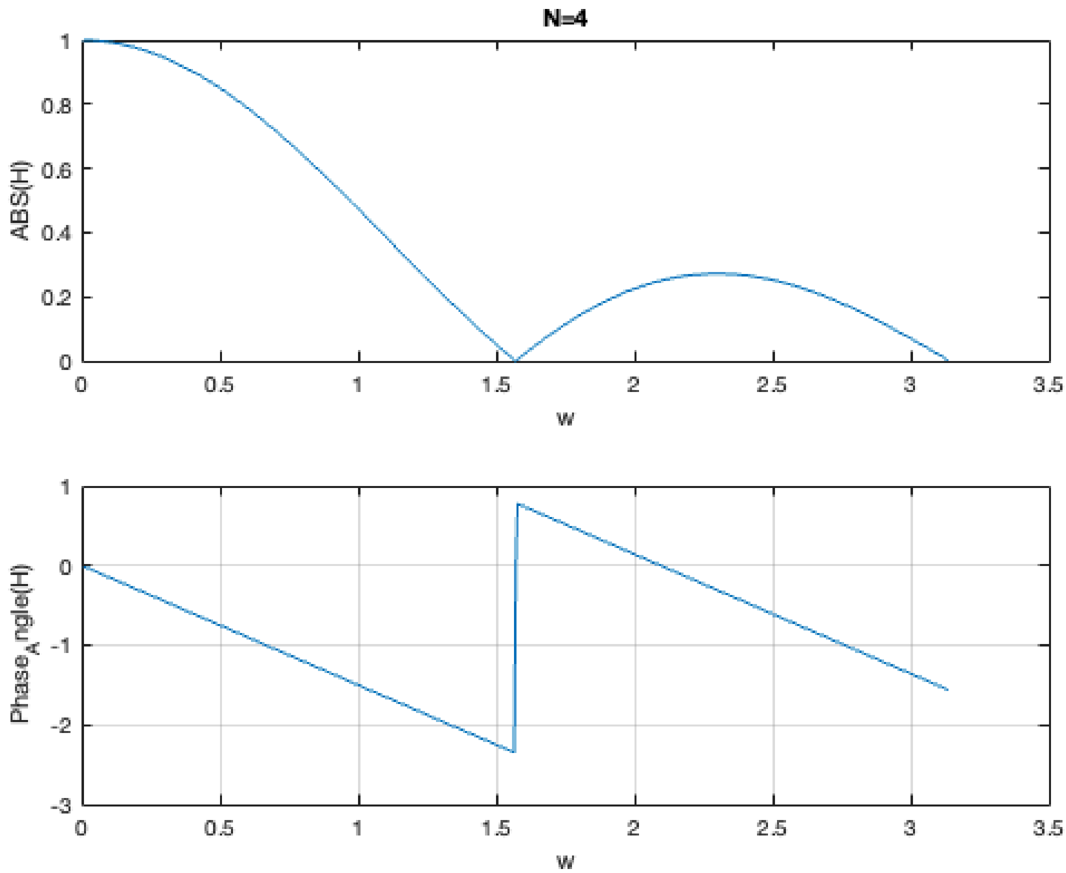
Table of Contents

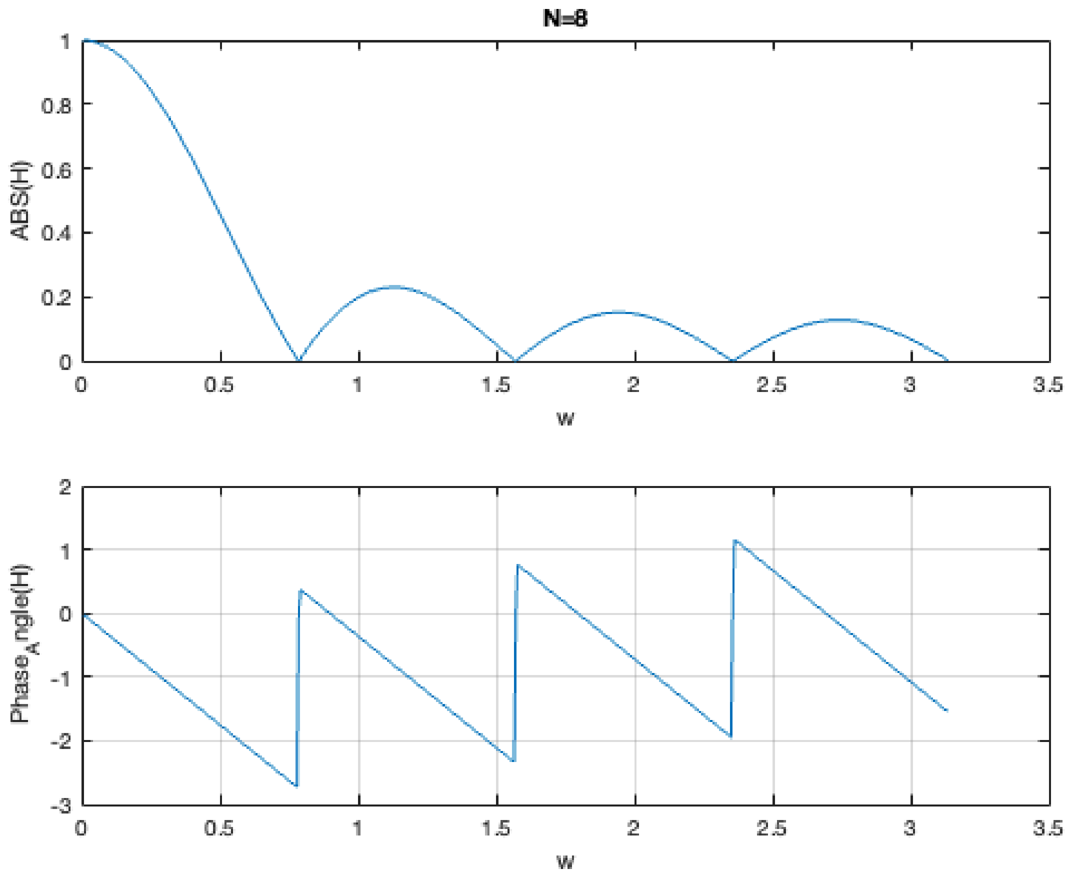

```
clc; close all; clear;

% Setting up required functions
u = @(n)double(n>=0);
del = @(n)double(n==0);
```

