RYERSON UNIVERSITY

Faculty of Engineering, Architecture and Science

Department of Electrical and Computer Engineering

Course Number	**ELE 532**
Course Title	**Signals and Systems I**
Semester/Year	**Fall 2021**
Instructor	**Dr. Javad Alirezaie**

ASSIGNMENT NO.	**1**

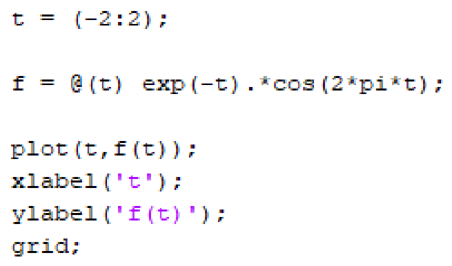
Report Title	Working with Matlab, Visualization of Signals

Section No.	02
Group No.	N/A
Submission Date	October 3rd, 2021
Due Date	October 3rd, 2021

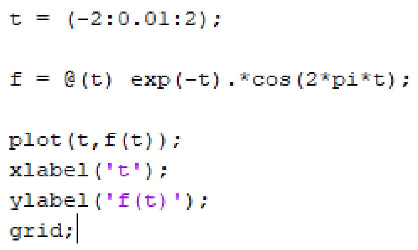
Name	Student No.	Signature
Naga Sreeja Kurra	xxxx20707	K.N.S
Moiz Kharodawala	xxxx68133	M.K.

PART A GRAPHS (MOIZ)

A.1



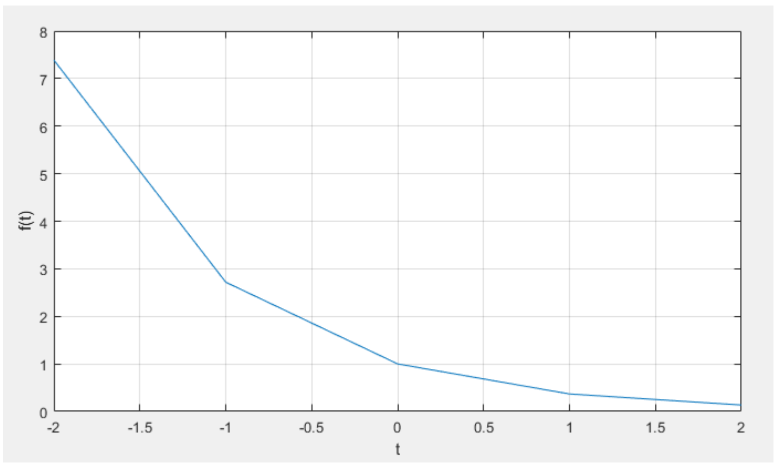
```
t = (-2:2);

f = @(t) exp(-t).*cos(2*pi*t);

plot(t,f(t));
xlabel('t');
ylabel('f(t)');
grid;
```



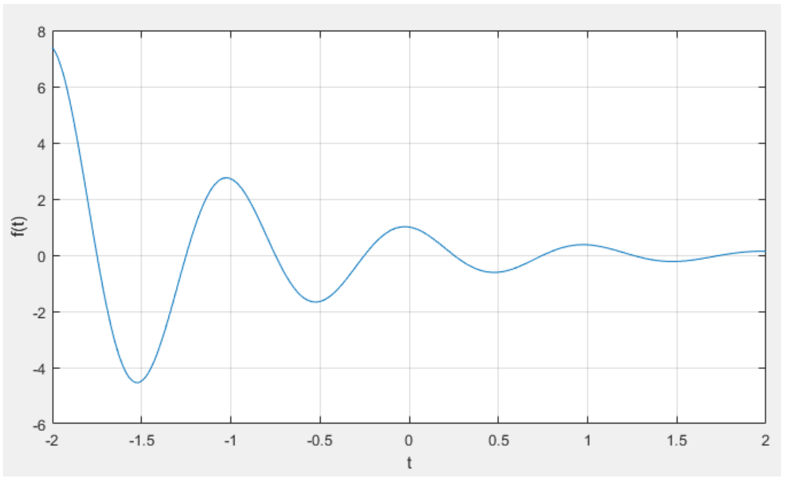
```
t = (-2:0.01:2);

f = @(t) exp(-t).*cos(2*pi*t);

plot(t,f(t));
xlabel('t');
ylabel('f(t)');
grid;
```

Figure 1: f(t) = e^(−t)cos(2πt) for t = (-2:2)

Figure 2: f(t) = e^(−t)cos(2πt) for t = (-2:0.01:2).



Graph 1: Graph output for Figure 1

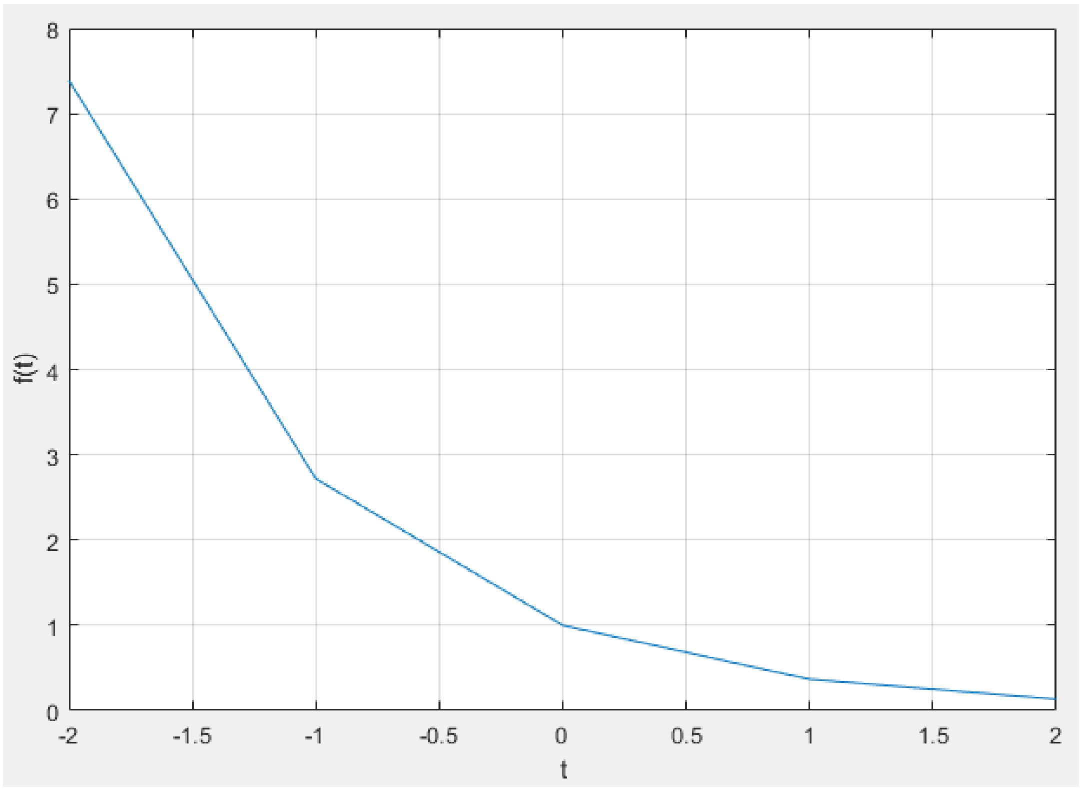


Graph 2: Graph output for Figure 2

A.2

```
t = (-2:2);

f = @(t) exp(-t);

plot(t,f(t));
xlabel('t');
ylabel('f(t)');
grid;
```

