# Table of Contents



```
% Matthew Lazarus 100962142
```

# Assignment 4

# Part 3

```
clearvars;
clear;
clear all;
close all;
clc;
```

Use stamps to generate the desired MNA matrices. These stamps were developed in ELEC 4506.

```
global G C b; %define global variables
G = zeros(6,6); % Define G, 5 node circuit (do not include additional
 variables)
C = zeros(6,6); % Define C, 5 node circuit (do not include additional
 variables)
b = zeros(6,1); % Define b, 5 node circuit (do not include additional
 variables)

vol(1,0,10);
cur(3,4,0.001)
% Use stamp for current controlled voltage source
ccvs(5,0, 4,0, 100);

res(1,2,1);
res(2,0,2);
res(3,4,10);
res(5,6,0.1);
res(6,0,1000);

cap(1,2,0.25);
cap(3,4,0.00001);
ind(2,3,0.2);
```

# A

The C matrix is now:

```
C
```

```
C =

  Columns 1 through 7

    0.2500   -0.2500         0         0         0         0         0
   -0.2500    0.2500         0         0         0         0         0
         0         0    0.0000   -0.0000         0         0         0
         0         0   -0.0000    0.0000         0         0         0
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0

  Columns 8 through 10

         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0         0
         0         0   -0.2000
```

# B

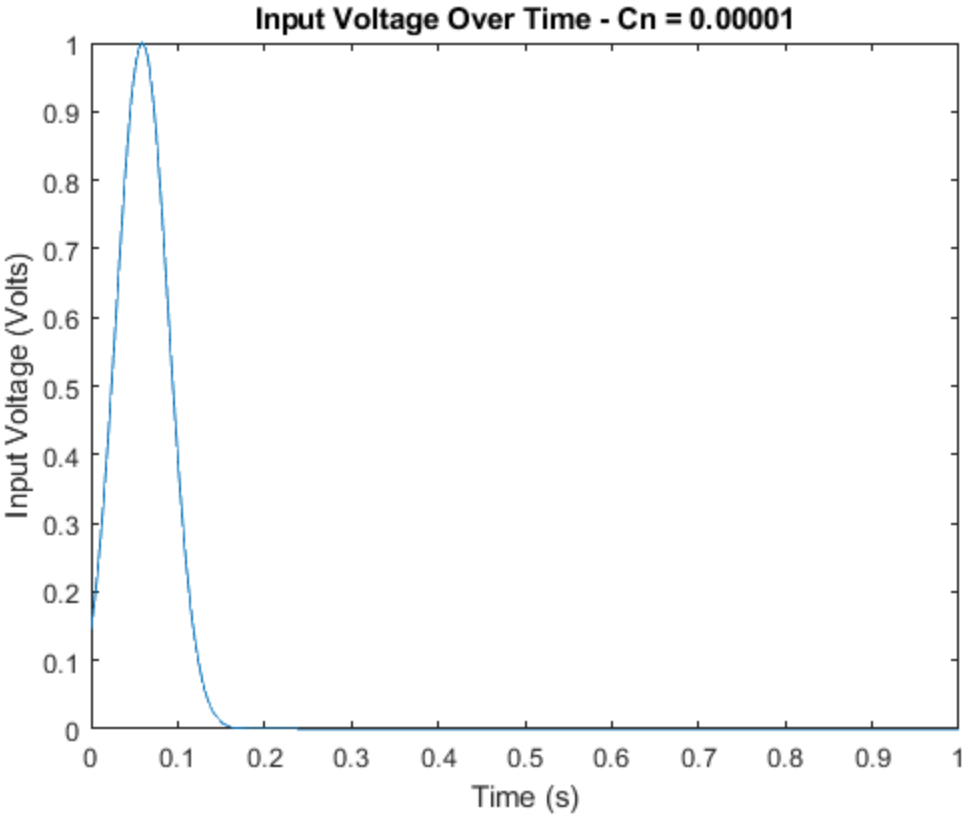
A noise source is added by setting In to 0.001*randn().

