Table of Contents

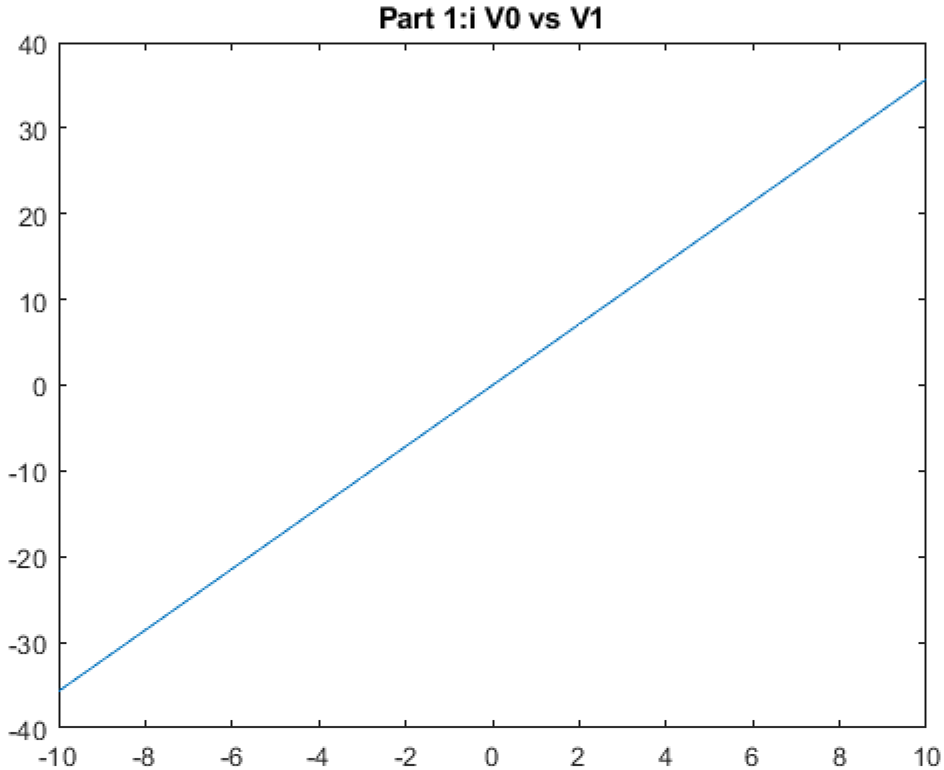

```
% Assignment 4
% By: Chantel Lepage 100999893

clear
clc
close all
```

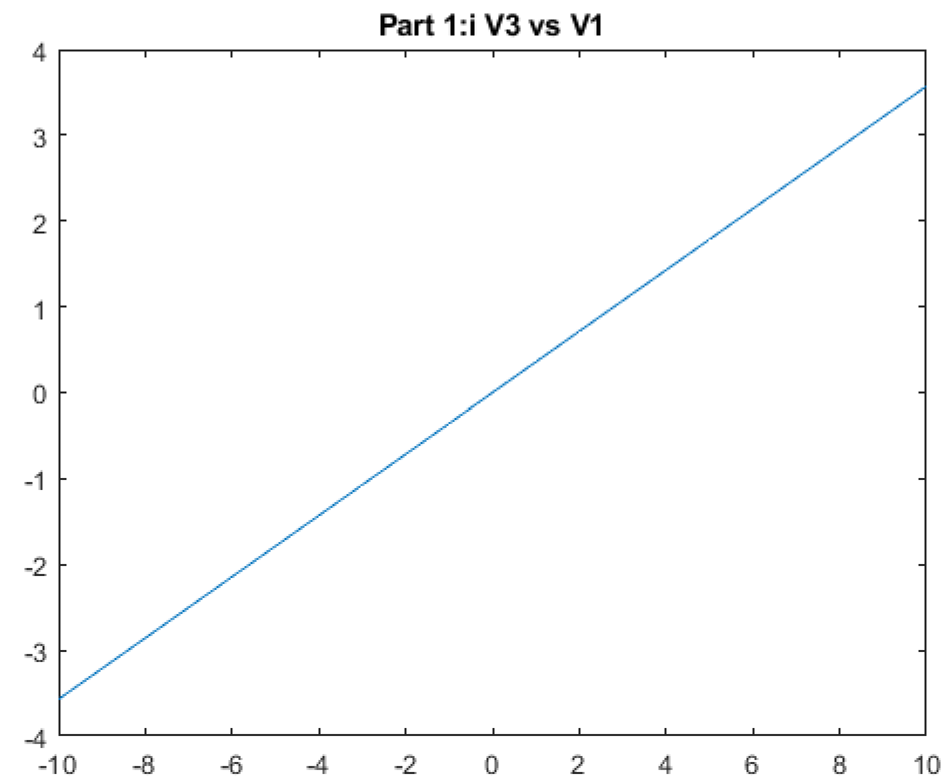
Part 1

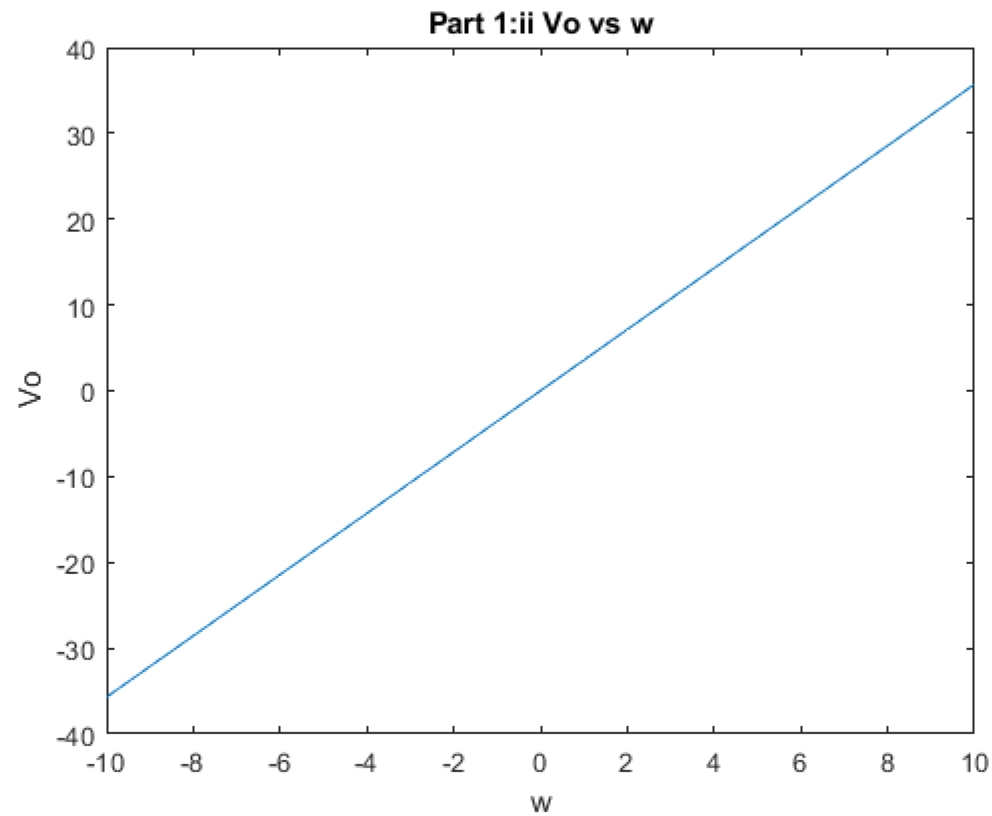
```
R = [1, 2, 10, 0.1, 1000];
CVals = 0.25;
LVal = 0.2;
alpha = 100;
omega = 0;

G = zeros(7);

%Row 1
G(1,1) = 1;

%Row 2
G(2,1) = -1./R(2);
G(2,2) = (1./R(1))+(1./R(2));
G(2,3) = -1;

%Row 3
G(3,2) = 1;
G(3,4) = -1;

%Row 4
G(4,3) = -1;
G(4,4) = 1./R(3);

%Row 5
G(5,5) = -alpha;
G(5,6) = 1;
```