Figure 3: e^(-t) between −2 ≤ t ≤ 2



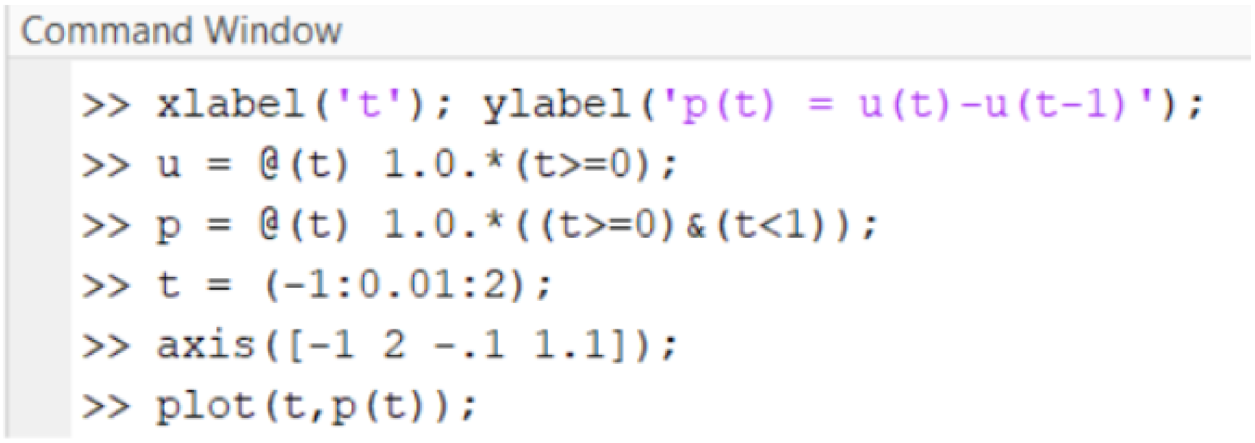
Graph 3: Graph output for Figure 3

A.3

There is no difference between the plots of A1 (Graph 1) and A2 (Graph 3). They both look the same.

PART B GRAPHS (MOIZ)

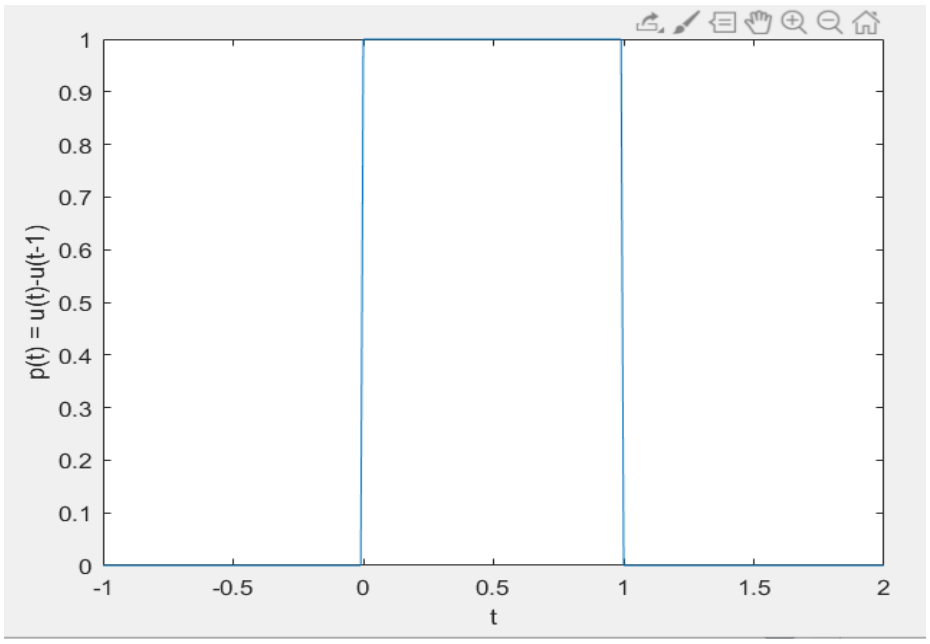
B.1



Command Window

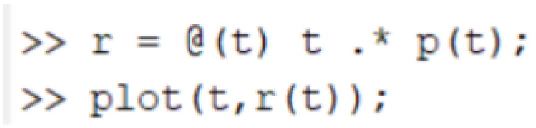
```
>> xlabel('t'); ylabel('p(t) = u(t)-u(t-1)');
>> u = @(t) 1.0.*(t>=0);
>> p = @(t) 1.0.*((t>=0)&(t<1));
>> t = (-1:0.01:2);
>> axis([-1 2 -.1 1.1]);
>> plot(t,p(t));
```

Figure 4: p(t)=u(t)-u(t-1) over −1 ≤ t ≤ 2



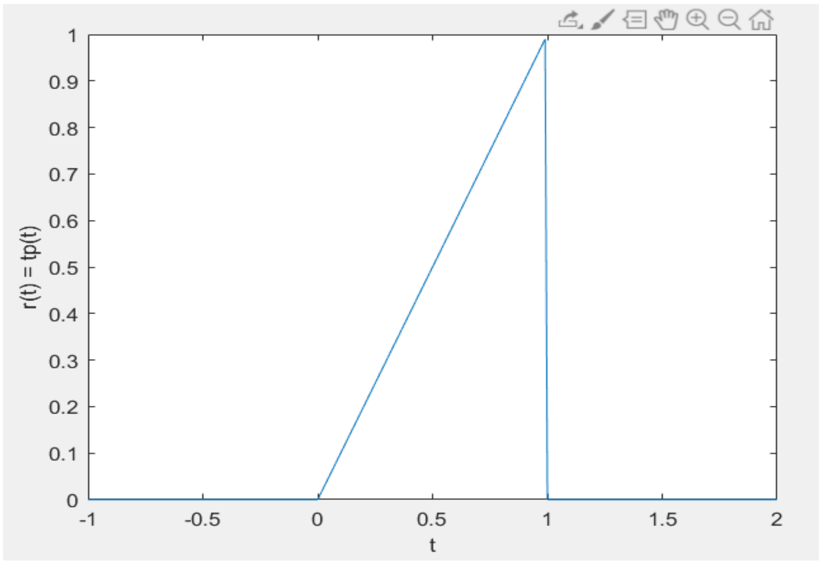
Graph 4: Graph output for Figure 4

B.2

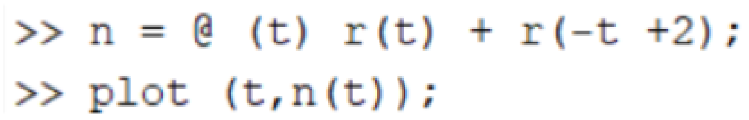


```
>> r = @(t) t .* p(t);
>> plot(t,r(t));
```