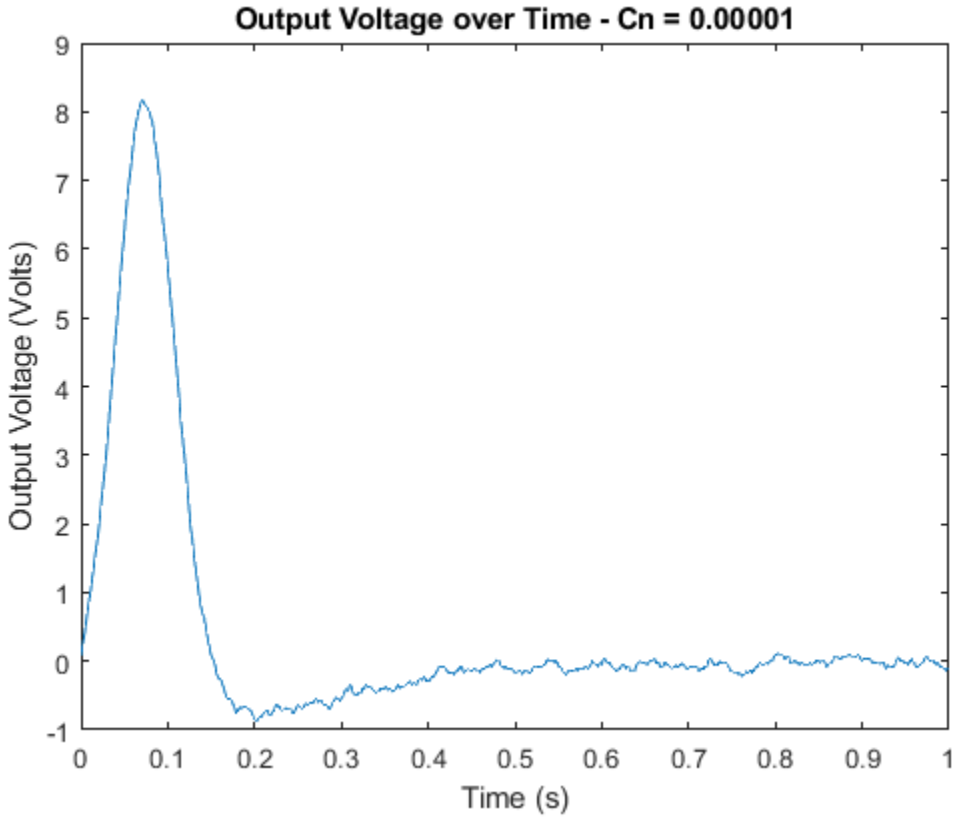
```
Xprev=zeros(10,1);

h=1/1000;
vInput = zeros(1000,1);
vOut = zeros(1000,1);
for count = 1:1000
    t=count*h;
    % Gaussian pulse, shifted by 0.06s and compressed to have std
 deviation
    % of 0.03.
    vInput(count) = exp(-0.5*((t-0.06)/0.03)^2);
end
```

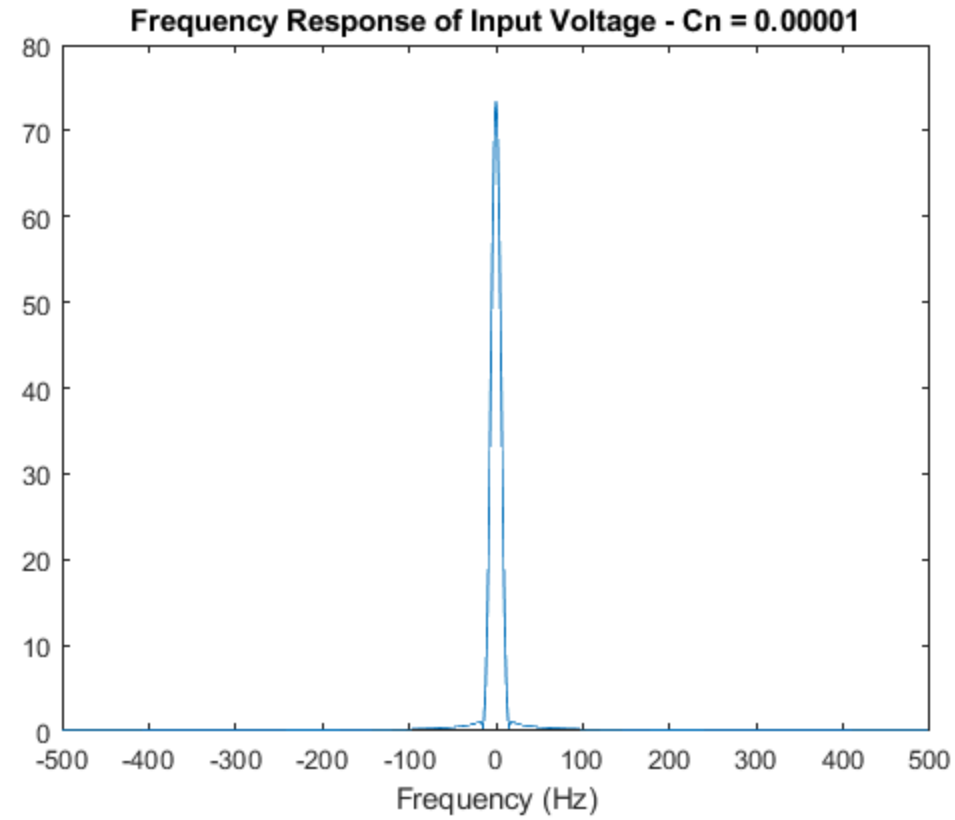
```
b(7) = vInput(1);
for count = 1:1000
    bNext = b;
    bNext(3)=0.005*randn();
    bNext(4)=-bNext(3);
    bNext(7) = vInput(count);
    Xnext = (G+(2*C/h))\((2*C/h - G)*Xprev+b+bNext);
    vOut(count) = Xnext(6);

    b = bNext;
    Xprev = Xnext;
end


fftVin = abs(fftshift(fft(vInput)));
fftVout = abs(fftshift(fft(vOut)));
n=length(fftVin);
fs=1/h;
fshift=(-n/2:n/2-1)*(fs/n);
```

