Assignment 4

Megan McEwen

ELEC 4700

Student # 101003509

5 April 2020

ASSIGNMENT 4 REPORT

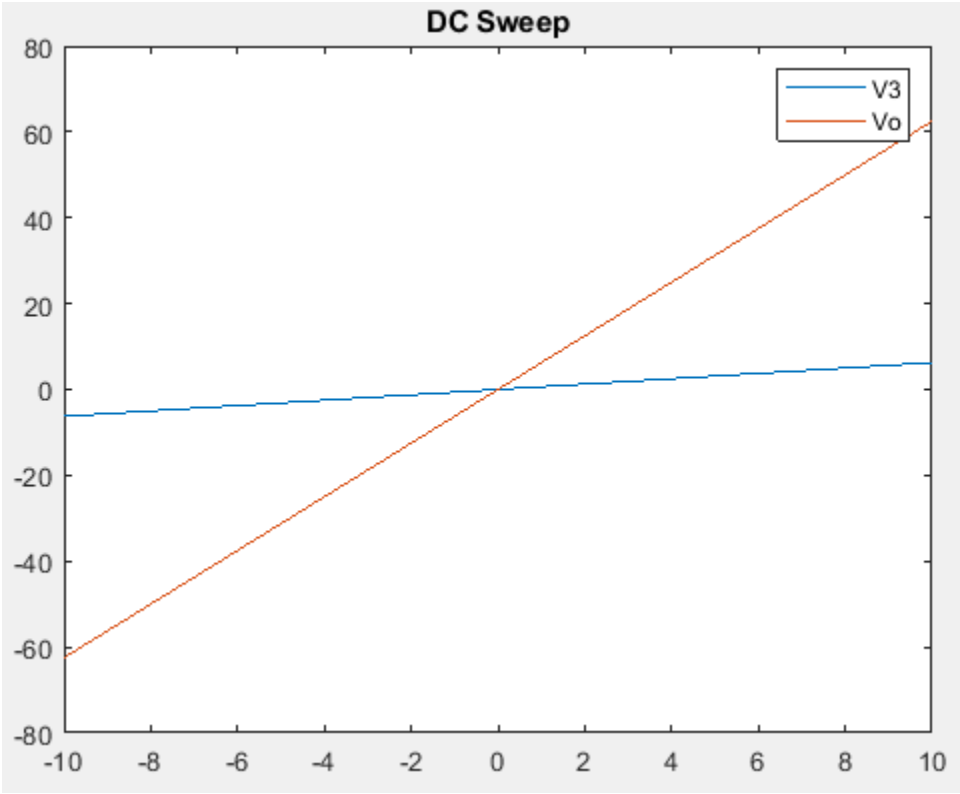
1.The linear fit simulation was not completed as the current sweep of the bottleneck didn't work in Assignment 3, so assume for the rest of the assignment that R3 remains equal to 10.

3. a) From PA7, this circuit can be described using 7 equations.

Based on this, the C, G, and F matrices are then written as shown:

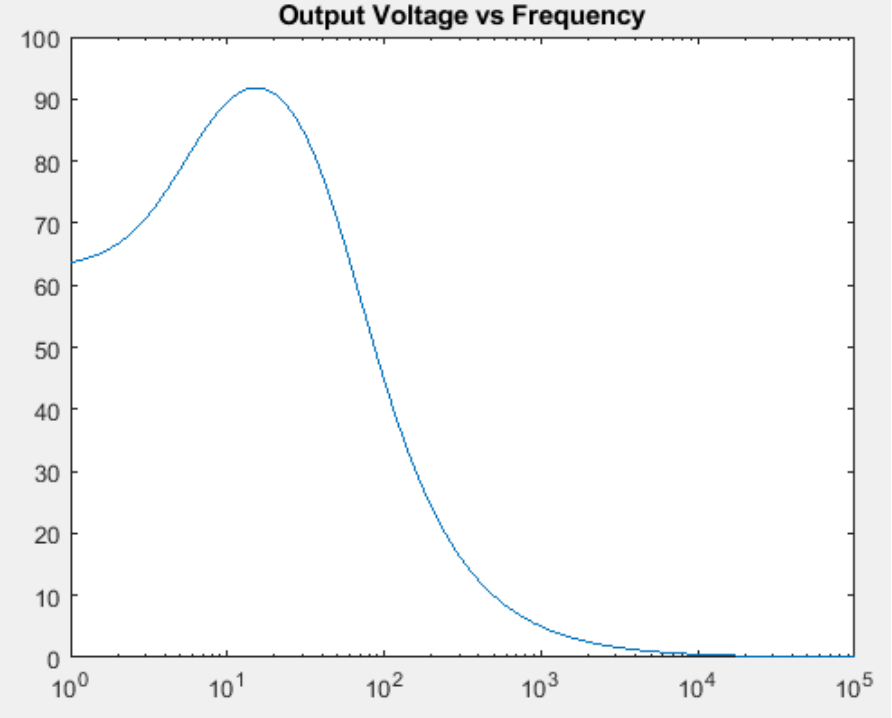
```
C =

    0.2500   -0.2500         0         0         0         0         0
   -0.2500    0.2500         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   -0.2000
         0         0         0         0         0         0         0

G =

    1.0000   -1.0000         0         0         0    1.0000         0
   -1.0000    1.5000         0         0         0         0    1.0000
         0         0    0.1000         0         0         0   -1.0000
         0         0         0    1.0000         0         0 -100.0000
         0         0         0  -10.0000   10.0010         0         0
         0    1.0000   -1.0000         0         0         0         0
    1.0000         0         0         0         0         0         0

F =

     0
     0
     0
     0
     0
     0
     1
```

b) The DC case results showing a sweep from -10 to 10V is shown:



