ELEC 4700 Assignment 4 Jalil (Rohana) Aalab #100995788 Apr/08/2018

Table of Contents

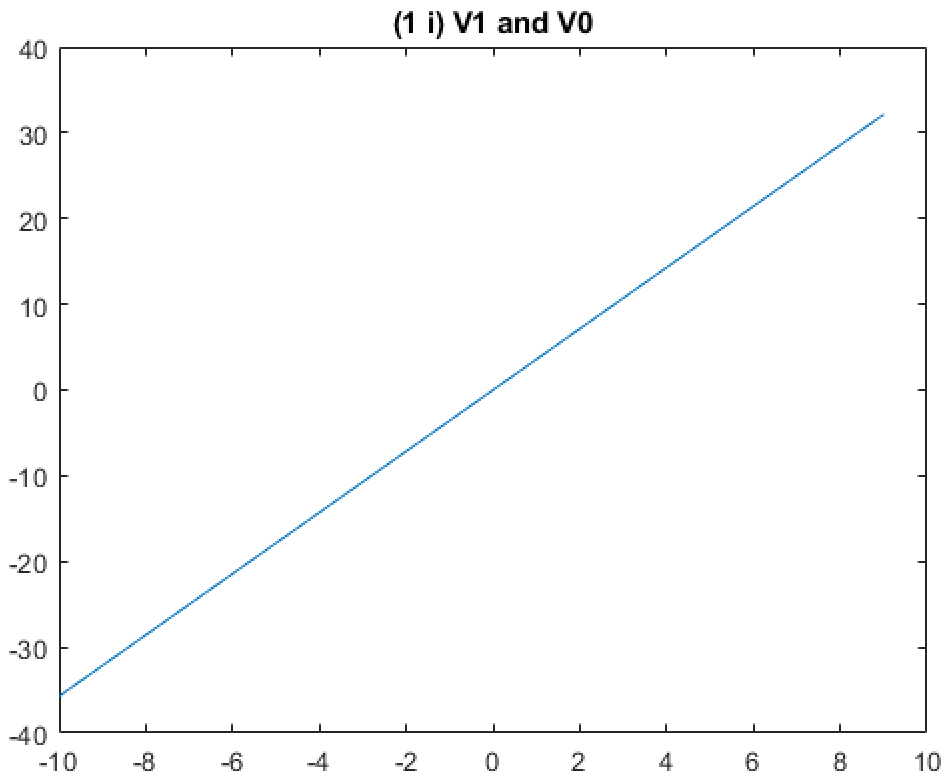
PART 1 (PA 9) 1
PART 2 (TRANSIENT CIRCUIT SIMULATION) 5
2A 5
2B 5
2D 5
PART 3 (CIRCUIT WITH NOISE) 15
3C 15
PART 4 (NON-LINEARITY) 28
4A 28
4B 28
4C 28

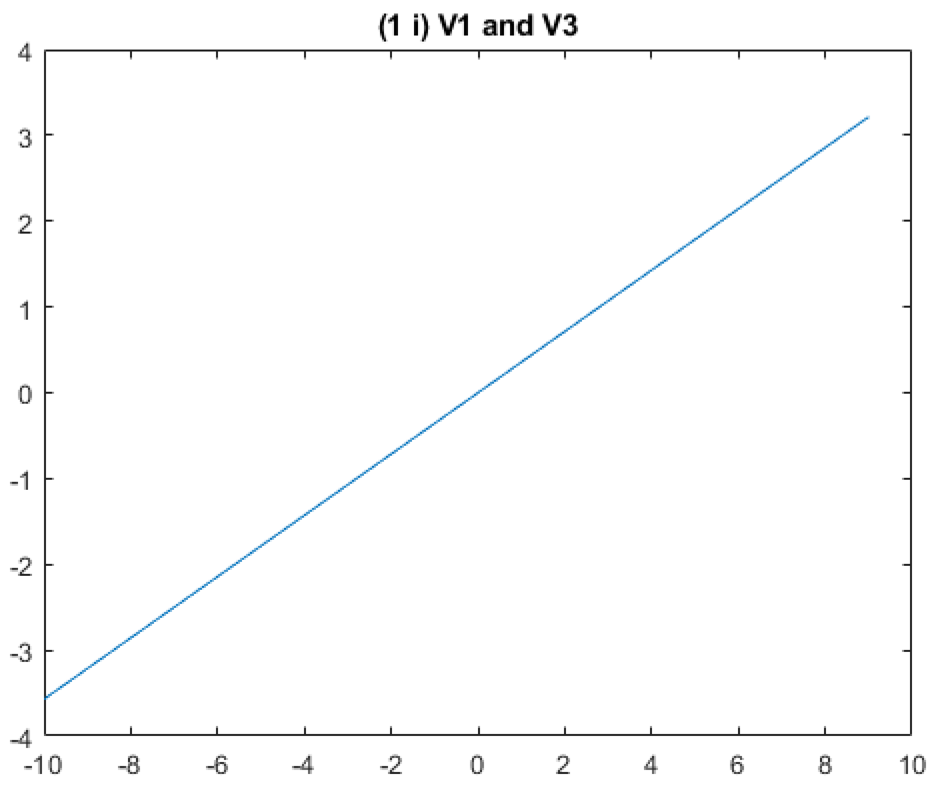
PART 1 (PA 9)

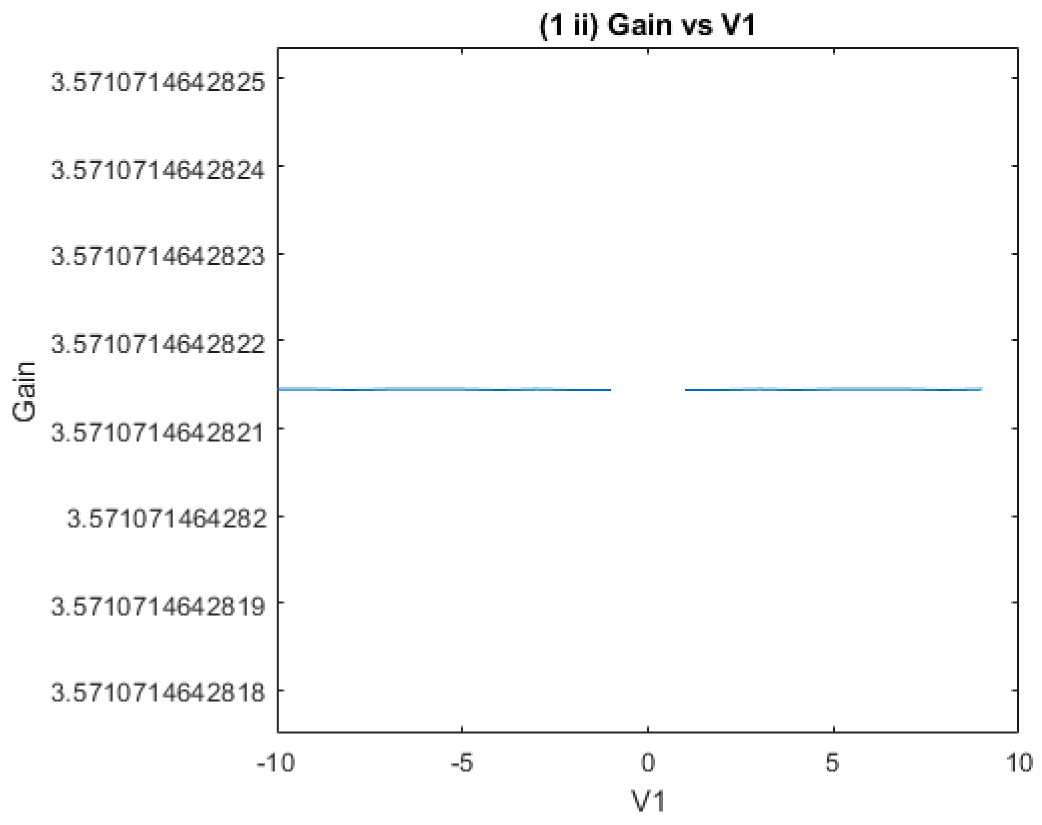
This section provides an introduction to the assignment by showing the code that was used in the 9th PA session. This code provides the basis for the rest of the assignment by establishing a foundation of the theory and skills that are behind the concept of circuit modeling.

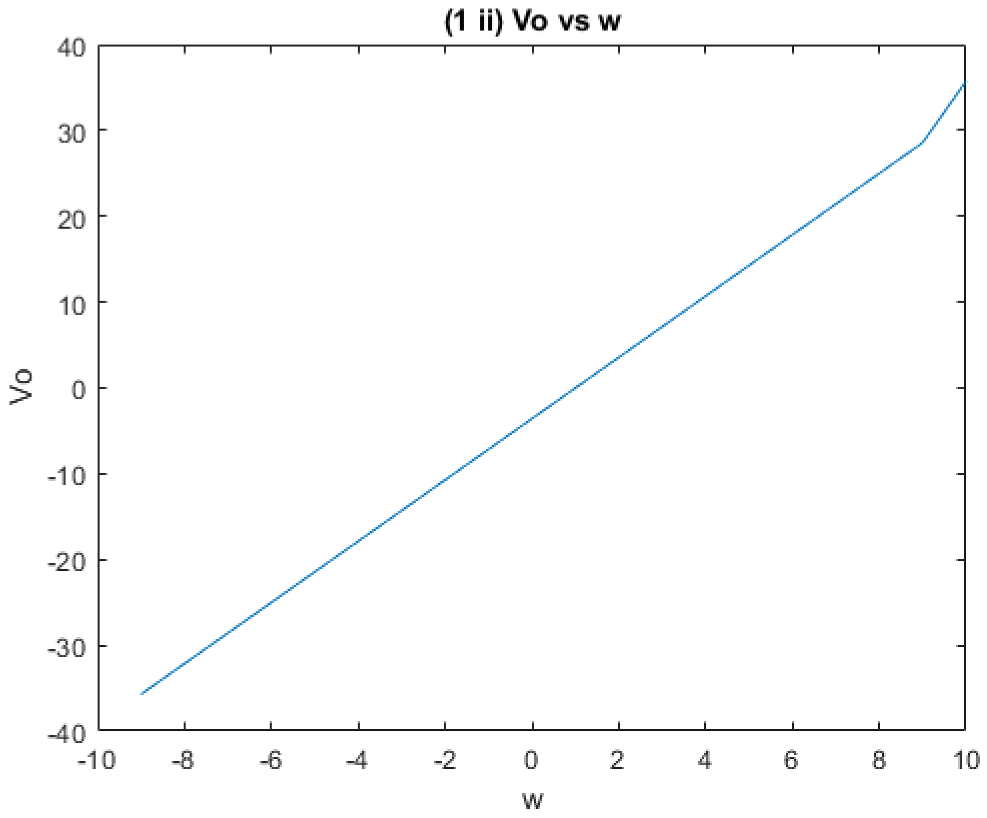
```
% Initialize variables
C = 0.25;
L = 0.2;
G1 = 1/1;
G2 = 1/2;
G3 = 1/10;
G4 = 1/0.1;
G0 = 1/1000;
a = 100;

% C Matrix
Cmat = zeros(7);
Cmat(2,1) = -C;
Cmat(2,2) = C;
Cmat(3,3) = -L;

% G Matrix
G = [1 0 0 0 0 0 0;...
    -G2 (G1+G2) -1 0 0 0 0;...
    0 1 0 -1 0 0 0;...
    0 0 -1 G3 0 0 0;...
    0 0 0 0 -a 1 0;...
    0 0 0 G3 -1 0 0;...
    0 0 0 0 0 -G4 (G4 +G0)];

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

% F Vector
V1 = -10;
F = [V1, 0, 0, 0, 0, 0, 0];

% Other Matrices
ALLv1 = zeros(1, 20);
ALLv3 = zeros(1, 20);
ALLv0 = zeros(1, 20);

% Part i DC CASE
% V1 is initially -10, then after 20 additions it will reach +10

for k = 1 : 20
    ALLv1(k) = V1;
    % The current value of V1 has been stored for plotting later
    F(1) = V1;
    V = G\F';
    ALLv3(k) = V(4);
    ALLv0(k) = V(7);
    % The current values of V3 and V0 have been stored for later
    V1 = V1 + 1;
end

% Now, all the various values for V3 and V0 have been collected
figure(1);
plot(ALLv1, ALLv0);
title('(1 i) V1 and V0');
figure(2);
plot(ALLv1, ALLv3);
title('(1 i) V1 and V3');

% Part ii AC CASE
% (G + jwC)V = F(w)
% gain = V0./V1;
gain = ALLv0 ./ ALLv1;
figure(3);
plot(ALLv1, gain);
title('(1 ii) Gain vs V1');
xlabel('V1');
ylabel('Gain');

for w = -10 : 10
    F(1) = w;
    V  = G \ F';
    ALLv0(k) = V(7);
    % The current value V0 has been stored for later
```

```
end

figure(4);
w = -9 : 10;
plot(w, ALLv0);
title('(1 ii) Vo vs w');
xlabel('w');
ylabel('Vo');
```




(1 i) V1 and V3
4
3
2
1
0
-1
-2
-3
-4
-10
-8
-6
-4
-2
0
2
4
6
8
10


(1 ii) Gain vs V1
Gain
3.5710714642825
3.5710714642824
3.5710714642823
3.5710714642822
3.5710714642821
3.571071464282
3.5710714642819
3.5710714642818
-10
-5
0
5
10
V1



PART 2 (TRANSIENT CIRCUIT SIMULATION)

2A

This section determines the type of circuit in Figure 1.

