## Table of Contents



```
% assignment4.m:
%
```

# ELEC4700, Assignment 4


Author: Jacob Godin Date: 2019/04/21


```
%--------------------------------------------------------------------------

global G C b;

% Generate G, C, and b matrices
netlist
```

# Part 1

```
%--------------------------------------------------------------------------
% a) G, C, and F (b matrix) matrices
%--------------------------------------------------------------------------
fprintf("The G, C, and F matrices:");
G
C
b

%--------------------------------------------------------------------------
% b) Plot of Vin vs Vout, DC case
%--------------------------------------------------------------------------

% For DC analysis, C = 0

Nrpt = 10000; % Number of points
OutputNode = 6;
Vin = linspace(-10,10,Nrpt);

Vout = zeros(1,Nrpt);
V3 = zeros(1,Nrpt);

for i=1:Nrpt
    b(7) = Vin(i);
```

```
    x = G\b;

    Vout(i) = x(OutputNode);
    V3(i) = x(3);
end

figure('Name','Vin-vs-Vout-and-V3-Response');
plot(Vin, Vout,'LineWidth',3);
hold on;
plot(Vin, V3,'LineWidth',3);
grid;
title('Vin vs. Vout/V3', 'FontSize',12);
xlabel('Vin  (Volts)','FontSize',20);
ylabel('Vout  (Volts)','FontSize',20);
legend('Vout', 'V3');

% The relationship between Vout, Vin, and V3 is linear for DC.

%---------------------------------------------------------------------------
% b) Plot of Vout as a function of w, AC case
%---------------------------------------------------------------------------

f = linspace(0,100,Nrpt);

V1 = zeros(1,Nrpt);

for i=1:Nrpt
    w = 2*pi*f(i);
    s=1i*w;

    A = G+s.*C;

    x = A\b;

    Vout(i) = x(OutputNode);
    V1(i) = x(1);
end

amplitude = Vout./V1;
gaindB = 20*log10(amplitude);

figure('Name','Vout-Function-of-w');
semilogx(f, Vout,'LineWidth',3);
grid;
title('Vout as a Function of w', 'FontSize',12);
xlabel('w','FontSize',20);
ylabel('Vout  (Volts)','FontSize',20);

figure('Name','Gain-Function-of-w');
plot(f, gaindB,'LineWidth',3);
grid;
title('Gain as a Function of w', 'FontSize',12);
xlabel('w','FontSize',20);
```

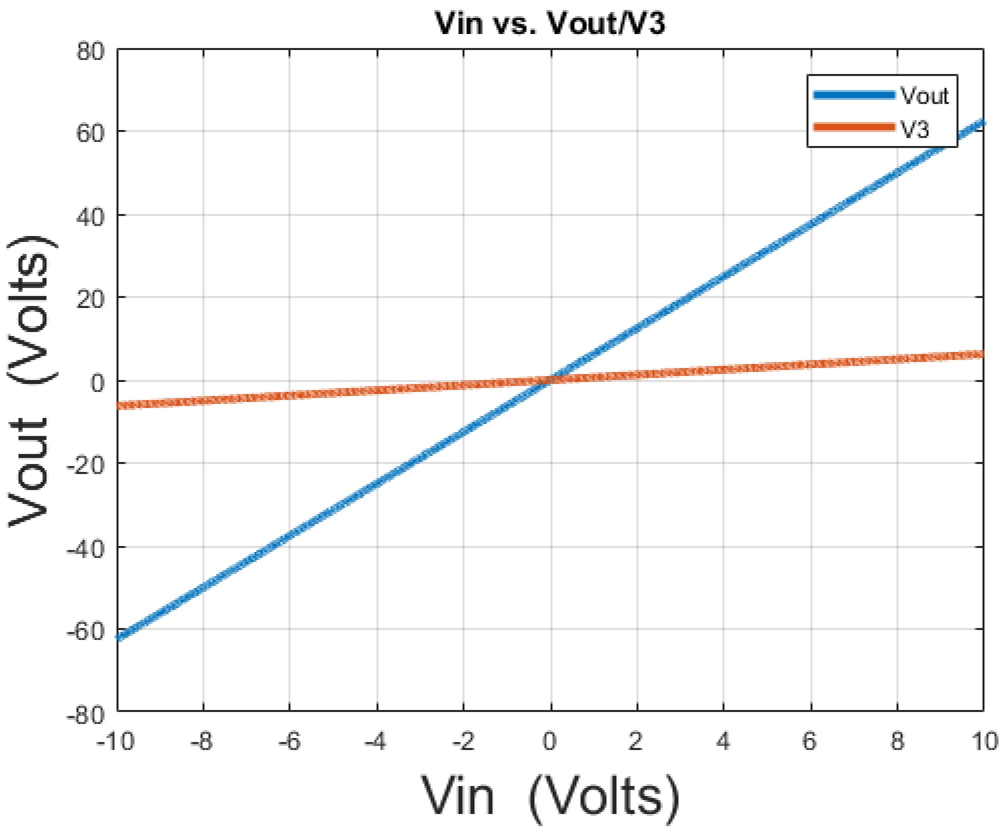
```
ylabel('Gain  (dB)','FontSize',20);

%-----------------------------------------------------------------------------
% b) Plot of Vout as a Function Random Pertubations of C
%-----------------------------------------------------------------------------

d = size(G,1);
std = 0.05;
x_plot = zeros(1,Nrpt);
count = 0;

for i=1:Nrpt
    w = 2*pi*f(i);
    s=1i*w;

    for row=1:d
        for col=1:d
            Cn = pi+rand()*std;
            C(row,col) = Cn;
        end
    end

    x_plot(i)=count+1;

    A = G+s.*C;

    x = A\b;

    Vout(i) = x(OutputNode);
    V1(i) = x(1);
end

amplitude = Vout./V1;
gaindB = 20*log10(amplitude);

figure('Name','Gain-Function-of-Random-Pert');
plot(1:Nrpt, gaindB,'LineWidth',1);
grid;
title('Gain as a Function of Random Pertubation Count in C
 Matrix', 'FontSize',12);
xlabel('Pertubation Count','FontSize',20);
ylabel('Gain (dB)','FontSize',20);

figure('Name','Hist-Gain-Function-of-Random-Pert');
histogram(abs(gaindB),25)
grid;
title('Histogram', 'FontSize',12);
xlabel('Gain (dB)','FontSize',20);
ylabel('Count','FontSize',20);
```