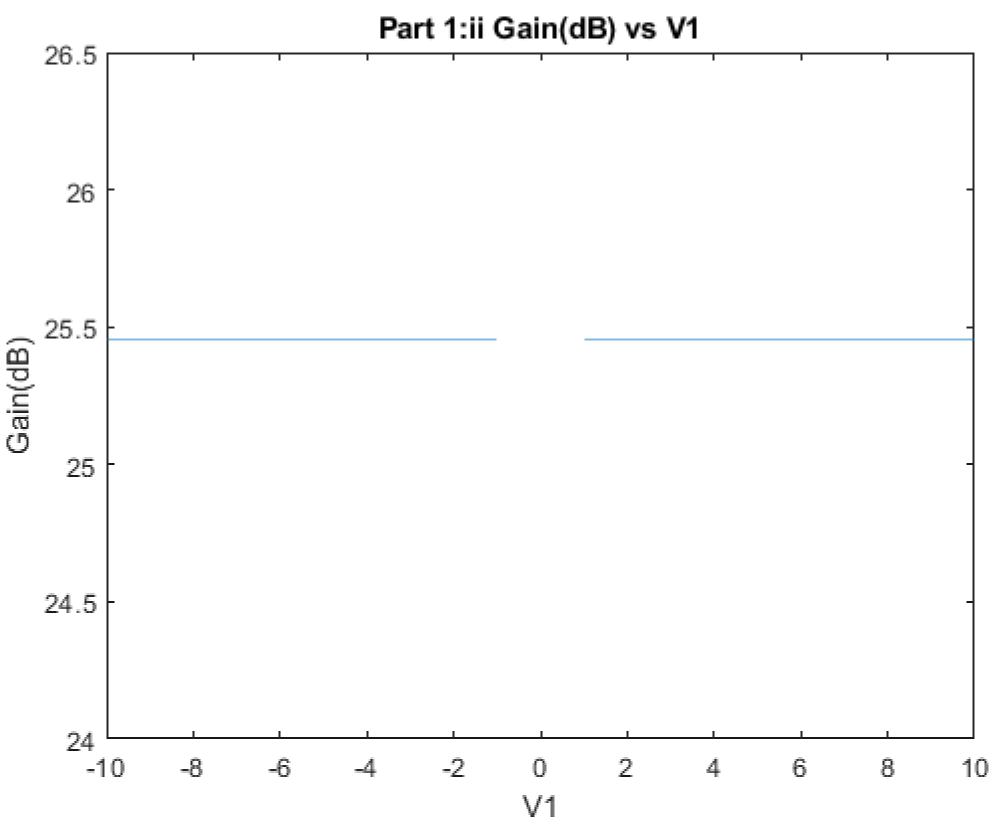
```
%Row 6
G(6,4) = 1./R(3);
G(6,5) = -1;

%Row 7
G(7,6) = -1./R(4);
G(7,7) = (1./R(4))+(1./R(5));


%C matrix
C = zeros(7);

%Row 1
C(2,1) = -CVals;

%Row 2
C(2,2) = CVals;

%Row 3
C(3,3) = -LVal;


%V = [V1; V2; IL; V3; I3; V4; V0];
V = zeros(1, 7);

%DC Case i
Vin = -10;
F = zeros(7,1);
F(1,1) = Vin;

V1 = zeros(1,21);
V3 = zeros(1,21);
V0 = zeros(1,21);

for i = 1:length(V1)
    V1(i)= Vin;
    F(1,1) = Vin;
    V = G\F;
    V3(i) = V(4);
    V0(i) = V(7);
    Vin = Vin+1;
end

figure(1);
plot(V1, V0);
title('Part 1:i V0 vs V1');

figure(2);
plot(V1, V3);
title('Part 1:i V3 vs V1');

%AC Case ii
```

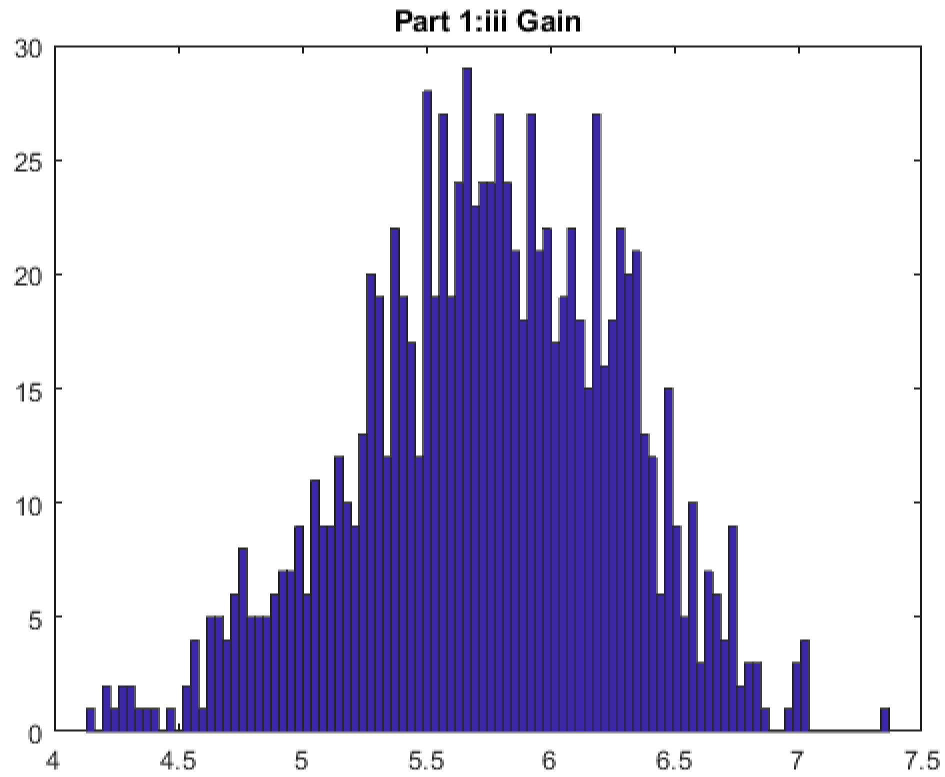
```
for w= -10:10
    F(1,1) = w;
    V = G\F;
    V0(i) = V(7);
end
figure(3);
w = -10 : 10;
plot(w, V0);
title('Part 1:ii Vo vs w');
xlabel('w');
ylabel('Vo');


gain = V0./V1;
gaindB = 20*log(abs(gain));

figure(4);
plot(V1, gaindB);
title('Part 1:ii Gain(dB) vs V1');
xlabel('V1');
ylabel('Gain(dB)');

%AC Case with random perturbations

Cstd =  0.25 + 0.05.*randn(1,1000);
w = pi;

gain = zeros(1000,1);

for m = 1:length(gain)
    c = Cstd(m);
    C(2,1) = -c;
    C(2,2) = c;
    V = (G+C*1j*w)\F;
    gain(m,1) = abs(V(7,1))/F(1);
end


figure (5)
hist(gain,100);
title('Part 1:iii Gain');
```