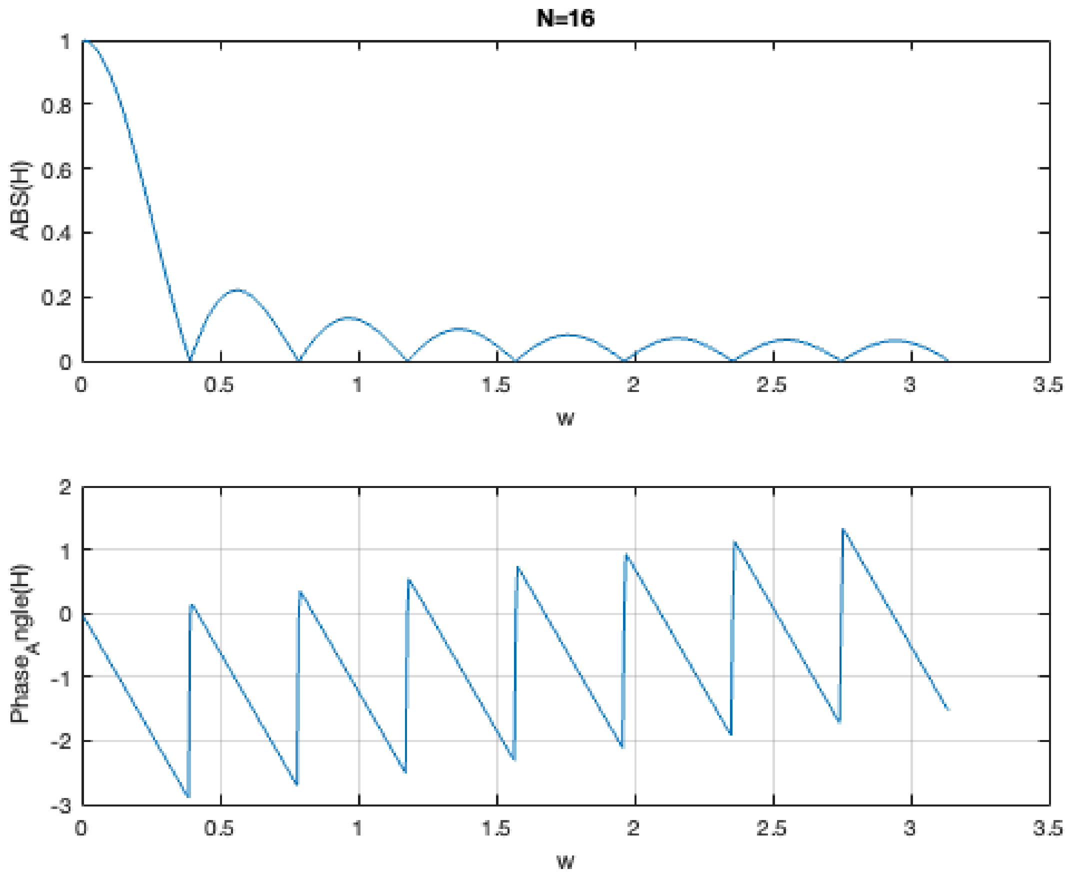
Question 1

```
% N = 4

N = 4;
b = zeros(1,N+1);
b(1) = 1;
b(N+1) = -1;
a = [1,-1];
[H, w] = freqz(b/N,a);

figure;
subplot(2,1,1);
plot(w,abs(H));
ylabel("ABS(H)")
xlabel("w")
title("N=4")
subplot(2,1,2);
plot(w,angle(H));
ylabel("Phase_Angle(H)");
xlabel("w");
grid on;

% What kind of filter is this? Low Pass Filter

% N = 8

N = 8;
b = zeros(1,N+1);
b(1) = 1;
b(N+1) = -1;
```

```
a = [1,-1];
[H, w] = freqz(b/N,a);

figure;
subplot(2,1,1);
plot(w,abs(H));
ylabel("ABS(H)")
xlabel("w")
title("N=8")
subplot(2,1,2);
plot(w,angle(H));
ylabel("Phase_Angle(H)")
xlabel("w")
grid on;

% N = 16

N = 16;
b = zeros(1,N+1);
b(1) = 1;
b(N+1) = -1;
a = [1,-1];
[H, w] = freqz(b/N,a);

figure;
subplot(2,1,1);
plot(w,abs(H));
ylabel("ABS(H)")
xlabel("w")
title("N=16")
subplot(2,1,2);
plot(w,angle(H));
ylabel("Phase_Angle(H)")
xlabel("w")
grid on;

% Whether or not you believe it is doing a good job of it? It depends on
% how you define a "good job". It is attenuating the amplitude of higher
% frequnecies as expected of a low pass filter but not enough as I see it.
% The moving filter is a good smoothing filter but a bad low-pass filter.
% (source:
% https://www.analog.com/media/en/technical-documentation/dsp-book/
dsp_book_ch15.pdf)
```