Figure 5: r(t)=tp(t) over −1 ≤ t ≤ 2

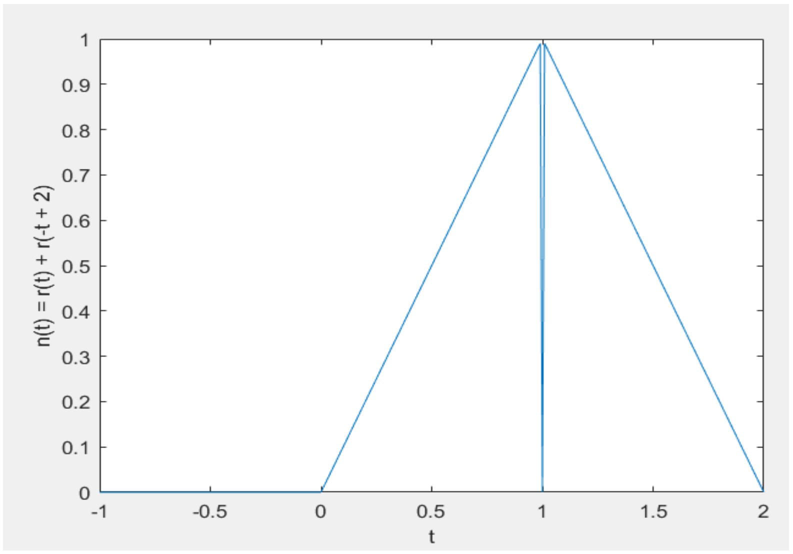


Graph 5: Graph output for Figure 5



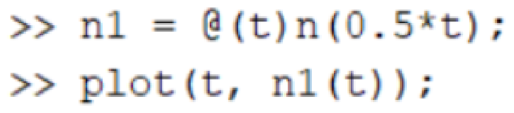
```
>> n = @ (t) r(t) + r(-t +2);
>> plot (t,n(t));
```

Figure 6: n(t) = r(t) + r(-t +2) over (-1 ≤ t ≤ 2)



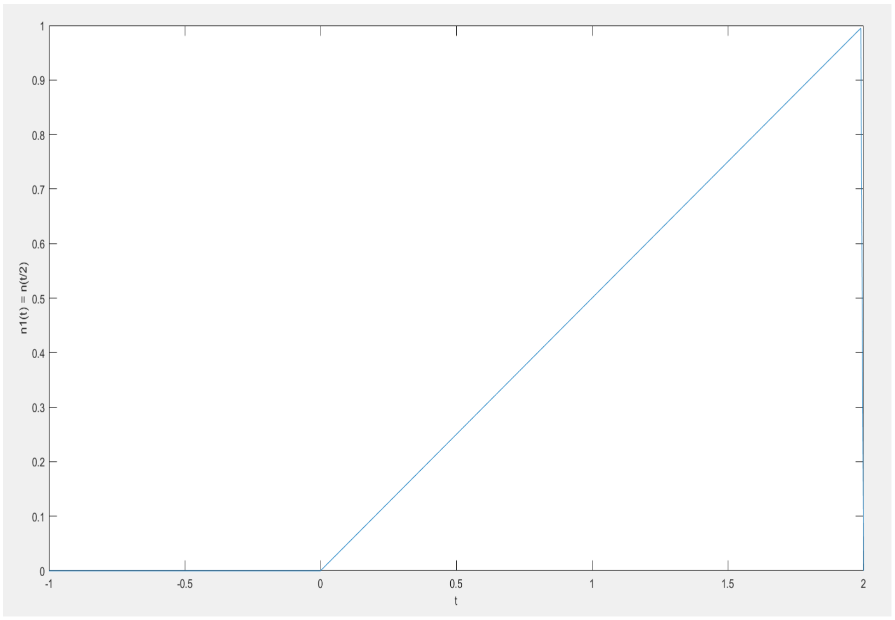
Graph 6: Graph output for Figure 6

B.3

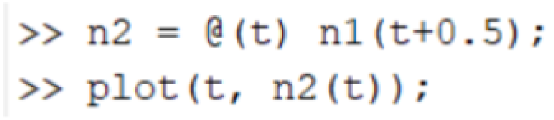


```
>> n1 = @(t)n(0.5*t);
>> plot(t, n1(t));
```

Figure 7: n1(t) = n(0.5 * t) over (-1 ≤ t ≤ 2)

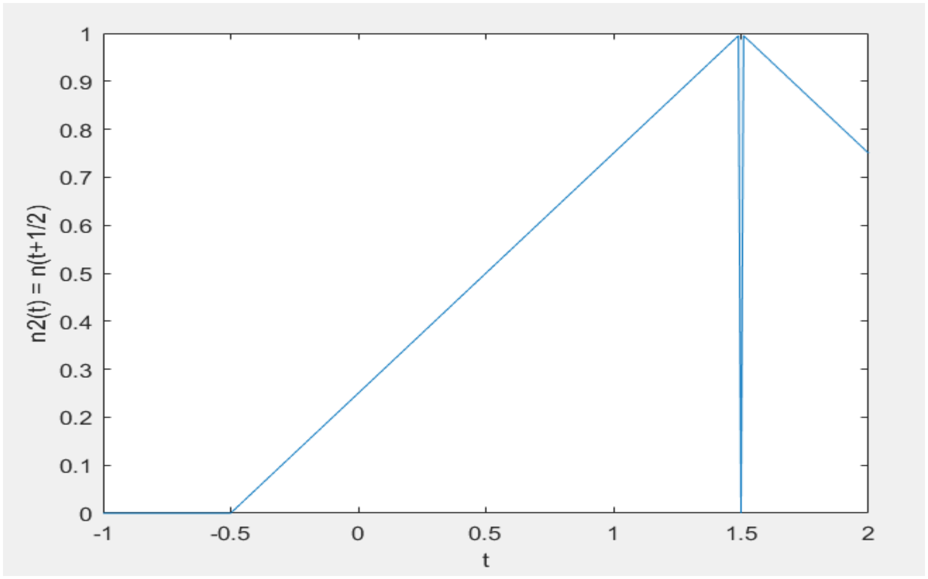


Graph 7: Graph output for Figure 7



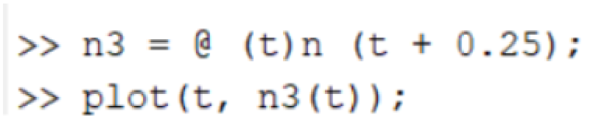
```
>> n2 = @(t) n1(t+0.5);
>> plot(t, n2(t));
```