Part 1:i V0 vs V1
40
30
20
10
0
-10
-20
-30
-40
-10
-8
-6
-4
-2
0
2
4
6
8
10


Part 1:i V3 vs V1
4
3
2
1
0
-1
-2
-3
-4
-10
-8
-6
-4
-2
0
2
4
6
8
10


Part 1:ii Vo vs w
Vo
w
40
30
20
10
0
-10
-20
-30
-40
-10
-8
-6
-4
-2
0
2
4
6
8
10


Part 1:ii Gain(dB) vs V1
Gain(dB)
V1
26.5
26
25.5
25
24.5
24
-10
-8
-6
-4
-2
0
2
4
6
8
10



Part 2

```
clear
clc
```

Part 2A

By inspection this circuit can be determined to be and RLC circuit due to the linear components used.

Part 2B

Part 2D

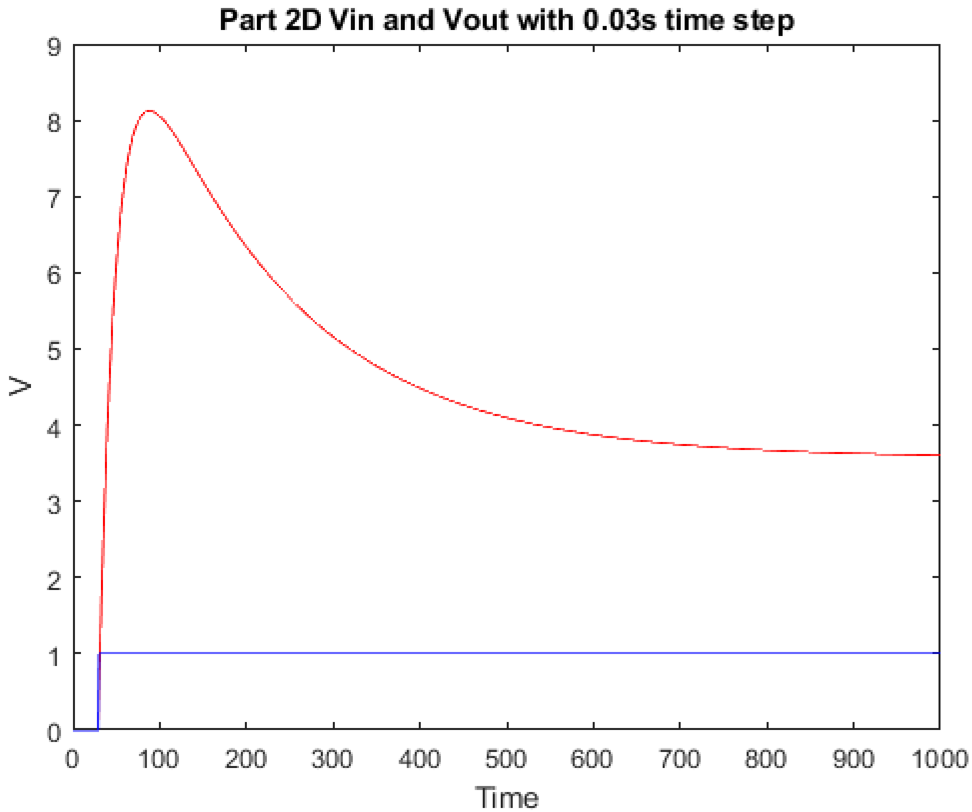
```
R = [1, 2, 10, 0.1, 1000];
CVals = 0.25;
LVal = 0.2;
alpha = 100;
omega = 0;

G = zeros(7);

%Row 1
G(1,1) = 1;
```

```
%Row 2
G(2,1) = -1./R(2);
G(2,2) = (1./R(1))+(1./R(2));
G(2,3) = -1;

%Row 3
G(3,2) = 1;
G(3,4) = -1;

%Row 4
G(4,3) = -1;
G(4,4) = 1./R(3);

%Row 5
G(5,5) = -alpha;
G(5,6) = 1;

%Row 6
G(6,4) = 1./R(3);
G(6,5) = -1;

%Row 7
G(7,6) = -1./R(4);
G(7,7) = (1./R(4))+(1./R(5));


%C matrix
C = zeros(7);

%Row 1
C(2,1) = -CVals;

%Row 2
C(2,2) = CVals;

%Row 3
C(3,3) = -LVal;


Vin = 1;
F = zeros(7,1);
F(1,1) = Vin;


Foff = zeros(7,1);
Foff(1,1) = Vin-Vin;

ts = 1000;

%2D:ii:A 0.03s time step
V1 = zeros(7, ts);
Vstart = zeros(7, 1);
dt=1e-3;
```