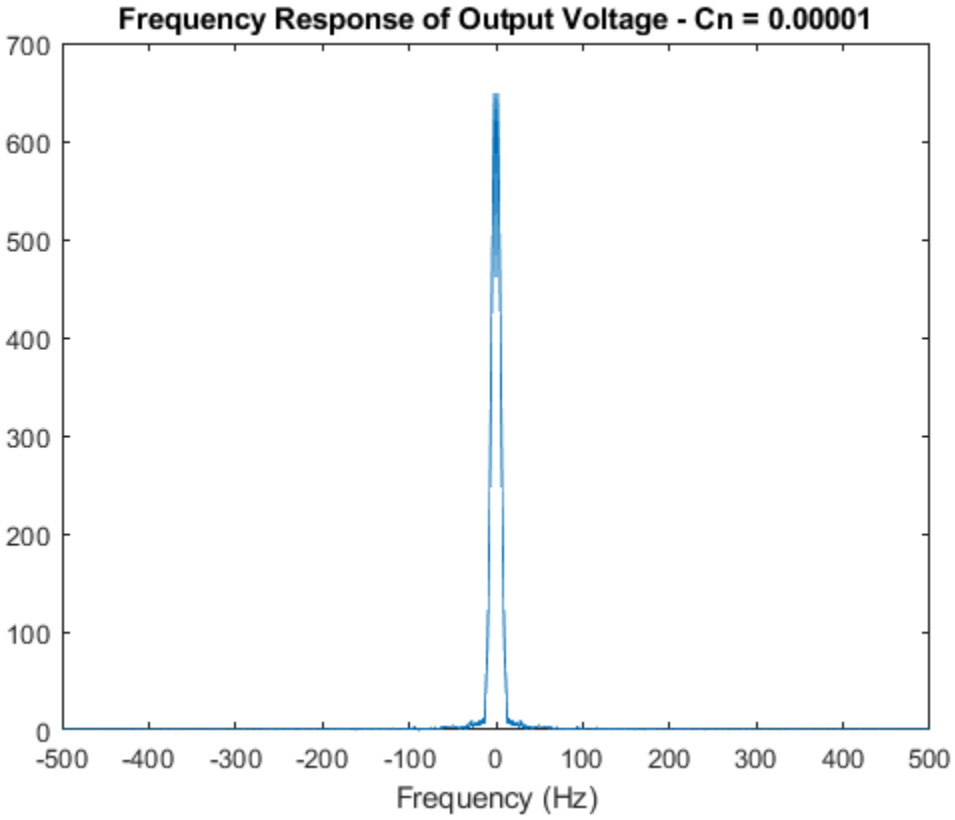
The figures below contain the time domain and frequency domain response of the circuit to a Gaussian input voltage with a noise source.

```
figure;
plot(linspace(0,1,1000),vInput)
xlabel('Time (s)')
ylabel('Input Voltage (Volts)')
title('Input Voltage Over Time - Cn = 0.00001')

figure;
plot(linspace(0,1,1000),vOut)
xlabel('Time (s)')
ylabel('Output Voltage (Volts)')
title('Output Voltage over Time - Cn = 0.00001')

figure;
plot(fshift, fftVin);
xlabel('Frequency (Hz)')
title('Frequency Response of Input Voltage - Cn = 0.00001')

figure;
plot(fshift, fftVout);
xlabel('Frequency (Hz)')
title('Frequency Response of Output Voltage - Cn = 0.00001')
```