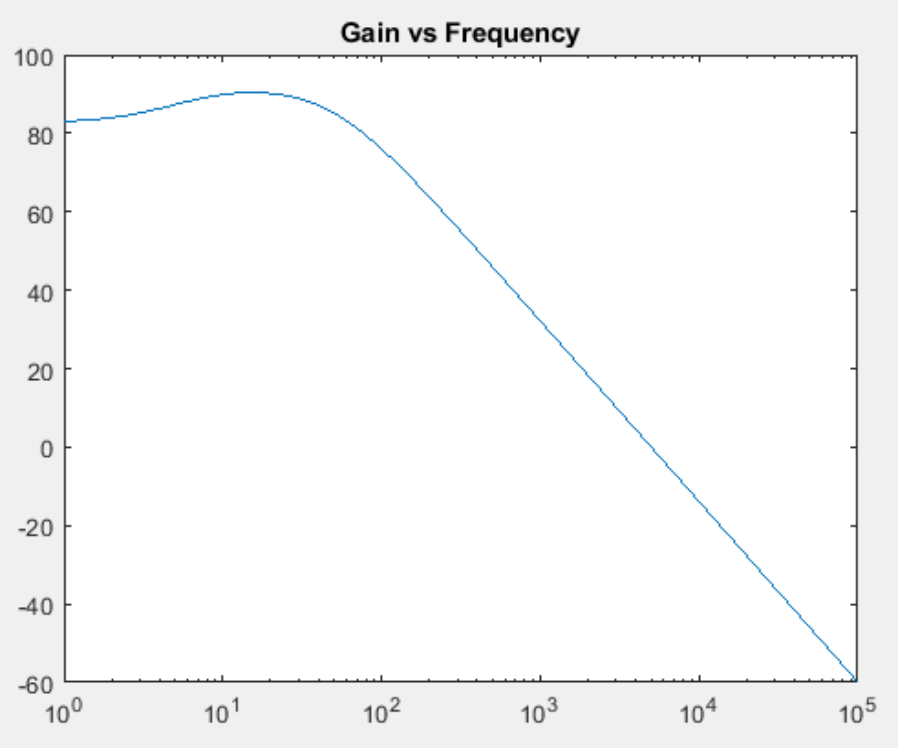
DC Sweep

The AC case showing output voltage vs frequency is shown below:



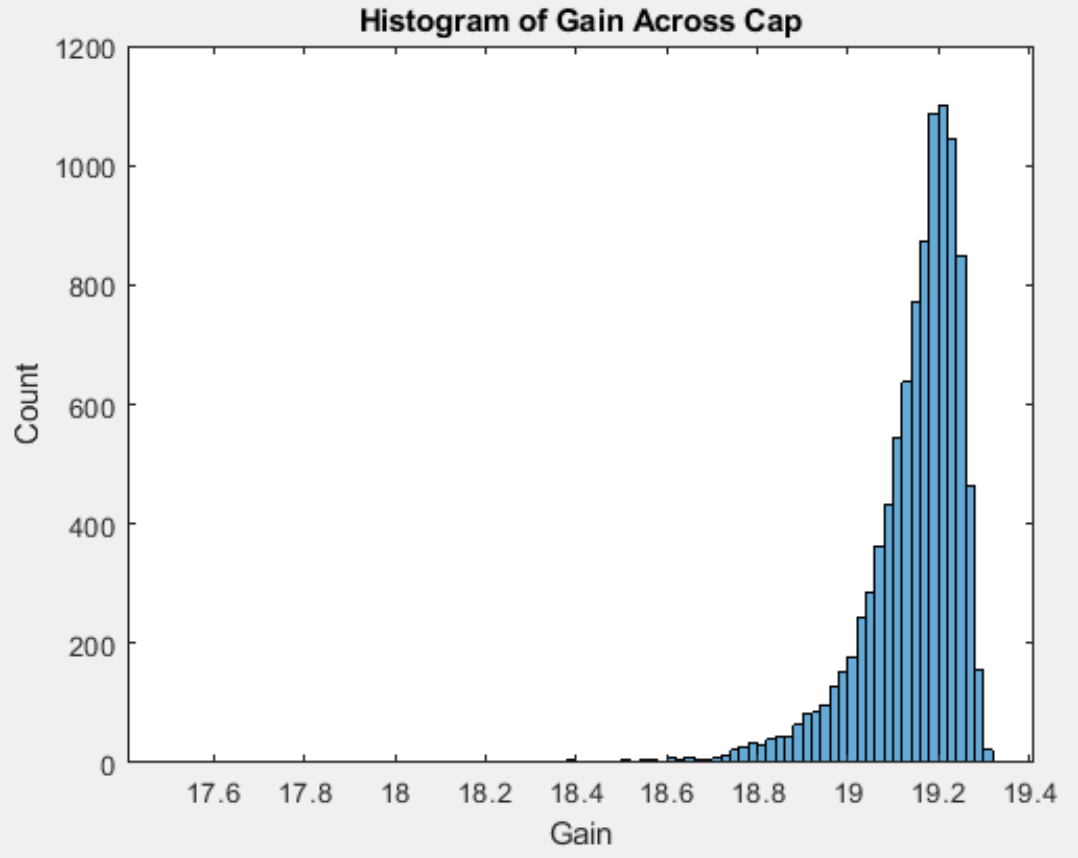
Output Voltage vs Frequency

The AC case showing gain with respect to frequency can be seen below:



Gain vs Frequency

Finally, a histogram showing the variation of gain as a function of C is shown:



