Use what is in the noisy C script to generate a noisy sine wave:

```
fs = 1e2;
t = 0:1/fs:1;
sw = sin(2*pi*262.62*t); % Middle C


n = 0.1*randn(size(sw));
swn = sw + n;
```

1. **Plot the magnitude and phase response oh the function**

```
t = 1:100;
y = 4 * sin(50 * t) ./ (6 * t);
figure, plot(t, abs(y)), title('Amplitude plot')
figure, plot(t, angle(y)), title('Phase plot')
```

2.

```
T = 1; % for a square wave of time period = 2*T = 2
secs

t = 0:0.001:2; % time base ranges from 0 to 2 secs -> 1 period of
wave

n = 1:2:39; % taking odd n to compute sine waves.
% the higher the value of n, the better the
square shape

wave = zeros(1,length(t));
s = wave;
% initialise arrays to store computed sine values
and
% running sum of Fourier series

for i = n % Perform summation specified by equation
wave = 1/i*sin(i*pi*t/T);
s = s + wave;
end

plot(t, s); % and finally...
```

3.

```
clear all
f = 500;
c = 4/pi;
w0 = 2*pi*f;
t=0:0.05e-3:4e-3;
s=zeros(1,length(t));
for n = 1: 12
s = s+c*(1/(2*n - 1))*sin((2*n - 1)*w0*t);
end
plot(t,s)
xlabel('Time, s')
ylabel('Amplitude, V')
title('Fourier series expansion')
```

4.

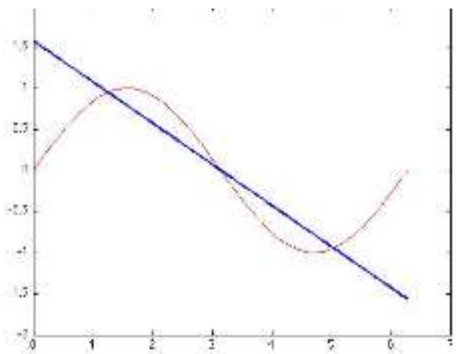
```
ezplot('sin(x) + sin(3*x)/3');

ezplot('sin(x) + sin(3*x)/3 + sin(5*x)/5 + sin(7*x)/7 + sin(9*x)/9');

ezplot('sum(sin([1:2:33] .* x) ./ [1:2:33])');
```

5. MATLAB Code for Fourier Sine Series,

```
x = linspace(0,2*pi,100);
sum = 0.*x;  %M = number of coefficients used
for j = 1:M
sum = sum + ((1/j)*sin(j*x));
end
F = ((1/2)*(pi-x));
plot(x, sum,’r’);
hold on
plot(x, F, ’LineWidth’, 2);
hold on
error = abs(sum - F);
plot(x, error,’m’)
hold on
```



ure 12: M = 1 Coefficient

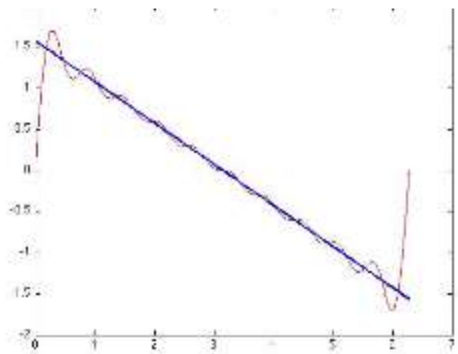
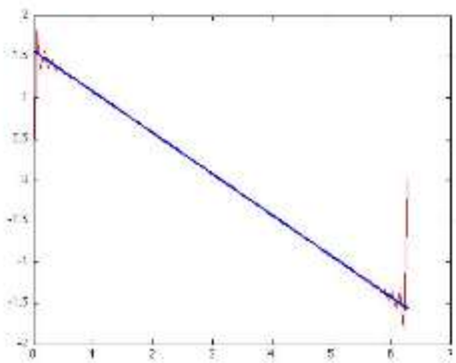