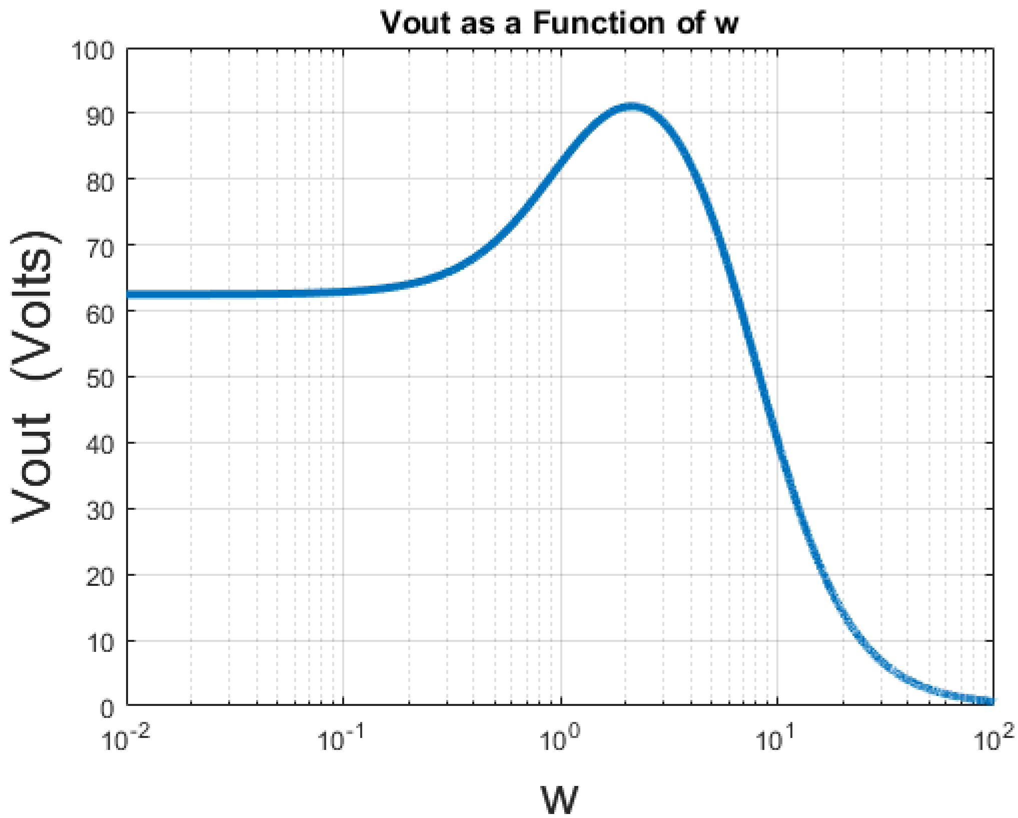
*The G, C, and F matrices:*
*G =*

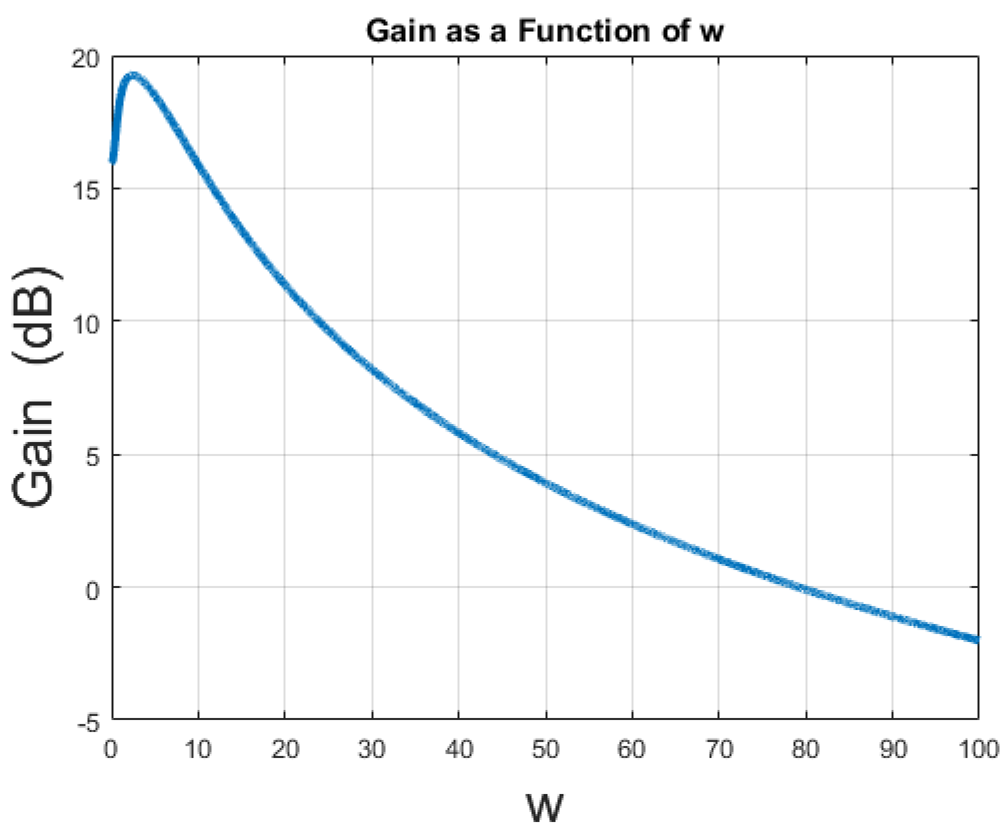
*Columns 1 through 7*

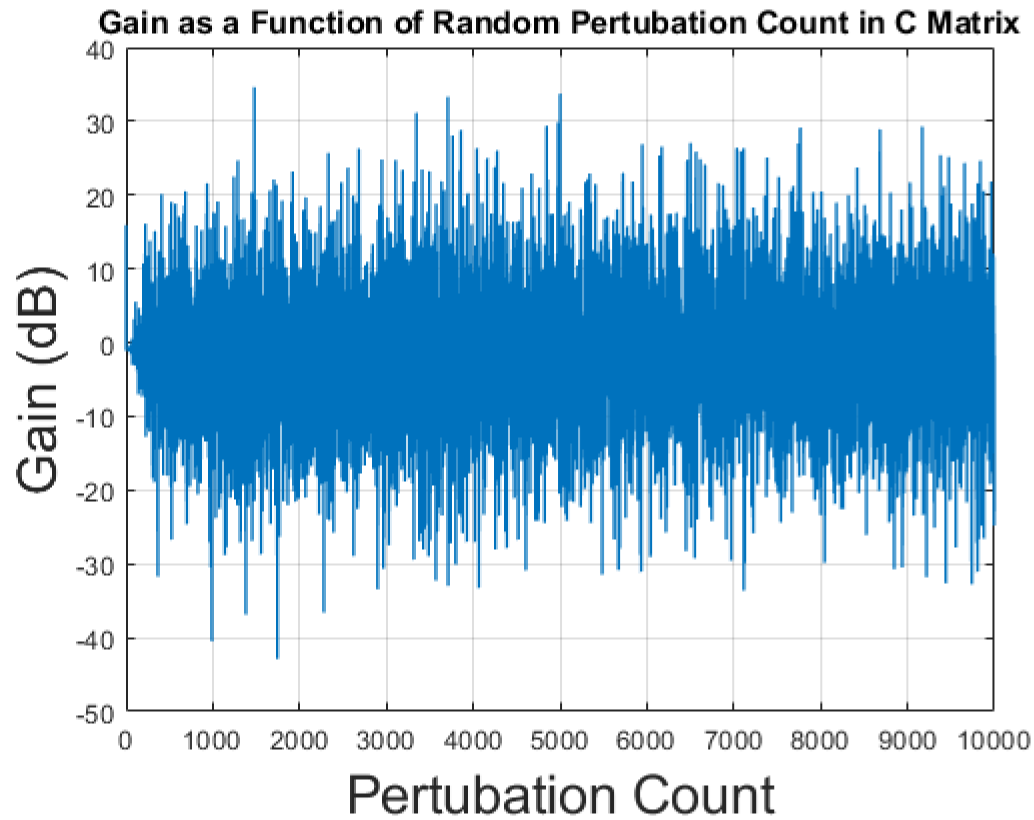
```
    1.0000   -1.0000         0         0         0         0    1.0000
   -1.0000    1.5000         0         0         0         0         0
         0         0    0.1000   -0.1000         0         0         0
         0         0   -0.1000    0.1000         0         0         0
         0  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

  Columns 8 through 10

         0         0         0
    1.0000         0         0
   -1.0000         0         0
         0    1.0000         0
         0         0   -1.0000
         0         0         0
         0         0         0
         0         0         0
         0 -100.0000         0
         0         0         0


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

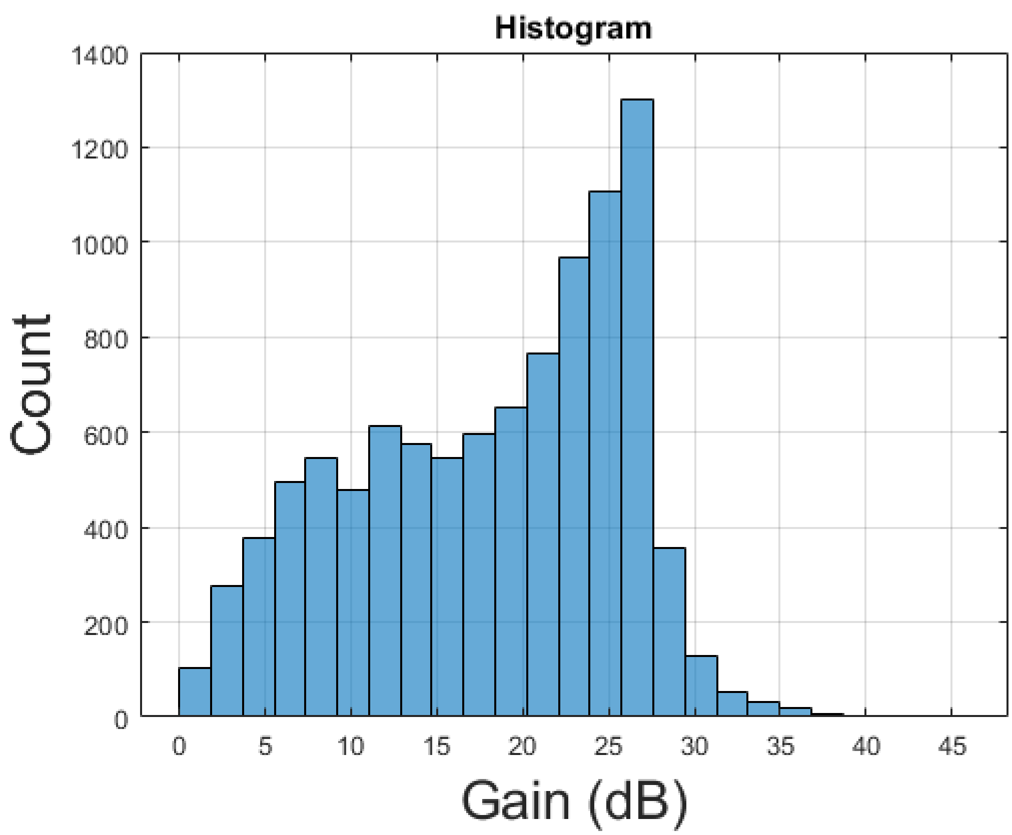
```
b =