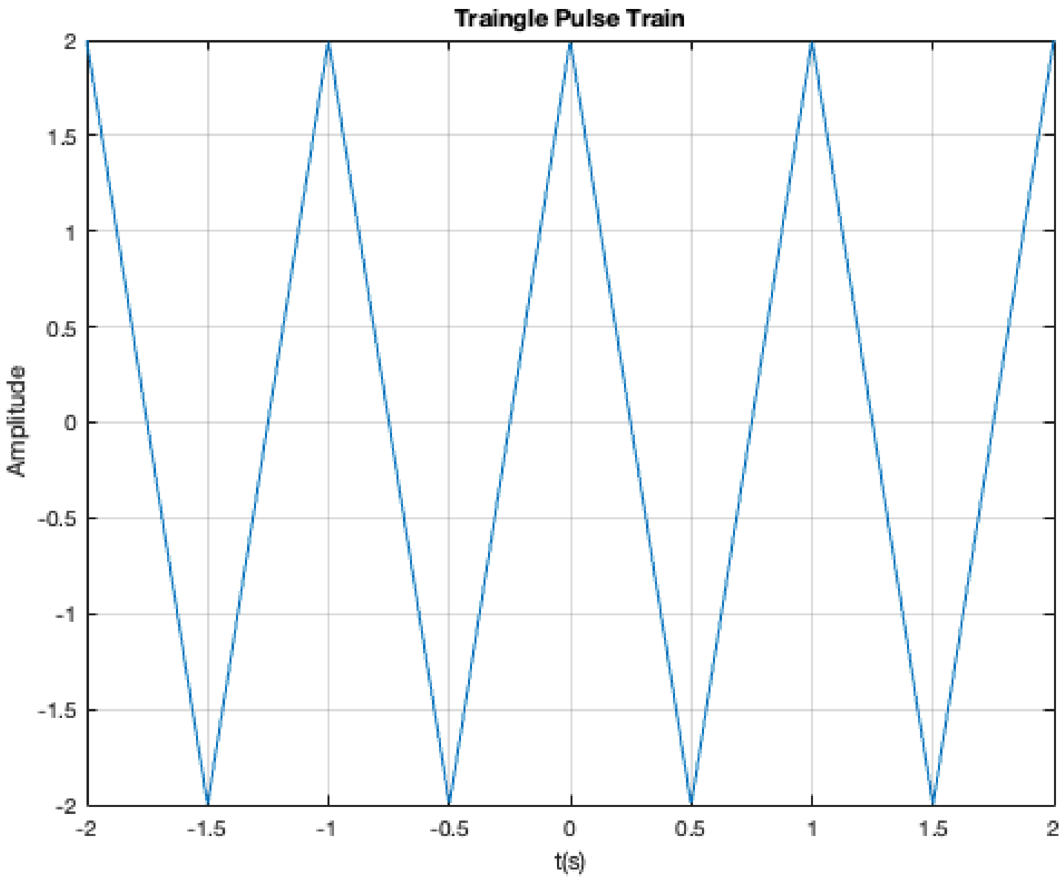

N=4
ABS(H)
w
Phase_Angle(H)
w
1
0.8
0.6
0.4
0.2
0
0.5
1.5
2
2.5
3
3.5
-1
-2
-3


N=8
ABS(H)
1
0.8
0.6
0.4
0.2
0
0
0.5
1
1.5
2
2.5
3
3.5
w
Phase_Angle(H)
2
1
0
-1
-2
-3
0
0.5
1
1.5
2
2.5
3
3.5
w



