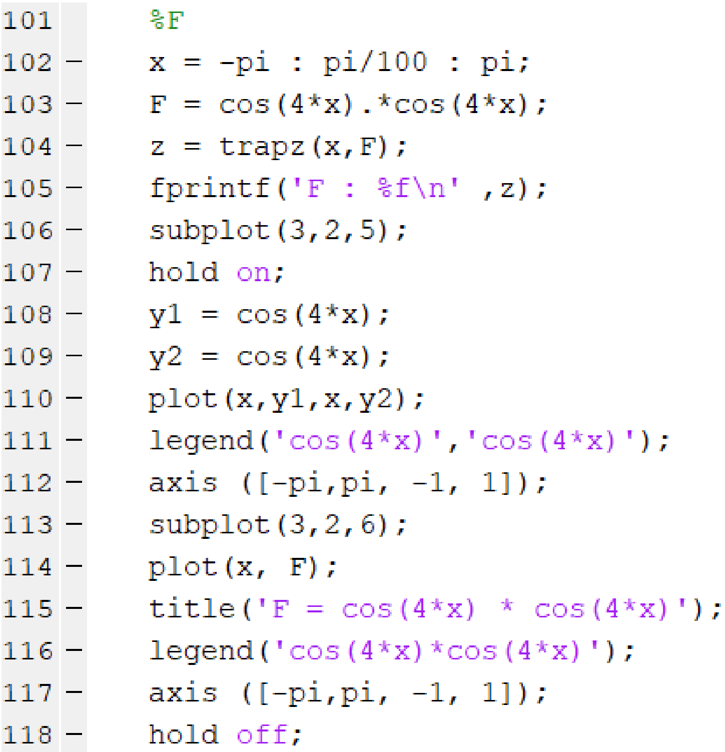
```
%C
x = -2*pi : pi/100 : 2*pi;
C = sin(x).*sin(x);
z = trapz(x,C);
fprintf('C : %f\n' ,z);
subplot(3,2,5);
hold on;
y1 = sin(x);
y2 = sin(x);
plot(x,y1,x,y2);
legend('sin(x)','sin(x)');
axis ([-2*pi,2*pi, -1, 1]);
subplot(3,2,6);
plot(x, C);
title('C = sin(x)  * sin(x)');
legend('sin(x)*sin(x)');
axis ([-2*pi,2*pi, -1, 1]);
hold off;
```



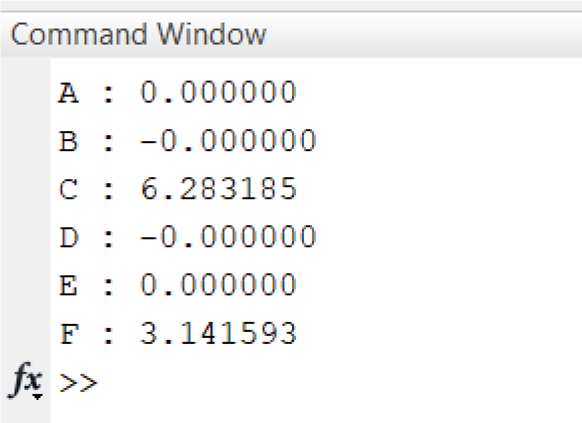
```
%D
x = -pi : pi/100 : pi;
D = sin(x).*cos(4*x);
z = trapz(x,D);
fprintf('D : %f\n' ,z);
figure('Name','D & E & F','NumberTitle','off');
subplot(3,2,1);
hold on;
y1 = sin(x);
y2 = cos(4*x);
plot(x,y1,x,y2);
legend('sin(x)','cos(4*x)');
axis ([-pi,pi, -1, 1]);
subplot(3,2,2);
plot(x, D);
title('D = sin(x)  * cos(4*x)');
legend('sin(x)*cos(4*x)');
axis ([-pi,pi, -1, 1]);
hold off;
```



```
%E
x = -pi : pi/100 : pi;
E = cos(x).*cos(4*x);
z = trapz(x,E);
fprintf('E : %f\n' ,z);
subplot(3,2,3);
hold on;
y1 = cos(x);
y2 = cos(4*x);
plot(x,y1,x,y2);
legend('cos(x)','cos(4*x)');
axis ([-pi,pi, -1, 1]);
subplot(3,2,4);
plot(x, E);
title('E = cos(x)  * cos(4*x)');
legend('cos(x)*cos(4*x)');
axis ([-pi,pi, -1, 1]);
hold off;
```



```
%F
x = -pi : pi/100 : pi;
F = cos(4*x).*cos(4*x);
z = trapz(x,F);
fprintf('F : %f\n' ,z);
subplot(3,2,5);
hold on;
y1 = cos(4*x);
y2 = cos(4*x);
plot(x,y1,x,y2);
legend('cos(4*x)','cos(4*x)');
axis ([-pi,pi, -1, 1]);
subplot(3,2,6);
plot(x, F);
title('F = cos(4*x)  * cos(4*x)');
legend('cos(4*x)*cos(4*x)');
axis ([-pi,pi, -1, 1]);
hold off;
```