     0
     0
     0
     0
     0
     0
     1
     0
     0
     0

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
```




Vout as a Function of w
Vout (Volts)
100
90
80
70
60
50
40
30
20
10
0
$10^{-2}$
$10^{-1}$
$10^{0}$
$10^{1}$
$10^{2}$
w


Gain as a Function of w
Gain (dB)
20
15
10
5
0
-5
0
10
20
30
40
50
60
70
80
90
100
w


Gain as a Function of Random Pertubation Count in C Matrix
40
30
20
10
0
-10
-20
-30
-40
-50
Gain (dB)
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Pertubation Count


Histogram
1400
1200
1000
800
600
400
200
0
Count
0
5
10
15
20
25
30
35
40
45
Gain (dB)

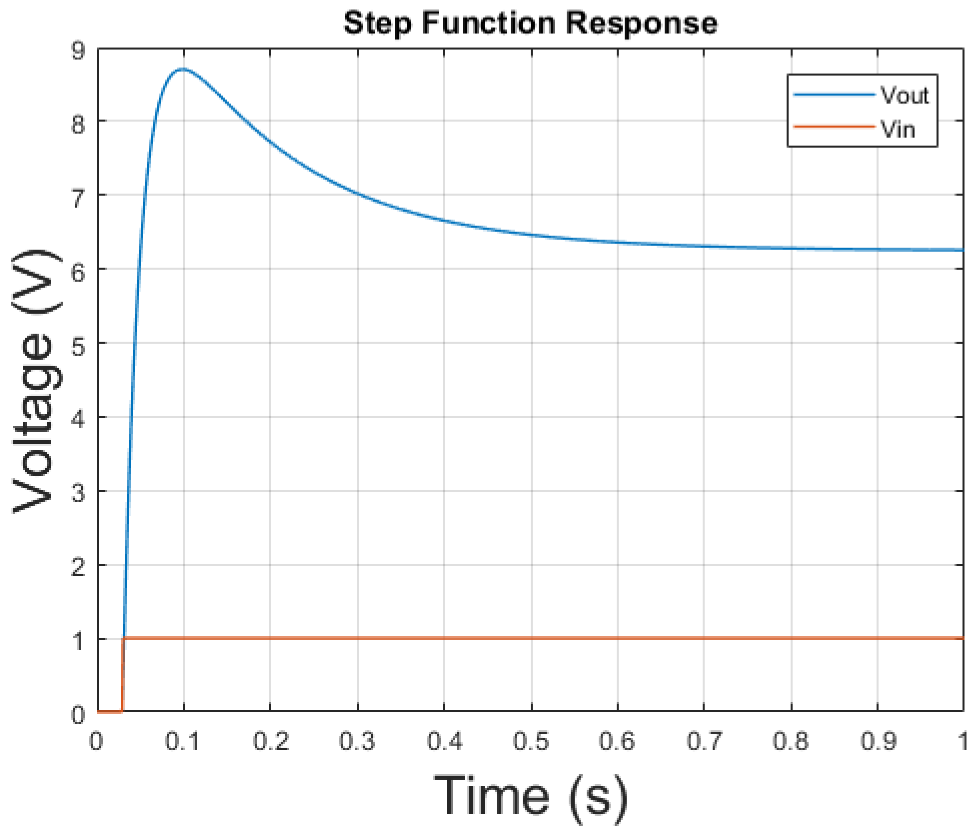
# Part 2: Transient Circuit Simulation

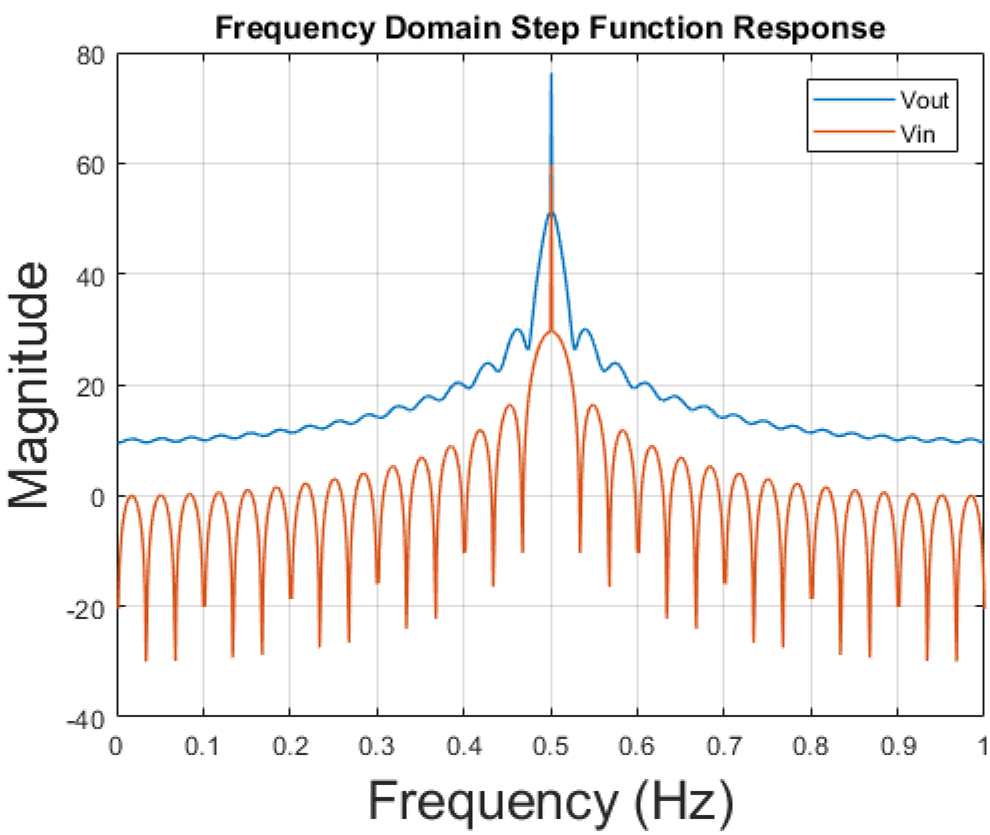
```
% The equation can be represented in the finite difference forumlation
 by
% the following equation:
%
% C*(Vnew - Vold)/delta_t + G*V = F
% Vnew(C/delta_t + G) = F + (C/delta_t)*Vold