Figure 13: M = 10 Coefficients



ure 14: M = 50 Coefficients

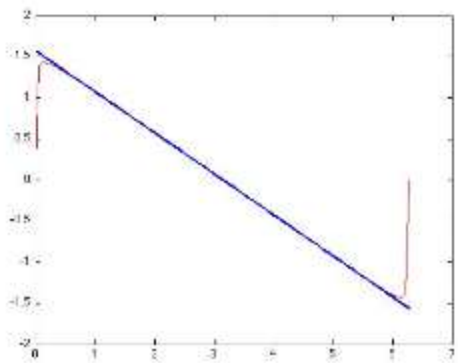


Figure 15: M = 100 Coefficients

$$F(x) = \sum_{-N}^{N} f(x)e^{i(k\pi x)}$$

6. Calculate Fourier coefficient

```
%task 01
clear all
clc
syms n aSym(n) bSym(n) x(t) t A(t)
To=4;
f=1/To;
wo=2*pi*f;
A(t)=1;
x(t)=A(t)*(1+2*heaviside(t+1)-2*heaviside(t+3)-2*heaviside(t-
1)+2*heaviside(t-3)+2*heaviside(t+5)-2*heaviside(t-5)-
2*heaviside(t+7)+2*heaviside(t-7));

a0=(int(x(t),0,To)/(To))/(2*pi)
aSym(n)=simplify(int(x(t)*cos(n.*wo.*t),t,0,To)/(To/2))
```

```
bSym(n)=simplify(int(x(t)*sin(n.*wo.*t),t,0,To)/(To/2))

nMax = 20;
n = 1:nMax;
an = eval(aSym(n));
bn=  eval(bSym(n));

subplot(411)
stem(a0)
ylabel('a0')

subplot(412)
stem(an)
ylabel('an')

subplot(413)
stem(bn)
ylabel('bn')

xApprox = a0;
for m = 1:nMax
    xApprox = xApprox + an(m)*cos(m*wo*t) + bn(m)*sin(m*wo*t);
end
t = -20*pi:0.1:20*pi;

xCalc = eval(xApprox);
subplot(4,1,4);
plot(t,xCalc)
grid on
ylabel('x(t)')
```

7.

```
clear all
clc
syms t n aSym(n) bSym(n) x(t) f(t)
tStart = -pi;
tStop = pi;
To = 10*pi;
f(t) = 1;
wo = 2*pi/To;
x(t) = f(t) *(heaviside(t-tStart)-heaviside(t-tStop));
a0 = (int(x(t),t,tStart,tStop)/To)/(2*pi)
aSym(n)  = simplify(int(x(t)*cos(n*wo*t),t,tStart,tStop)/(To/2))
bSym(n)  = simplify(int(x(t)*sin(n*wo*t),t,tStart,tStop)/(To/2))

nMax = 20;
n = 1:nMax;
a = eval(aSym(n));
b = eval(bSym(n));

figure
```

```
subplot(3,1,1)
stem(n,a)
ylabel('a(n)');
subplot(3,1,2);
stem(n,b)
ylabel('b(n)');

xApprox = a0;
for m = 1:nMax
    xApprox = xApprox + a(m)*cos(m*wo*t) + b(m)*sin(m*wo*t);
end
t = -20*pi:0.1:20*pi;

xCalc = eval(xApprox);
subplot(3,1,3);
plot(t,xCalc)
grid on
ylabel('x(t)')
```

8.

```
clear all
clc
syms n aSym(n) bSym(n) x(t) t A(t)
To=2*pi;
f=1/To;
wo=2*pi*f;
A(t)=1;
x(t)=A(t)*t*heaviside(t)-(t*heaviside(t-(2*pi)));

a0=(int(x(t),0,To)/(To))/(2*pi)
aSym(n)=simplify(int(x(t)*cos(n.*wo.*t),t,0,To)/(To/2))
bSym(n)=simplify(int(x(t)*sin(n.*wo.*t),t,0,To)/(To/2))

nMax = 20;
n = 1:nMax;
an = eval(aSym(n));
bn=  eval(bSym(n));

subplot(411)
stem(a0)
ylabel('a0')

subplot(412)
stem(an)
ylabel('an')

subplot(413)
stem(bn)
ylabel('bn')

xApprox = a0;
```

```
for m = 1:nMax
    xApprox = xApprox + an(m)*cos(m*wo*t) + bn(m)*sin(m*wo*t);
end
t = -20*pi:0.1:20*pi;

xCalc = eval(xApprox);
subplot(4,1,4);
plot(t,xCalc)
grid on
ylabel('x(t)')
```

9.

```
clear all
close all
syms n aSym(n) bSym(n) x(t) t A(t)
To=pi;
f=1/To;
wo=2*pi*f;
A(t)=1;
x(t)=A(t)*(t*heaviside(t+pi/4)-t*heaviside(t-pi/4));

a0=(int(x(t),-pi/4,3*pi/4)/To)/(2*pi)
aSym(n)=simplify(int(x(t)*cos(n.*wo.*t),t,-pi/4,3*pi/4)/(To/2))
bSym(n)=simplify(int(x(t)*sin(n.*wo.*t),t,-pi/4,3*pi/4)/(To/2))

nMax = 20;
n = 1:nMax;
a = eval(aSym(n));
b=  eval(bSym(n));

subplot(411)
stem(a0)
ylabel('a0')

subplot(412)
stem(a)
ylabel('an')

subplot(413)
stem(b)
ylabel('bn')

xApprox = a0;
for m = 1:nMax
    xApprox = xApprox + a(m)*cos(m*wo*t) + b(m)*sin(m*wo*t);
end
t = -20*pi:0.1:20*pi;

xCalc = eval(xApprox);
subplot(4,1,4);
plot(t,xCalc)
grid on
```

```
ylabel('x(t)')
```