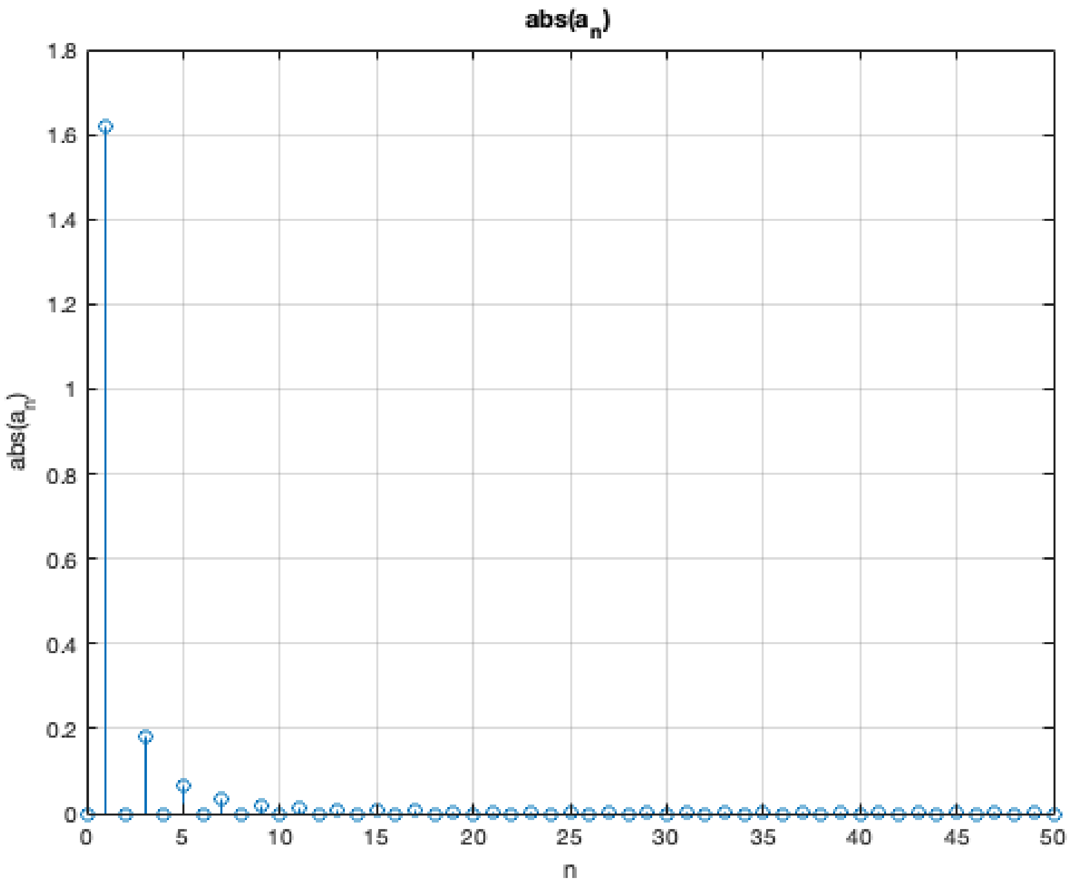
Question 2 - a)

```
figure;
T = 10*(1/50);
t = linspace(-2,2,1000);
x = 2*sawtooth(2*pi*(t-0.5),1/2);
plot(t,x)
xlabel("t(s)");
ylabel("Amplitude")
title("Traingle Pulse Train")
grid on
```



Question 2 - b)

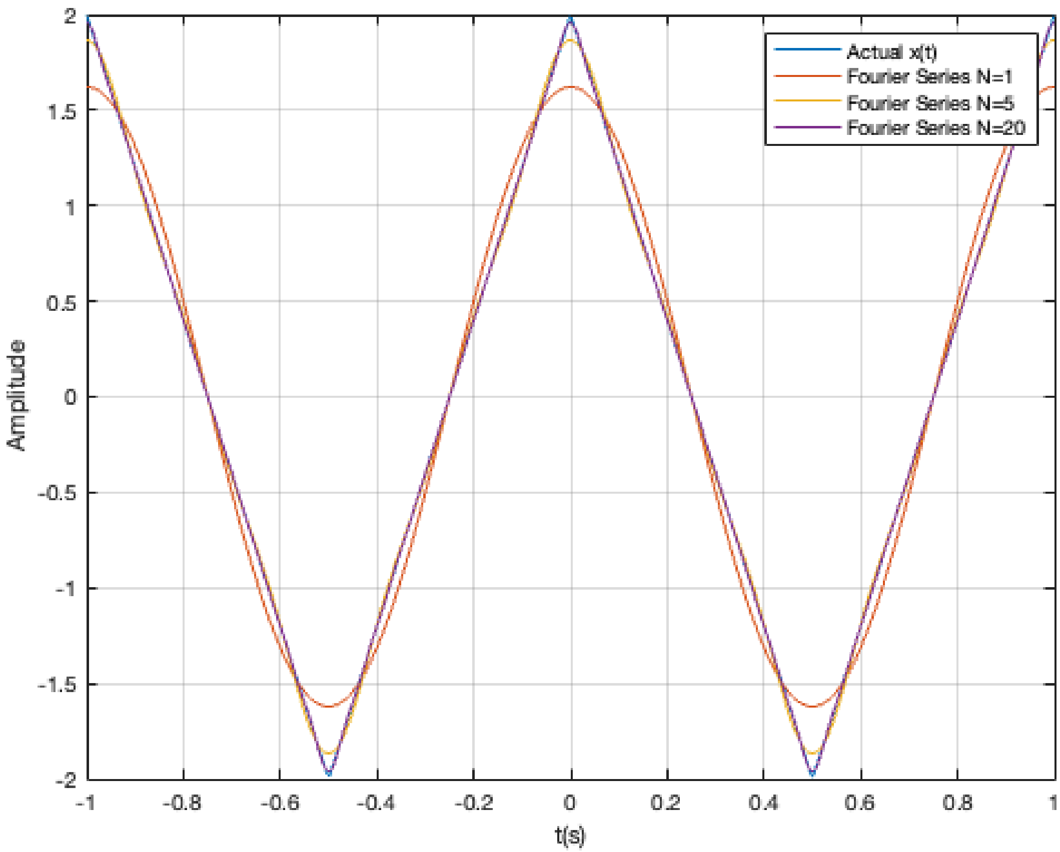
```
A = 2;
n = 0:50;
a = [];

for c = n
    if mod(c, 2) == 1
        a = [a,8*A/(pi*pi*c*c)];
    else
        a = [a,0];
    end
end

figure;
stem(n,abs(a));
xlabel("n");
ylabel("abs(a_n)")
title("abs(a_n)")
grid on
```