%----------------------------------------------------------------------
% d) Circuit Simulation
%----------------------------------------------------------------------
netlist % regen circuit

% Step function
Nrpt = 1000;
t_final = 1;
delta_t = t_final/Nrpt;
time_vec = linspace(0,1,Nrpt);

VinStep = zeros(1,Nrpt);
Vold = zeros(length(b),1);
VoutStep = zeros(length(b),1);

OutputNode = 6;
t=0;

for i=1:Nrpt
    if t >= 0.03
        VinStep(i) = 1;
    else
        VinStep(i) = 0;
    end

    b(7) = VinStep(i);

    A = C/delta_t + G;
    B = b + C*Vold/delta_t;
    x = A\B;

    VoutStep(i) = x(OutputNode);

    Vold = x;

    t = t+delta_t;
end

figure('Name','Step-Func-Response');
plot(time_vec, VoutStep,'LineWidth',1);
grid;
hold on;
plot(time_vec, VinStep,'LineWidth',1);
title('Step Function Response', 'FontSize',12);
```

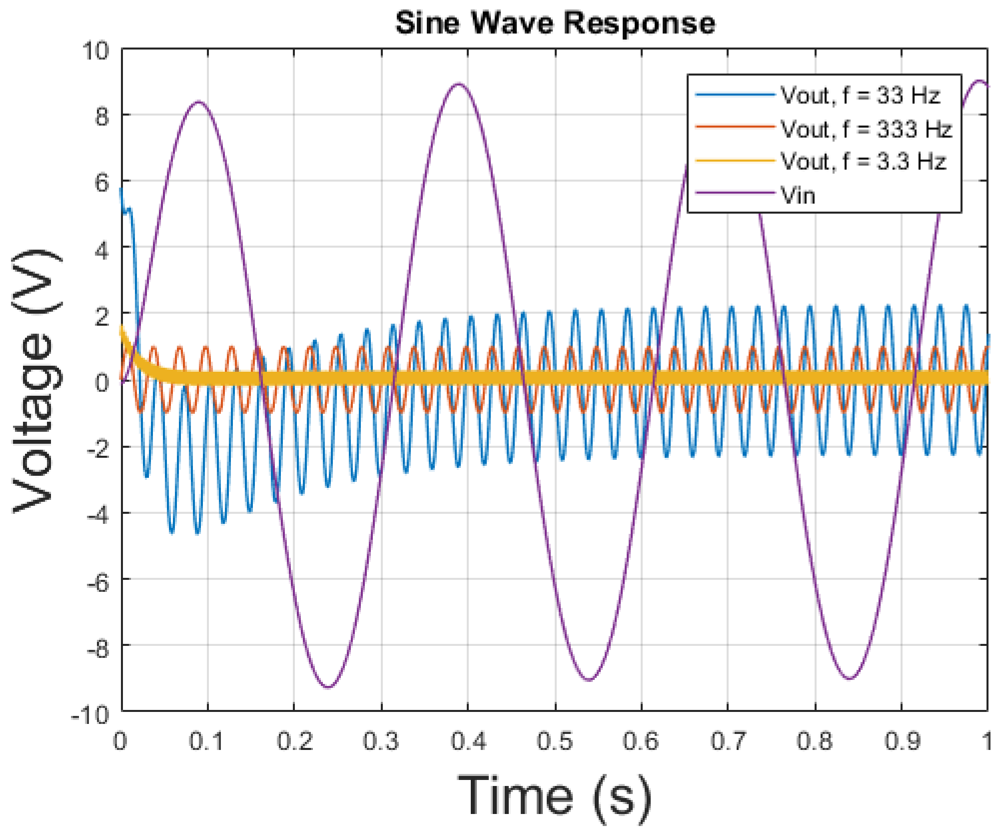
```
xlabel('Time (s)','FontSize',20);
ylabel('Voltage (V)','FontSize',20);
legend('Vout','Vin');

F = abs(fftshift(fft(VoutStep)));
magVout = 20*log10(F);

figure('Name','Freq-Response');
plot((1:length(F))/Nrpt, magVout,'LineWidth',1);
grid;
hold on;

F = abs(fftshift(fft(VinStep)));
magVin = 20*log10(F);

plot((1:length(F))/Nrpt, magVin,'LineWidth',1);
title('Frequency Domain Step Function Response', 'FontSize',12);
xlabel('Frequency (Hz)','FontSize',20);
ylabel('Magnitude','FontSize',20);
legend('Vout','Vin');

% Sine function
netlist

f = 1/0.03;

t=0;
VinSin1 = zeros(1,Nrpt);
VoutSin1 = zeros(1,Nrpt);
for i=1:Nrpt
    VinSin1(i) = sin(2*pi*f*t);

    b(7) = VinSin1(i);

    A = C/delta_t + G;
    B = b + C*Vold/delta_t;
    x = A\B;

    VoutSin1(i) = x(OutputNode);

    Vold = x;

    t = t+delta_t;
end

figure('Name','Sin-Func-Response');
plot(time_vec, VoutSin1,'LineWidth',1);
grid;
hold on;
plot(time_vec, VinSin1,'LineWidth',1);
hold on;

f = 1/0.003;
```

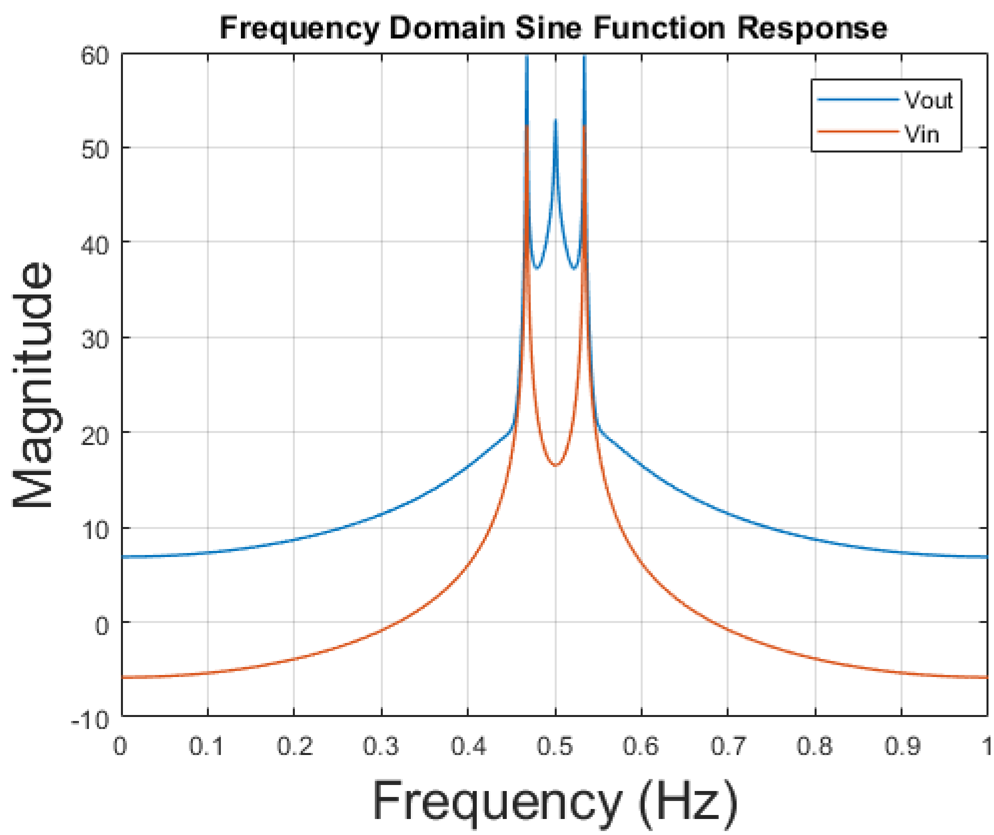
```
t=0;
VinSin2 = zeros(1,Nrpt);
VoutSin2 = zeros(1,Nrpt);
for i=1:Nrpt
    VinSin2(i) = sin(2*pi*f*t);

    b(7) = VinSin2(i);