By inspection, the circuit in Figure 1 uses resistors, capacitors, and inductors with voltage sources to form a closed circuit. The use of these linear components would determine that the circuit is an 'RLC', or resistor-inductor-capacitor circuit.

2B

This section states the expected type of frequency response.

As opposed to the resistors, the impedances of the capacitor and inductor are influenced by the frequency. The impedance of the capacitor is given as 1 / (jwC) , and for the inductor it is (jwL) , where 'j' is the square root of -1 and w represents the angular frequency (2 * pi * f). These formulas show that as the frequency increases, the impedance of the capacitor decreases but that of the inductor increases. When there is a frequency of zero (or a DC case), the impedance of the capacitor is extremely large whereas that of the inductor is zero. At an infinite frequency, the capacitor has zero impedance, but the inductor will have infinite impedance.

2D

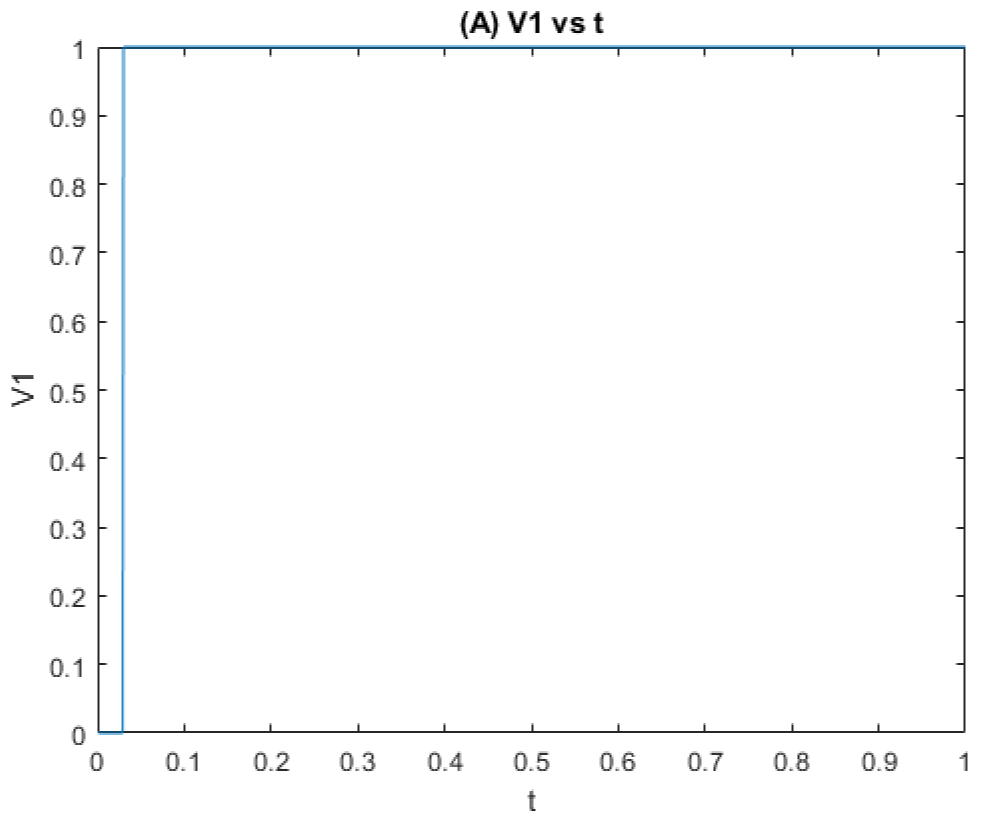
This section covers the simulation of the circuit. Initialize variables

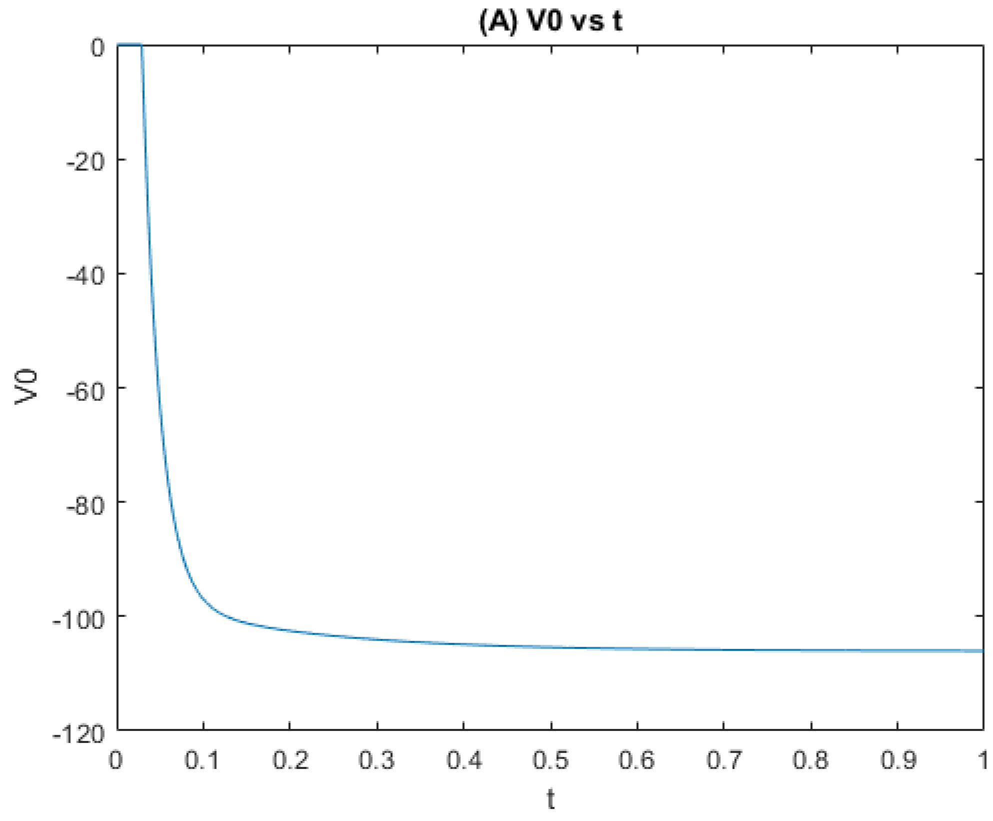
```
C = 0.25;
L = 0.2;
G1 = 1/1;
G2 = 1/2;
G3 = 1/10;
G4 = 1/0.1;
G0 = 1/1000;
a = 100;

% C Matrix
Cmat = zeros(7);
Cmat(2,1) = -C;
Cmat(2,2) = C;
Cmat(3,3) = -L;

% G Matrix
G = [1 0 0 0 0 0 0;...
    -G2 (G1+G2) -1 0 0 0 0;...
    0 1 0 -1 0 0 0;...
    0 0 -1 G3 0 0 0;...
    0 0 0 0 -a 1 0;...
    0 0 0 G3 -1 0 0;...
    0 0 0 0 0 -G4 (G4 +G0)];

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

% F Vector
% F = [V1, 0, 0, 0, 0, 0, 0];
F = zeros(1, 7);

% Other Matrices
ALLv1 = zeros(1, 1000);
ALLv3 = zeros(1, 1000);
ALLv0 = zeros(1, 1000);

% Q2 A (Step Input)
A = zeros(7,7);
sim_time = 1;
steps = 1000;
delta_t = 0.001;
counter = 1;
Vin = 0;
for t = 0.001 : 0.001 : 1
    Vold = Vnew;
    if t < 0.03
        Vin = 0;
    else
        Vin = 1;
    end
    F(1) = Vin;
    A = (Cmat ./ delta_t) + G;
    Vnew = inv(A) * ((Cmat * (Vold / delta_t)) + F);
    ALLv1(counter) = Vnew(1);
    ALLv0(counter) = Vnew(7);
    counter = counter + 1;
end
t = 0.001 : 0.001 : 1;
figure(5);
plot(t, ALLv1);
title('(A) V1 vs t');
ylabel('V1');
xlabel('t');

figure(6);
plot(t, ALLv0);
title('(A) V0 vs t');
ylabel('V0');
xlabel('t');

Y = fft(ALLv1);
Y = fftshift(Y);
figure(7);
plot(t, Y);
title('(A) Frequency V1');
xlabel('t');
ylabel('Frequency Content');

Y = fft(ALLv0);
Y = fftshift(Y);
figure(8);
plot(t, Y);
title('(A) Frequency V0');
xlabel('t');
ylabel('Frequency Content');

% Q2 B (Sinusoid Input)
f = 1 / 0.03;
counter = 1;
for t = 0.001 : 0.001 : 1
    Vold = Vnew;
    Vin = sin(2 * pi * f * t);
    F(1) = Vin;
    A = (Cmat ./ delta_t) + G;
    Vnew = inv(A) * ((Cmat * (Vold / delta_t)) + F);
    ALLv1(counter) = Vnew(1);
    ALLv0(counter) = Vnew(7);
    counter = counter + 1;
end
t = 0.001 : 0.001 : 1;
figure(9);
plot(t, ALLv1);
title('(B) V1 and t');
xlabel('t');
ylabel('V1');
```