Figure 8: n2(t) = n1(t + 0.5) over (-1 ≤ t ≤ 2)



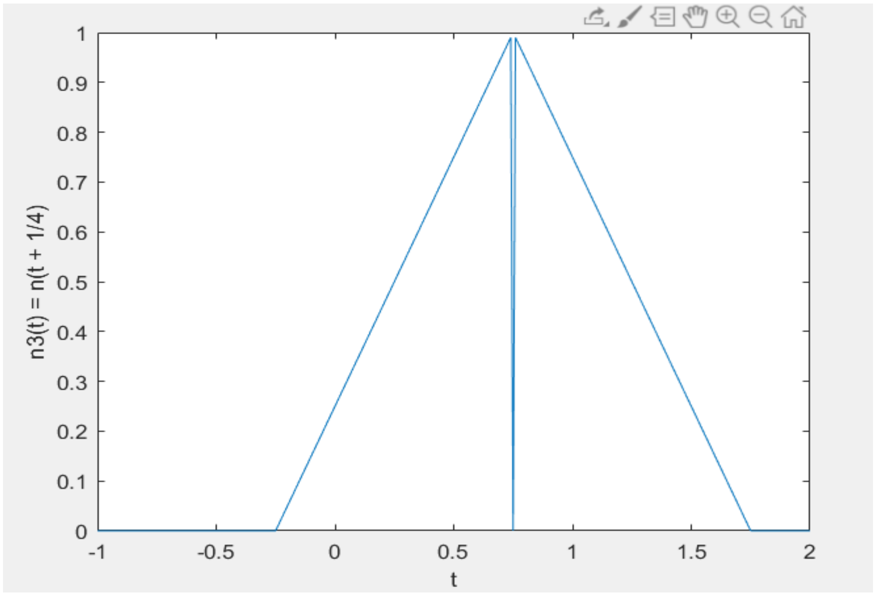
Graph 8: Graph output for Figure 8

B.4

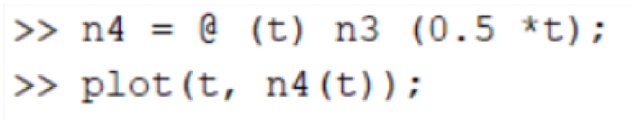


```
>> n3 = @ (t)n (t + 0.25);
>> plot(t, n3(t));
```

Figure 9: n3(t) = n(t + 0.25) over (-1 ≤ t ≤ 2)

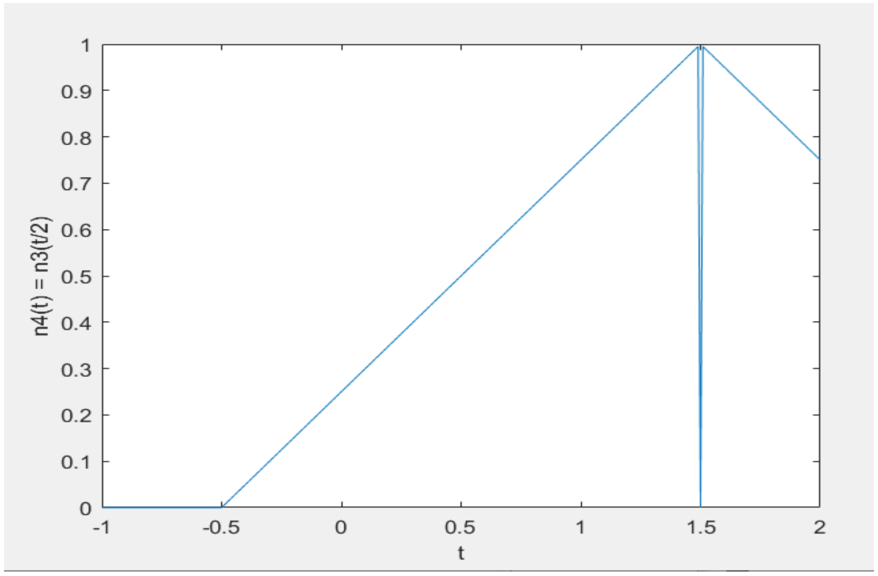


Graph 9: Graph output for Figure 9



```
>> n4 = @ (t) n3 (0.5 *t);
>> plot(t, n4(t));
```

Figure 10: n4(t) = n3(0.5 * t) over (-1 ≤ t ≤ 2)



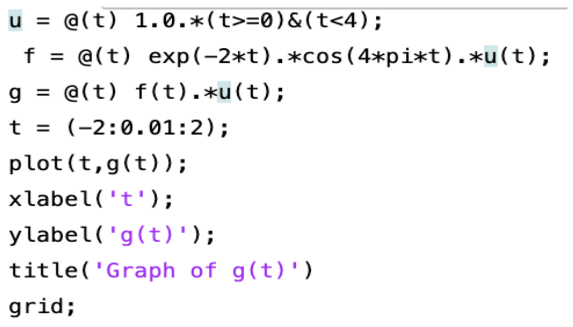
Graph 10: Graph output for Figure 10

B.5

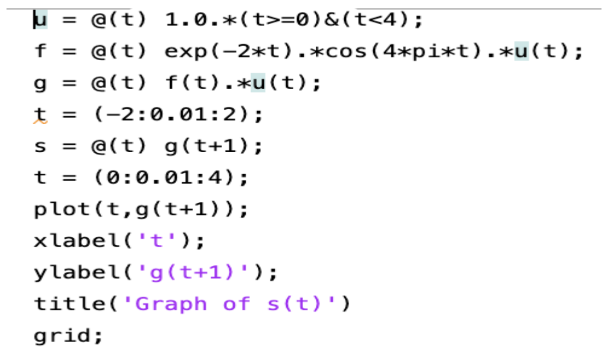
There is one big difference between the plots of n3(t) and n4(t) i.e. the plot for n4(t) is horizontally stretched by 2 relative to the plot of n3(t).