    A = C/delta_t + G;
    B = b + C*Vold/delta_t;
    x = A\B;

    VoutSin2(i) = x(OutputNode);

    Vold = x;

    t = t+delta_t;
end
plot(time_vec, VoutSin2,'LineWidth',1);
hold on;

f = 1/0.3;

t=0;
VinSin3 = zeros(1,Nrpt);
VoutSin3 = zeros(1,Nrpt);
for i=1:Nrpt
    VinSin3(i) = sin(2*pi*f*t);

    b(7) = VinSin3(i);

    A = C/delta_t + G;
    B = b + C*Vold/delta_t;
    x = A\B;

    VoutSin3(i) = x(OutputNode);

    Vold = x;

    t = t+delta_t;
end
plot(time_vec, VoutSin3,'LineWidth',1);
title('Sine Wave Response', 'FontSize',12);
xlabel('Time (s)','FontSize',20);
ylabel('Voltage (V)','FontSize',20);
legend('Vout, f = 33 Hz','Vout, f = 333 Hz','Vout, f = 3.3 Hz','Vin');

F = abs(fftshift(fft(VoutSin1)));
magVout = 20*log10(F);

figure('Name','Freq-Response');
plot((1:length(F))/Nrpt, magVout,'LineWidth',1);
grid;
hold on;
```

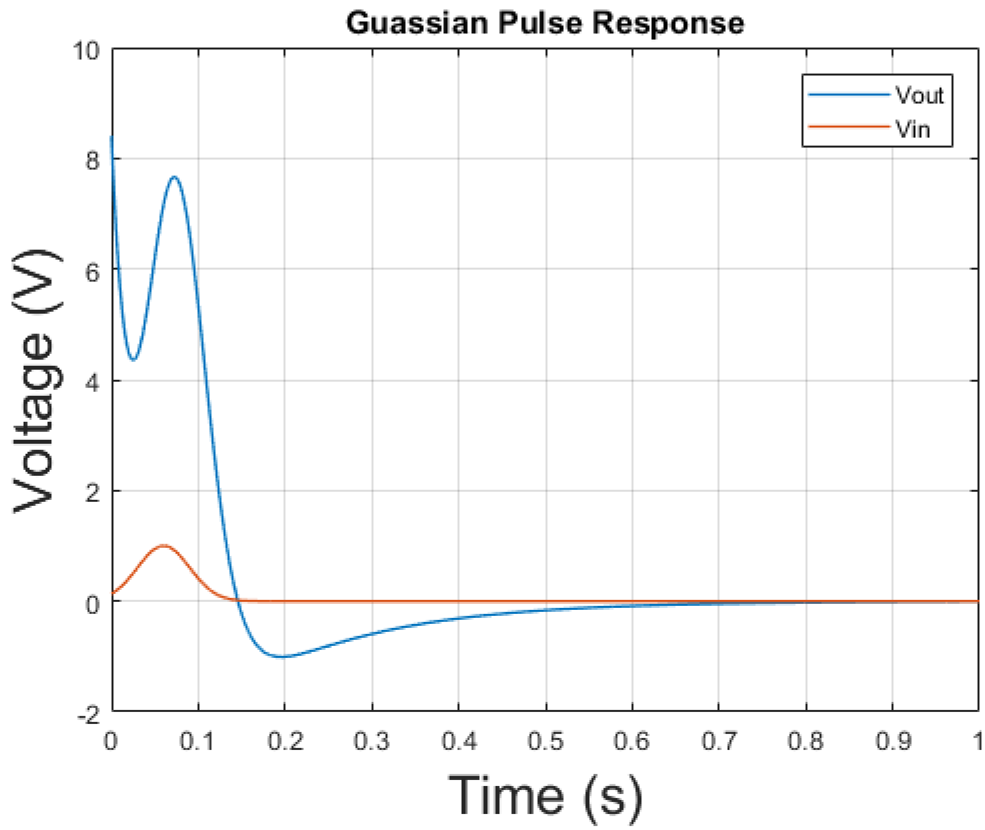
```
F = abs(fftshift(fft(VinSin1)));
magVin = 20*log10(F);

plot((1:length(F))/Nrpt, magVin,'LineWidth',1);
title('Frequency Domain Sine Function Response', 'FontSize',12);
xlabel('Frequency (Hz)','FontSize',20);
ylabel('Magnitude','FontSize',20);
legend('Vout','Vin');

% Gaussian Pulse
netlist
std = 0.03;
mean = 0.06;
VinGaus = gaussmf(linspace(0,1,Nrpt),[std mean]);
VoutGaus = zeros(1,Nrpt);
for i=1:Nrpt
    b(7) = VinGaus(i);

    A = C/delta_t + G;
    B = b + C*Vold/delta_t;
    x = A\B;

    VoutGaus(i) = x(OutputNode);

    Vold = x;

    t = t+delta_t;
end
figure('Name', 'Gaus-Response');
plot(time_vec, VoutGaus,'LineWidth',1);
grid;
hold on;
plot(time_vec, VinGaus,'LineWidth',1);
title('Guassian Pulse Response', 'FontSize',12);
xlabel('Time (s)','FontSize',20);
ylabel('Voltage (V)','FontSize',20);
legend('Vout', 'Vin');
```


Step Function Response
Voltage (V)
Time (s)
Vout
Vin
9
8
7
6
5
4
3
2
1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Frequency Domain Step Function Response
Magnitude
Frequency (Hz)
Vout
Vin
80
60
40
20
0
-20
-40
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Sine Wave Response
Vout, f = 33 Hz
Vout, f = 333 Hz
Vout, f = 3.3 Hz
Vin
Voltage (V)
Time (s)
10
8
6
4
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


Frequency Domain Sine Function Response
Vout
Vin
Magnitude
Frequency (Hz)
60
50
40
30
20
10
0
-10
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

```
netlist % regenerate original C matrix

%--------------------------------------------------------------------------
% a) Add Current Source In to Global MNA Network
%--------------------------------------------------------------------------

cur(3,4,1);

%--------------------------------------------------------------------------
% b) Add Capacitor Cn to Global MNA Network
%--------------------------------------------------------------------------

Cn = 0.0001;
cap(3,4, Cn);

fprintf("The updated C matrix:");
C
%--------------------------------------------------------------------------
% c) Simulation
%--------------------------------------------------------------------------

clear Vout;
```

```
Nrpt = 1000;

% Time variables
t = 1; % s
delta_t = t/Nrpt;
time_vec = linspace(0,1,Nrpt);
% Generate noisy In and Vin
mean = 0.06;
std = 0.03;
InMag = 0.0001;
In = InMag*rand(Nrpt,1);
Vin = gaussmf(linspace(0,1,Nrpt),[std mean]);

Vold = zeros(length(b),1);
Vout = zeros(length(b),1);

OutputNode = 6;

for i=1:Nrpt
    % Overwrite b matrix with noisy Vin
    b(7) = Vin(i);