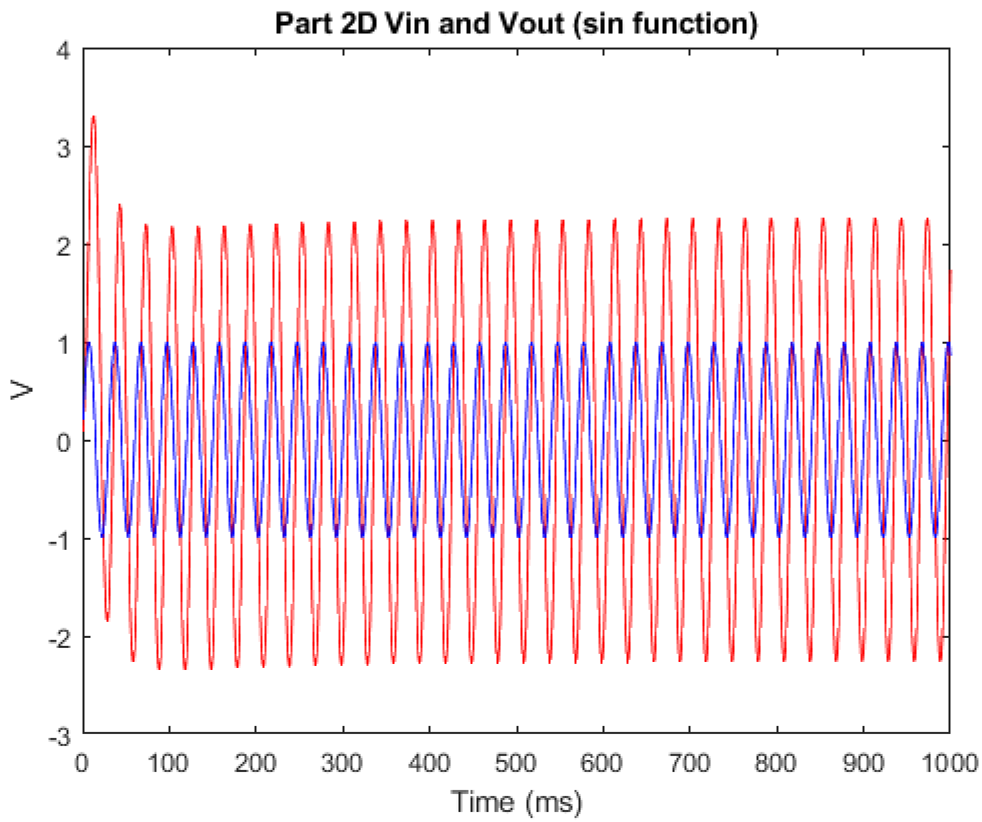
```
for i = 1:ts
    if i < 30
        V1(:,i) = (C./dt+G)\(Foff+C*Vstart/dt);
    elseif i == 30
        V1(:,i) = (C./dt+G)\(F+C*Vstart/dt);
    else
        V1(:,i) = (C./dt+G)\(F+C*Vpast/dt);
    end
    Vpast = V1(:, i);
end

figure(6)
plot(1:ts, V1(7,:), 'r')
hold on
plot(1:ts, V1(1,:), 'b')
title('Part 2D Vin and Vout with 0.03s time step')
xlabel('Time')
ylabel('V')

%2D:ii:B sin function
V2 = zeros(7, ts);
Fsin = zeros(7,1);
for j = 1:ts

    Vsin = sin(2*pi*(1/0.03)*j/ts);
    Fsin(1,1) = Vsin;
    if j == 1
        V2(:,j) = (C./dt+G)\(Fsin+C*Vstart/dt);
    else
        V2(:,j) = (C./dt+G)\(Fsin+C*Vpast/dt);
    end
    Vpast = V2(:, j);

end

figure(7)
plot(1:ts, V2(7,:), 'r')
hold on
plot(1:ts, V2(1,:), 'b')
title('Part 2D Vin and Vout (sin function)')
xlabel('Time (ms)')
ylabel('V')

%2D:ii:C gauss function
V3 = zeros(7, ts);
Fgauss = zeros(7,1);
for k = 1:ts

    Vgauss = exp(-1/2*((k/ts-0.06)/(0.03))^2);
    Fgauss(1,1) = Vgauss;
    if k == 1
        V3(:,k) = (C./dt+G)\(Fgauss+C*Vstart/dt);
    else
```

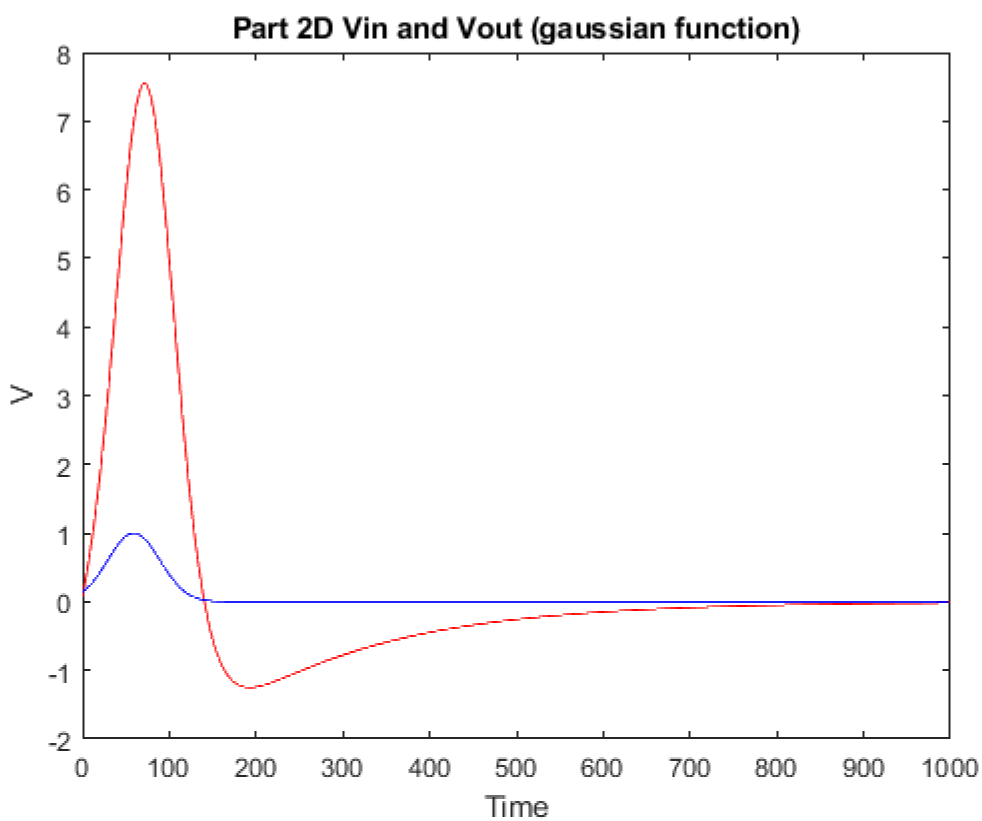
```
        V3(:,k) = (C./dt+G)\(Fgauss+C*Vpast/dt);
    end
    Vpast = V3(:, k);

end

figure(8)
plot(0:ts-1, V3(7,:), 'r')
hold on
plot(0:ts-1, V3(1,:), 'b')
title('Part 2D Vin and Vout (gaussian function)')
xlabel('Time')
ylabel('V')


%%2D:iv
f = (-ts/2:ts/2-1);

%step function
fV1in = fft(V1(1, :));
fV1out = fft(V1(7, :));
fsV1in = fftshift(fV1in);
fsV1out = fftshift(fV1out);
figure(9)
plot(f, abs(fsV1in), 'r')
hold on
plot(f, abs(fsV1out), 'b')
xlim([-150,150]);
title('Vin and Vout frequency domain (step function)')
xlabel('frequency')
ylabel('V')

%sine function
fV2 = fft(V2.');
fsV2 = fftshift(fV2);
figure(10)
plot(f, abs(fsV2(:, 1)), 'r')
hold on
plot(f, abs(fsV2(:, 7)), 'b')
xlim([-150,150]);
title('Vin and Vout frequency domain (sin function)')
xlabel('frequency')
ylabel('V')

%guass function
fV3 = fft(V3.');
fsV3 = fftshift(fV3);
figure(11)
plot(f, abs(fsV3(:, 1)), 'r')
hold on
plot(f, abs(fsV3(:, 7)), 'b')
xlim([-150,150]);
title('Vin and Vout frequency domain (gauss function)')
xlabel('frequency')
```