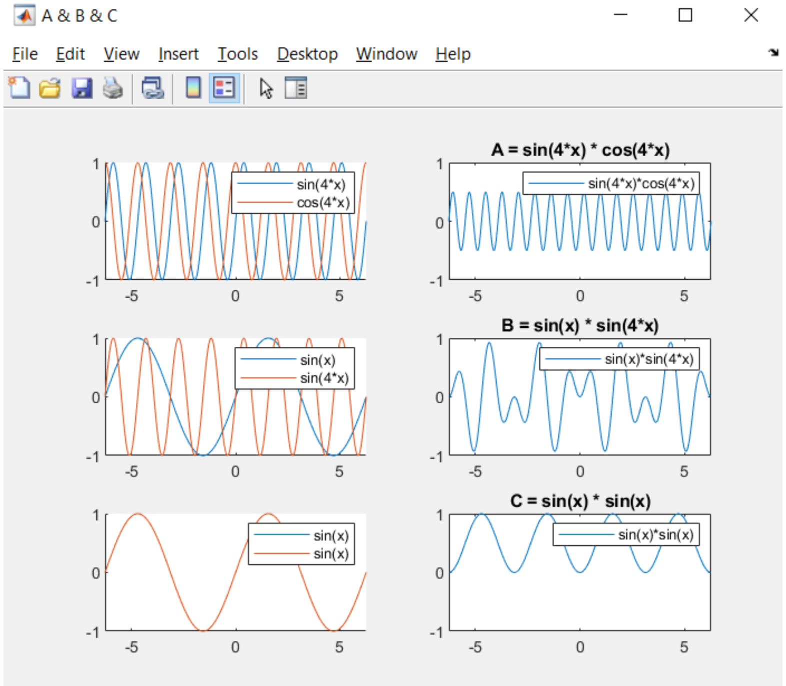
Command Window

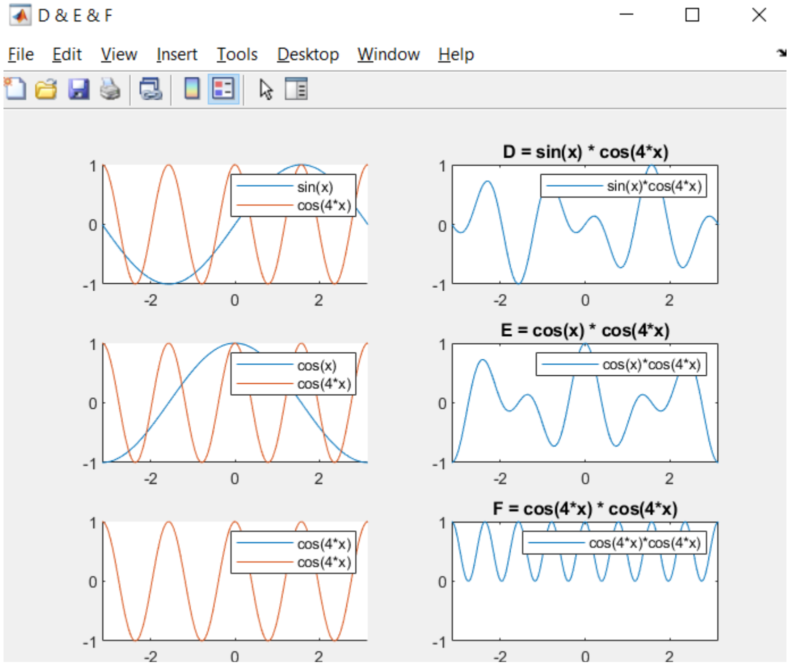
```
A : 0.000000
B : -0.000000
C : 6.283185
D : -0.000000
E : 0.000000
F : 3.141593
fx >>
```