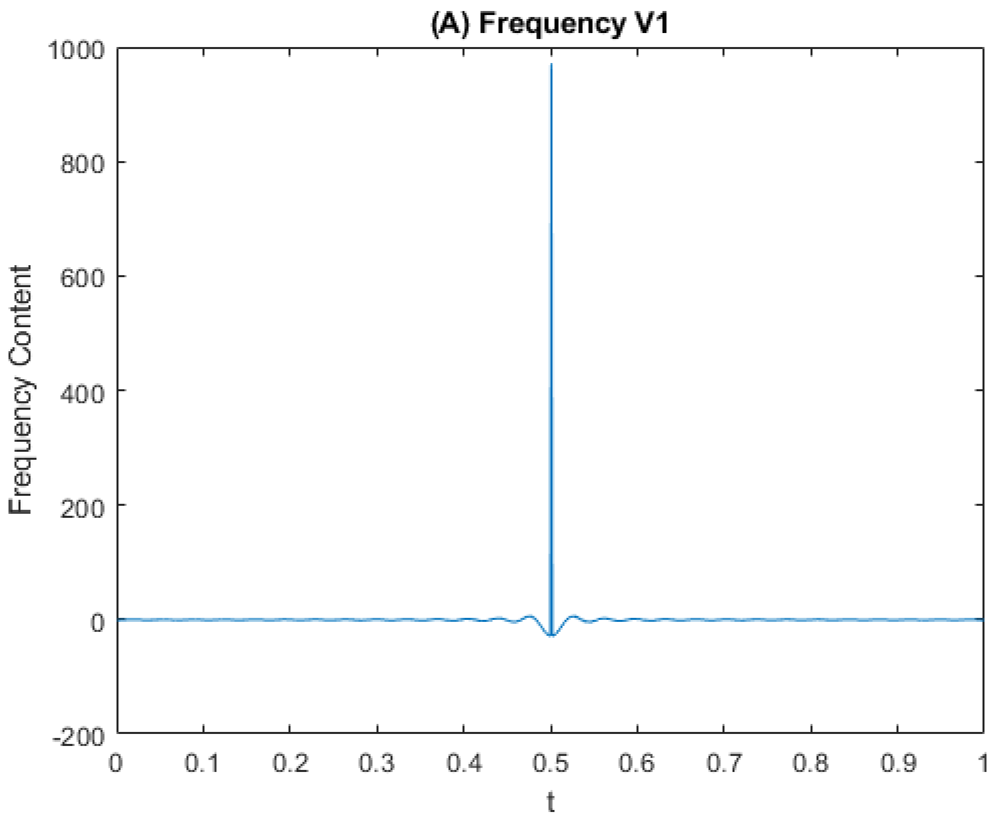
```
figure(10);
plot(t, ALLv0);
title('(B) V0 vs t');
ylabel('V0');
xlabel('t');

Y = fft(ALLv1);
Y = fftshift(Y);
figure(11);
plot(t, Y);
title('(B) Frequency V1');
xlabel('t');
ylabel('Frequency Content');

Y = fft(ALLv0);
Y = fftshift(Y);
figure(12);
plot(t, Y);
title('(B) Frequency V0');
xlabel('t');
ylabel('Frequency Content');

% Q2 C (Gaussian Input)
counter = 1;
Vin = 0;
for t = 0.001 : 0.001 : 1
    Vold = Vnew;
    Vin = (1 / (0.03 * sqrt(2*pi))) * (exp((-1/2) *
 (((t-0.06)/0.03)^2)));
    F(1) = Vin;
    A = (Cmat ./ delta_t) + G;
    Vnew = inv(A) * ((Cmat * (Vold / delta_t)) + F);
    ALLv1(counter) = Vnew(1);
    ALLv0(counter) = Vnew(7);
    counter = counter + 1;
end
t = 0.001 : 0.001 : 1;
figure(13);
plot(t, ALLv1);
title('(C) V1 and t');
xlabel('t');
ylabel('V1');

figure(14);
plot(t, ALLv0);
title('(C) V0 vs t');
ylabel('V0');
xlabel('t');

Y = fft(ALLv1);
Y = fftshift(Y);
figure(15);
plot(t, Y);
title('(C) Frequency V1');
```

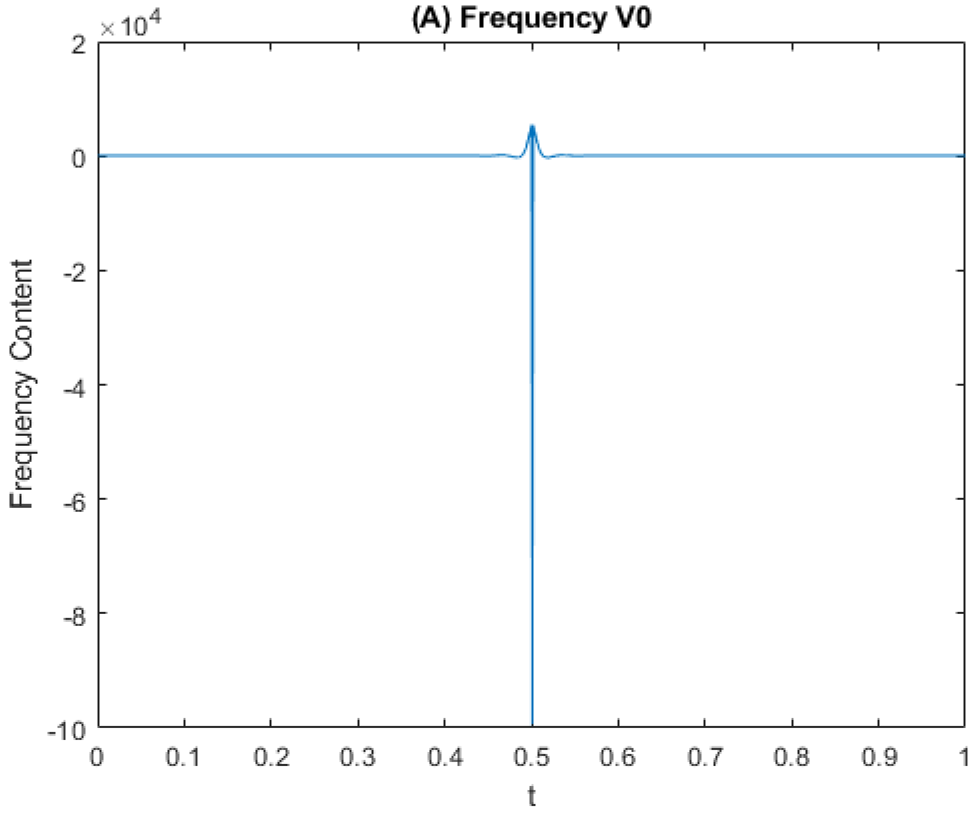
```
xlabel('t');
ylabel('Frequency Content');

Y = fft(ALLv0);
Y = fftshift(Y);
figure(16);
plot(t, Y);
title('(C) Frequency V0');
xlabel('t');
ylabel('Frequency Content');
```

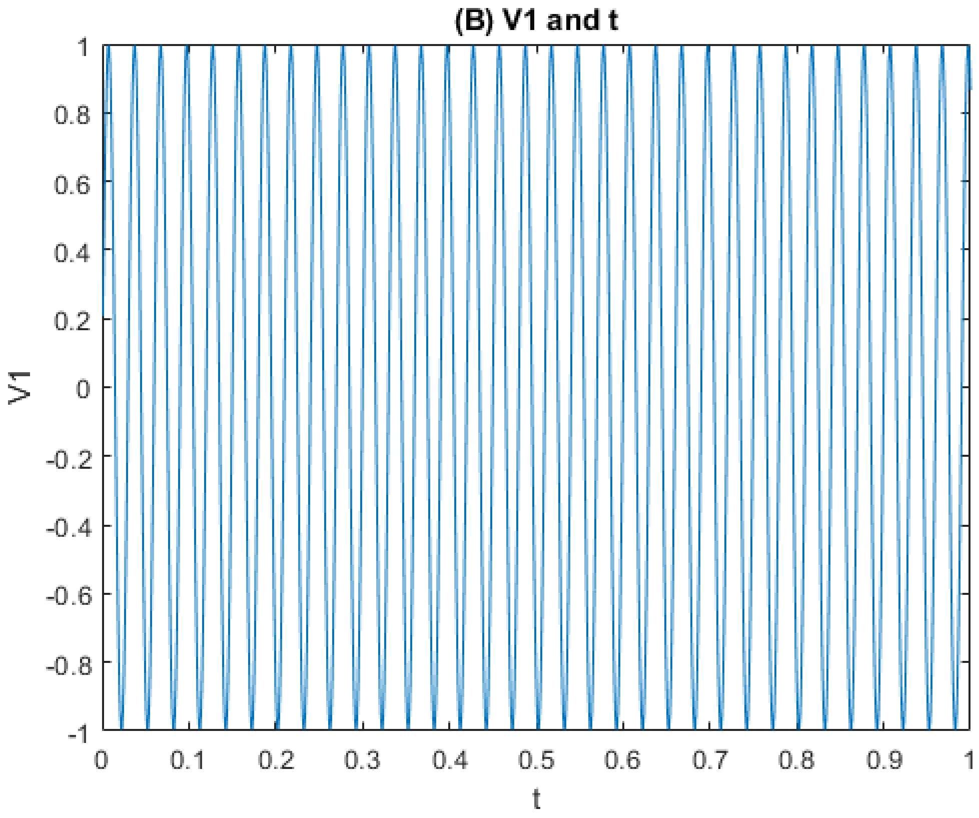
```
Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored
```

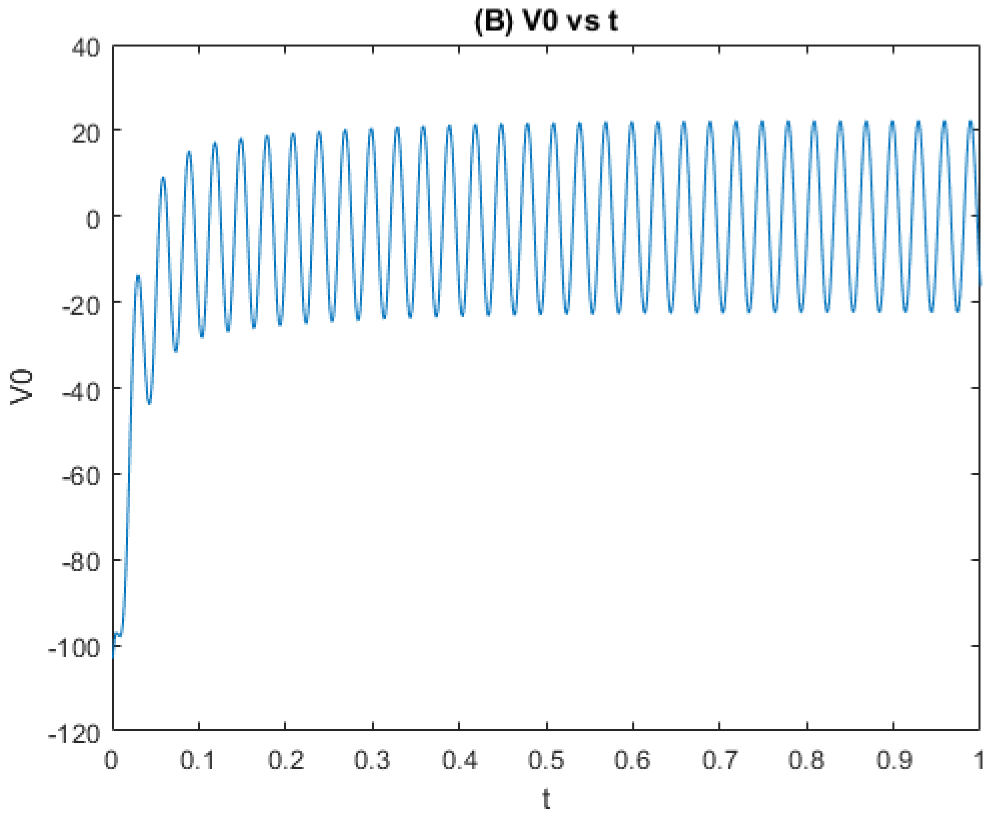


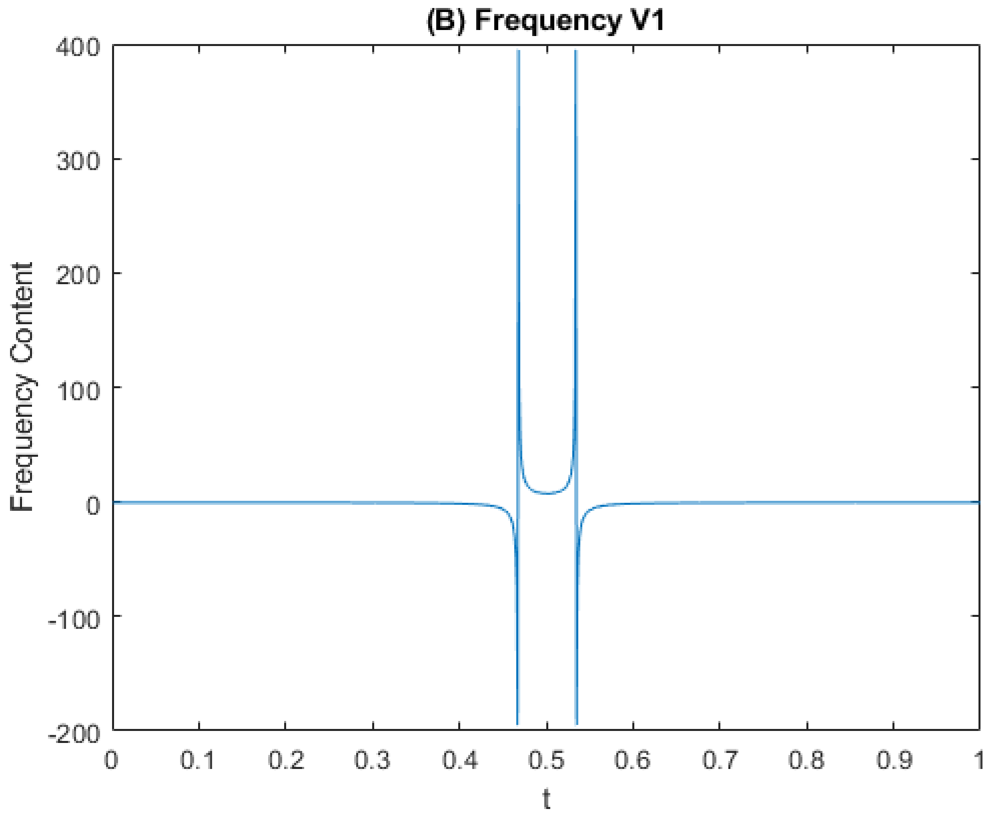

(A) V0 vs t
V0
t
0
-20
-40
-60
-80
-100
-120
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