Input Voltage Over Time - Cn = 0.00001
Input Voltage (Volts)
Time (s)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Output Voltage over Time - Cn = 0.00001
Output Voltage (Volts)
Time (s)
-1
0
1
2
3
4
5
6
7
8
9
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

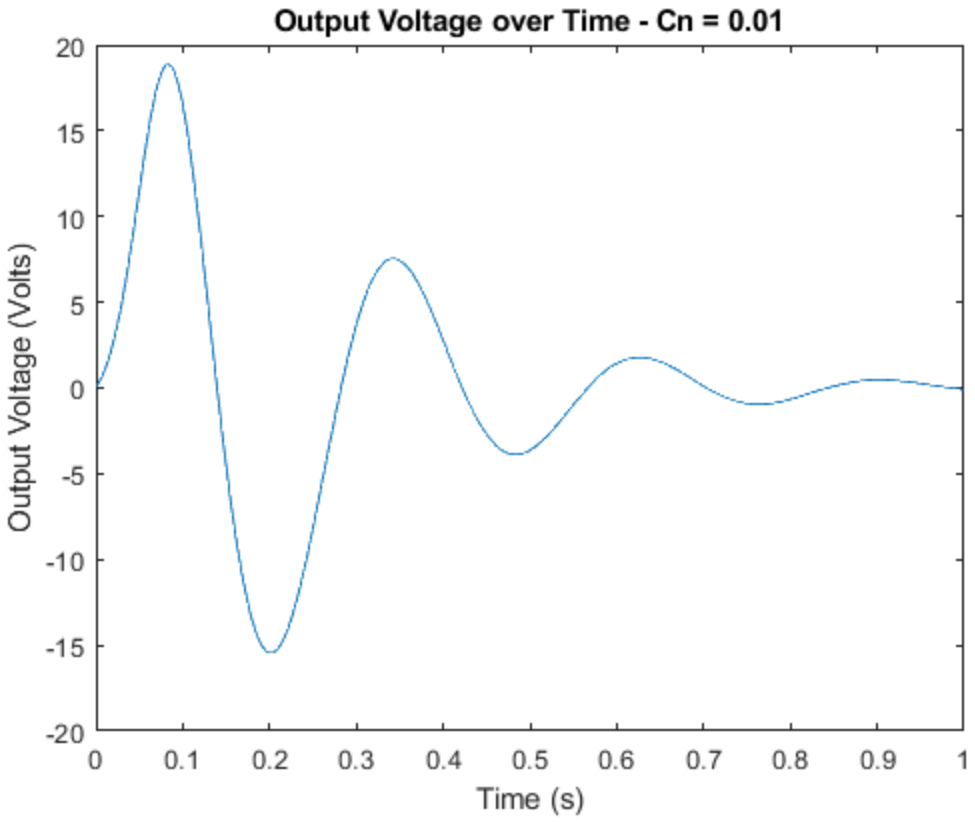

Frequency Response of Input Voltage - Cn = 0.00001
80
70
60
50
40
30
20
10
0
-500
-400
-300
-200
-100
0
100
200
300
400
500
Frequency (Hz)