$$\int_{-p}^{p} \sin\left(\frac{m\pi x}{p}\right)\cos\left(\frac{n\pi x}{p}\right)dx = 0, for\ all\ m\ and\ n$$ -----> A = 0 & D = 0

$$\int_{-p}^{p} \sin\left(\frac{m\pi x}{p}\right)\sin\left(\frac{n\pi x}{p}\right)dx = \begin{cases} 0,\ \text{if m} \neq \text{n} & \text{-----> B = 0} \\ \text{P , if m =n} & \text{----> C = } 2\pi \end{cases}$$

$$\int_{-p}^{p} \cos\left(\frac{m\pi x}{p}\right)\cos\left(\frac{n\pi x}{p}\right)dx = \begin{cases} 0,\ \text{if m} \neq \text{n} & \text{----> E = 0} \\ \text{P , if m =n} & \text{----> F = } \pi \end{cases}$$



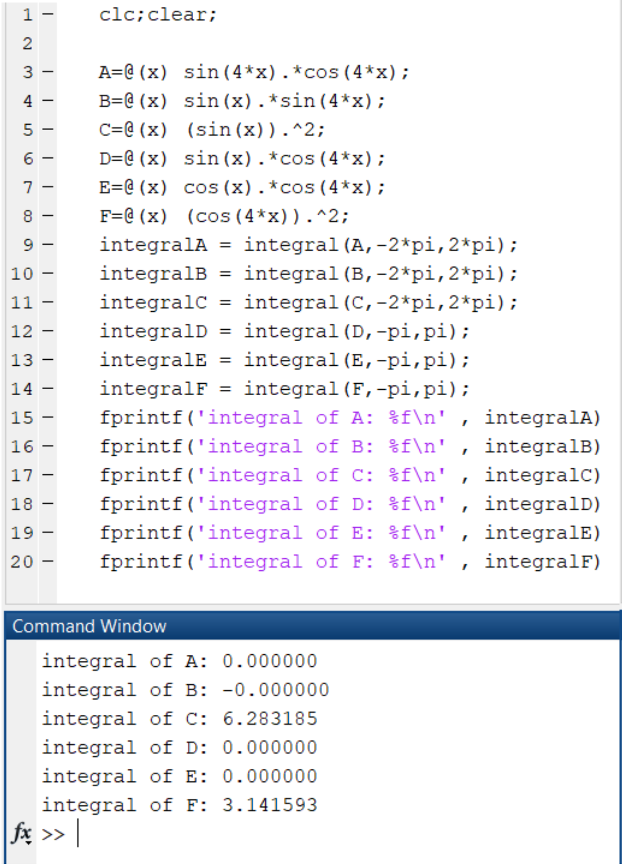


5-3)

	Syntax	Description
integral	integral(fun,xmin,xmax)	This will return the integration of function fun from xmin to xmax using global adaptive quadrature and default error tolerances.
	integral(fun,xmin,xmax,Name,Value)	This function works like the previous one, except it has 2 more arguments in order to specify additional options.

	Syntax	Description
quad	quad(fun,a,b)	quad function is very similar to the integral function. It is used to find the area under the graph of function fun in the range of a and b.
	quad(fun,a,b,tol)	In this syntax, tol represents an absolute error tolerance and it will use this instead of default error 1.0e-6.
	[q, fcnt] = quad(…)	This function returns the number of function evaluations.