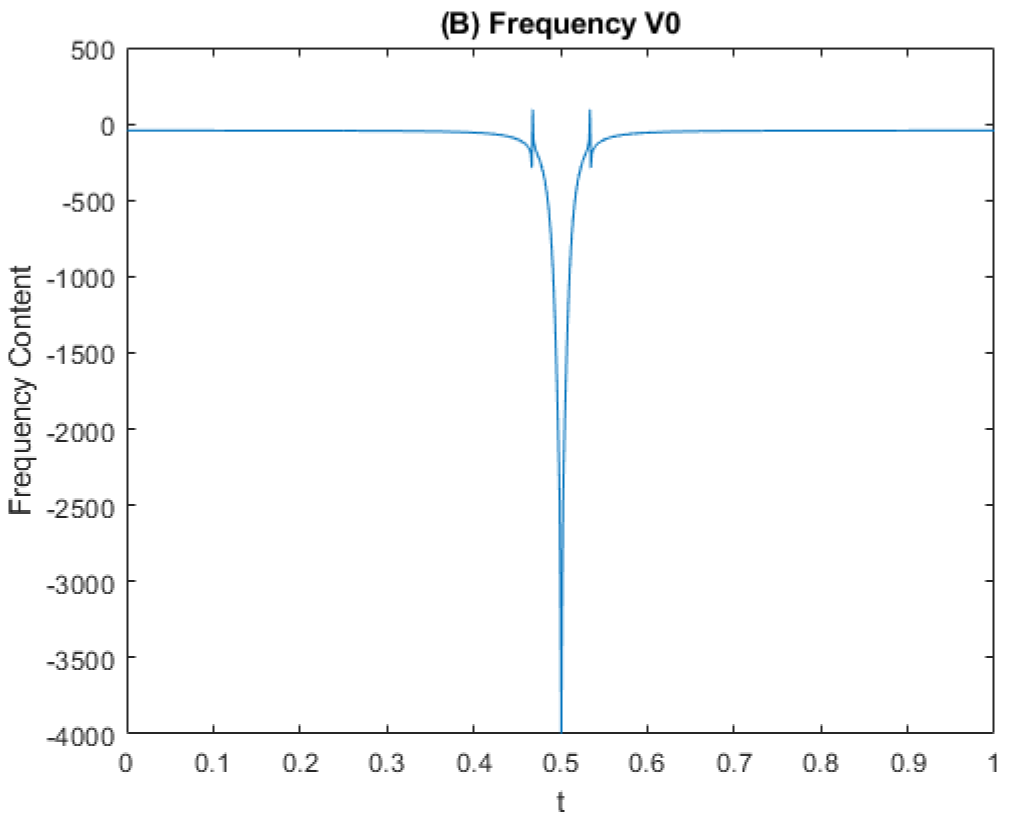

(A) Frequency V1
Frequency Content
t
1000
800
600
400
200
0
-200
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

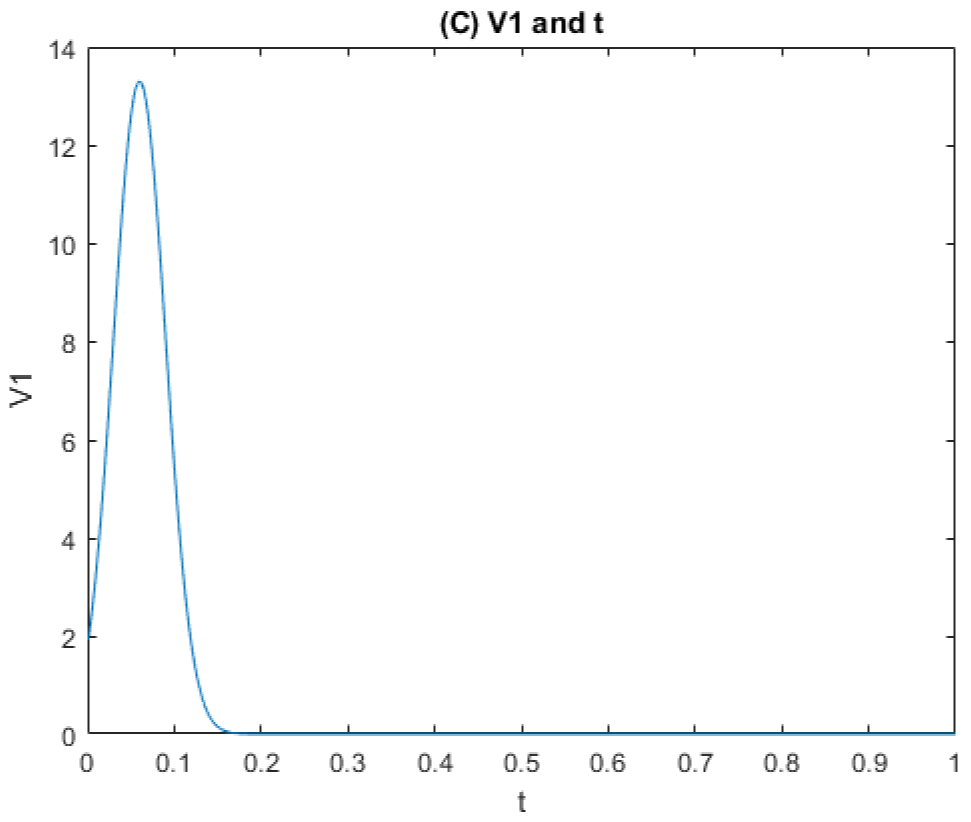

(A) Frequency V0
×10^4
Frequency Content
t
2
0
-2
-4
-6
-8
-10
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

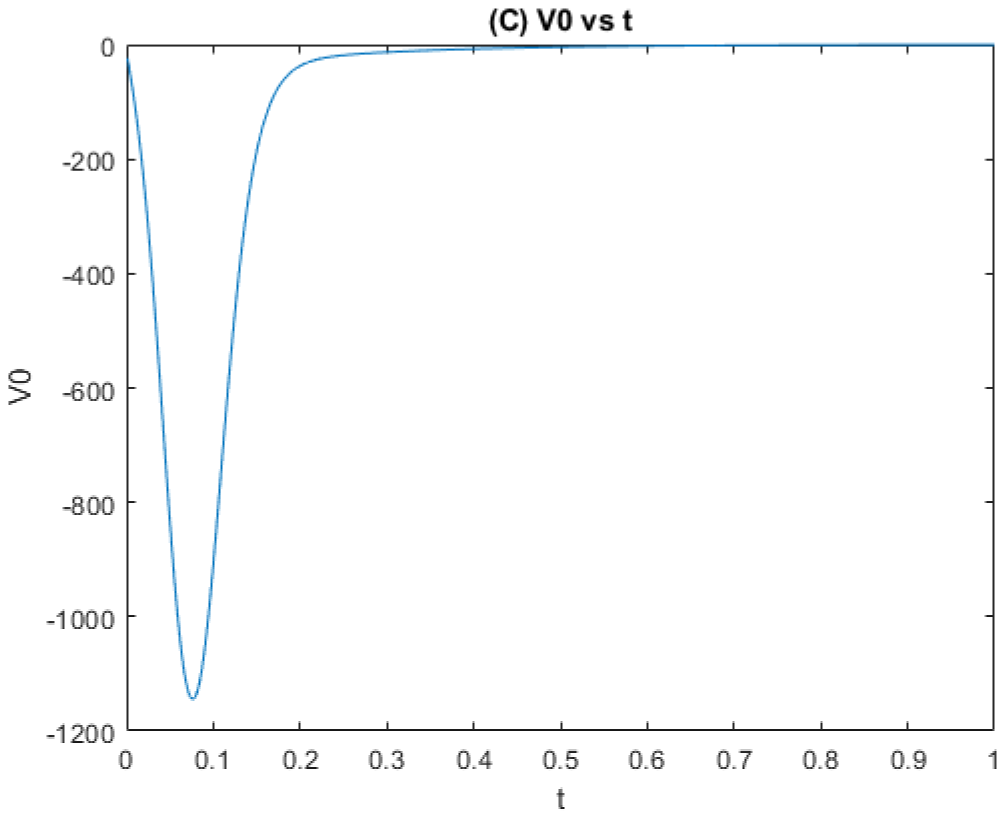

(B) V1 and t
V1
t
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

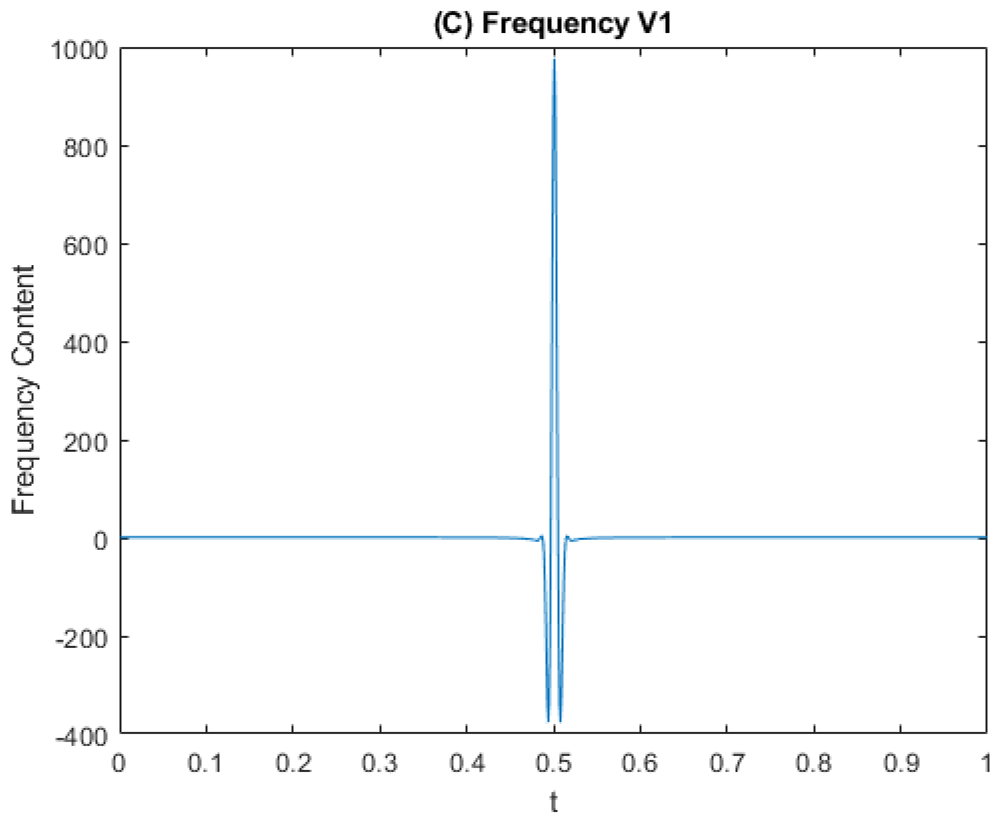

(B) V0 vs t
V0
t
40
20
0
-20
-40
-60
-80
-100
-120
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