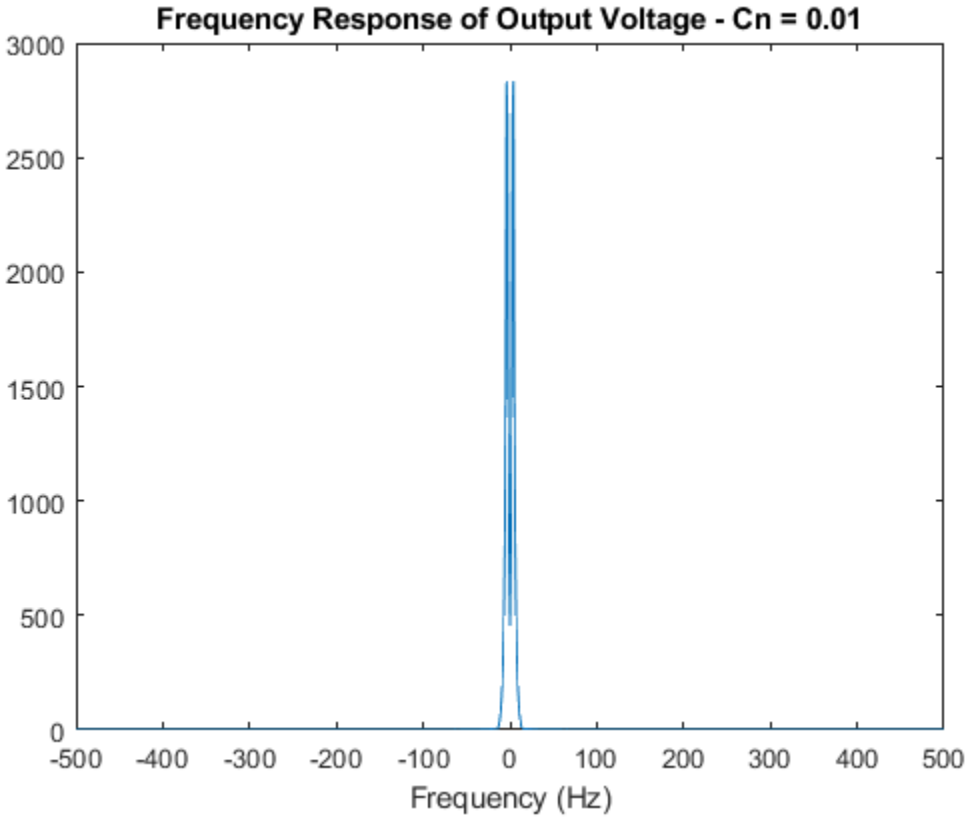

Frequency Response of Output Voltage - Cn = 0.00001
700
600
500
400
300
200
100
0
-500
-400
-300
-200
-100
0
100
200
300
400
500
Frequency (Hz)

# E

Obtain 2nd plot of Vout. Use Cn=0.01.

```
C(3,3)=0.01;
C(4,4)=0.01;
C(3,4)=-0.01;
C(4,3)=-0.01;

Xprev=zeros(10,1);
b(7) = vInput(1);
for count = 1:1000
    bNext = b;
    bNext(3)=0.005*randn();
    bNext(4)=-bNext(3);
    bNext(7) = vInput(count);
    Xnext = (G+(2*C/h))\((2*C/h - G)*Xprev+b+bNext);
    vOut(count) = Xnext(6);

    b = bNext;
    Xprev = Xnext;
end

fftVin = abs(fftshift(fft(vInput)));
fftVout = abs(fftshift(fft(vOut)));
n=length(fftVin);
fs=1/h;
fshift=(-n/2:n/2-1)*(fs/n);

figure;
plot(linspace(0,1,1000),vOut)
xlabel('Time (s)')
ylabel('Output Voltage (Volts)')
title('Output Voltage over Time - Cn = 0.01')

figure;
plot(fshift, fftVout);
xlabel('Frequency (Hz)')
title('Frequency Response of Output Voltage - Cn = 0.01')
```