PART C GRAPHS (SREEJA)

C.1 & C.2



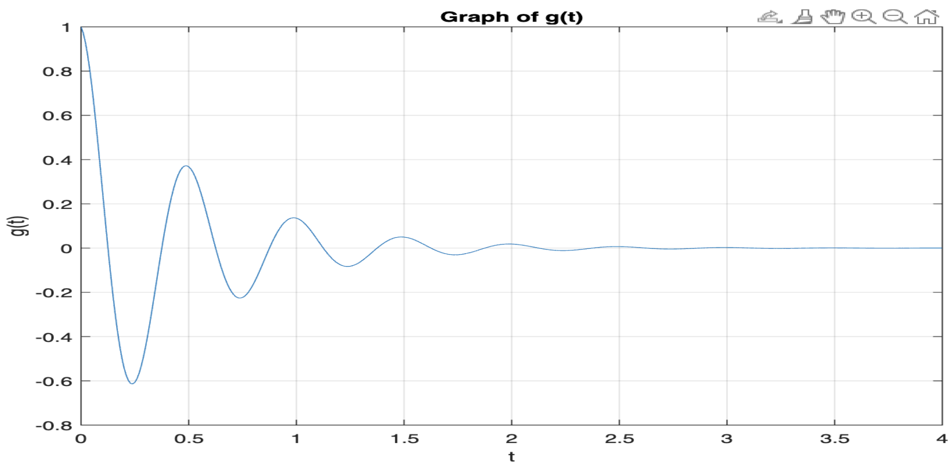
```
u = @(t) 1.0.*(t>=0)&(t<4);
 f = @(t) exp(-2*t).*cos(4*pi*t).*u(t);
g = @(t) f(t).*u(t);
t = (-2:0.01:2);
plot(t,g(t));
xlabel('t');
ylabel('g(t)');
title('Graph of g(t)')
grid;
```



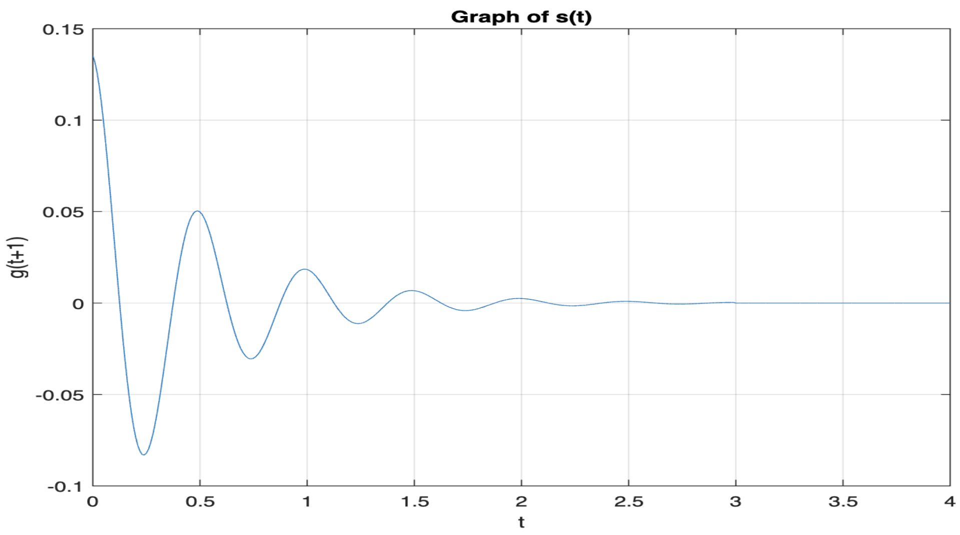
```
u = @(t) 1.0.*(t>=0)&(t<4);
f = @(t) exp(-2*t).*cos(4*pi*t).*u(t);
g = @(t) f(t).*u(t);
t = (-2:0.01:2);
s = @(t) g(t+1);
t = (0:0.01:4);
plot(t,g(t+1));
xlabel('t');
ylabel('g(t+1)');
title('Graph of s(t)')
grid;
```

Figure 11: g(t) = e^(-2t) cos(4πt) * u(t); t[-2:0.01:2]

Figure 12 : s(t) = g(t + 1); t[0:0.01:4]

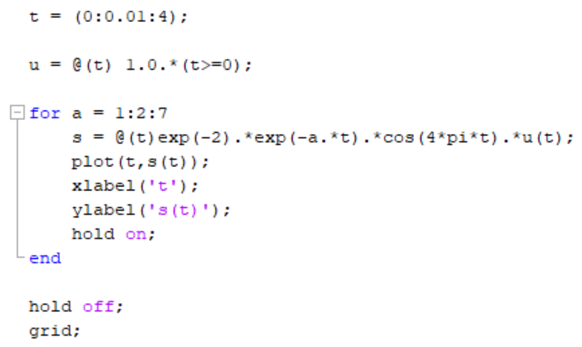


Graph 11: Graph output for the Figure 11

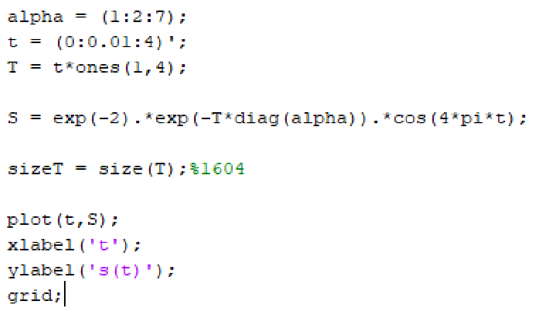


Graph 12: Graph output for the Figure 12

C.3



```
t = (0:0.01:4);

u = @(t) 1.0.*(t>=0);

for a = 1:2:7
    s = @(t)exp(-2).*exp(-a.*t).*cos(4*pi*t).*u(t);
    plot(t,s(t));
    xlabel('t');
    ylabel('s(t)');
    hold on;
end

hold off;
grid;
```