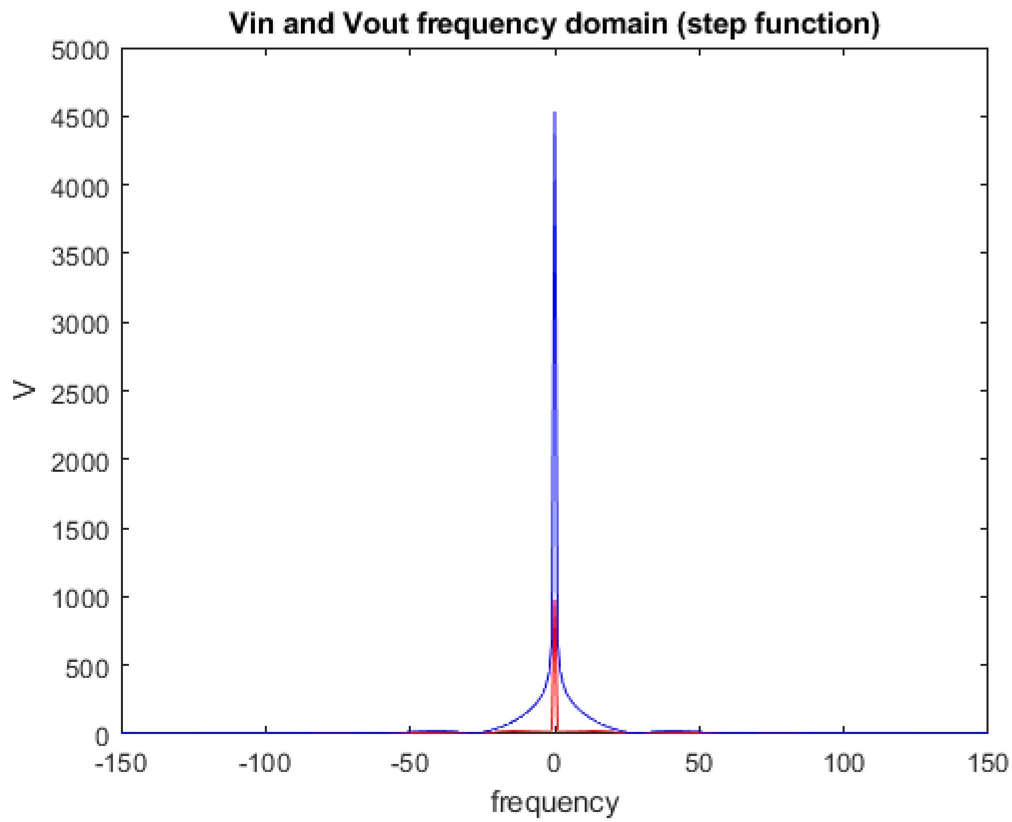
```
ylabel('V')
```

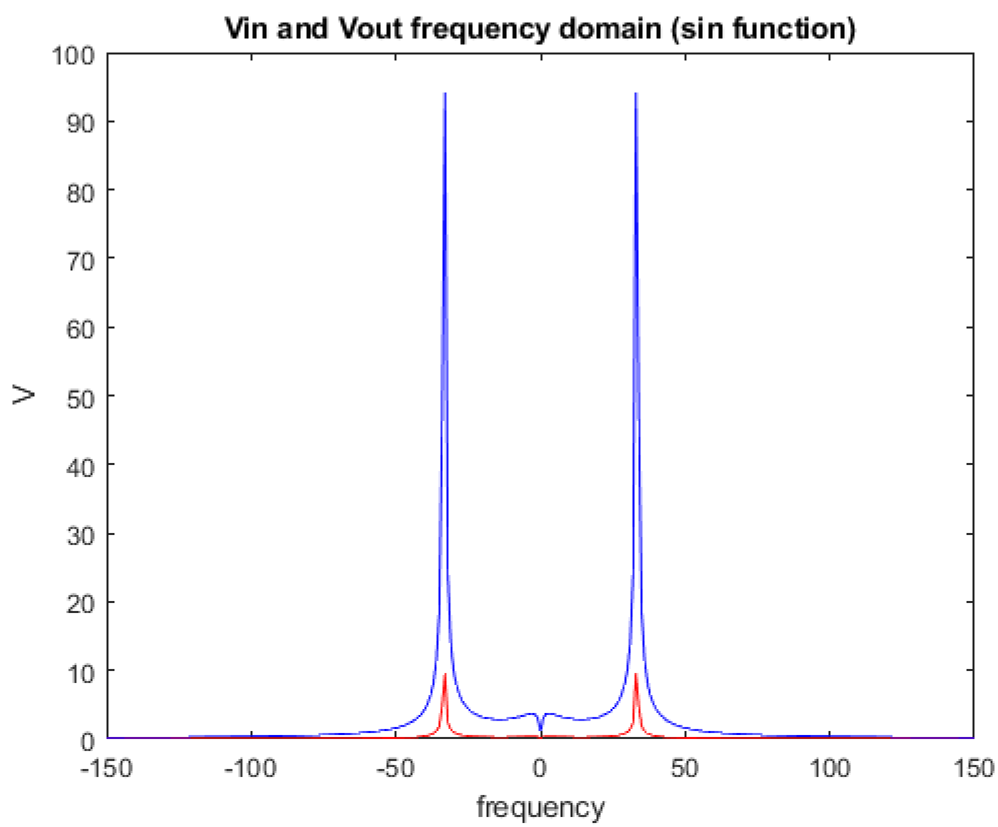


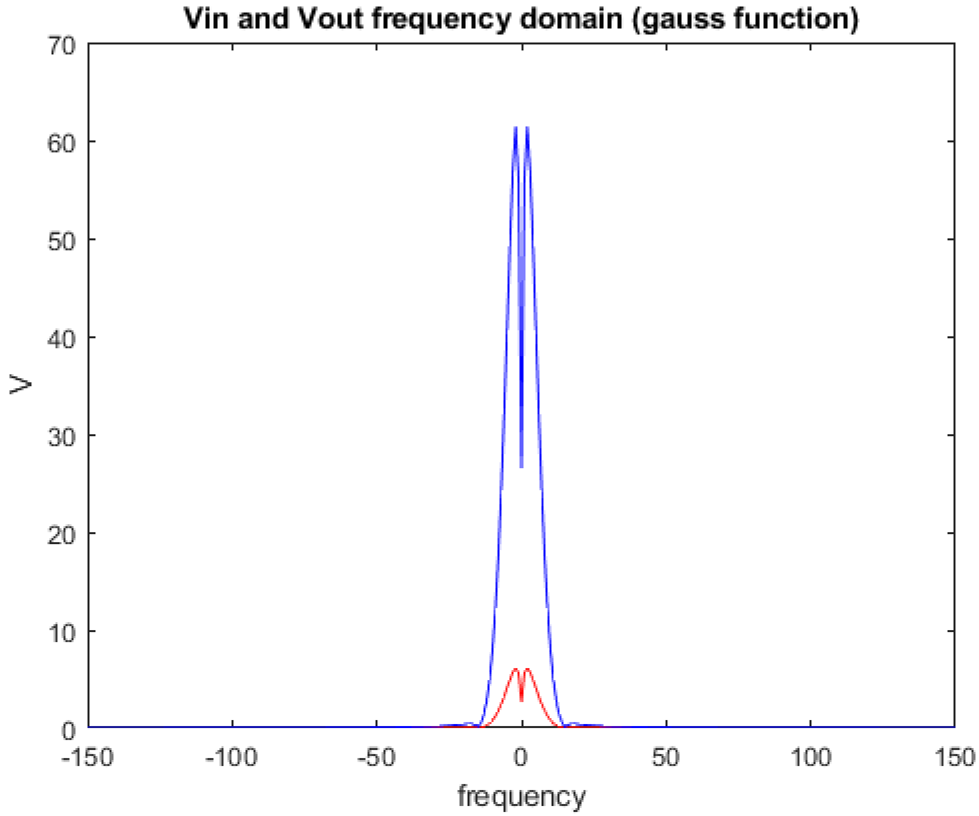
Part 2D Vin and Vout with 0.03s time step
V
Time
0
1
2
3
4
5
6
7
8
9
0
100
200
300
400
500
600
700
800
900
1000