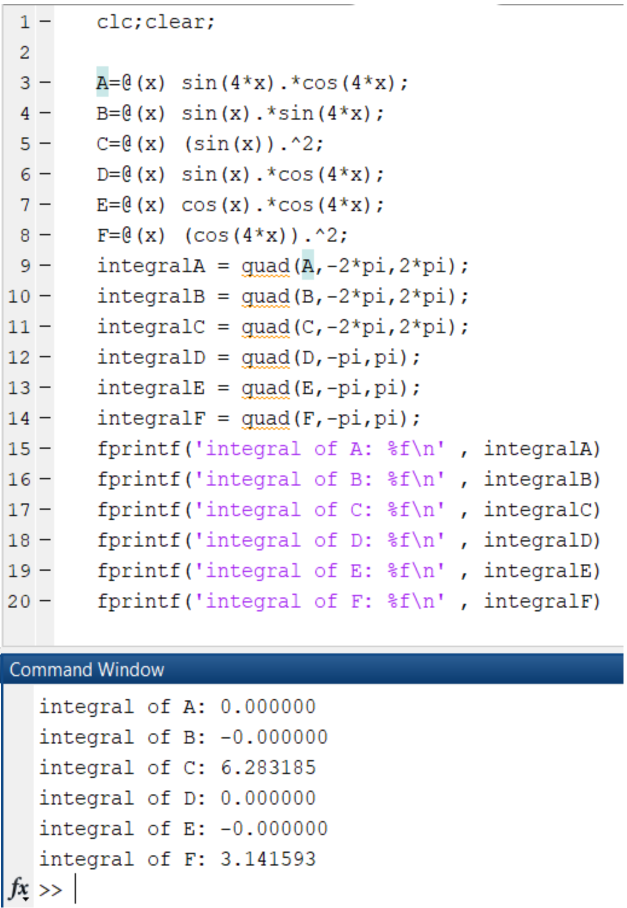
Integral with integral function



```
clc;clear;

A=@(x) sin(4*x).*cos(4*x);
B=@(x) sin(x).*sin(4*x);
C=@(x) (sin(x)).^2;
D=@(x) sin(x).*cos(4*x);
E=@(x) cos(x).*cos(4*x);
F=@(x) (cos(4*x)).^2;
integralA = integral(A,-2*pi,2*pi);
integralB = integral(B,-2*pi,2*pi);
integralC = integral(C,-2*pi,2*pi);
integralD = integral(D,-pi,pi);
integralE = integral(E,-pi,pi);
integralF = integral(F,-pi,pi);
fprintf('integral of A: %f\n' , integralA)
fprintf('integral of B: %f\n' , integralB)
fprintf('integral of C: %f\n' , integralC)
fprintf('integral of D: %f\n' , integralD)
fprintf('integral of E: %f\n' , integralE)
fprintf('integral of F: %f\n' , integralF)
```

```
integral of A: 0.000000
integral of B: -0.000000
integral of C: 6.283185
integral of D: 0.000000
integral of E: 0.000000
integral of F: 3.141593
fx >>
```

Integral with quad function



```
clc;clear;

A=@(x) sin(4*x).*cos(4*x);
B=@(x) sin(x).*sin(4*x);
C=@(x) (sin(x)).^2;
D=@(x) sin(x).*cos(4*x);
E=@(x) cos(x).*cos(4*x);
F=@(x) (cos(4*x)).^2;
integralA = quad(A,-2*pi,2*pi);
integralB = quad(B,-2*pi,2*pi);
integralC = quad(C,-2*pi,2*pi);
integralD = quad(D,-pi,pi);
integralE = quad(E,-pi,pi);
integralF = quad(F,-pi,pi);
fprintf('integral of A: %f\n' , integralA)
fprintf('integral of B: %f\n' , integralB)
fprintf('integral of C: %f\n' , integralC)
fprintf('integral of D: %f\n' , integralD)
fprintf('integral of E: %f\n' , integralE)
fprintf('integral of F: %f\n' , integralF)
```

```
integral of A: 0.000000
integral of B: -0.000000
integral of C: 6.283185
integral of D: 0.000000
integral of E: -0.000000
integral of F: 3.141593
fx >>
```

We use fprintf to easily view the results (like how they are equal to zero). By using fprintf to compare the outputs we'll see no difference but if we were to use display, we'll see that the result will vary only a small amount. For example, Integral of A will be 5.2042e-16 by using integral and 6.9389e-17 by using quad, but basically that amount is so little that we'll still stay it's zero.

5-4-1)

```
clc;clear;

%question's inputs
N = 101; k = 3;
x = linspace(-8,8);

%an = 0 & a0 = 0 & bn = (2*k/(n*pi))(1-cos(n*pi))
%Fourier Serie
an = 0;
a0 = 0;
sn = 0;
bn = zeros(1,N);
n = 1:N;
for n = 1 : N
    bn(n) = ((2*k) / (n*pi)) * (1 - cos(n*pi));
    %an = 0
    sn = sn + bn(n) .* sin(n*x);
end

figure('Name' , 'Main function and fourier serie');
%limited range [-pi,pi] to show a part of periodic function f(x) with
%T = 2*pi
%if -pi<x<0 ---> -k
%else 0<x<pi ---> k
fx = 2*k*((sin(x)>0) - 0.5);
subplot(2,1,1);
plot(x, fx);
title('f(x)');
xlabel('x');
ylabel('f(x)');
axis ([-8, 8, -k-1, k+1]);

subplot(2,1,2);
plot(x, sn ,'r');
title(['SN{f(x)}    N = ', num2str(N)]);
xlabel('x');
ylabel('Sn');
axis ([-8, 8, -k-1, k+1]);
```