    % Overwrite b matrix with noisy In
    b(3) = -In(i);
    b(4) = In(i);

    A = C/delta_t + G;
    B = b + C*Vold/delta_t;
    x = A\B;

    Vout(i) = x(OutputNode);
    V1(i) = x(1);

    Vold = x;
end

figure('Name','Gauss-Func-Response');
plot(time_vec, Vout,'LineWidth',1);
grid;
hold on;
plot(time_vec, Vin,'LineWidth',1);
title('Gaussian Function Response', 'FontSize',12);
xlabel('Time (s)','FontSize',20);
ylabel('Voltage (V)','FontSize',20);
legend('Vout','Vin');

% Using fft function to calculate the spectrum on the output

F = abs(fftshift(fft(Vout)));
magVout = 20*log10(F);

figure('Name','Freq-Response');
plot((1:length(F))/Nrpt, magVout,'LineWidth',1);
grid;
```

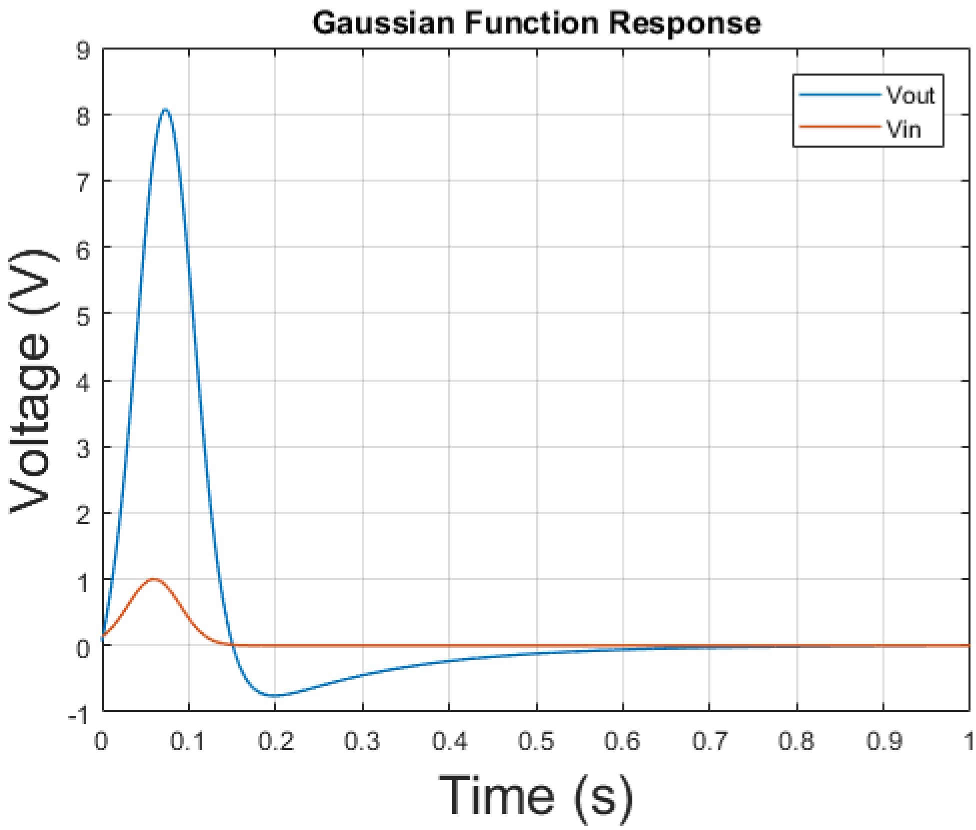
```
hold on;

F = abs(fftshift(fft(Vin)));
magVin = 20*log10(F);

plot((1:length(F))/Nrpt, magVin,'LineWidth',1);
title('Frequency Domain Gaussian Function Response', 'FontSize',12);
xlabel('Frequency (Hz)','FontSize',20);
ylabel('Magnitude','FontSize',20);
legend('Vout','Vin');

% Vary Cn to see how bandwidth changes

% Cn 1
Cn = 0.001;
clear G C b;
netlist;
cap(3,4,Cn);
Vout = generate(G, C, b, Vin, In, Nrpt, delta_t, OutputNode);

F = abs(fftshift(fft(Vout)));
magVout = 20*log10(F);

figure('Name','Freq-Response');
plot((1:length(F))/Nrpt, magVout,'LineWidth',1);
grid;
hold on;

plot((1:length(F))/Nrpt, magVin,'LineWidth',1);
title('Frequency Domain Gaussian Function Response, with Cn =
 0.001', 'FontSize',12);
xlabel('Frequency (Hz)','FontSize',20);
ylabel('Magnitude','FontSize',20);
legend('Vout','Vin');

% Cn 2
Cn = 1e-10;
clear G C b;
netlist;
cap(3,4,Cn);
Vout = generate(G, C, b, Vin, In, Nrpt, delta_t, OutputNode);

F = abs(fftshift(fft(Vout)));
magVout = 20*log10(F);

figure('Name','Freq-Response');
plot((1:length(F))/Nrpt, magVout,'LineWidth',1);
grid;
hold on;

plot((1:length(F))/Nrpt, magVin,'LineWidth',1);
title('Frequency Domain Gaussian Function Response, with Cn =
 0.0000000001', 'FontSize',12);
xlabel('Frequency (Hz)','FontSize',20);
```

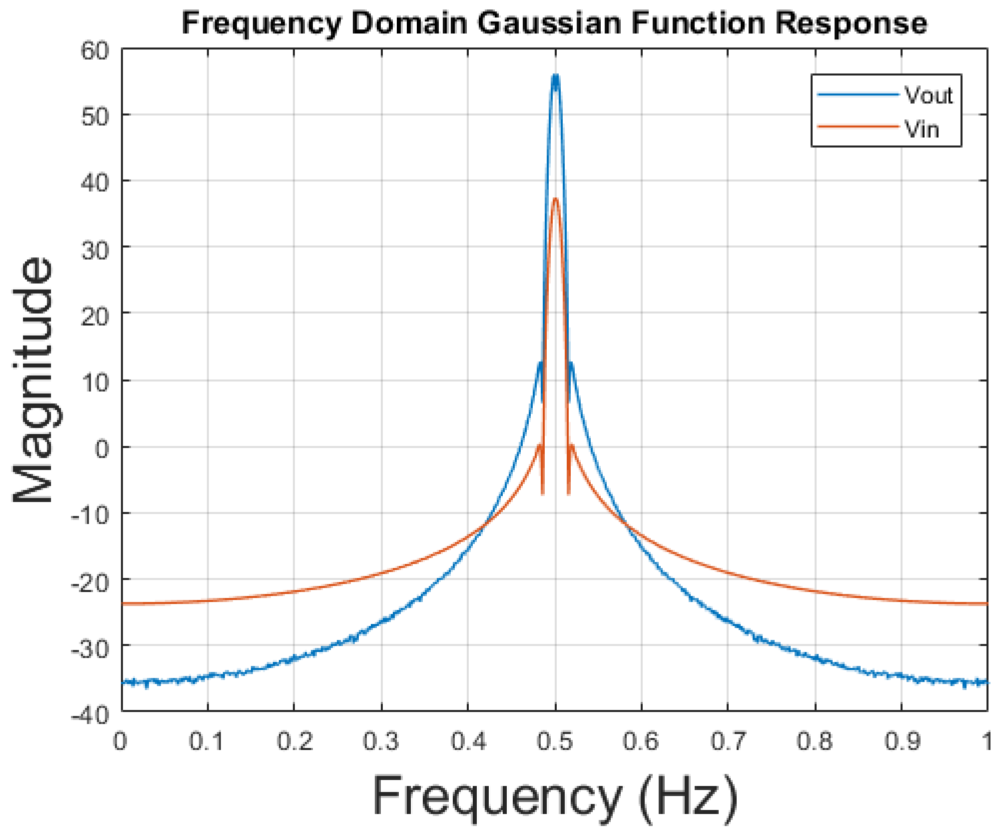
```
ylabel('Magnitude','FontSize',20);
legend('Vout','Vin');

% Cn 3
Cn = 0.01;
clear G C b;
netlist;
cap(3,4,Cn);
Vout = generate(G, C, b, Vin, In, Nrpt, delta_t, OutputNode);

F = abs(fftshift(fft(Vout)));
magVout = 20*log10(F);

figure('Name','Freq-Response');
plot((1:length(F))/Nrpt, magVout,'LineWidth',1);
grid;
hold on;

plot((1:length(F))/Nrpt, magVin,'LineWidth',1);
title('Frequency Domain Gaussian Function Response, with Cn =
 0.01', 'FontSize',12);
xlabel('Frequency (Hz)','FontSize',20);
ylabel('Magnitude','FontSize',20);
legend('Vout','Vin');

% Decreasing Cn produced more noise in the output freqency response.