10.

```
clear all
close all
syms n aSym(n) bSym(n) x(t) t A(t)
To=4;
f=1/To;
wo=2*pi*f;
A(t)=1;
x(t)=A(t)*t*heaviside(t)-(t*heaviside(t-1));

a0=(int(x(t),0,1)/To)
aSym(n)=simplify(int(x(t)*cos(n.*wo.*t),t,0,1)/(To/2))
bSym(n)=simplify(int(x(t)*sin(n.*wo.*t),t,0,1)/(To/2))

nMax = 20;
n = 1:nMax;
a = eval(aSym(n));
b=  eval(bSym(n));

subplot(411)
stem(a0)
ylabel('a0')

subplot(412)
stem(a)
ylabel('an')

subplot(413)
stem(b)
ylabel('bn')

xApprox = a0;
for m = 1:nMax
    xApprox = xApprox + a(m)*cos(m*wo*t) + b(m)*sin(m*wo*t);
end
t = -20*pi:0.1:20*pi;

xCalc = eval(xApprox);
subplot(4,1,4);
plot(t,xCalc)
grid on
ylabel('x(t)')
```

Approximating f(x) = x:

In order to actually approximatef(x) =x using fourier exponential coefficients we created a set of MATLAB codes.They were to be run one after the other as each code called some parameter from the previous

```
clc
np = 100
nc = 30
k = -nc:nc;
x = linspace(-1,1,np);
f = @(x) x;
for j = 1:length(k)
C = @(x) f(x).*exp(-1i*x*k(j)*pi);
fk(j) = quad(C,-1,1);  endplot(k, fk);
his first code is used to calculate the coefficients to be used later.function
F = reconstruction(y,fk,k)
for x = 1:length(y)
F(x) = 0;
for j= 1:length(k)
F(x) = F(x) + fk(j)*exp(1i*k(j)*y(x)*pi);
end
end
end
```

```
clc np = 100;nc = 5;f = @(x) x;npy = 101;x = linspace(-1,1,np);y = linspace(-1,1,npy);k = -nc:nc;for j = 1:length(k)C = @(x) f(x).*exp(-1i*pi*x*k(j));fk(j) = quad(C,-1,1,1e-12);endF = reconstructionexpo(y,fk,k);F = F./(2);plot(x,f(x),'--b')hold onplot(y,F,'-xr')
```

About Fourier Series Models

The Fourier series is a sum of sine and cosine functions that describes a periodic signal. It is represented in either the trigonometric form or the exponential form. The toolbox provides this trigonometric Fourier series form

$$y=a_0+\sum_{i=1}^{n} a_i\cos(iwx)+b_i\sin(iwx)$$

Fit a Two-Term Fourier Model

```
load enso;
f = fit(month,pressure,'fourier2')
```

```
plot(f,month,pressure)

f2 = fit(month,pressure,'fourier8')
plot(f2,month,pressure)

f3 = fit(month,pressure,'fourier8', 'StartPoint', coeffs);
plot(f3,month,pressure)
hold on
plot(f2, 'b')
hold off
legend( 'Data', 'f3', 'f2')
```

A. Write a matlab code to perform Fourier Series analysis of a given periodic time domain signal. The inputs for the matlab function are:

1. time domain signal,

2. signal periodicity

3. number of freq domain harmonics considered.

Apply your code to analyse a sinusoidal signal of your choice. present the graphs of the input and the frequency domain output (magnitude and phase)

B. Using the applet in the website (http://falstad.com/fourier/) identify the gibbs phenomenon for atleast 2 type of input signals. Show atleast 3 cases of modifying the Fourier coefficients and explain how that effects the gibbs phenomenon.

dft:

```
fs = 100; % It is sampling frequency
t=0:1/fs:5; % It is time series used to generate signal x
x = 4 * cos(2 * pi * 10 * t + pi / 6); % x is function of t
X = fft(x); % This statement computes Fourier transform of x
n = length(x); % length(x) gives the array length of signal x
c = (-1 * fs) / 2:fs / n:fs / 2 - fs / n; % It generates the frequency series to plot X in frequency
domain
subplot(4, 1, 1),plot(t,x); % This subplot shows the signal x vs. time series t
subplot(4, 1 ,2),plot(c,fftshift(abs(X))); % This subplot shows the Fourier spectrum of x with
zero frequency component shifted to center
subplot(4, 1, 3),plot(c,phase(X)); % This subplot shows the phase distribution of X (Fourier
transform of x)
subplot(4,1 ,4),plot(c,real(X)); % This subplot shows the real component of X spectrum
```

using for loop of dft:

```
%%using loop
%a is a constant
clear
a=2;
N=100;
for k=1:N
X(k)=0;
for n=1:N %this include u(n)
x(n)=a.^n;
X(k)= X(k)+(x(n)*exp(-j*2*pi*(k-1)*(n-1)/N));
end
end
subplot(2,1,1),plot(abs(X)) %magnitude
subplot(2,1,2),plot(angle(X)) %phase
%%using FFT
%a is a constant
clear
a=2;
N=100;
n=[1:1:N];
x=a.^n;
X=fftshift(fft(x),N);
subplot(2,1,1),plot(n,abs(X)) %magnitude
subplot(2,1,2),plot(n,angle(X)) %phase
```

https://www.projectrhea.org/rhea/index.php/Category:Signals_and_systems