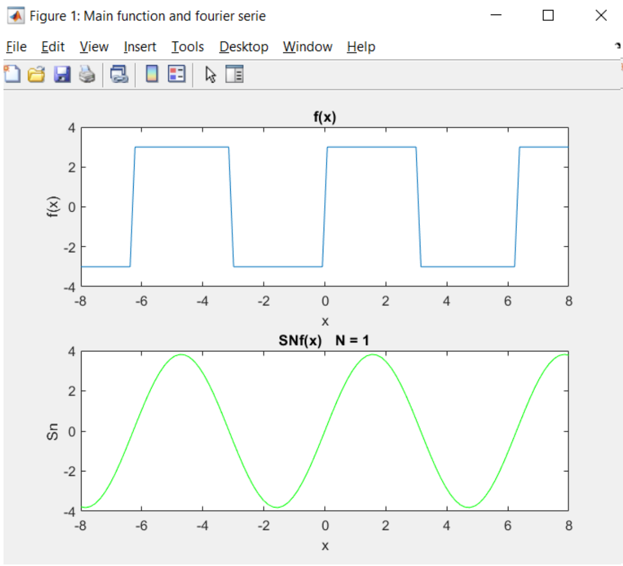


Figure 1: Main function and fourier serie
File Edit View Insert Tools Desktop Window Help
f(x)
f(x)
4
2
0
-2
-4
-8 -6 -4 -2 0 2 4 6 8
x
SNf(x)   N = 1
Sn
4
2
0
-2
-4
-8 -6 -4 -2 0 2 4 6 8
x

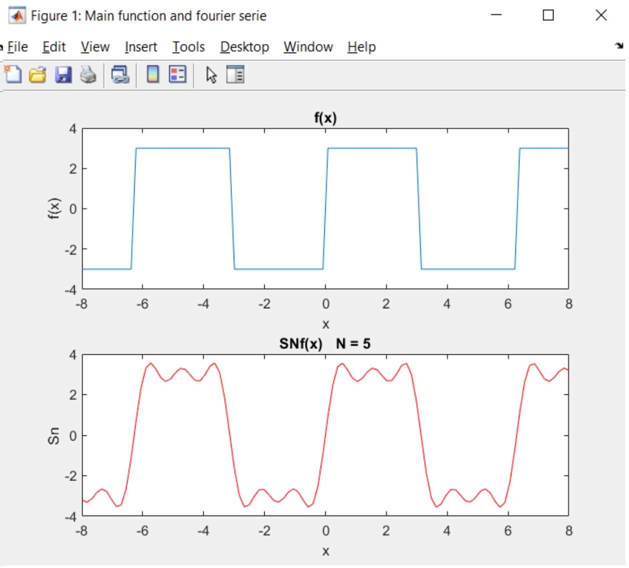


Figure 1: Main function and fourier serie
File Edit View Insert Tools Desktop Window Help
f(x)
f(x)
4
2
0
-2
-4
-8 -6 -4 -2 0 2 4 6 8
x
SNf(x)   N = 5
Sn
4
2
0
-2
-4
-8 -6 -4 -2 0 2 4 6 8
x

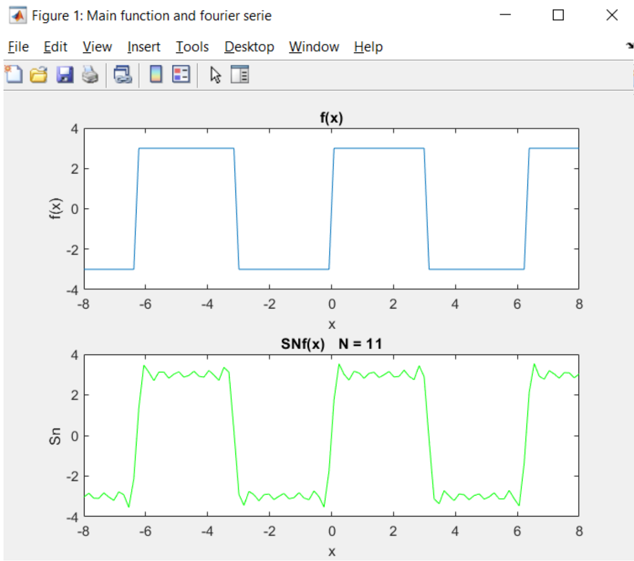


Figure 1: Main function and fourier serie
File Edit View Insert Tools Desktop Window Help
f(x)
4
2
0
-2
-4
f(x)
-8 -6 -4 -2 0 2 4 6 8
x
SNf(x)   N = 11
Sn
-8 -6 -4 -2 0 2 4 6 8
x