% Vary timestep
Cn = 0.00001;
clear G C b;
netlist;
cap(3,4,Cn);
Nrpt = 10000;
delta_t = 1/Nrpt;
In = InMag*rand(Nrpt,1);
Vin = gaussmf(linspace(0,1,Nrpt),[std mean]);
Vout = generate(G, C, b, Vin, In, Nrpt, delta_t, OutputNode);
time_vec = linspace(0,1,Nrpt);
figure('Name','Gauss-Func-Response');
plot(time_vec, Vout,'LineWidth',1);
grid;
hold on;
title('Gaussian Function Response With 10000 Time
 Steps', 'FontSize',12);
xlabel('Time (s)','FontSize',20);
ylabel('Voltage (V)','FontSize',20);

Nrpt = 100;
delta_t = 1/Nrpt;
In = InMag*rand(Nrpt,1);
Vin = gaussmf(linspace(0,1,Nrpt),[std mean]);
Vout = generate(G, C, b, Vin, In, Nrpt, delta_t, OutputNode);
time_vec = linspace(0,1,Nrpt);
plot(time_vec, Vout,'LineWidth',1);
```

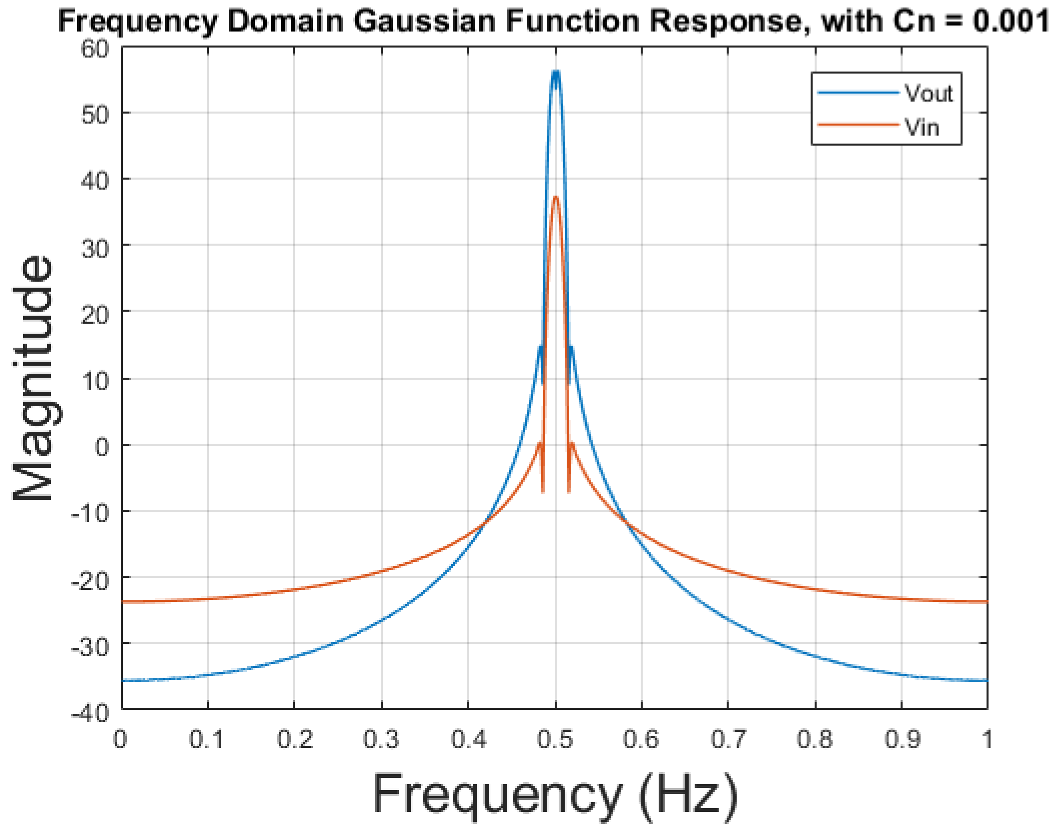
```
hold on;

Nrpt = 1000;
delta_t = 1/Nrpt;
In = InMag*rand(Nrpt,1);
Vin = gaussmf(linspace(0,1,Nrpt),[std mean]);
Vout = generate(G, C, b, Vin, In, Nrpt, delta_t, OutputNode);
time_vec = linspace(0,1,Nrpt);
plot(time_vec, Vout,'LineWidth',1);
legend('10000 Steps','100 Steps', '1000 Steps');