Histogram of Gain

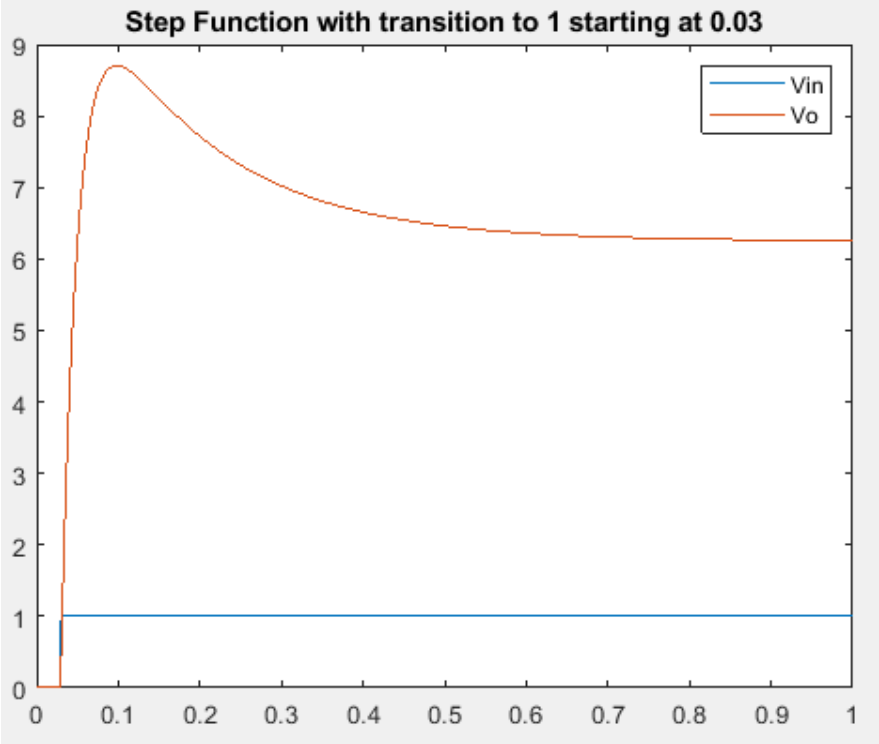
This MNA code is used in subsequent sections to model the circuit behaviour.

4) This section focuses on a transient analysis of the circuit.

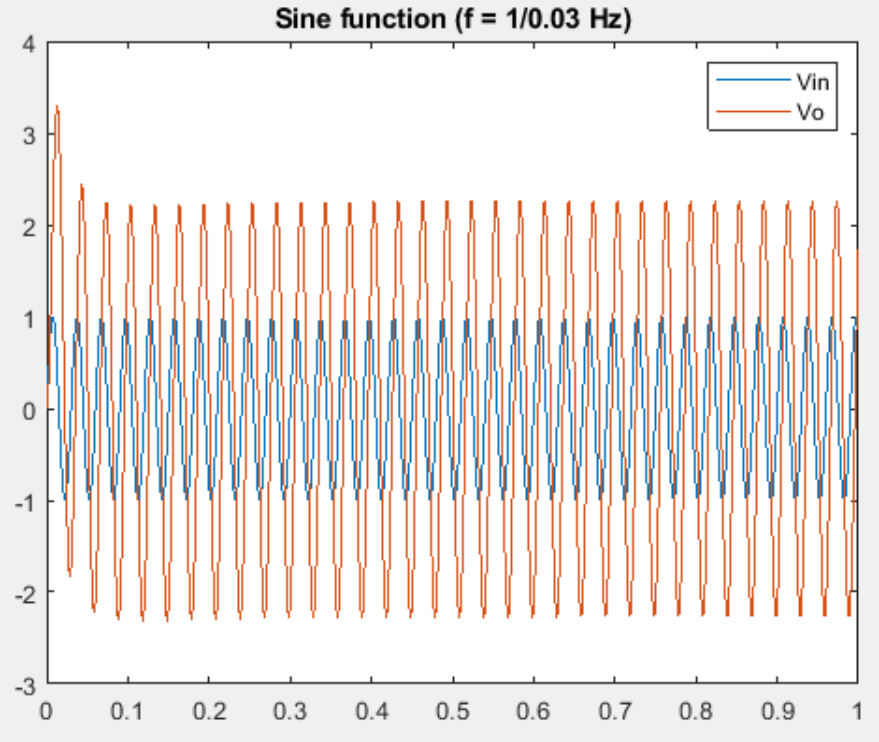
a) By inspection of the gain plot above, this circuit is behaving like a low pass filter.

b) The expected frequency response for low-pass filters is a constant gain at low frequencies, followed by a constant drop in gain starting at a -3dB cutoff point.

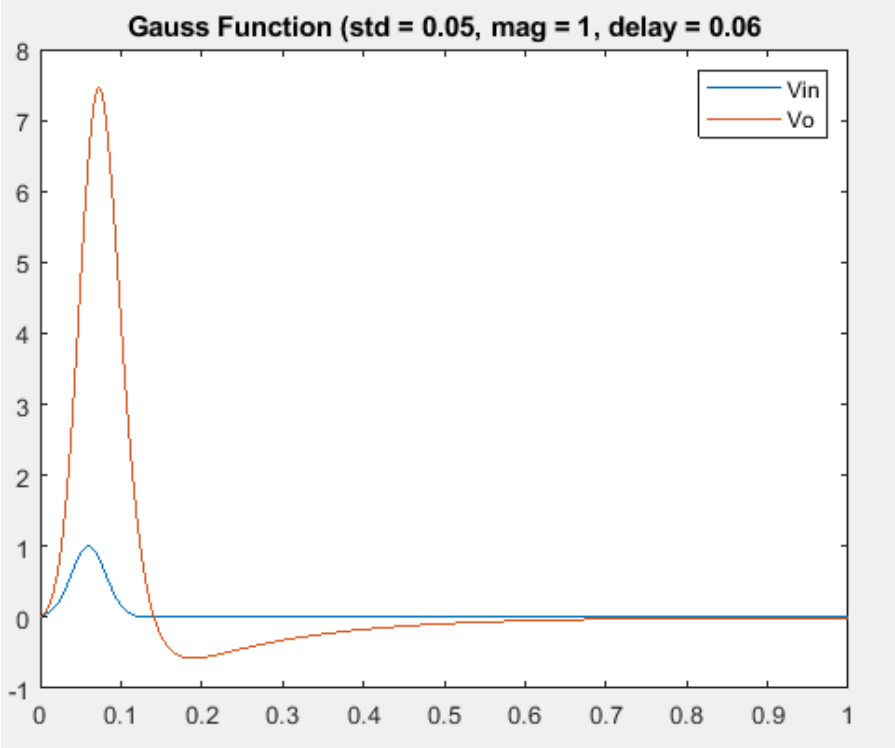
c) Using the finite difference method as described in Assignments 2 and 3, a numerical solution for the circuit is derived using three different input types: a step function, a sine function, and a Gaussian function. The resulting plots of Vin and Vout from these inputs can be seen below:



Vin/Vout with Step Function



Vin/Vout with Sine Function



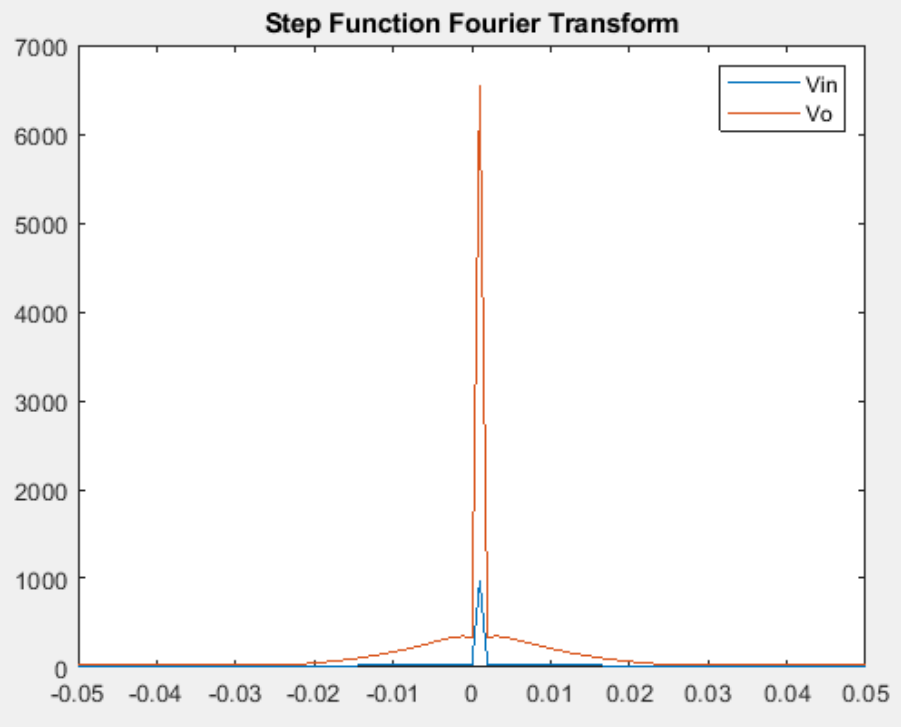
Vin/Vout with Gauss Function

d) The numerical solution with each input type was repeated, but was plotted using fft() to use the Fourier transform. An example of the code setup for the step input is shown:

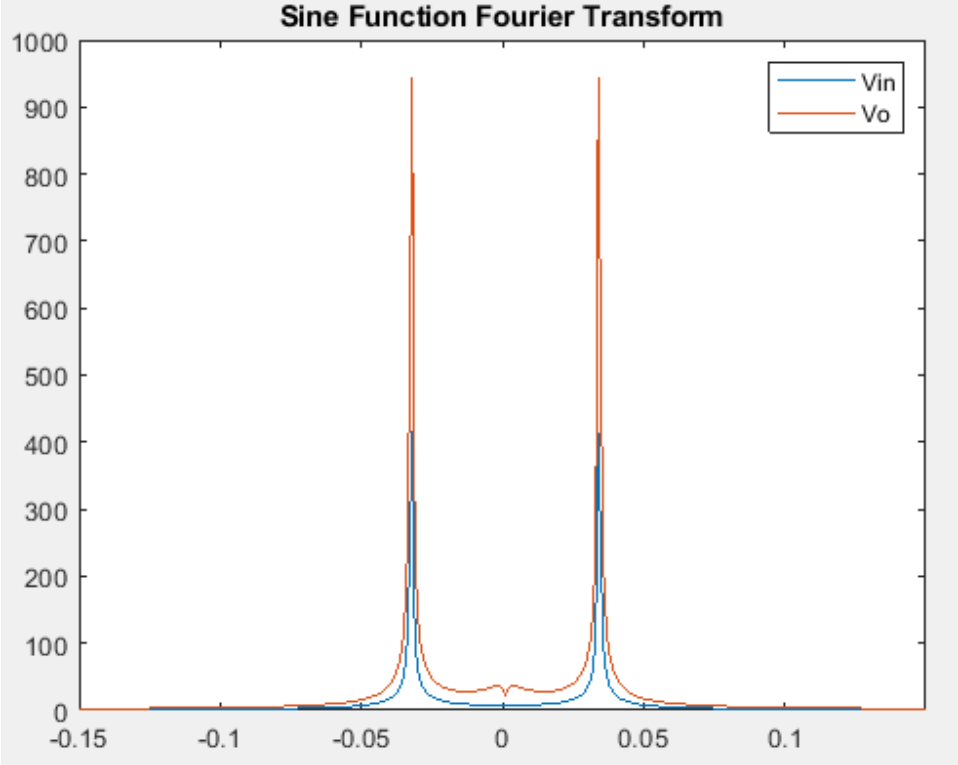
```
%set up stepping of input for V
V_step = zeros(7,1,step); %all inputs are initially 0, set up 3x3 to add in
step
%set up stepping of input for F
F_step = zeros(7,1,step);
%we want to simulate 1000 steps for 1s and change F_step to 1 at 0.03s, so
%.03*1000 = 30 (= first step we need to set to 1).
F_step(7,1,30:step) = 1;

%step through FD solution
for i = 2:1:step %need to start at 2 to get indices for V_step to work
    S = C/del + G;
    A = C*V_step(:,:,i-1)/del+F_step(:,:,i);
    V_step(:,:,i) = S\A; %set up V for each step
end

%extract Vin and Vout
Vo = V_step(5,1,:);
Vin = V_step(1,1,:);

%plot Fourier
figure(2)
ff = abs(fftshift(fft(Vin(1,:))));
FF = abs(fftshift(fft(Vo(1,:))));
plot(((1:length(ff))/step)-0.5,ff)
```

```
hold on
plot(((1:length(FF))/step)-0.5,FF);
title('Step Function Fourier Transform')
legend('Vin','Vo');
xlim([-0.05 0.05]); %zoom in to area
```