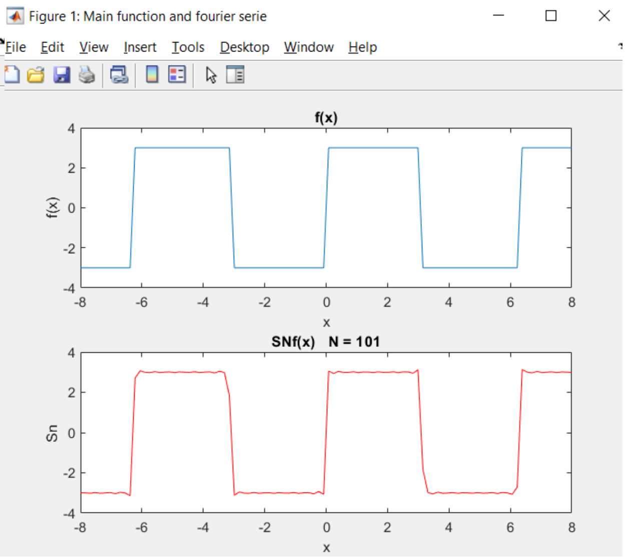


Figure 1: Main function and fourier serie
File Edit View Insert Tools Desktop Window Help
f(x)
4
2
0
-2
-4
f(x)
-8 -6 -4 -2 0 2 4 6 8
x
SNf(x)   N = 101
Sn
-8 -6 -4 -2 0 2 4 6 8
x

5-4-2)

```
%eN{f(x)} = |f(x) - SN{f(x)}
en = abs(fx - sn);
figure('Name' , 'Approxiamte error function')
plot(x, en);
title(['eN     N = ' num2str(N)]);
xlabel('x');
ylabel('eN');
axis ([-8, 8, 0, 3]);
```

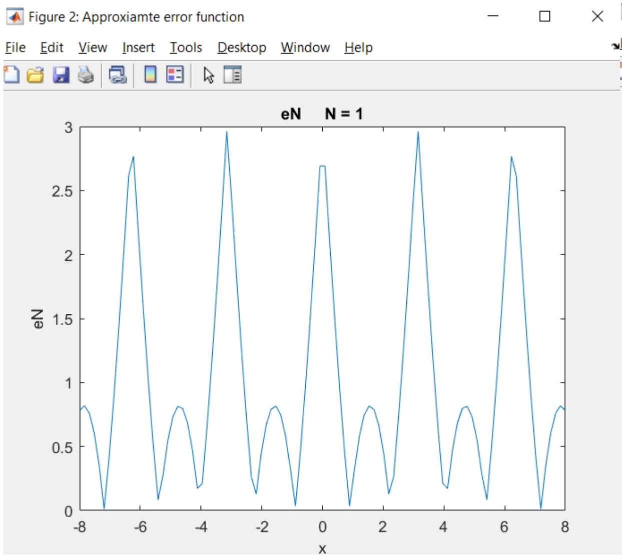


Figure 2: Approxiamte error function
File Edit View Insert Tools Desktop Window Help
eN   N = 1
3
2.5
2
1.5
1
0.5
0
eN
-8 -6 -4 -2 0 2 4 6 8
x