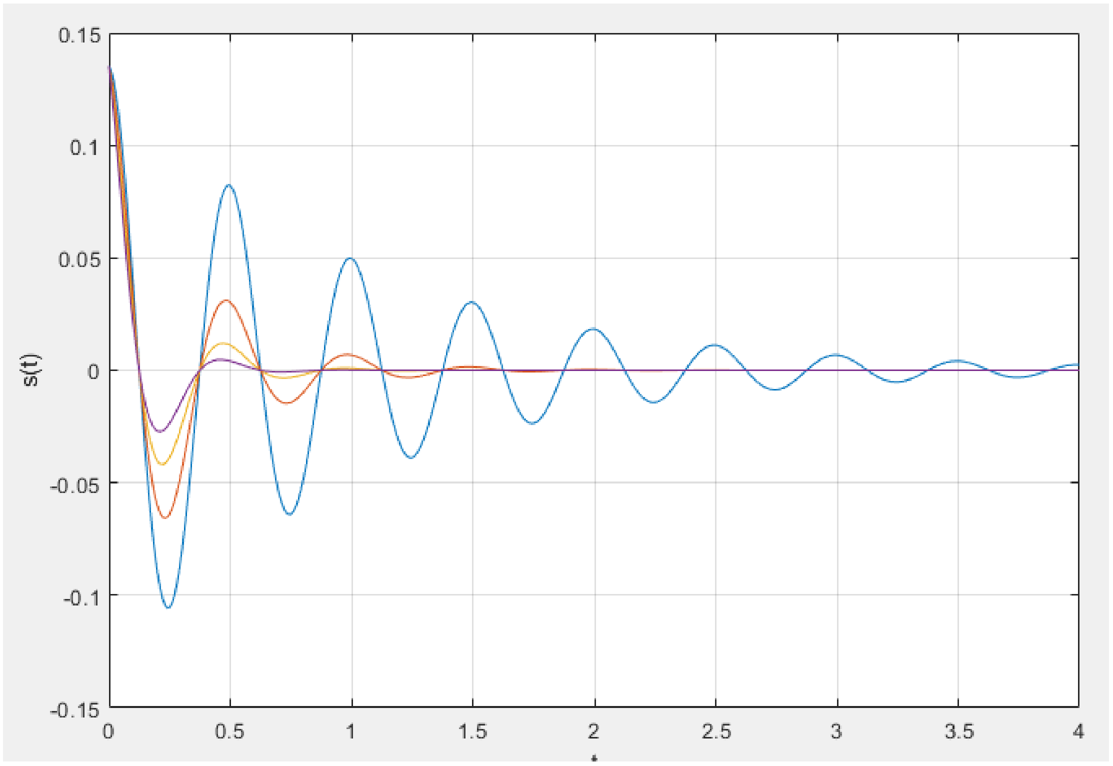
```
alpha = (1:2:7);
t = (0:0.01:4)';
T = t*ones(1,4);

S = exp(-2).*exp(-T*diag(alpha)).*cos(4*pi*t);

sizeT = size(T);%1604

plot(t,S);
xlabel('t');
ylabel('s(t)');
grid;
```

Figure 11: $s_{\alpha}(t) = e^{-2}e^{-\alpha t}cos(4\pi t)u(t)$

$\alpha = 1, 3, 5, 7$

Figure 12: $s_{\alpha}(t) = e^{-2}e^{-\alpha t}cos(4\pi t)u(t)$

using matrix method



Graph 13: Graph output for Figure 11 and Figure 12

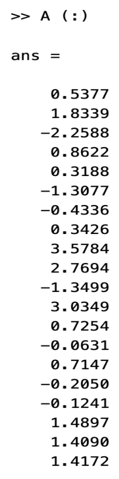
C.4
The size of the matrix generated in C.3 is 1604

PART D (SREEJA)

D.1

(a) A (:)

This operator is used to create a row of vectors. In this case, the row of vectors is from the data in matrix A. With this operation, we have not specified any value to be placed in a row of vectors. Thus, it takes all the values from matrix A.

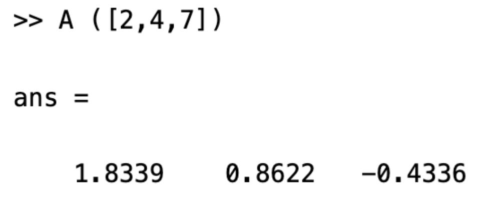


```
>> A (:)

ans =

    0.5377
    1.8339
   -2.2588
    0.8622
    0.3188
   -1.3077
   -0.4336
    0.3426
    3.5784
    2.7694
   -1.3499
    3.0349
    0.7254
   -0.0631
    0.7147
   -0.2050
   -0.1241
    1.4897
    1.4090
    1.4172
```

Figure 15 : Matrix A showing all the values in a row of vectors

(b) A([2,4,7])

This prints the second, fourth and seventh values of the matrix A.

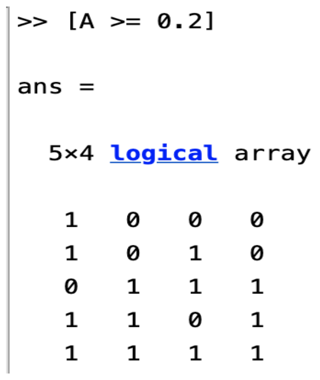


```
>> A ([2,4,7])

ans =

    1.8339    0.8622   -0.4336
```

Figure 16 : Matrix A showing the values of 2,4,7.

(c) [A >= 0.2]

This value represents the 5 x 4 logical array where all the values above or equal to 0.2 get the value of 1. While the values below 0.2 will be 0.



```
>> [A >= 0.2]

ans =

  5×4 logical array

   1   0   0   0
   1   0   1   0
   0   1   1   1
   1   1   0   1
   1   1   1   1
```

Figure 17 : Matrix A showing the values above or equal to 0.2

(d) A ([A >= 0.2])

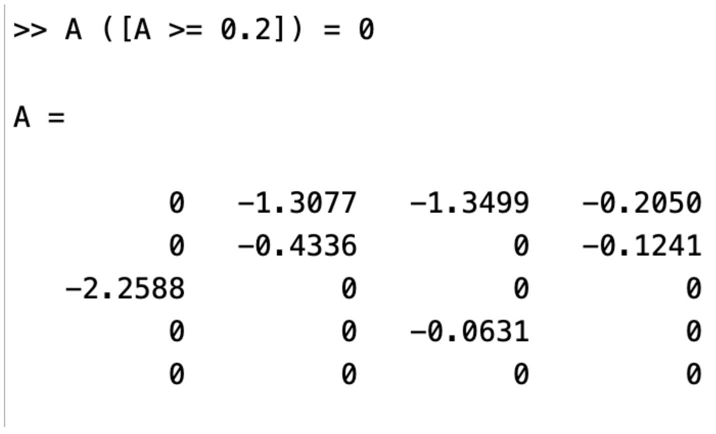
This prints the values above or equal to 0.2 in matrix A. It creates a row of vectors of those values in matrix A.

```
>> A ([A >= 0.2])

ans =

    0.5377
    1.8339
    0.8622
    0.3188
    0.3426
    3.5784
    2.7694
    3.0349
    0.7254
    0.7147
    1.4897
    1.4090
    1.4172
```

Figure 18 : Matrix A showing the row of values in the vector.