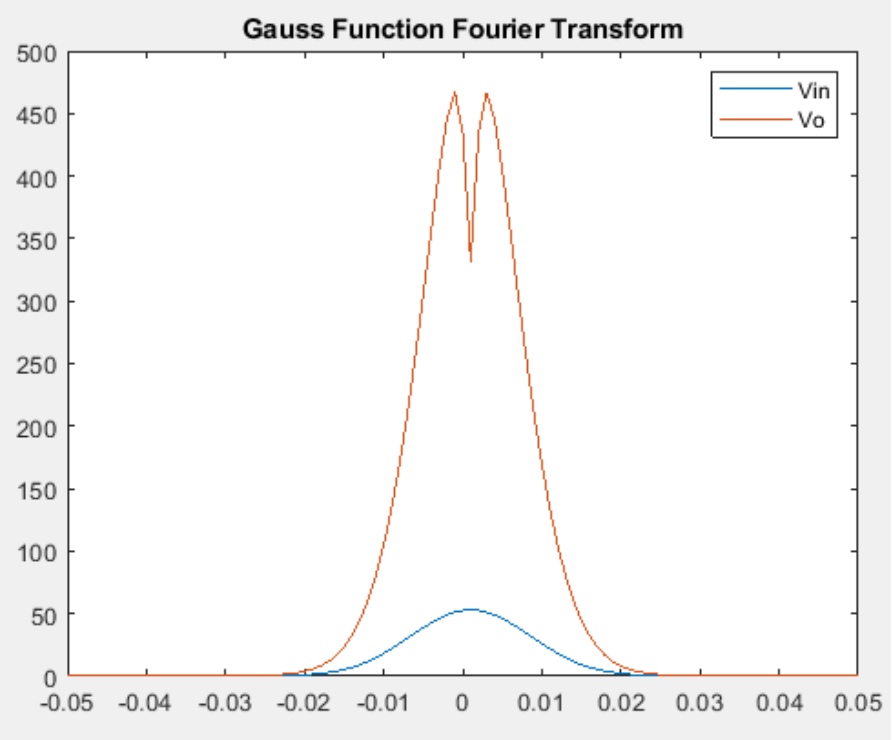
The results can be seen below:



Vin/Vout for Step Function with Fourier



Vin/Vout for Sine Function with Fourier



Vin/Vout for Gauss Function with Fourier

e) As the time step increases, the signal stretches out and the frequency of the signal decreases.

5. This section adds a noise element to the circuit and examines how noise impacts the signal.

a) The addition of a current source and a new capacitor Cn changes the C matrix. The updated matrices are shown below:

```
C =

    0.2500   -0.2500         0         0         0         0         0
   -0.2500    0.2500         0         0         0         0         0
         0         0    0.0001         0         0         0         0
         0         0         0         0         0         0         0
         0         0         0         0         0         0   -0.2000
         0         0         0         0         0         0         0
         0         0         0         0         0         0         0

G =

    1.0000   -1.0000         0         0         0    1.0000         0
   -1.0000    1.5000         0         0         0         0    1.0000
         0         0    0.1000         0         0         0   -1.0000
         0         0         0    1.0000         0         0 -100.0000
         0         0         0  -10.0000   10.0010         0         0
         0    1.0000   -1.0000         0         0         0         0
    1.0000         0         0         0         0         0         0
```

b) In is simulated using a random number function:

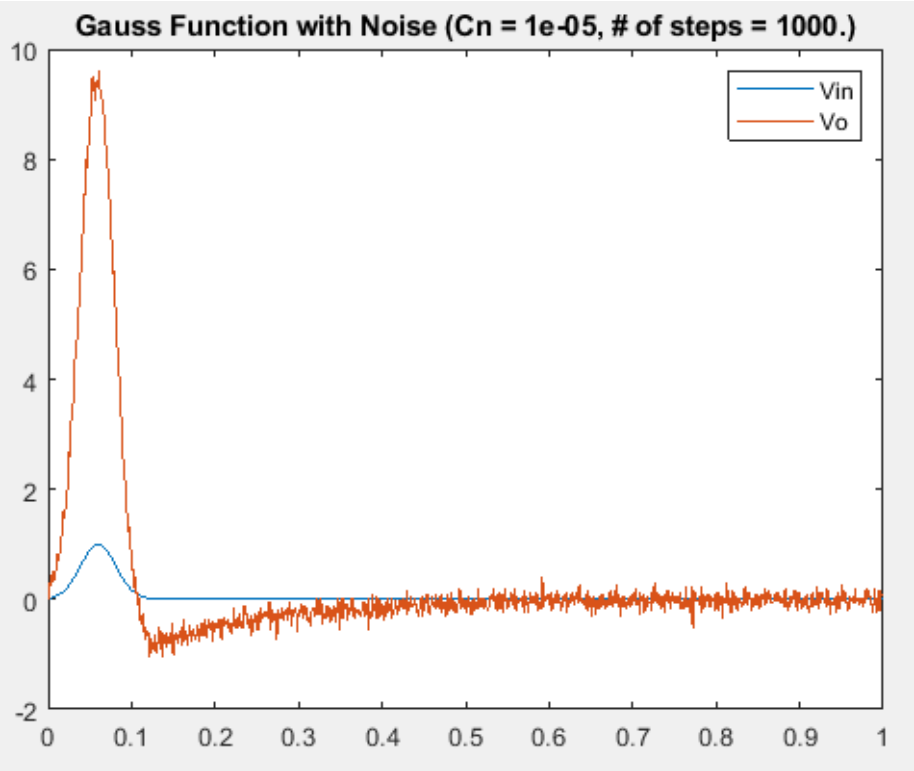
```
%set up F with Gaussian and random noise
for i = 1:1:step
    F_gauss(3,1,i) = In*randn;
    F_gauss(7,1,i) = exp(-1*((del*i - delay)/std)^2); %sine population
End
```