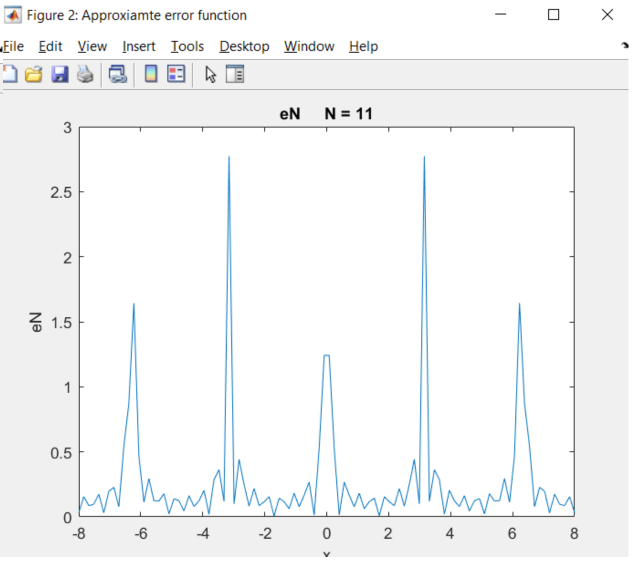
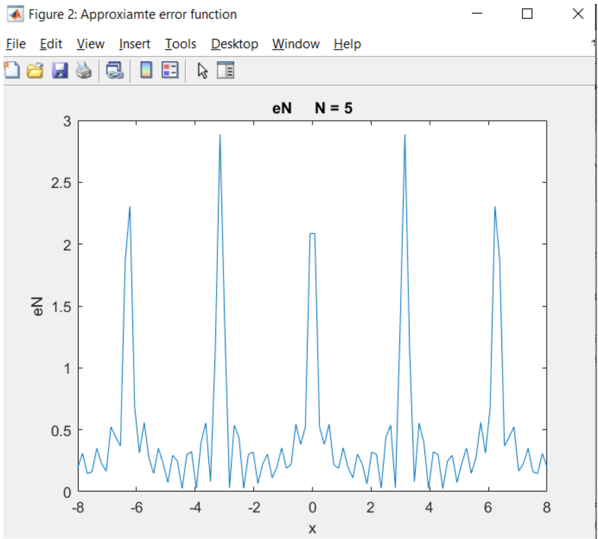
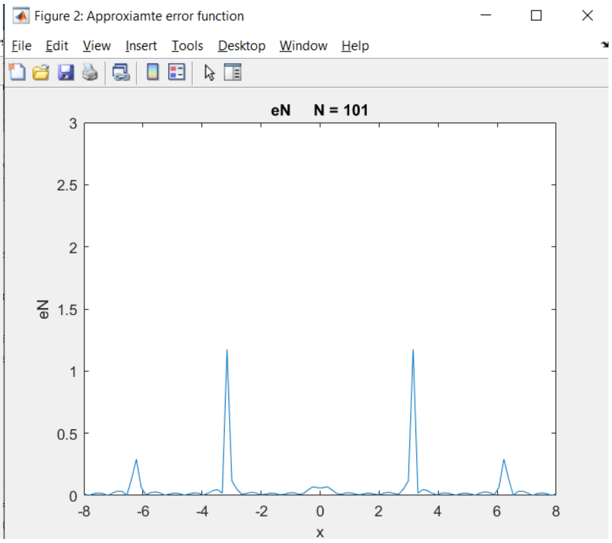


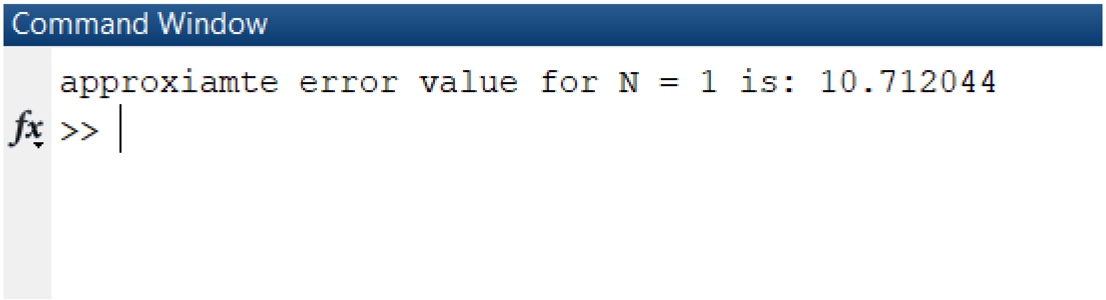
Figure 2: Approxiamte error function
File Edit View Insert Tools Desktop Window Help
eN   N = 11
3
2.5
2
1.5
1
0.5
0
eN
-8 -6 -4 -2 0 2 4 6 8
x