Question 2 - c)

```
% N = 1
n1 = 0;
for c = 0:1
    n1 = n1 + a(c+1)*cos(2*pi*c*t/1);
end

% N = 5
n2 = 0;
for c = 0:5
    n2 = n2 + a(c+1)*cos(2*pi*c*t/1);
end

% N = 20
n3 = 0;
for c = 0:20
    n3 = n3 + a(c+1)*cos(2*pi*c*t/1);
end


figure;
```

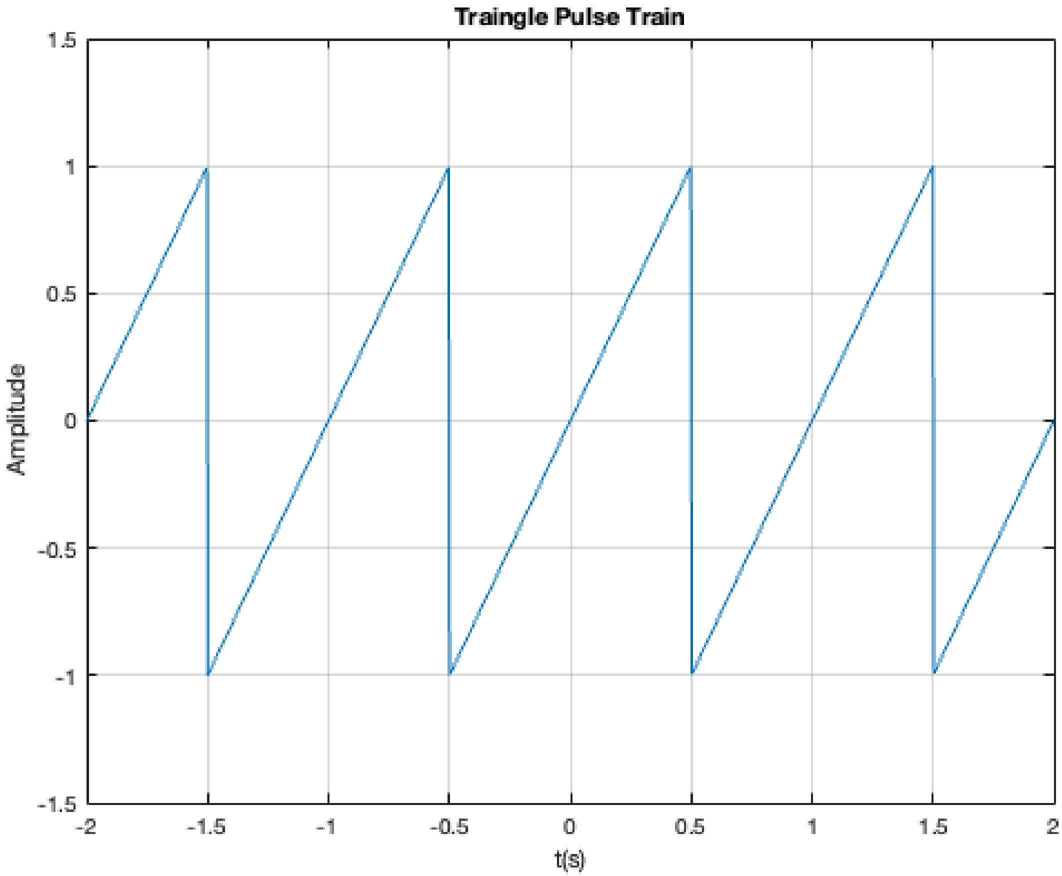
```
plot(t,x);
hold on;
plot(t,n1);
hold on;
plot(t,n2);
hold on;
plot(t,n3);
xlim([-1,1])
xlabel("t(s)");
ylabel("Amplitude")
legend('Actual x(t)','Fourier Series N=1','Fourier Series N=5','Fourier Series
 N=20')
grid on

% What happens as N increases?
% The foureir series gets closer and closer to the actual graph of x(t)
```