Part 2D Vin and Vout (sin function)
V
Time (ms)
4
3
2
1
0
-1
-2
-3
0
100
200
300
400
500
600
700
800
900
1000


Part 2D Vin and Vout (gaussian function)
V
Time
8
7
6
5
4
3
2
1
0
-1
-2
0
100
200
300
400
500
600
700
800
900
1000


Vin and Vout frequency domain (step function)
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
V
-150
-100
-50
0
50
100
150
frequency


Vin and Vout frequency domain (sin function)
100
90
80
70
60
50
40
30
20
10
0
V
-150
-100
-50
0
50
100
150
frequency



Comments on increased time step

By increasing and decreasing the time step, the simulations will become respcetively less and more defined. However, the simulation will start to break when the time step is too small.

Part 3

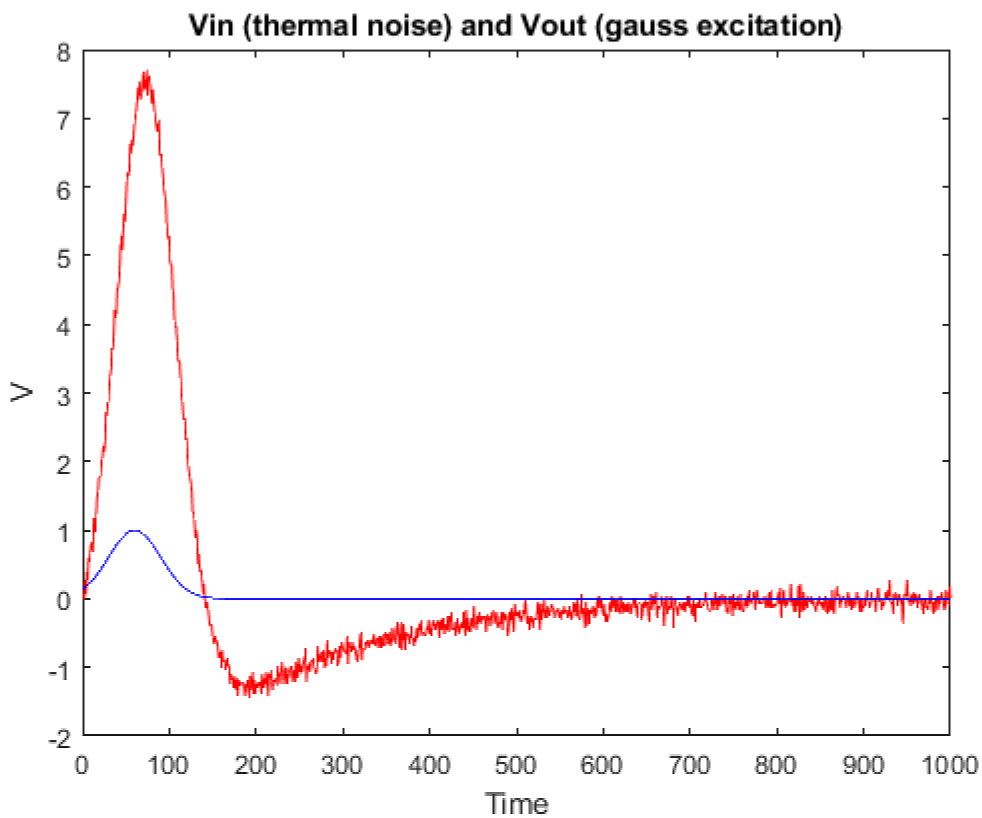
```
R = [1, 2, 10, 0.1, 1000];
C1Val = 0.25;
C2Val = 1e-5;
LVal = 0.2;
alpha = 100;
omega = 0;

G = zeros(7);

%Row 1
G(1,1) = 1;

%Row 2
G(2,1) = -1./R(2);
G(2,2) = (1./R(1))+(1./R(2));
G(2,3) = -1;

%Row 3
```

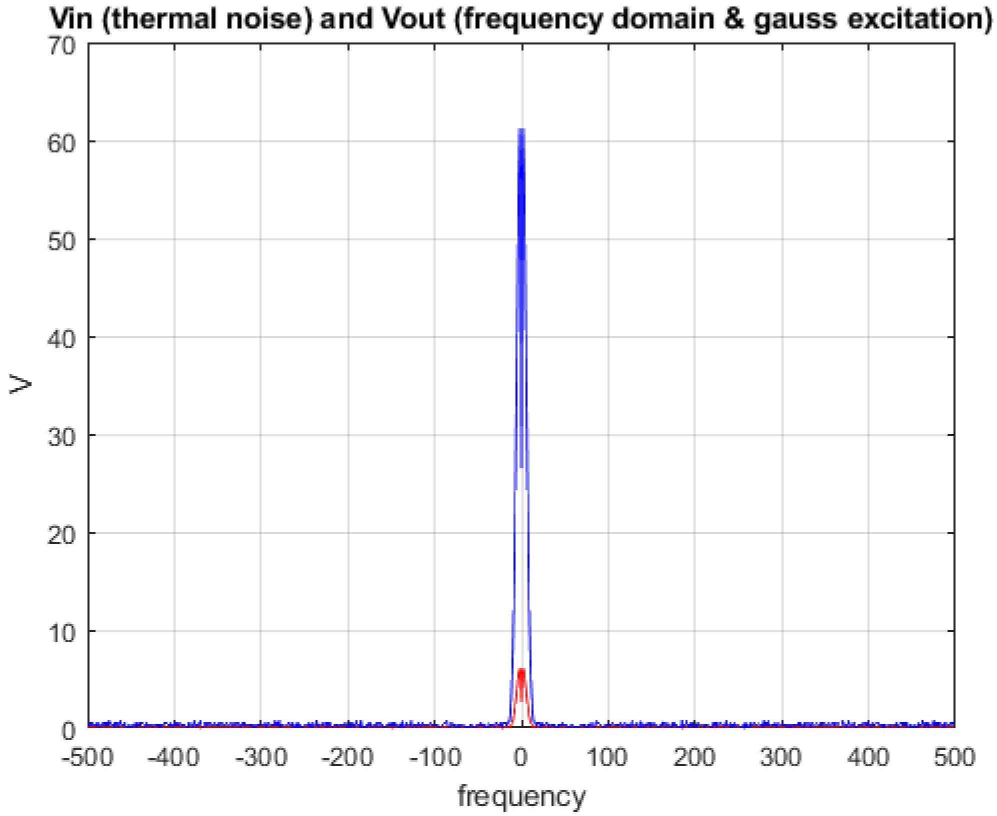
```
G(3,2) = 1;
G(3,4) = -1;

%Row 4
G(4,3) = -1;
G(4,4) = 1./R(3);

%Row 5
G(5,5) = -alpha;
G(5,6) = 1;

%Row 6
G(6,4) = 1./R(3);
G(6,5) = -1;

%Row 7
G(7,6) = -1./R(4);
G(7,7) = (1./R(4))+(1./R(5));


%C matrix
C = zeros(7);

%Row 1
C(2,1) = -C1Val;

%Row 2
C(2,2) = C1Val;

%Row 3
C(3,3) = -LVal;

%Row 4
C(4,4) = -C2Val;

%Row 6
C(6,4) = -C2Val;


Vin = 1;
F = zeros(7,1);
F(1,1) = Vin;

Foff = zeros(7,1);
Foff(1,1) = Vin-Vin;

ts1 = 1000;
ts2 = 1.9898e4;

Vstart = zeros(7, 1);
dt1 = 1e-3;
dt2 = 1.9898e-4;

% Noise simulation and default time step
```