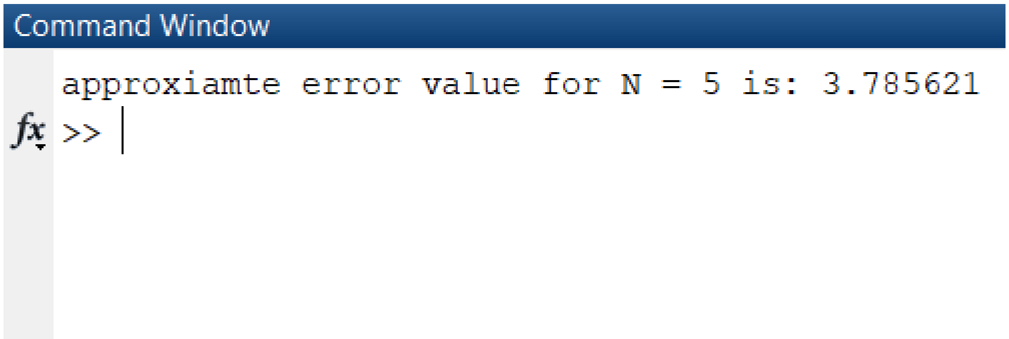


5-4-3)

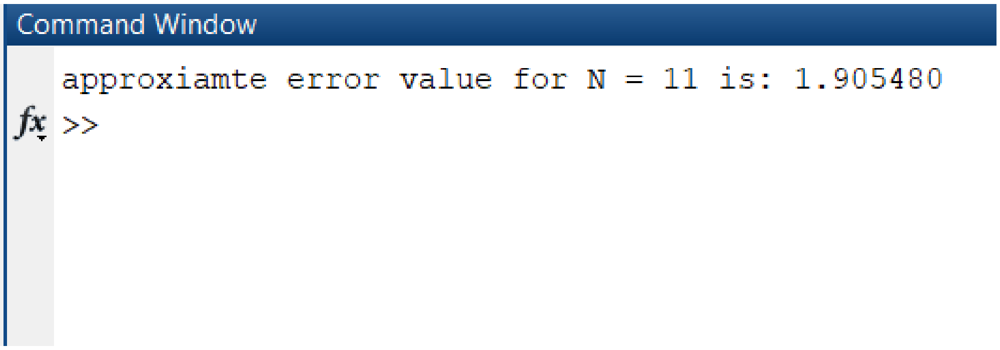
```
%En{f(x)} = integral(f(x)^2,-pi,pi) - 2*pi*a0^2 - pi*sigma(an^2+bn^2,1,N)
integralf2 = integral(@(x) (2*k*((sin(x)>0) - 0.5)) .^2, -pi, pi);
sigma = 0;
for n = 1 : N
    sigma = sigma + bn(n) ^ 2;
end
% a0 =0 and an =0
answer = integralf2 - pi*sigma;
fprintf('approxiamte error value for N = %d is: %f\n',N,answer);
```



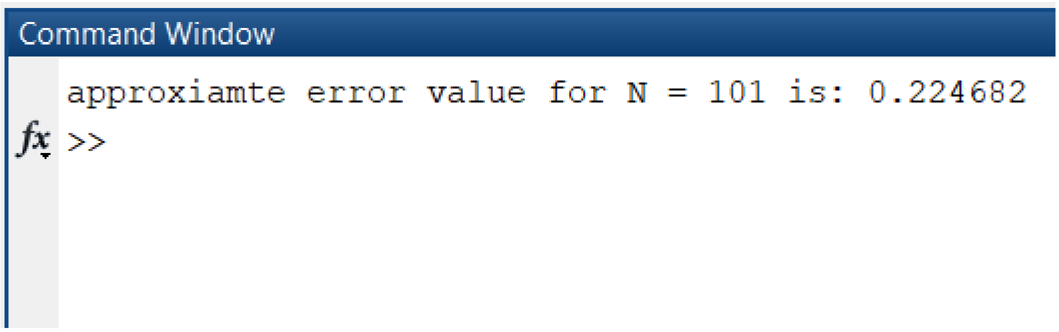
```
approxiamte error value for N = 1 is: 10.712044
fx >>
```



```
approxiamte error value for N = 5 is: 3.785621
fx >>
```



```
approxiamte error value for N = 11 is: 1.905480
>>
```



```
approxiamte error value for N = 101 is: 0.224682
>>
```

5-4-4)

```
%finding the first and least N for(E=<0.01)
N= 1;
%the previous value of approximate error
E = answer;
while E > 0.001
    sigma = 0;
    for n = 1 : N
        bn(n) = ((2*k) / (n*pi)) * (1 - cos(n*pi));
        sigma = sigma + bn(n)^2;
    end
    E = integralf2 - pi*sigma;
    N = N + 1;
end
N = N-1;
disp("N{min} for (E<=0.01)  =  " +N)
```

Command Window : N{min} for (E<=0.01) = 22919