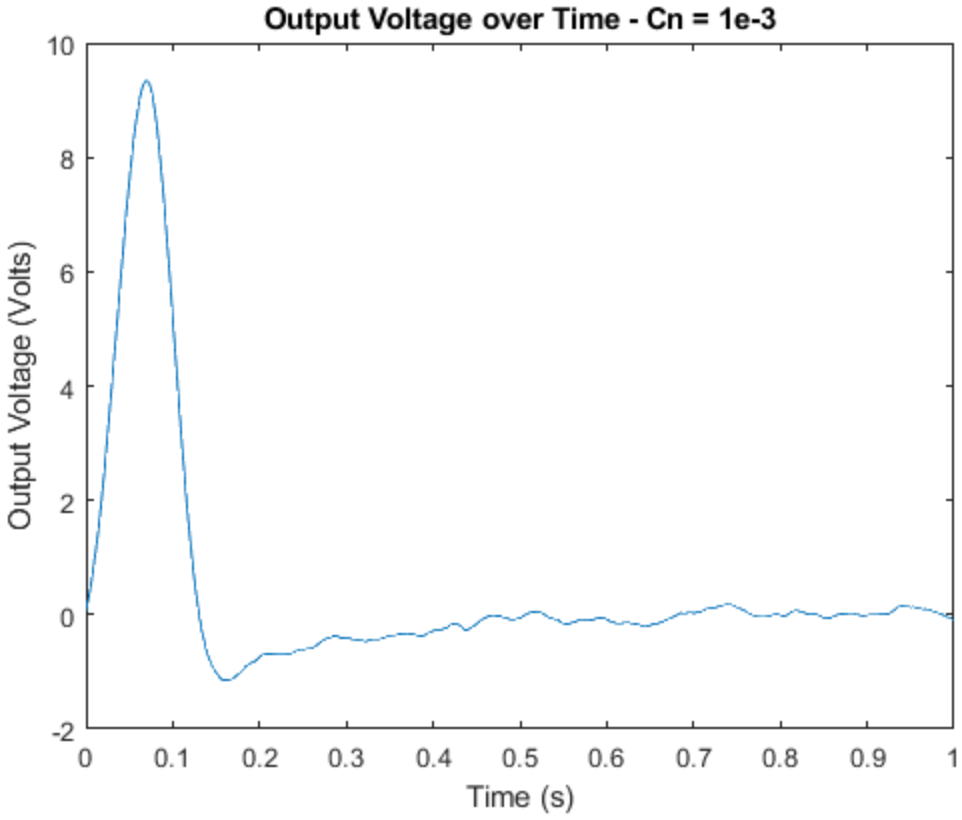

Output Voltage over Time - Cn = 0.01
Output Voltage (Volts)
Time (s)
20
15
10
5
0
-5
-10
-15
-20
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

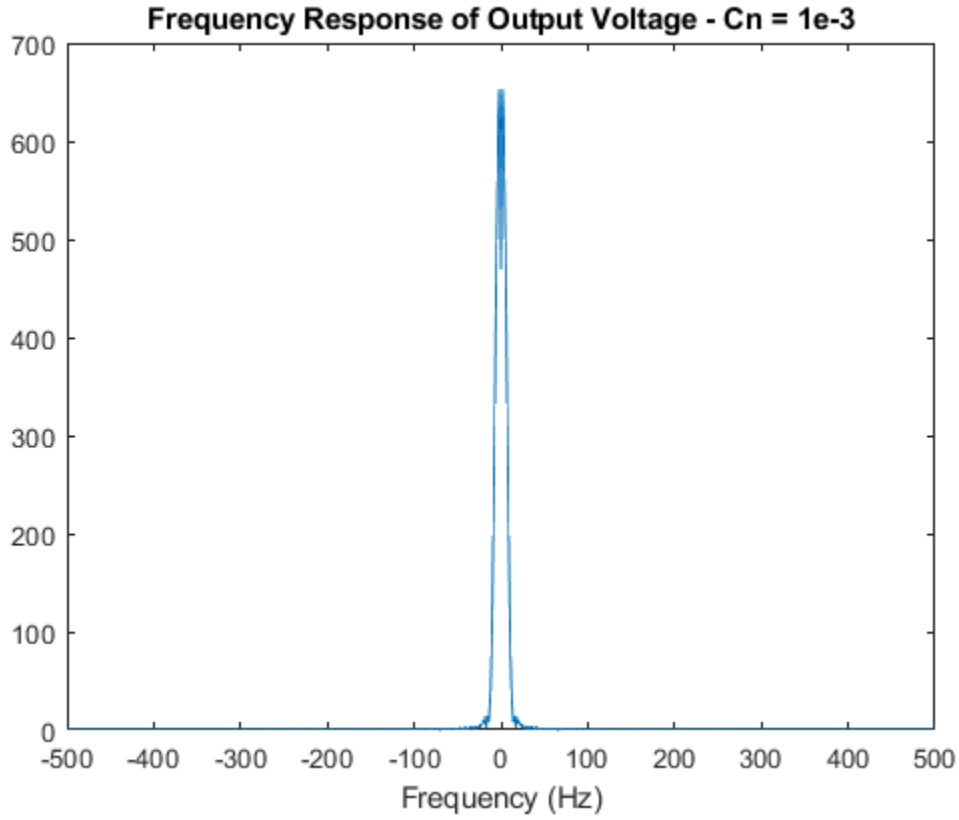

Frequency Response of Output Voltage - Cn = 0.01
Frequency (Hz)
3000
2500
2000
1500
1000
500
0
-500
-400
-300
-200
-100
0
100
200
300
400
500