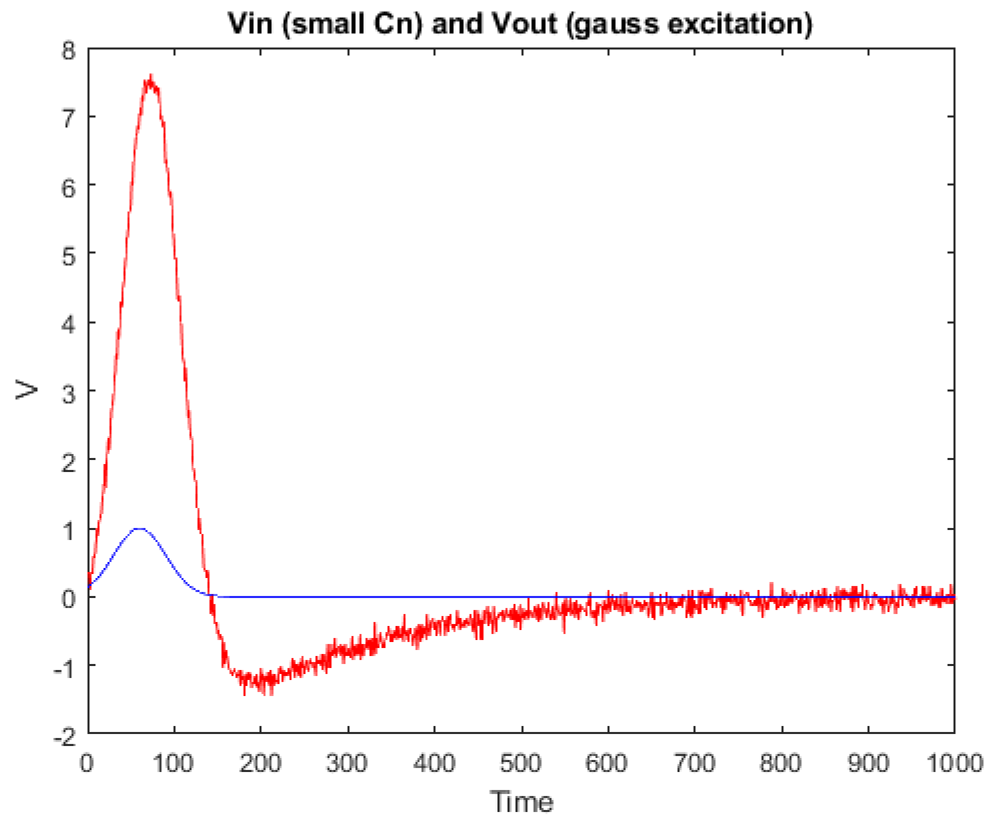
```
% Time domain simulation
V1 = zeros(7, ts1);
Fgauss = zeros(7,1);
for i = 1:ts1

    Fgauss(1,1) = exp(-1/2*((i/ts1-0.06)/(0.03))^2);
    Fgauss(4,1) = 0.001*randn();
    Fgauss(7,1) = 0.001*randn();
    if i == 1
        V1(:,i) = (C./dt1+G)\(Fgauss+C*Vstart/dt1);
    else
        V1(:,i) = (C./dt1+G)\(Fgauss+C*Vpast/dt1);
    end
    Vpast = V1(:, i);

end
figure(12)
plot(1:ts1, V1(7,:), 'r')
hold on
plot(1:ts1, V1(1,:), 'b')
title('Vin (thermal noise) and Vout (gauss excitation)')
xlabel('Time')
ylabel('V')

% Frequency domain simulation
f = (-ts1/2:ts1/2-1);

fV1 = fft(V1.');
fsV1 = fftshift(fV1);
figure(13)
plot(f, abs(fsV1(:, 1)), 'r')
hold on
plot(f, abs(fsV1(:, 7)), 'b')
title('Vin (thermal noise) and Vout (frequency domain & gauss
 excitation)')
xlabel('frequency')
ylabel('V')
grid on

% Simulation with smaller capacitance
V2 = zeros(7, ts1);
Fgauss = zeros(7,1);

%C matrix with smaller Cn values
Csmaller = zeros(7);

%Row 1
Csmaller(2,1) = -C1Val;

%Row 2
Csmaller(2,2) = C1Val;

%Row 3
Csmaller(3,3) = -LVal;
```