Question 3 - a

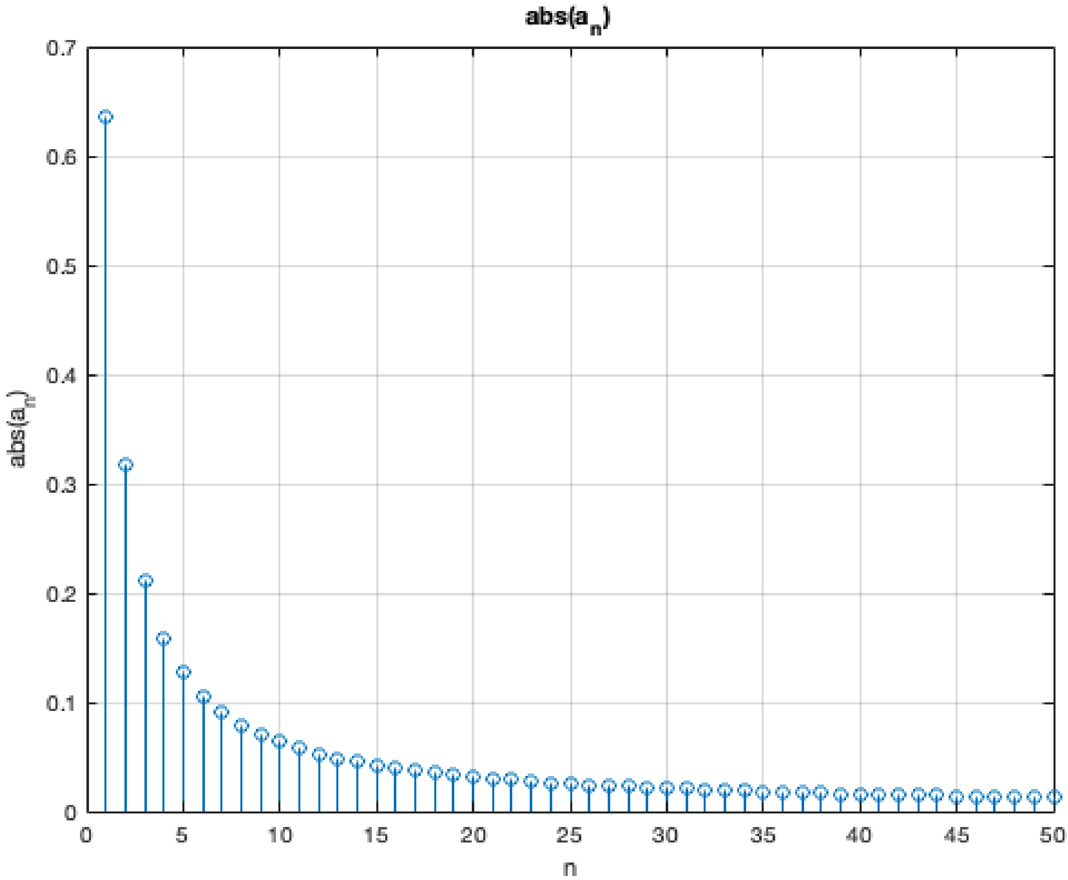
```
figure;
T = 10*(1/50);
t = linspace(-2,2,1000);
x = sawtooth(2*pi*(t-0.5));
```

```
plot(t,x)
ylim([-1.5,1.5])
xlabel("t(s)");
ylabel("Amplitude")
title("Traingle Pulse Train")
grid on
```



Question 3 - b

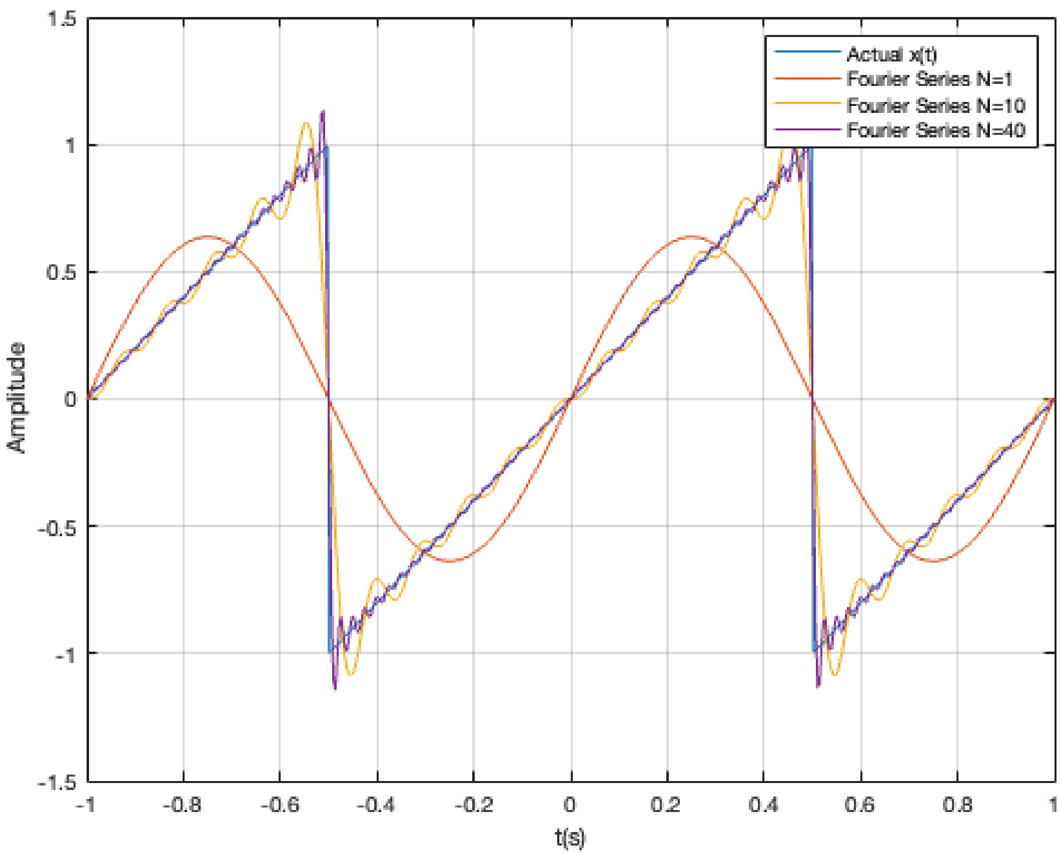
```
n = 1:50;
a = [];

for c = n
    a = [a, ((2)/(pi*c))*(-1)^(c+1)];
end

figure;
stem(n,abs(a))
xlabel("n");
ylabel("abs(a_n)")
title("abs(a_n)")
grid on
```