# E (2)

Obtain 3rd plot of Vout. Use Cn = 1e-3.

```
C(3,3)=1e-3;
C(4,4)=1e-3;
C(3,4)=-1e-3;
C(4,3)=-1e-3;

Xprev=zeros(10,1);
b(7) = vInput(1);
for count = 1:1000
    bNext = b;
    bNext(3)=0.005*randn();
    bNext(4)=-bNext(3);
    bNext(7) = vInput(count);
    Xnext = (G+(2*C/h))\((2*C/h - G)*Xprev+b+bNext);
    vOut(count) = Xnext(6);

    b = bNext;
    Xprev = Xnext;
end

fftVin = abs(fftshift(fft(vInput)));
fftVout = abs(fftshift(fft(vOut)));
n=length(fftVin);
fs=1/h;
fshift=(-n/2:n/2-1)*(fs/n);

figure;
plot(linspace(0,1,1000),vOut)
xlabel('Time (s)')
ylabel('Output Voltage (Volts)')
title('Output Voltage over Time - Cn = 1e-3')

figure;
plot(fshift, fftVout);
xlabel('Frequency (Hz)')
title('Frequency Response of Output Voltage - Cn = 1e-3')
```


Output Voltage over Time - Cn = 1e-3
Output Voltage (Volts)
Time (s)
10
8
6
4
2
0
-2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Frequency Response of Output Voltage - Cn = 1e-3
Frequency (Hz)
700
600
500
400
300
200
100
0
-500
-400
-300
-200
-100
0
100
200
300
400
500

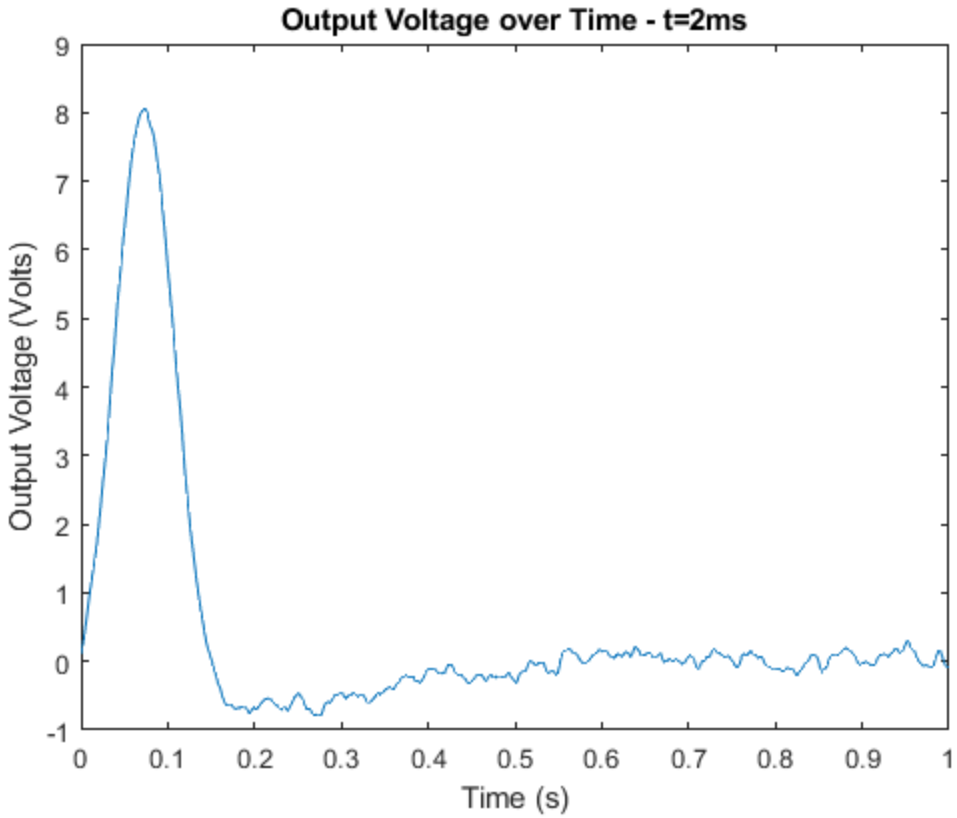
As can be seen in the plots above, the thermal noise is able to be seen. As Cn increases, the bandwidth of the noise decreases.