(e) A ([A >= 0.2]) = 0

This operation is able to override the values of the original matrix A. This operation says that any value greater than or equal to 0.2 in the matrix sets them to 0.

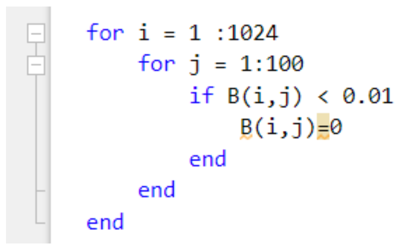


```
>> A ([A >= 0.2]) = 0

A =

         0   -1.3077   -1.3499   -0.2050
         0   -0.4336         0   -0.1241
   -2.2588         0         0         0
         0         0   -0.0631         0
         0         0         0         0
```

Figure 19 : Matrix A showing the operation

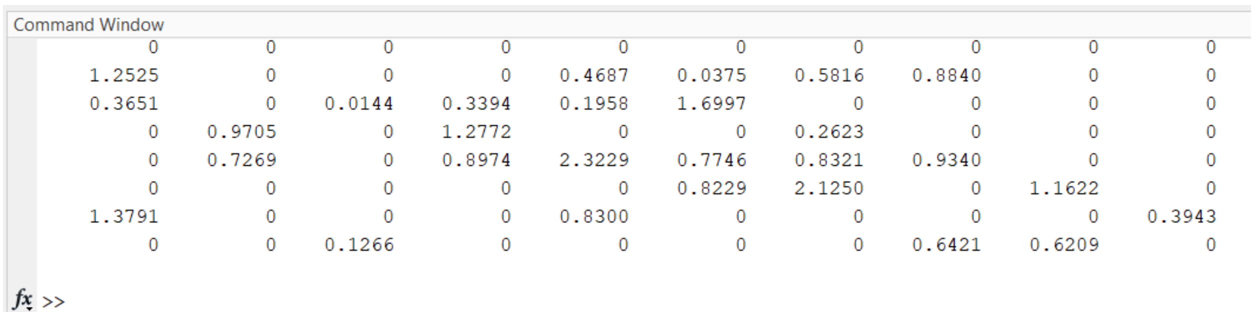
D.2

a) MATLAB using nested loops for B(i,j) = 0, if **|B(i,j)| < 0.01.**



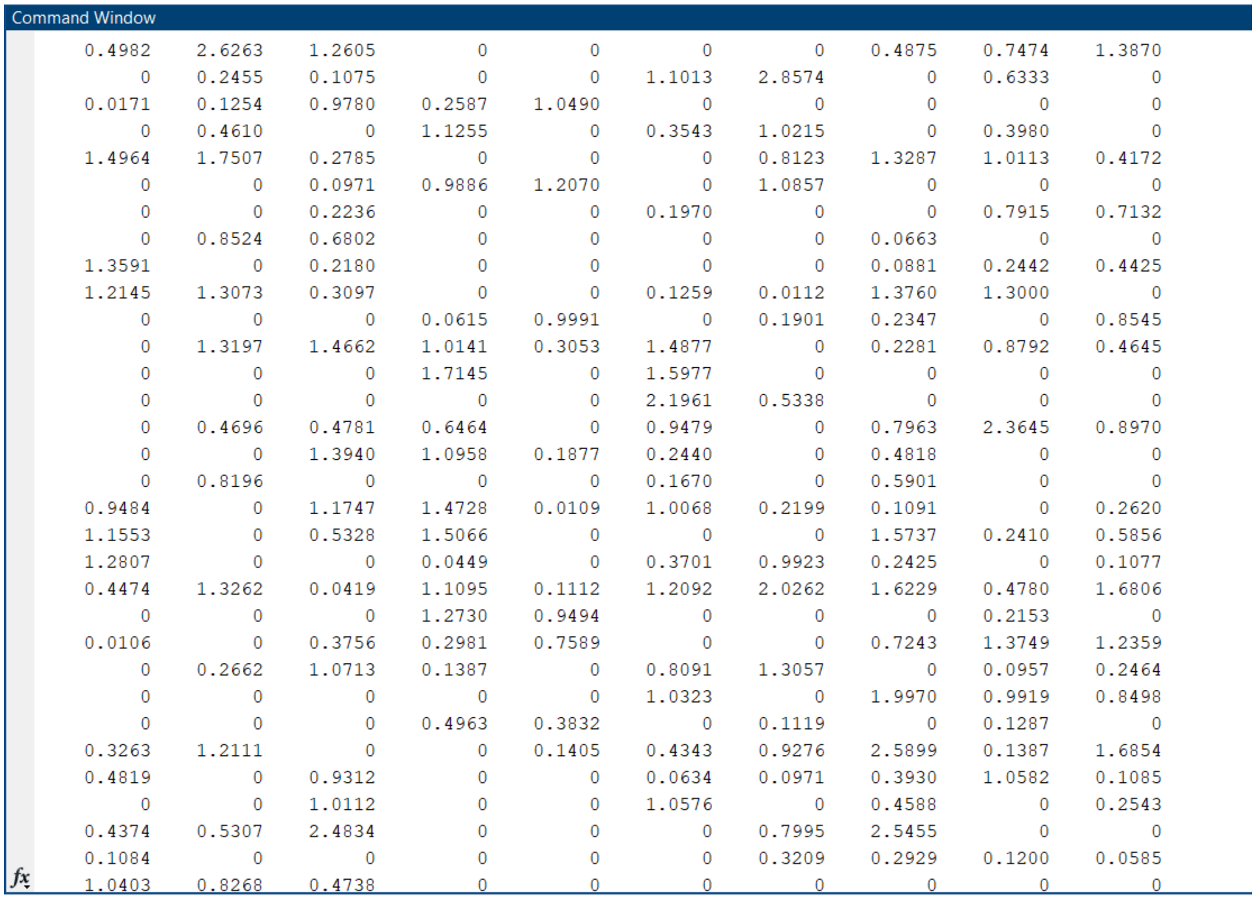
```
for i = 1 :1024
    for j = 1:100
        if B(i,j) < 0.01
            B(i,j)=0
        end
    end
end
```

Figure 20 : MATLAB code for the above equation.



```
         0         0         0         0         0         0         0         0         0         0
    1.2525         0         0         0    0.4687    0.0375    0.5816    0.8840         0         0
    0.3651         0    0.0144    0.3394    0.1958    1.6997         0         0         0         0
         0    0.9705         0    1.2772         0         0    0.2623         0         0         0
         0    0.7269         0    0.8974    2.3229    0.7746    0.8321    0.9340         0         0
         0         0         0         0         0    0.8229    2.1250         0    1.1622         0
    1.3791         0         0         0    0.8300         0         0         0         0    0.3943
         0         0    0.1266         0         0         0         0    0.6421    0.6209         0

fx >>
```