Question 3 - c

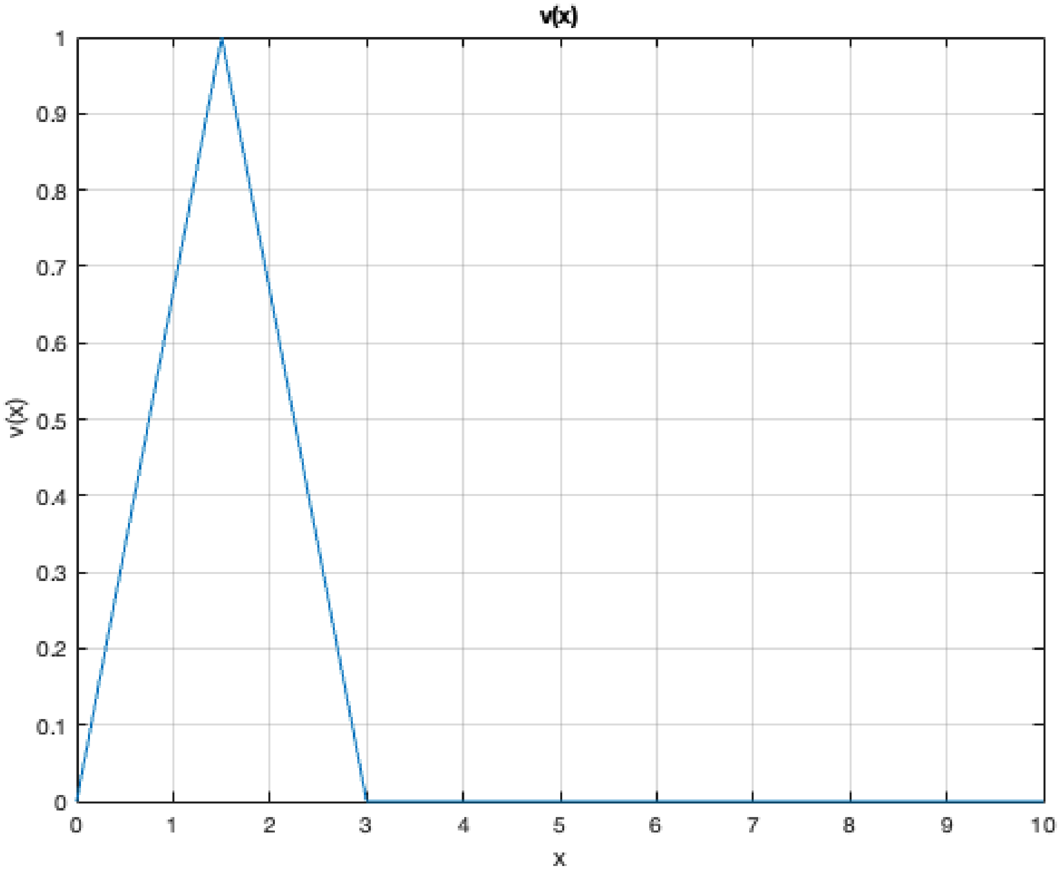
```
% N = 1
n1 = 0;
for c = 1:1
    n1 = n1 + a(c)*sin(2*pi*c*t/1);
end

% N = 10
n2 = 0;
for c = 1:10
    n2 = n2 + a(c)*sin(2*pi*c*t/1);
end

% N = 40
n3 = 0;
for c = 1:40
    n3 = n3 + a(c)*sin(2*pi*c*t/1);
end

figure;
plot(t,x);
```