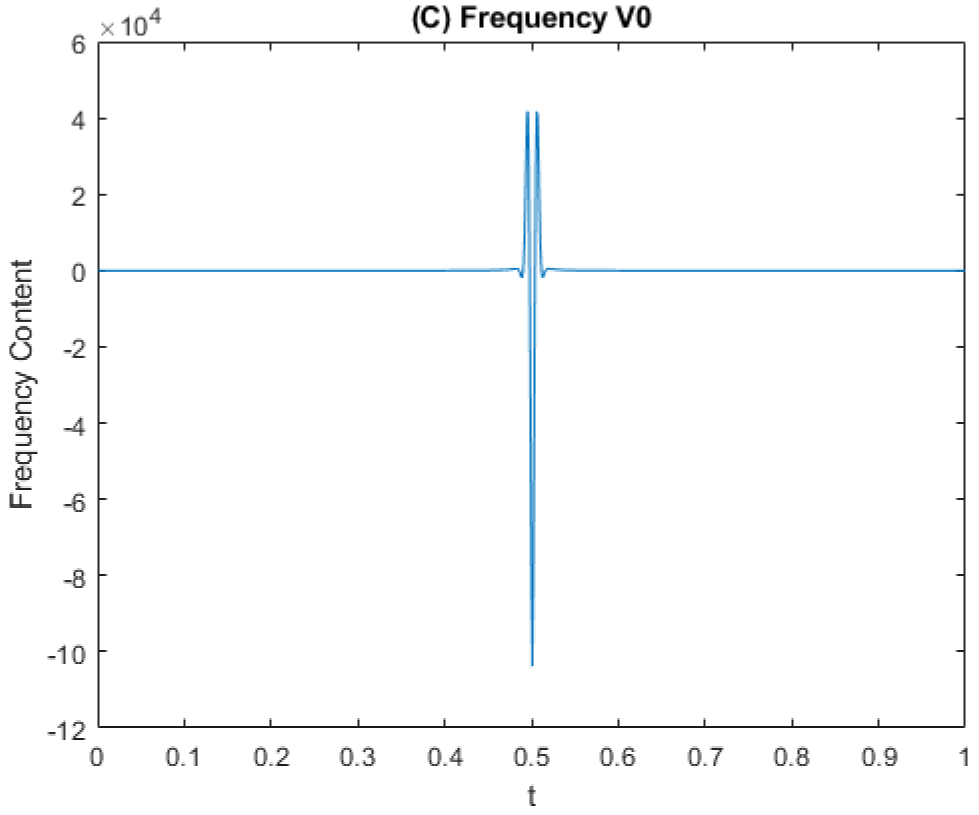

(B) Frequency V1
Frequency Content
t
400
300
200
100
0
-100
-200
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


(B) Frequency V0
Frequency Content
t
500
0
-500
-1000
-1500
-2000
-2500
-3000
-3500
-4000
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


(C) V1 and t
V1
t
14
12
10
8
6
4
2
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


(C) V0 vs t
V0
t
0
-200
-400
-600
-800
-1000
-1200
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


(C) Frequency V1
Frequency Content
t
1000
800
600
400
200
0
-200
-400
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



PART 3 (CIRCUIT WITH NOISE)

The simulation in this section attempts to mimic the effects of adding noise to disrupt the circuit.

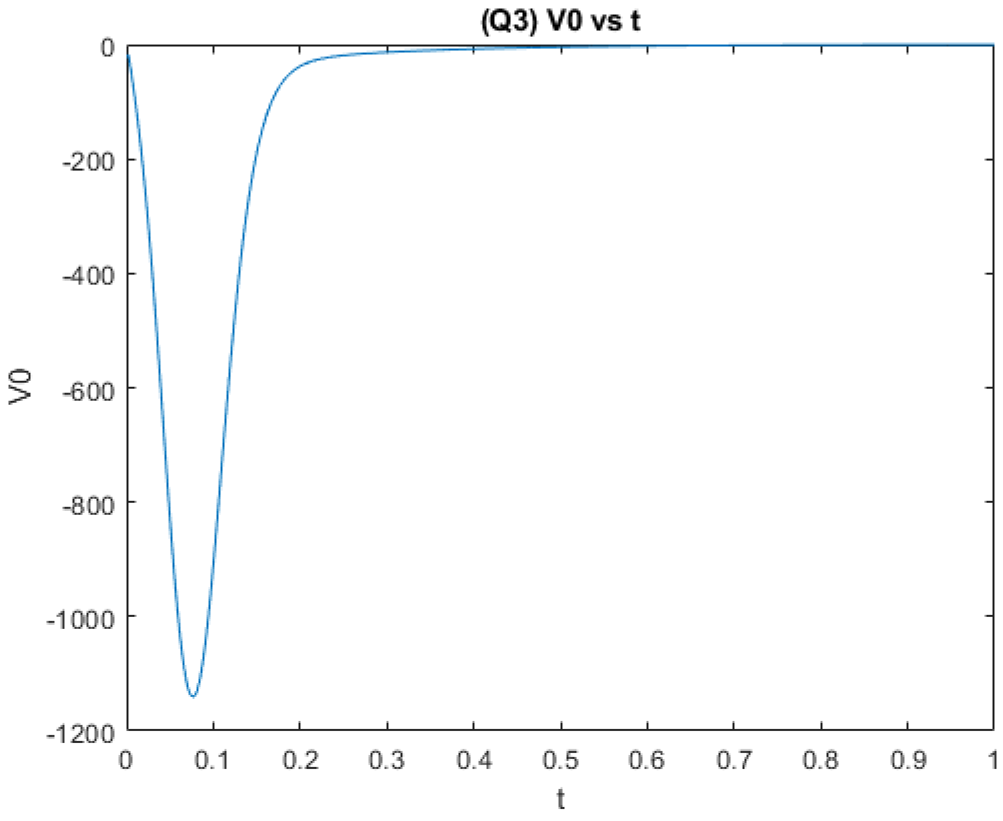
3C

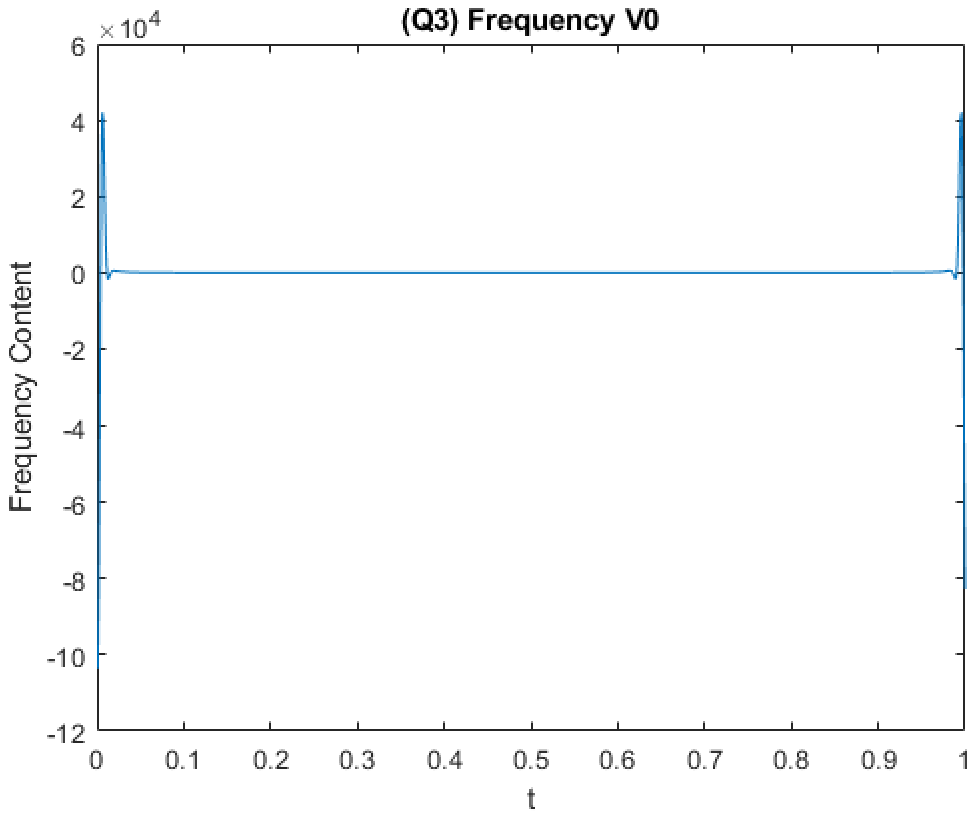
This code covers the simulation of the circuit with noise.

```
Cn = 0.00001;
Cmat(4,4) = Cn;

% Random number for In
In = (1 / (0.03 * sqrt(2*pi))) * (exp((-1/2) * ((((rand /10) -
 0.001)/0.03)^2))) / 1000;

G = [1 0 0 0 0 0 0;...
    -G2 (G1+G2) -1 0 0 0 0;...
    0 1 0 -1 0 0 0;...
    0 0 -1 G3 In 0 0;...
    0 0 0 0 -a 1 0;...
    0 0 0 G3 -1 0 0;...
    0 0 0 0 0 -G4 (G4 +G0)];

% Calculation
delta_t = 0.001;
```

```
counter = 1;
Vin = 0;
for t = 0.001 : 0.001 : 1
    Vold = Vnew;
    Vin = (1 / (0.03 * sqrt(2*pi))) * (exp((-1/2) *
 (((t-0.06)/0.03)^2)));
    F(1) = Vin;
    A = (Cmat ./ delta_t) + G;
    Vnew = inv(A) * ((Cmat * (Vold / delta_t)) + F);
    ALLv0(counter) = Vnew(7);
    counter = counter + 1;
end

t = 0.001 : 0.001 : 1;
figure(17);
plot(t, ALLv0);
title('(Q3) V0 vs t');
ylabel('V0');
xlabel('t');

Y = fft(ALLv0);
figure(18);
plot(t, Y);
title('(Q3) Frequency V0');
xlabel('t');
ylabel('Frequency Content');
```

```
Warning: Imaginary parts of complex X and/or Y arguments ignored
```


(Q3) V0 vs t
V0
t
0
-200
-400
-600
-800
-1000
-1200
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