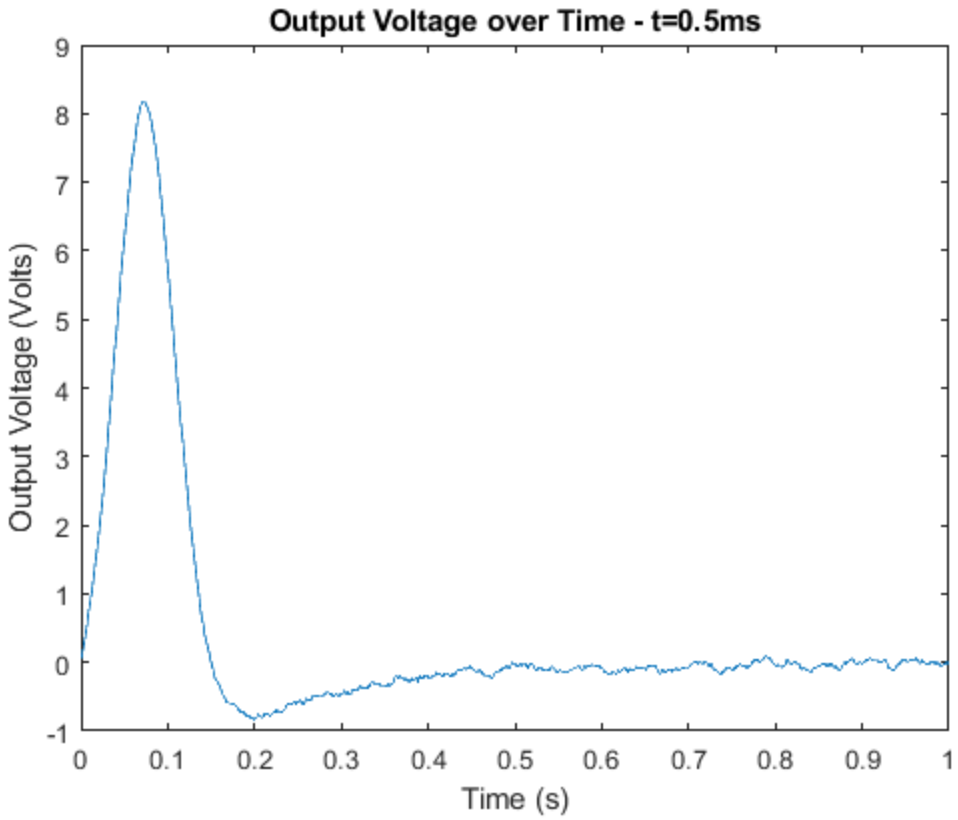
# F

Increase the time step.

```
h=1/500;
vInput = zeros(1/h,1);
vOut = zeros(1/h,1);
for count = 1:length(vOut)
    t=count*h;
    % Gaussian pulse, shifted by 0.06s and compressed to have std
 deviation
    % of 0.03.
    vInput(count) = exp(-0.5*((t-0.06)/0.03)^2);
end

C(3,3)=1e-5;
C(4,4)=1e-5;
C(3,4)=-1e-5;
C(4,3)=-1e-5;

Xprev=zeros(10,1);
b(7) = vInput(1);
for count = 1:length(vOut)
    bNext = b;
    bNext(3)=0.005*randn();
    bNext(4)=-bNext(3);
    bNext(7) = vInput(count);
    Xnext = (G+(2*C/h))\((2*C/h - G)*Xprev+b+bNext);
    vOut(count) = Xnext(6);

    b = bNext;
    Xprev = Xnext;
end

figure;
plot(linspace(0,1,length(vOut)),vOut)
xlabel('Time (s)')
ylabel('Output Voltage (Volts)')
title('Output Voltage over Time - t=2ms')

% Decrease the time step.
h=1/2000;
vInput = zeros(1/h,1);
vOut = zeros(1/h,1);
for count = 1:(1/h)
    t=count*h;
    % Gaussian pulse, shifted by 0.06s and compressed to have std
 deviation
    % of 0.03.
    vInput(count) = exp(-0.5*((t-0.06)/0.03)^2);
```

```
end

Xprev=zeros(10,1);
b(7) = vInput(1);
for count = 1:(1/h)
    bNext = b;
    bNext(3)=0.005*randn();
    bNext(4)=-bNext(3);
    bNext(7) = vInput(count);
    Xnext = (G+(2*C/h))\((2*C/h - G)*Xprev+b+bNext);
    vOut(count) = Xnext(6);

    b = bNext;
    Xprev = Xnext;
end

figure;
plot(linspace(0,1,length(vOut)),vOut)
xlabel('Time (s)')
ylabel('Output Voltage (Volts)')
title('Output Voltage over Time - t=0.5ms')
```






*Published with MATLAB® R2018a*