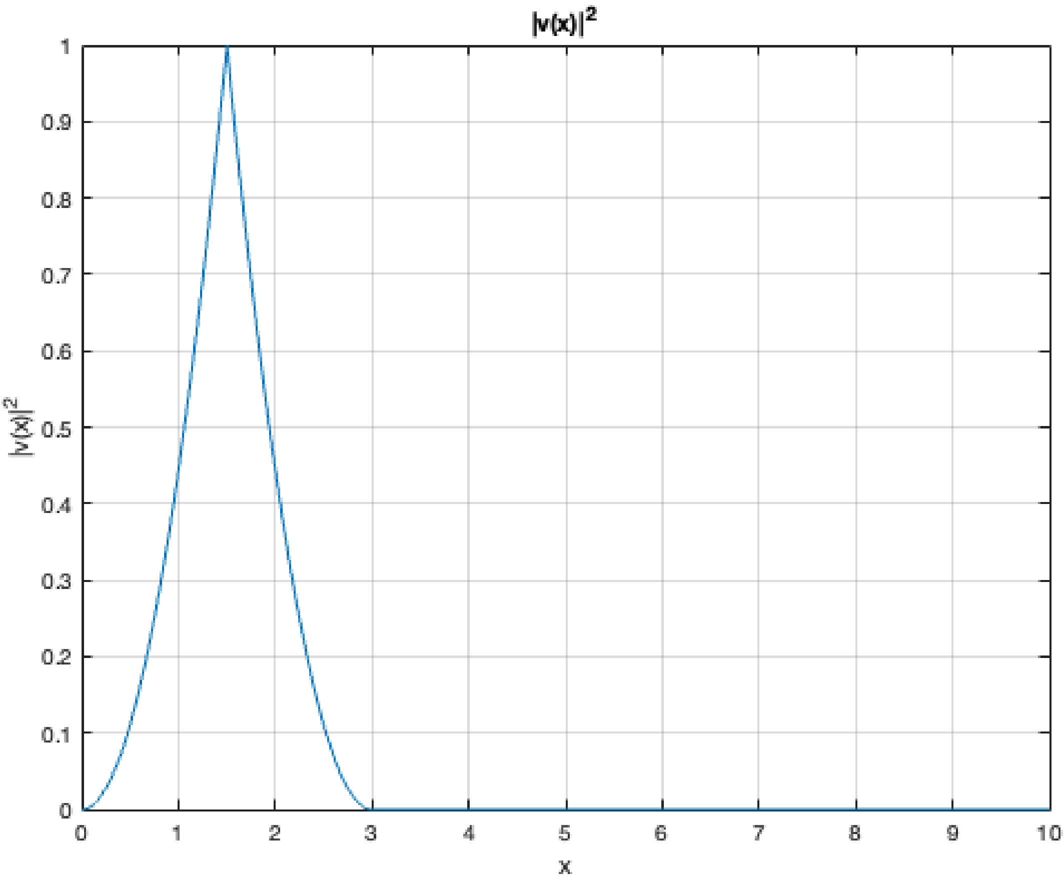
```
hold on;
plot(t,n1);
hold on;
plot(t,n2);
hold on;
plot(t,n3);
xlim([-1,1])
xlabel("t(s)");
ylabel("Amplitude")
legend('Actual x(t)','Fourier Series N=1','Fourier Series N=10','Fourier
 Series N=40')
grid on

% What happens as N increases?
% The foureir series gets closer and closer to the actual graph of x(t)

% Do you notice any behavior that wasn t present in the triangular pulse
 trainDs Fourier series?
% The reason behind this could be the jumps in the graph from -1 to 1. Large
 spikes appear
% at the points of discontinuities during the period when the value
% transitions from -1 to 1. This is known as gibbs phenomenon. This is due
% to the oscillations overshooting.
```



Question 4 - a

```
step_x = 0.0001;
x = 0:step_x:10;
v = zeros(size(x));

for i = 1:length(x)
    if x(i) >= 0 && x(i) <= 1.5
       v(i) = (2/3) * x(i);
    elseif x(i) > 1.5 && x(i) <= 3
       v(i) = 2*(1-(x(i)/3));
    else
       v(i) = 0;
    end
end

figure;
plot(x,v)
xlabel("x");
ylabel("v(x)")
title("v(x)")
grid on;

pdf = abs(v).*abs(v);
figure;
plot(x,pdf)
xlabel("x");
ylabel("|v(x)|^2")
title("|v(x)|^2")
grid on
```


v(x)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
v(x)
0
1
2
3
4
5
6
7
8
9
10
x



Question 4 - b

```
area = 0;
for i = 1:(length(x)-1)
    height = pdf(i);
    height_after = pdf(i+1);
    trapz_area = (height+height_after)*step_x*0.5;
    area = area + trapz_area;
end;

disp("Area (Numerially Calculated): " + area);
syms x_symbolic
disp("Area (Using Matlab for verification): " + trapz(x,pdf));

% Area of 1. Make sense as pdf is always 1 (sum of probs equal to 1)
```

Area (Numerially Calculated): 1
Area (Using Matlab for verification): 1

Question 4 - c

```
n = 1:50;
```

```
a = [];

for c = n
    a = [a,8*sin(c*pi*0.5)/(pi*pi*c*c)];
end

comp = 0;
L = 3;

for c = n
    comp = comp + (L/2)*(a(c)*a(c));
end

disp("Part C = " + comp);

% Compare your answers to parts (b) and (c).
% The same answer. both 1. Probabilty distribution is always 1 as it is the
% sum of probabilites.

% Can you give any reasoning to justify your answers?
% Both sides have the same area under the graph of 1.
% To simplify, the equality arises because both reflect the total probability
 of locating the particle.
% In the first case it was the sum of probabilites for all possible
% positions in and the second being the summationf of contributions from
% all possibles states.
```

```
Part C = 1
```

Published with MATLAB® R2023a