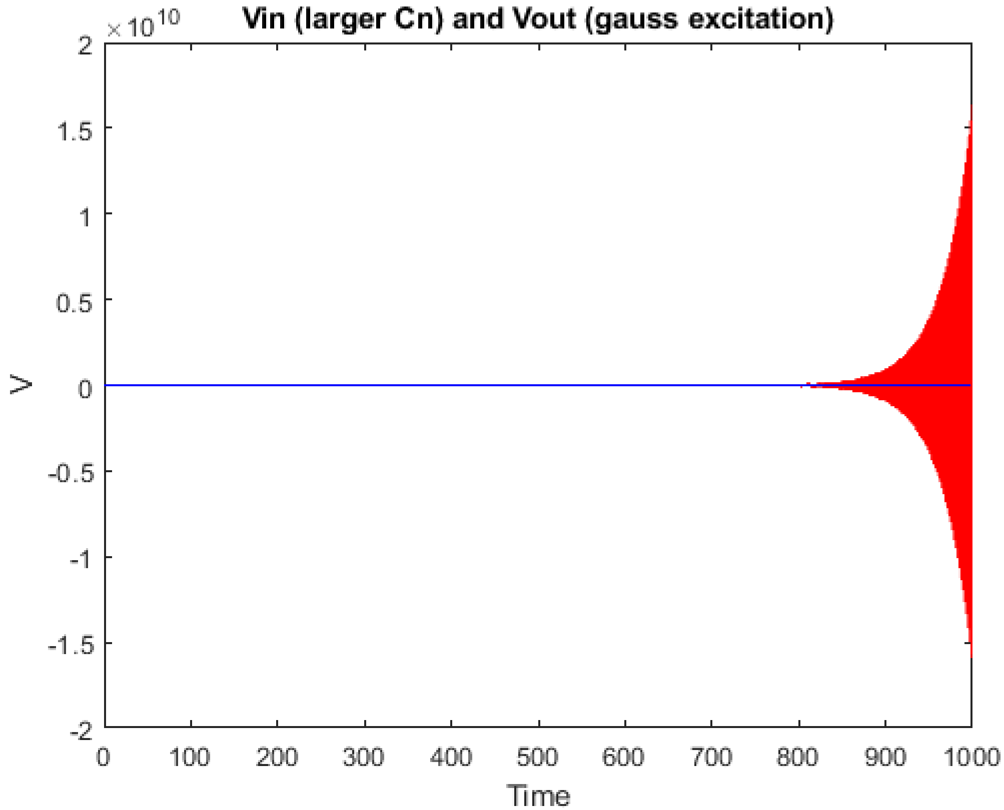
```
%Row 4
Csmaller(4,4) = -1e-12;

%Row 6
Csmaller(6,4) = -1e-12;

for j = 1:ts1

    Fgauss(1,1) = exp(-1/2*((j/ts1-0.06)/(0.03))^2);
    Fgauss(4,1) = 0.001*randn();
    Fgauss(7,1) = 0.001*randn();
    if j == 1
        V2(:,j) = (Csmaller./dt1+G)\(Fgauss+Csmaller*Vstart/dt1);
    else
        V2(:,j) = (Csmaller./dt1+G)\(Fgauss+Csmaller*Vpast/dt1);
    end
    Vpast = V2(:, j);

end
figure(14)
plot(1:ts1, V2(7,:), 'r')
hold on
plot(1:ts1, V2(1,:), 'b')
title('Vin (small Cn) and Vout (gauss excitation)')
xlabel('Time')
ylabel('V')

% Simulation with larger capacitance
V3 = zeros(7, ts1);
Fgauss = zeros(7,1);

%C matrix with larger Cn value
Clager = zeros(7);

%Row 1
Clager(2,1) = -C1Val;

%Row 2
Clager(2,2) = C1Val;

%Row 3
Clager(3,3) = -LVal;

%Row 4
Clager(4,4) = -5.2e-5;

%Row 6
Clager(6,4) = -5.2e-5;

for k = 1:ts1

    Fgauss(1,1) = exp(-1/2*((k/ts1-0.06)/(0.03))^2);
    Fgauss(4,1) = 0.001*randn();
```

```
    Fgauss(7,1) = 0.001*randn();
    if k == 1
        V3(:,k) = (Clager./dt1+G)\(Fgauss+Clager*Vstart/dt1);
    else
        V3(:,k) = (Clager./dt1+G)\(Fgauss+Clager*Vpast/dt1);
    end
    Vpast = V3(:, k);

end
figure(15)
plot(1:ts1, V3(7,:), 'r')
hold on
plot(1:ts1, V3(1,:), 'b')
title('Vin (larger Cn) and Vout (gauss excitation)')
xlabel('Time')
ylabel('V')
```




Vin (thermal noise) and Vout (frequency domain & gauss excitation)
V
frequency
70
60
50
40
30
20
10
0
-500
-400
-300
-200
-100
0
100
200
300
400
500


Vin (small Cn) and Vout (gauss excitation)
V
Time
8
7
6
5
4
3
2
1
0
-1
-2
0
100
200
300
400
500
600
700
800
900
1000

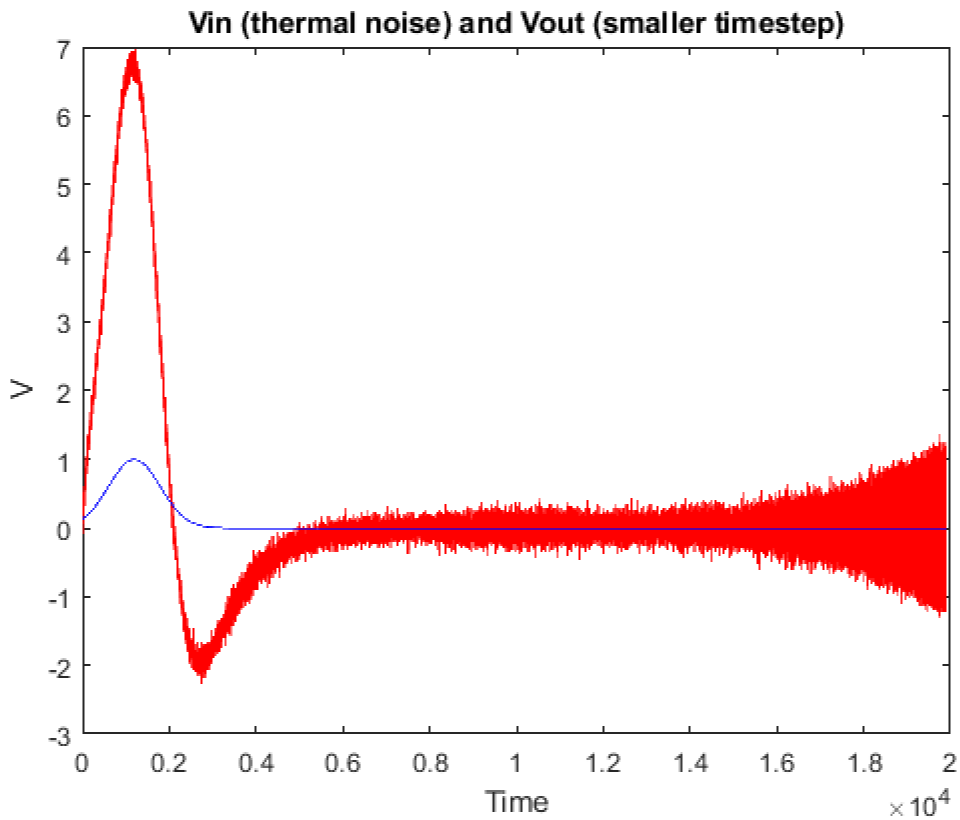


Comments on capacitance changes

When the capacitance is increased slightly the simulation breaks since the it gets suck in a feedback loop and there is no change when the capacitance is decreased.

```
% Simulation with smaller time step
V4 = zeros(7, ts2);
Fgauss = zeros(7,1);
for m = 1:ts2

    Fgauss(1,1) = exp(-1/2*((m/ts2-0.06)/(0.03))^2);
    Fgauss(4,1) = 0.001*randn();
    Fgauss(7,1) = 0.001*randn();
    if m == 1
        V4(:,m) = (C./dt2+G)\(Fgauss+C*Vstart/dt2);
    else
        V4(:,m) = (C./dt2+G)\(Fgauss+C*Vpast/dt2);
    end
    Vpast = V4(:, m);

end
figure(16)
plot(1:ts2, V4(7,:), 'r')
hold on
plot(1:ts2, V4(1,:), 'b')
title('Vin (thermal noise) and Vout (smaller timestep)')
```

```
xlabel('Time')
ylabel('V')
```



Part 4

Part 4A

If the voltage was modeled by the new equation given then this would cause the problem to be non-linear. With the non-linearity it means we can no longer use exact numerical solutions and more than likely need to use an iterative solution. This iterative solution means we would need to introduce a new vector into the simulator.

Part 4B

To implement the non-linear portion of the system, a new 'B' vector must be created to store these values. The new vector will be used as an additional term that is summed with the other terms in the equation for voltage. After taking into account the new vector, a mathemaical technique is used to convert the non-linear terms into a form that can be applied to linear operations. To do this it would involed producing a Jacobian matrix. After this step the calculations would be relatively the same except the Jacobian matrix would be need to evlaute voltages.

Published with MATLAB® R2017b