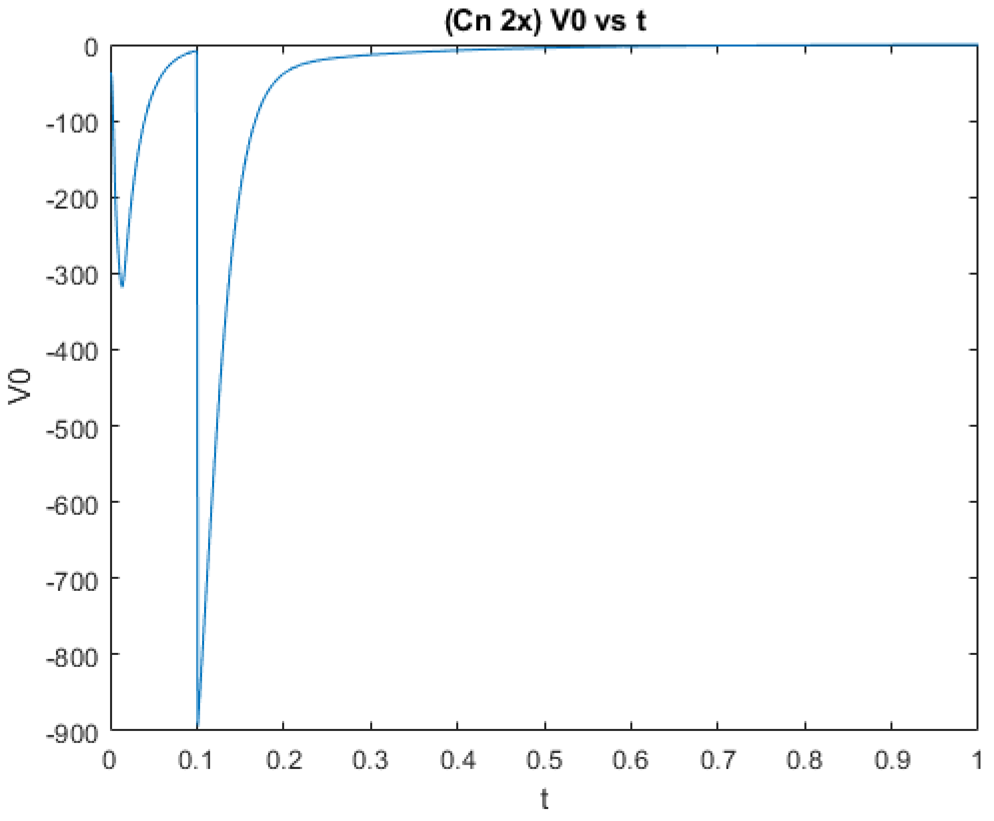

(Q3) Frequency V0
×10^4
Frequency Content
t
6
4
2
0
-2
-4
-6
-8
-10
-12
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

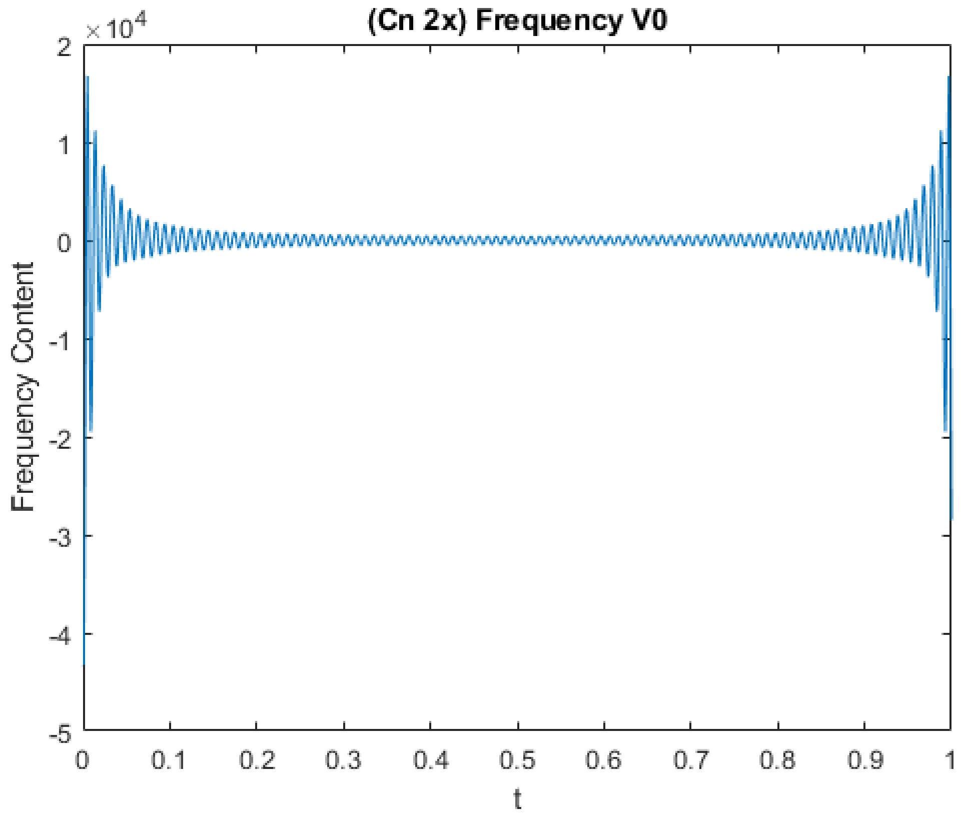
In this section, 3 different values of Cn will be used to evaluate how it affects the simulation, the first case will increase the value of Cn by a factor the 2, the second case by a factor of 5, and the third case by a factor of 10. The C matrix is updated with the new values of Cn.

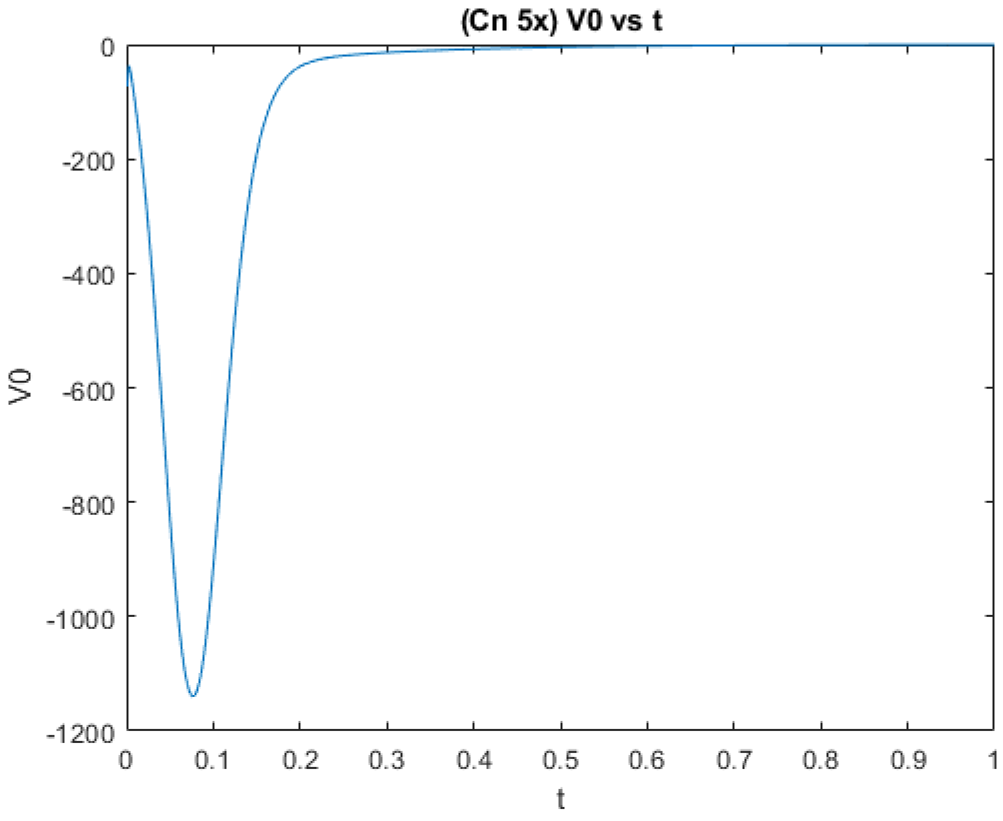
```
% First case 2x
Cn = 2 * 0.00001;
Cmat(4,4) = Cn;

% Random number for In
In = (1 / (0.03 * sqrt(2*pi))) * (exp((-1/2) * ((((rand /10) -
 0.001)/0.03)^2))) / 1000;

G = [1 0 0 0 0 0 0;...
    -G2 (G1+G2) -1 0 0 0 0;...
    0 1 0 -1 0 0 0;...
    0 0 -1 G3 In 0 0;...
    0 0 0 0 -a 1 0;...
    0 0 0 G3 -1 0 0;...
    0 0 0 0 0 -G4 (G4 +G0)];

% Calculation
delta_t = 0.001;
counter = 1;
Vin = 0;
for t = 0.001 : 0.01 : 1
    Vold = Vnew;
    Vin = (1 / (0.03 * sqrt(2*pi))) * (exp((-1/2) *
 (((t-0.06)/0.03)^2)));
    F(1) = Vin;
    A = (Cmat ./ delta_t) + G;
    Vnew = inv(A) * ((Cmat * (Vold / delta_t)) + F);
    ALLv0(counter) = Vnew(7);
    counter = counter + 1;
end

t = 0.001 : 0.001 : 1;
figure(19);
plot(t, ALLv0);
title('(Cn 2x) V0 vs t');
ylabel('V0');
xlabel('t');

Y = fft(ALLv0);
figure(20);
plot(t, Y);
title('(Cn 2x) Frequency V0');
xlabel('t');
ylabel('Frequency Content');

% Second case 5x
Cn = 5 * 0.00001;
Cmat(4,4) = Cn;

% Random number for In
```

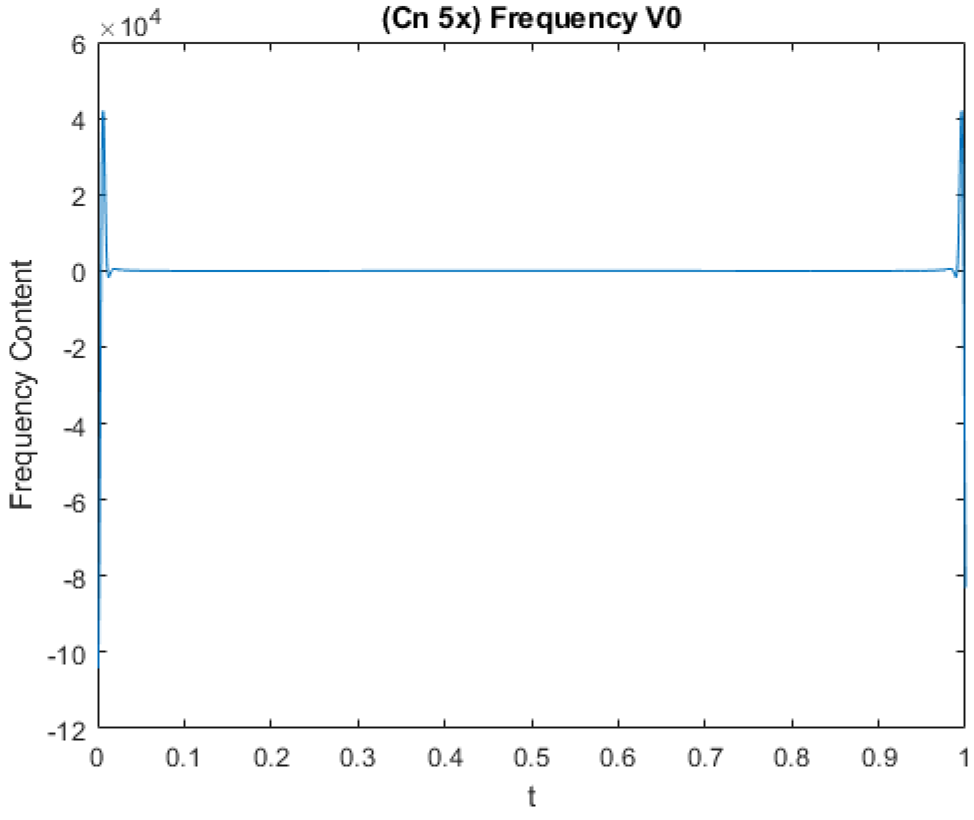
```
In = (1 / (0.03 * sqrt(2*pi))) * (exp((-1/2) * ((((rand /10) -
 0.001)/0.03)^2))) / 1000;

G = [1 0 0 0 0 0 0;...
    -G2 (G1+G2) -1 0 0 0 0;...
    0 1 0 -1 0 0 0;...
    0 0 -1 G3 In 0 0;...
    0 0 0 0 -a 1 0;...
    0 0 0 G3 -1 0 0;...
    0 0 0 0 0 -G4 (G4 +G0)];

% Calculation
delta_t = 0.001;
counter = 1;
Vin = 0;
for t = 0.001 : 0.001 : 1
    Vold = Vnew;
    Vin = (1 / (0.03 * sqrt(2*pi))) * (exp((-1/2) *
 (((t-0.06)/0.03)^2)));
    F(1) = Vin;
    A = (Cmat ./ delta_t) + G;
    Vnew = inv(A) * ((Cmat * (Vold / delta_t)) + F);
    ALLv0(counter) = Vnew(7);
    counter = counter + 1;
end

t = 0.001 : 0.001 : 1;
figure(21);
plot(t, ALLv0);
title('(Cn 5x) V0 vs t');
ylabel('V0');
xlabel('t');

Y = fft(ALLv0);
figure(22);
plot(t, Y);
title('(Cn 5x) Frequency V0');
xlabel('t');
ylabel('Frequency Content');

% Third case 10x
Cn = 10 * 0.00001;
Cmat(4,4) = Cn;

% Random number for In
In = (1 / (0.03 * sqrt(2*pi))) * (exp((-1/2) * ((((rand /10) -
 0.001)/0.03)^2))) / 1000;

G = [1 0 0 0 0 0 0;...
    -G2 (G1+G2) -1 0 0 0 0;...
    0 1 0 -1 0 0 0;...
    0 0 -1 G3 In 0 0;...
    0 0 0 0 -a 1 0;...
    0 0 0 G3 -1 0 0;...
    0 0 0 0 0 -G4 (G4 +G0)];

% Calculation
delta_t = 0.001;
counter = 1;
Vin = 0;
for t = 0.001 : 0.001 : 1
    Vold = Vnew;
    Vin = (1 / (0.03 * sqrt(2*pi))) * (exp((-1/2) *
 (((t-0.06)/0.03)^2)));
    F(1) = Vin;
    A = (Cmat ./ delta_t) + G;
    Vnew = inv(A) * ((Cmat * (Vold / delta_t)) + F);
    ALLv0(counter) = Vnew(7);
    counter = counter + 1;
end

t = 0.001 : 0.001 : 1;
figure(23);
plot(t, ALLv0);
title('(Cn 10x) V0 vs t');
ylabel('V0');
xlabel('t');

Y = fft(ALLv0);
figure(24);
plot(t, Y);
title('(Cn 10x) Frequency V0');
xlabel('t');
ylabel('Frequency Content');
```