A Gaussian distribution is also set up and plotted using the finite difference method:

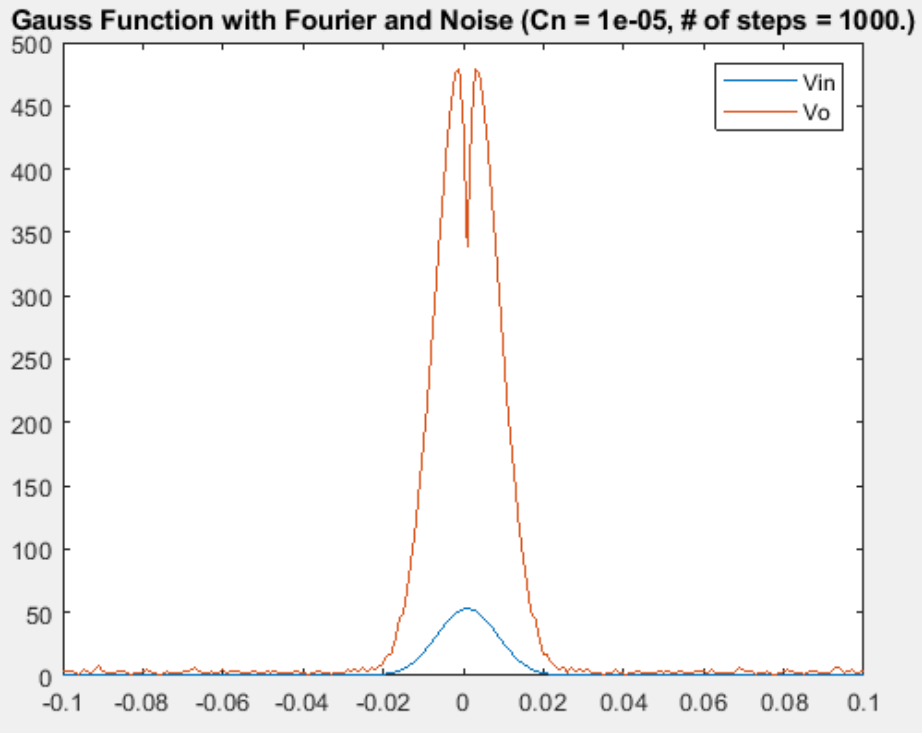
```
%step through FD solution
for i = 2:1:step %need to start at 2 to get indices for V_step to work
    S = C/del + G;
    A = C*V_gauss(:,:,i-1)/del+F_gauss(:,:,i);
    V_gauss(:,:,i) = S\A; %set up V for each step
end

%extract Vin and Vout
Vo = V_gauss(5,1,:);
Vin = V_gauss(1,1,:);

%plot Gauss function and Fourier
figure(1)
plot((1:step).*del, Vin(1,:))
```

```
hold on
plot((1:step).*del, Vo(1,:))
title('Gauss Function (std = 0.05, mag = 1, delay = 0.06')
legend('Vin','Vo')

figure(2)
ff = abs(fftshift(fft(Vin(1,:))));
FF = abs(fftshift(fft(Vo(1,:))));
plot(((1:length(ff))/step)-0.5,ff)
hold on
plot(((1:length(FF))/step)-0.5,FF);
title('Gauss Function Fourier Transform')
legend('Vin','Vo');
xlim([-0.1 0.1]); %zoom in to area
```

Plots of the new output Vo and the Fourier transform result can be seen below:

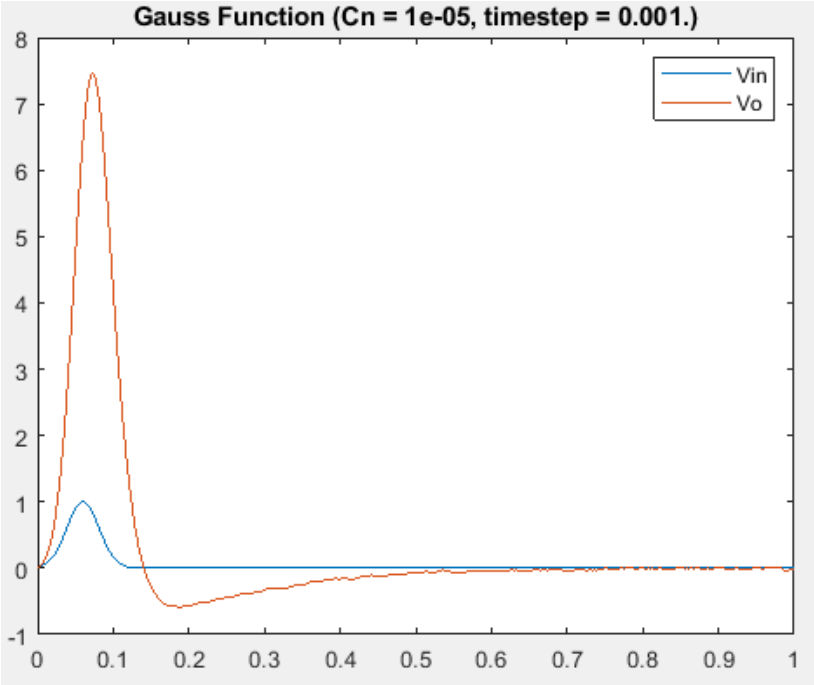


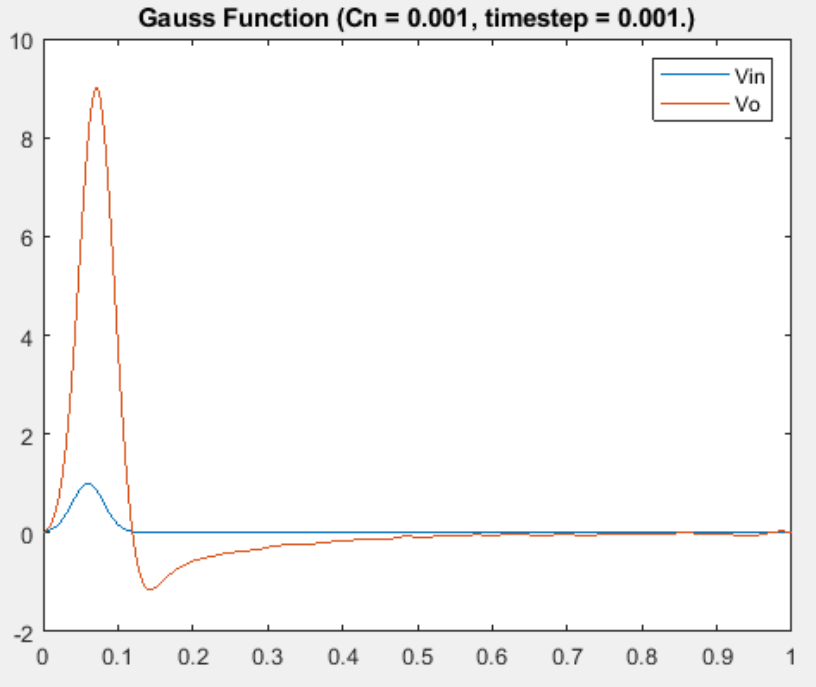
Vout with Gauss Input and Noise

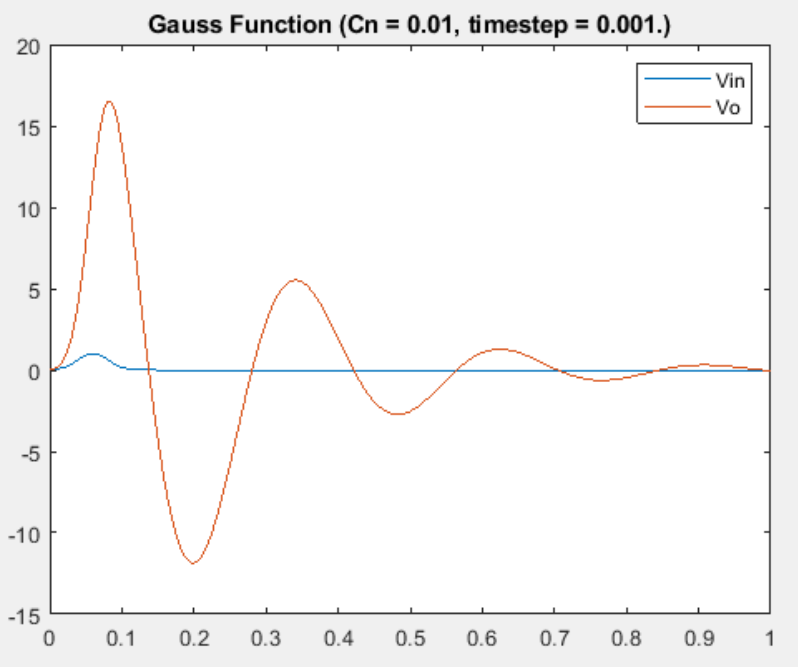


Vout with Fourier and Noise

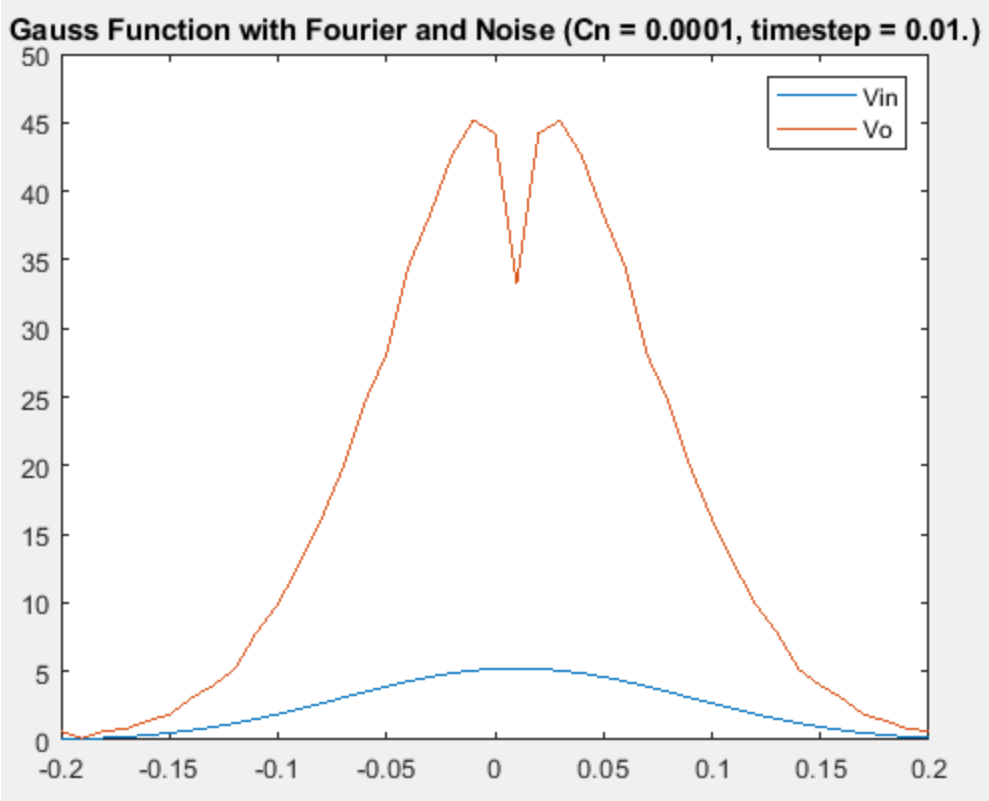
c) As the value of Cn is varied, the bandwidth of the signal varies. The figures below show how changing Cn affects bandwidth:

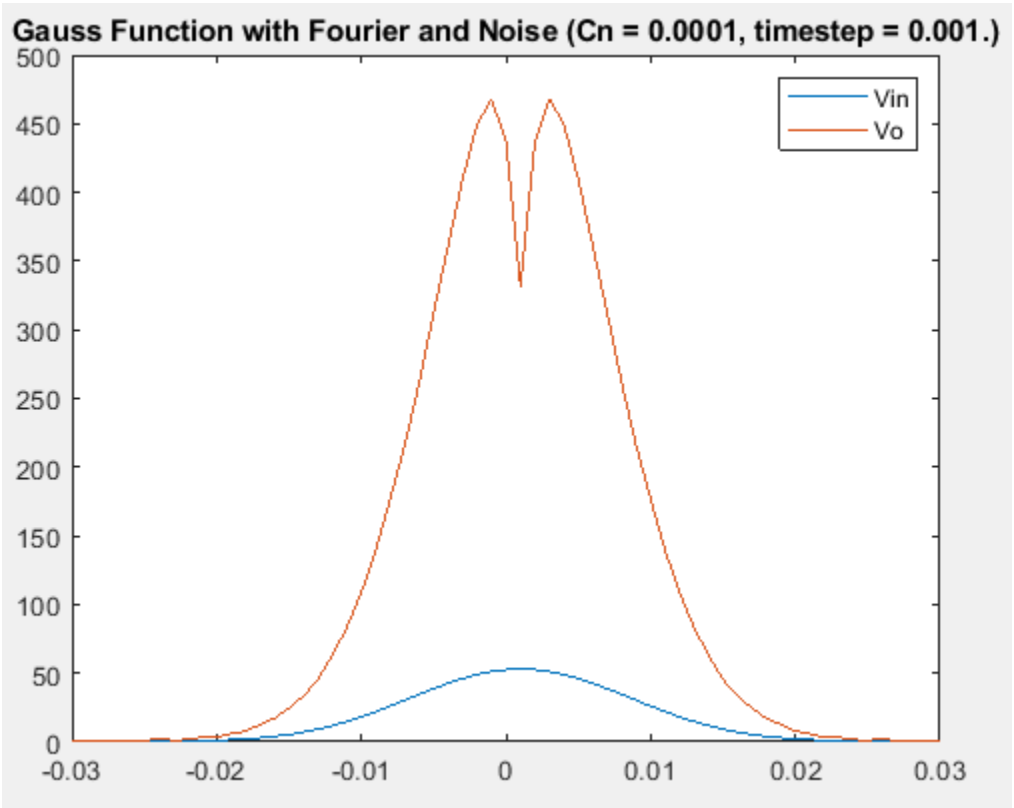


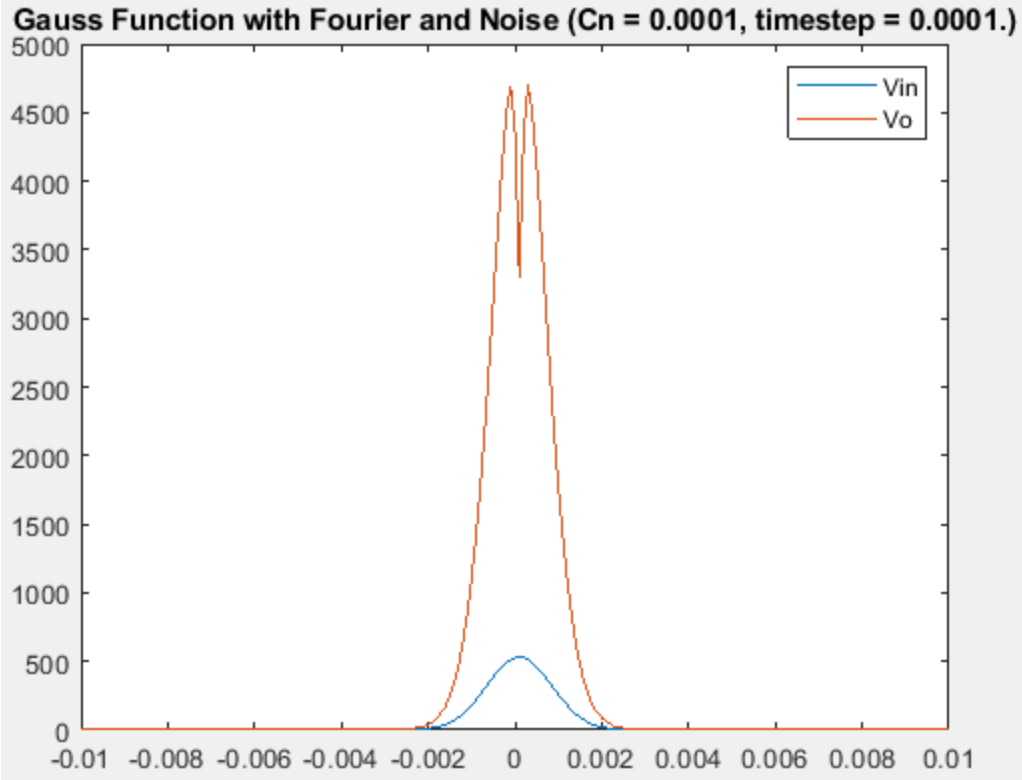




d) As the timestep is varied, the Fourier signal bandwidth shrinks proportionally, as shown:


Gauss Function with Fourier and Noise (Cn = 0.0001, timestep = 0.01.)
Vin
Vo
50
45
40
35
30
25
20
15
10
5
0
-0.2
-0.15
-0.1
-0.05
0
0.05
0.1
0.15
0.2


Gauss Function with Fourier and Noise (Cn = 0.0001, timestep = 0.001.)
Vin
Vo
500
450
400
350
300
250
200
150
100
50
0
-0.03
-0.02
-0.01
0
0.01
0.02
0.03


Gauss Function with Fourier and Noise (Cn = 0.0001, timestep = 0.0001.)
Vin
Vo
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
-0.01
-0.008
-0.006
-0.004
-0.002
0
0.002
0.004
0.006
0.008
0.01