% Varying the timestep did not seem to produce a noticable effect on
 the
% noise of the circuit. A larger timestep produced a less accurate
 model of
% the transient response.
```

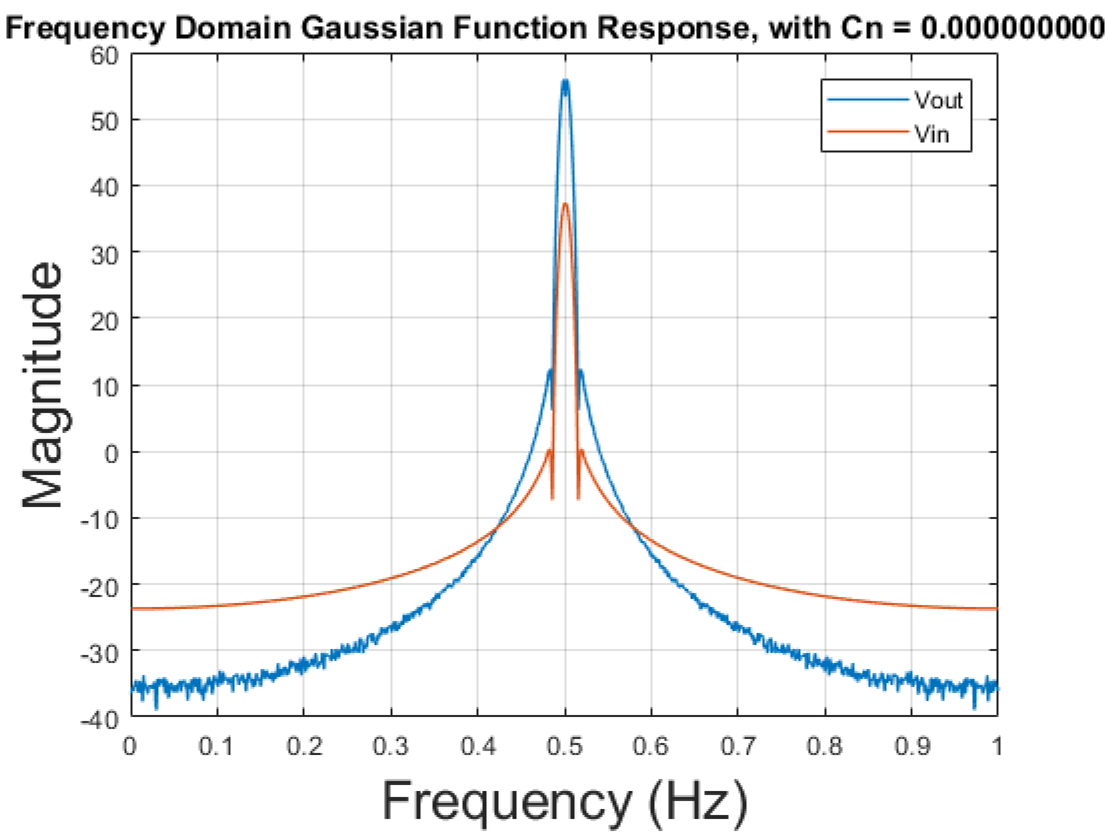
*The updated C matrix:*

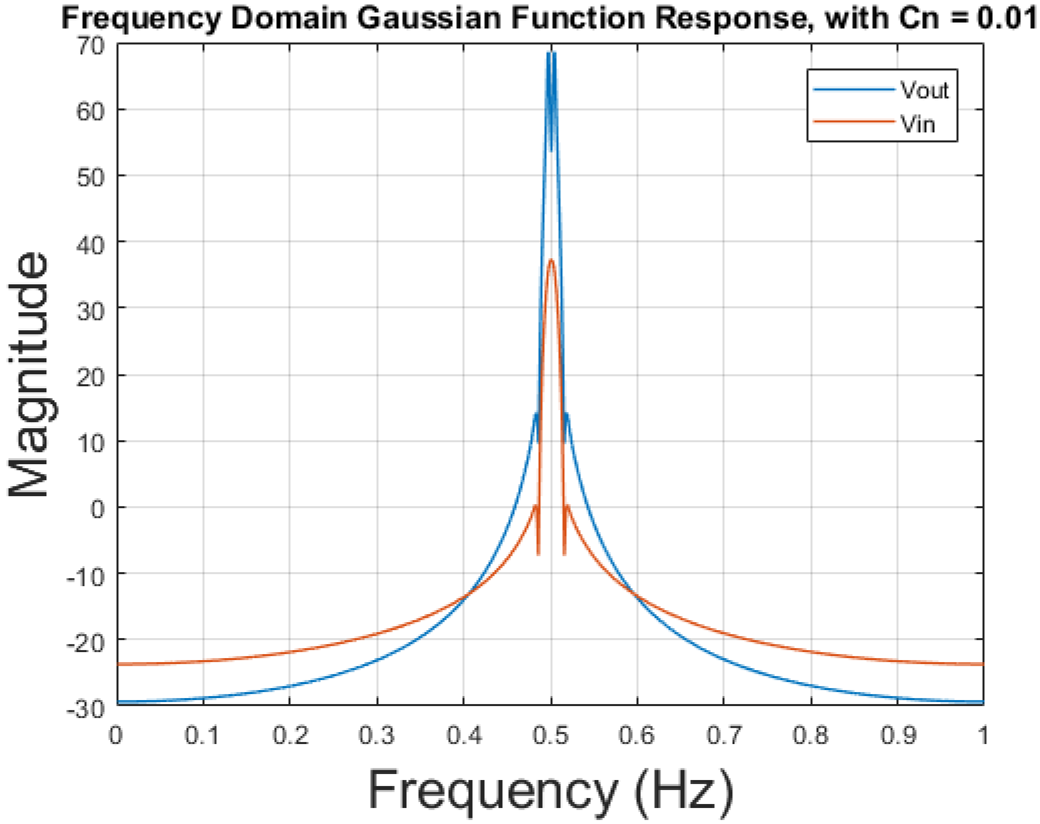
```
C =

  Columns 1 through 7

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


Gaussian Function Response
Vout
Vin
Voltage (V)
Time (s)
9
8
7
6
5
4
3
2
1
0
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

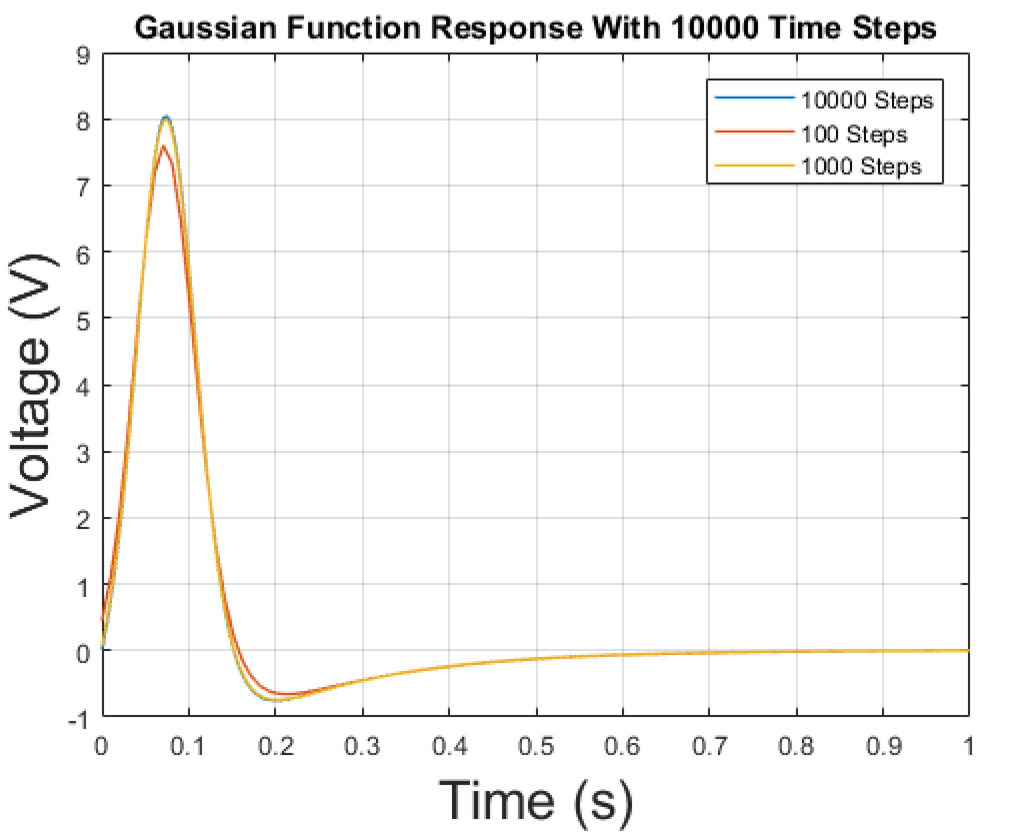

Frequency Domain Gaussian Function Response
Vout
Vin
Magnitude
Frequency (Hz)
60
50
40
30
20
10
0
-10
-20
-30
-40
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Frequency Domain Gaussian Function Response, with Cn = 0.001
Vout
Vin
Magnitude
Frequency (Hz)
60
50
40
30
20
10
0
-10
-20
-30
-40
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Frequency Domain Gaussian Function Response, with Cn = 0.000000000
Vout
Vin
Magnitude
Frequency (Hz)
60
50
40
30
20
10
0
-10
-20
-30
-40
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



Frequency Domain Gaussian Function Response, with Cn = 0.01
Magnitude
Frequency (Hz)
Vout
Vin
70
60
50
40
30
20
10
0
-10
-20
-30
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Gaussian Function Response With 10000 Time Steps
Voltage (V)
Time (s)
10000 Steps
100 Steps
1000 Steps
9
8
7
6
5
4
3
2
1
0
-1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

# Part 4

```
% a) An additional matrix must be added to the time domain equation to
% solve for a non-linear output.

% The new MNA equation can be represented by the following

% C*(dV/dt) + GV + F(V) = b(t)

% b) In order to solve this equation, it must be done iteratively. The
% Newton-Raphson method can be used to converge on a threshold voltage
% difference error.
```

*Published with MATLAB® R2017b*