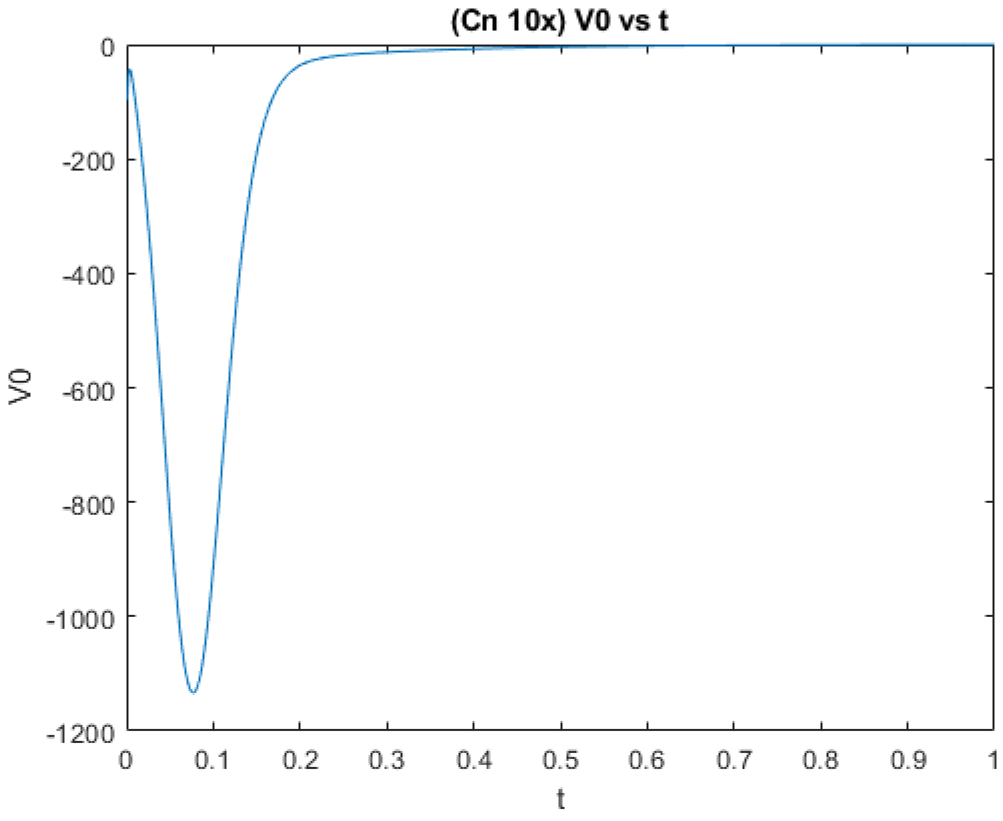
```
Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored
```

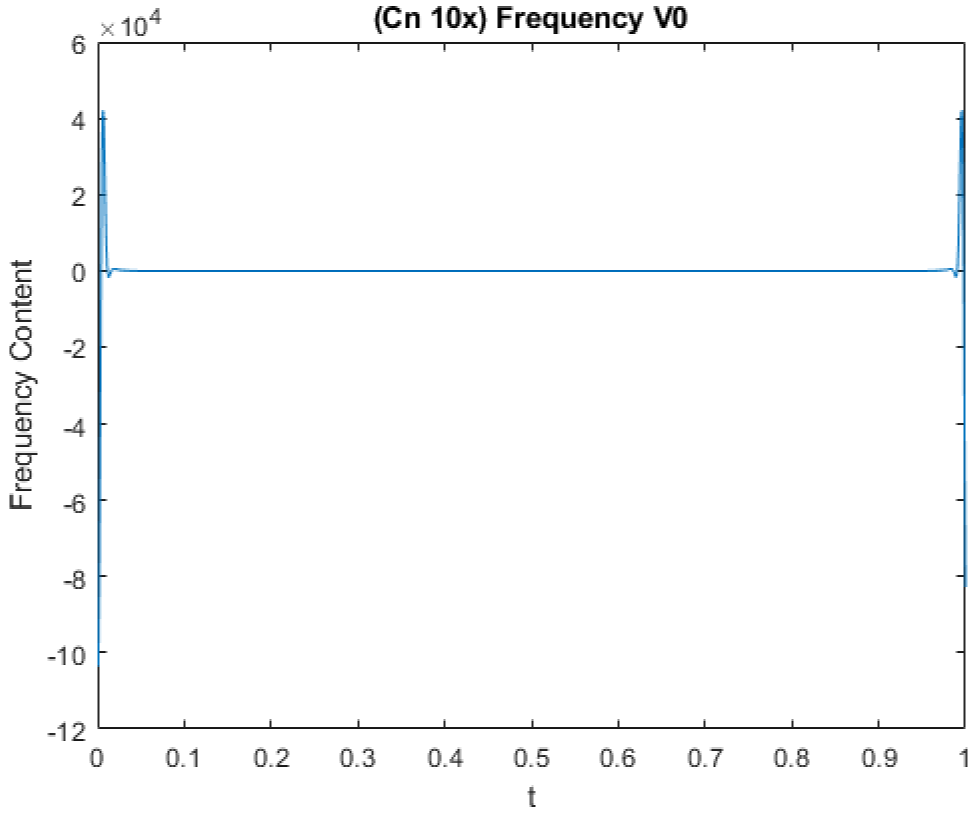

(Cn 2x) V0 vs t
V0
t
0
-100
-200
-300
-400
-500
-600
-700
-800
-900
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


(Cn 2x) Frequency V0
×10^4
Frequency Content
t
2
1
0
-1
-2
-3
-4
-5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


(Cn 5x) V0 vs t
V0
t
0
-200
-400
-600
-800
-1000
-1200
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


(Cn 5x) Frequency V0
×10^4
Frequency Content
t
6
4
2
0
-2
-4
-6
-8
-10
-12
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

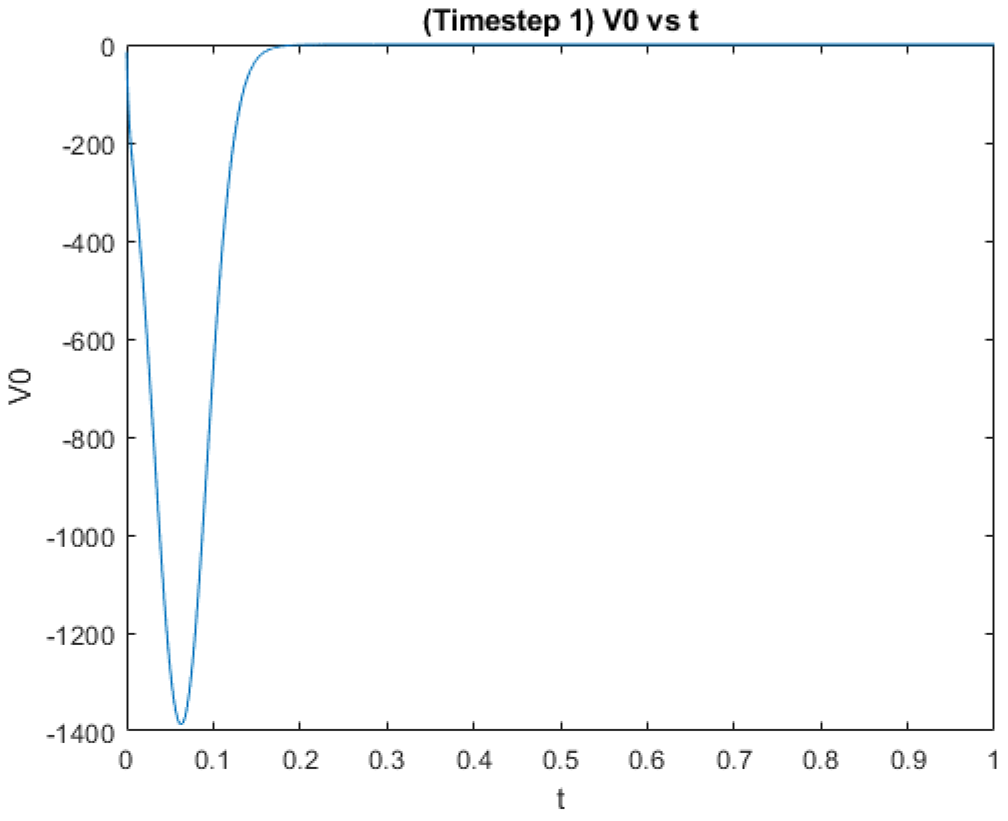

(Cn 10x) V0 vs t
V0
t
0
-200
-400
-600
-800
-1000
-1200
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

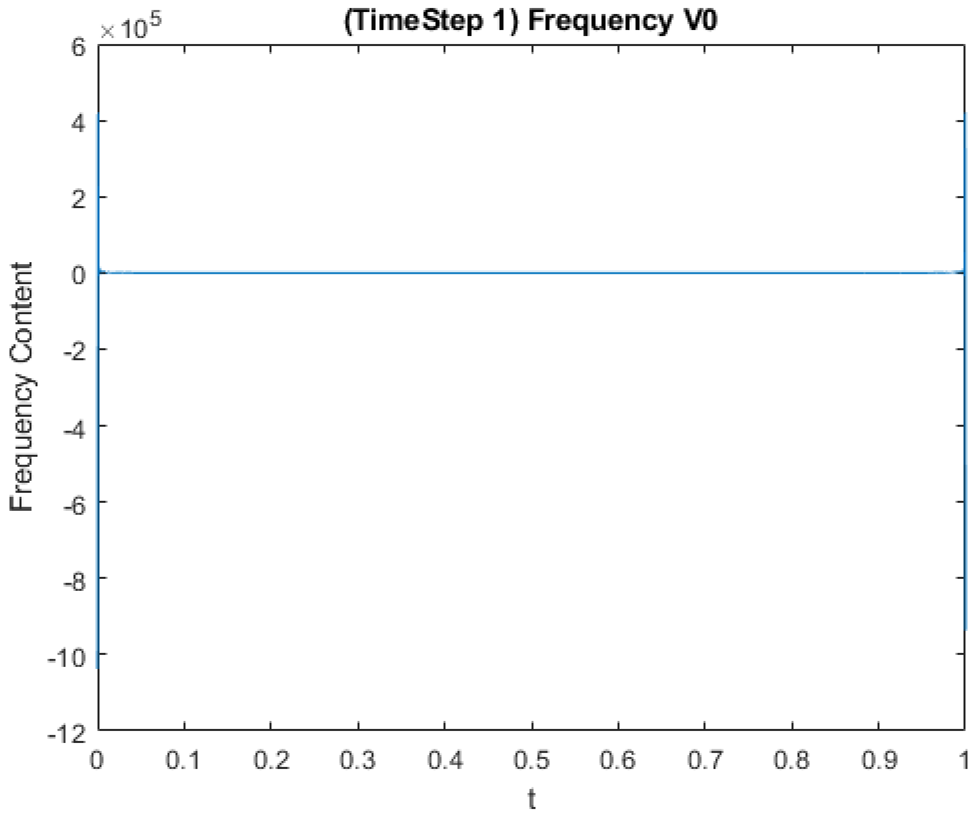

(Cn 10x) Frequency V0
×10^4
Frequency Content
t
6
4
2
0
-2
-4
-6
-8
-10
-12
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

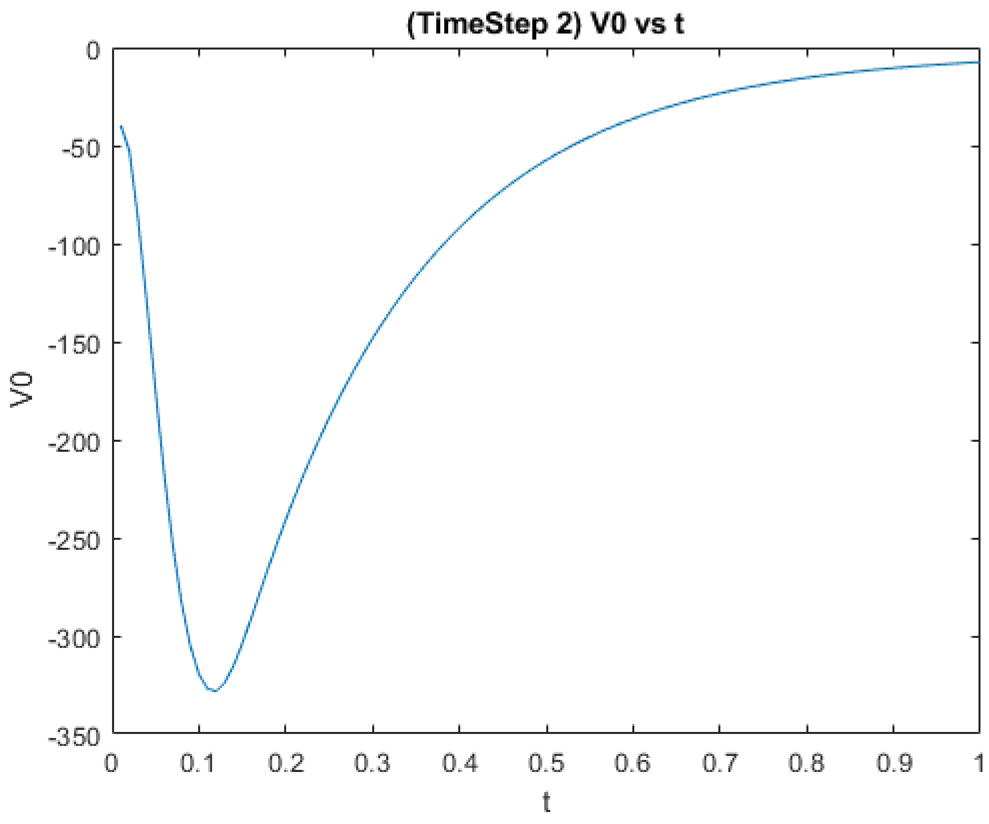
This section observes the simulation by using 2 different timesteps, the first of which increases the timesteps by a factor of 10, and the second of which decreases it by a factor of 10.

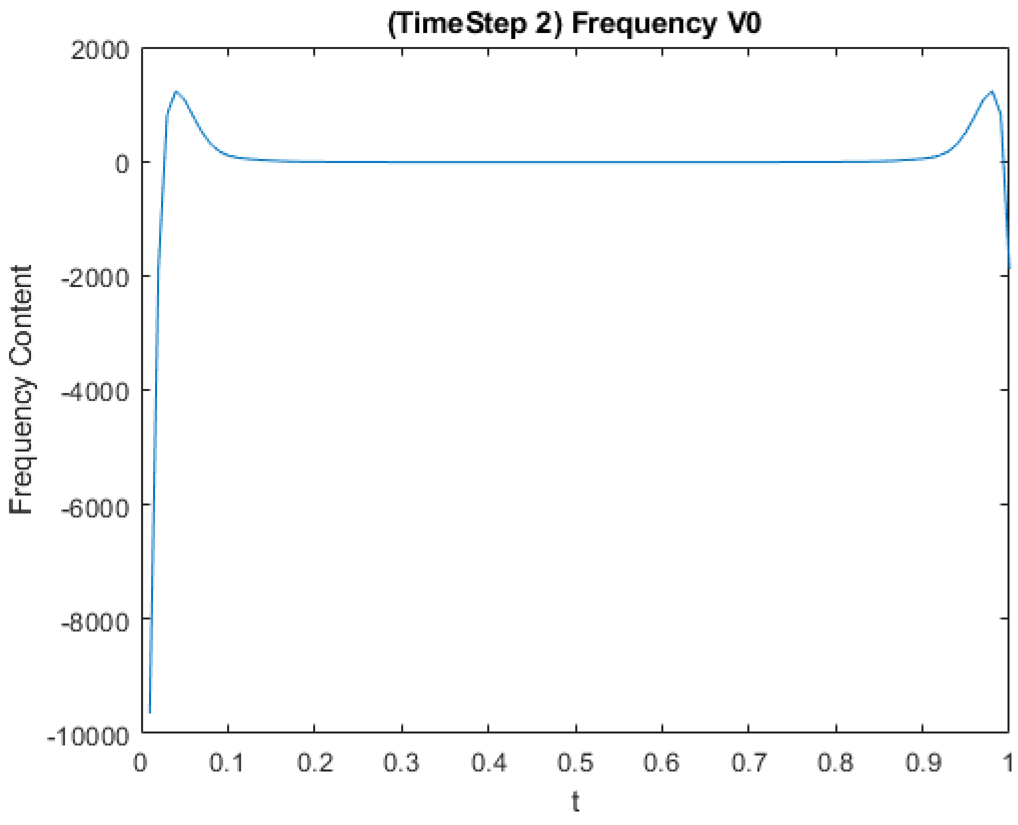
```
% First case 10x MORE
Cn = 0.00001;
Cmat(4,4) = Cn;
ALLv0 = zeros(1, 10000);

% Random number for In
In = (1 / (0.03 * sqrt(2*pi))) * (exp((-1/2) * ((((rand /10) -
 0.001)/0.03)^2))) / 1000;

G = [1 0 0 0 0 0 0;...
    -G2 (G1+G2) -1 0 0 0 0;...
    0 1 0 -1 0 0 0;...
    0 0 -1 G3 In 0 0;...
    0 0 0 0 -a 1 0;...
    0 0 0 G3 -1 0 0;...
    0 0 0 0 0 -G4 (G4 +G0)];

% Calculation
delta_t = 0.001;
counter = 1;
Vin = 0;
for t = 0.0001 : 0.0001 : 1
    Vold = Vnew;
    Vin = (1 / (0.03 * sqrt(2*pi))) * (exp((-1/2) *
 (((t-0.06)/0.03)^2)));
    F(1) = Vin;
    A = (Cmat ./ delta_t) + G;
    Vnew = inv(A) * ((Cmat * (Vold / delta_t)) + F);
    ALLv0(counter) = Vnew(7);
    counter = counter + 1;
end

t = 0.0001 : 0.0001 : 1;
figure(25);
plot(t, ALLv0);
title('(Timestep 1) V0 vs t');
ylabel('V0');
xlabel('t');

Y = fft(ALLv0);
figure(26);
plot(t, Y);
title('(TimeStep 1) Frequency V0');
xlabel('t');
ylabel('Frequency Content');

% Second case 10x LESS
Cn = 0.00001;
Cmat(4,4) = Cn;
ALLv0 = zeros(1, 100);
```

```
% Random number for In
In = (1 / (0.03 * sqrt(2*pi))) * (exp((-1/2) * ((((rand /10) -
 0.001)/0.03)^2))) / 1000;

G = [1 0 0 0 0 0 0;...
    -G2 (G1+G2) -1 0 0 0 0;...
    0 1 0 -1 0 0 0;...
    0 0 -1 G3 In 0 0;...
    0 0 0 0 -a 1 0;...
    0 0 0 G3 -1 0 0;...
    0 0 0 0 0 -G4 (G4 +G0)];

% Calculation
delta_t = 0.001;
counter = 1;
Vin = 0;
for t = 0.01 : 0.01 : 1
    Vold = Vnew;
    Vin = (1 / (0.03 * sqrt(2*pi))) * (exp((-1/2) *
 (((t-0.06)/0.03)^2)));
    F(1) = Vin;
    A = (Cmat ./ delta_t) + G;
    Vnew = inv(A) * ((Cmat * (Vold / delta_t)) + F);
    ALLv0(counter) = Vnew(7);
    counter = counter + 1;
end

t = 0.01 : 0.01 : 1;
figure(27);
plot(t, ALLv0);
title('(TimeStep 2) V0 vs t');
ylabel('V0');
xlabel('t');

Y = fft(ALLv0);
figure(28);
plot(t, Y);
title('(TimeStep 2) Frequency V0');
xlabel('t');
ylabel('Frequency Content');

Warning: Imaginary parts of complex X and/or Y arguments ignored
Warning: Imaginary parts of complex X and/or Y arguments ignored
```


(Timestep 1) V0 vs t
V0
t
0
-200
-400
-600
-800
-1000
-1200
-1400
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


(TimeStep 1) Frequency V0
×10^5
Frequency Content
t
6
4
2
0
-2
-4
-6
-8
-10
-12
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


(TimeStep 2) V0 vs t
V0
t
0
-50
-100
-150
-200
-250
-300
-350
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


(TimeStep 2) Frequency V0
Frequency Content
t
2000
0
-2000
-4000
-6000
-8000
-10000
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

PART 4 (NON-LINEARITY)

4A

If the voltage source was modeled differently, the following would occur. The present voltage source is modeled using the current I3 and the coefficient 'a', which are included in the G matrix. Adding new terms to the equation for the voltage that involve the current being squared or cubed introduces non-linearity into the problem. The presence of non-linearity prevents the use of an exact numerical solution, therefore the best possible solution is found through iterative calculations. The simulator must introduce a new vector to accomodate the non-linear terms aside from the linear terms. The simulator must also involve new mathematical techniques of iteration to find a solution that differs from the ideal solution by a small, specified tolerance level.

4B

The following section describes the changes and the implementation. In order to allow the simulator to work with non-linearity, a 'B' vector must be created that will store the values of the non-linear terms of the circuit. This new 'B' vector will be an additional term that is summed with the other terms in the equation for the voltage. After rearranging the formulas to include 'B', a new mathematical technique is used to convert the non-linearities into a suitable form to apply linear operations. This technique involves producing the Jacobian matrix, which, in a very general and basic sense, evaluates how the non-linearities change with respect to the voltage. Then, the calculation of the values of the V vector greatly resembles the techniques employed in previous sections, however, each iteration will use the Jacobian matrix to evaluate again the value of Vn. Later, the formula for the V vector will again be used, this time with the inclusion of the values that are present in the 'B' matrix.

4C

This is the implementation of the above code. There is currently no implementation.

Published with MATLAB® R2017b