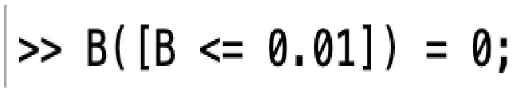
Figure 21: Results from Figure 20.



```
    0.4982    2.6263    1.2605         0         0         0         0    0.4875    0.7474    1.3870
         0    0.2455    0.1075         0         0    1.1013    2.8574         0    0.6333         0
    0.0171    0.1254    0.9780    0.2587    1.0490         0         0         0         0         0
         0    0.4610         0    1.1255         0    0.3543    1.0215         0    0.3980         0
    1.4964    1.7507    0.2785         0         0         0    0.8123    1.3287    1.0113    0.4172
         0         0    0.0971    0.9886    1.2070         0    1.0857         0         0         0
         0         0    0.2236         0         0    0.1970         0         0    0.7915    0.7132
         0    0.8524    0.6802         0         0         0         0    0.0663         0         0
    1.3591         0    0.2180         0         0         0         0    0.0881    0.2442    0.4425
    1.2145    1.3073    0.3097         0         0    0.1259    0.0112    1.3760    1.3000         0
         0         0         0    0.0615    0.9991         0    0.1901    0.2347         0    0.8545
         0    1.3197    1.4662    1.0141    0.3053    1.4877         0    0.2281    0.8792    0.4645
         0         0         0    1.7145         0    1.5977         0         0         0         0
         0         0         0         0         0    2.1961    0.5338         0         0         0
         0    0.4696    0.4781    0.6464         0    0.9479         0    0.7963    2.3645    0.8970
         0         0    1.3940    1.0958    0.1877    0.2440         0    0.4818         0         0
         0    0.8196         0         0         0    0.1670         0    0.5901         0         0
    0.9484         0    1.1747    1.4728    0.0109    1.0068    0.2199    0.1091         0    0.2620
    1.1553         0    0.5328    1.5066         0         0         0    1.5737    0.2410    0.5856
    1.2807         0         0    0.0449         0    0.3701    0.9923    0.2425         0    0.1077
    0.4474    1.3262    0.0419    1.1095    0.1112    1.2092    2.0262    1.6229    0.4780    1.6806
         0         0         0    1.2730    0.9494         0         0         0    0.2153         0
    0.0106         0    0.3756    0.2981    0.7589         0         0    0.7243    1.3749    1.2359
         0    0.2662    1.0713    0.1387         0    0.8091    1.3057         0    0.0957    0.2464
         0         0         0         0         0    1.0323         0    1.9970    0.9919    0.8498
         0         0         0    0.4963    0.3832         0    0.1119         0    0.1287         0
    0.3263    1.2111         0         0    0.1405    0.4343    0.9276    2.5899    0.1387    1.6854
    0.4819         0    0.9312         0         0    0.0634    0.0971    0.3930    1.0582    0.1085
         0         0    1.0112         0         0    1.0576         0    0.4588         0    0.2543
    0.4374    0.5307    2.4834         0         0         0    0.7995    2.5455         0         0
    0.1084         0         0         0         0         0    0.3209    0.2929    0.1200    0.0585
fx  1.0403    0.8268    0.4738         0         0         0         0         0         0         0
```

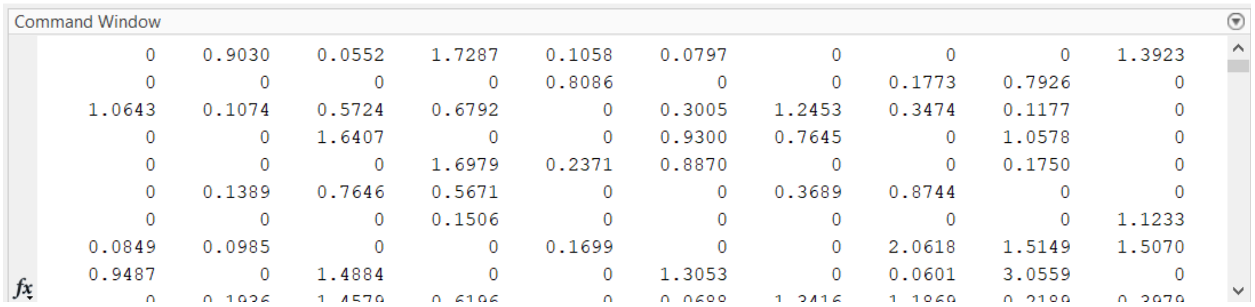
Figure 22: Results from Figure 20.

b) Repeat part (a) using MATLAB's indexing features.



```
>> B([B <= 0.01]) = 0;
```

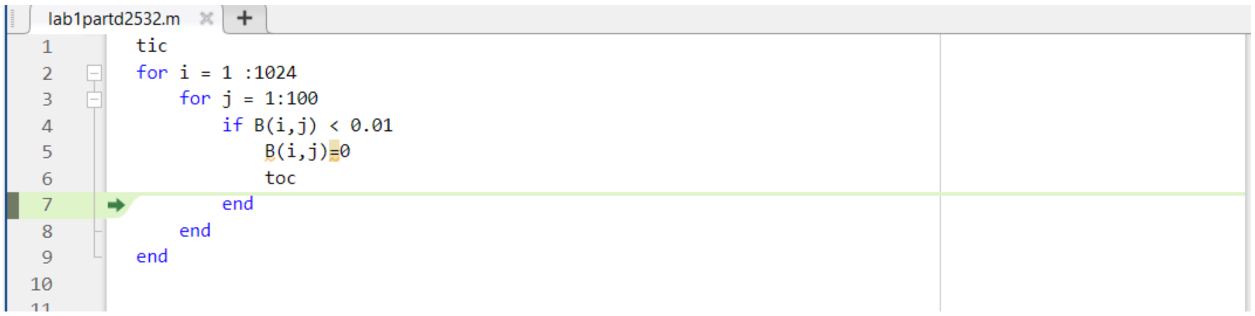
Figure 23: Matlab code for the above Matlab's indexing features.



```
         0    0.9030    0.0552    1.7287    0.1058    0.0797         0         0         0    1.3923
         0         0         0         0    0.8086         0         0    0.1773    0.7926         0
    1.0643    0.1074    0.5724    0.6792         0    0.3005    1.2453    0.3474    0.1177         0
         0         0    1.6407         0         0    0.9300    0.7645         0    1.0578         0
         0         0         0    1.6979    0.2371    0.8870         0         0    0.1750         0
         0    0.1389    0.7646    0.5671         0         0    0.3689    0.8744         0         0
         0         0         0    0.1506         0         0         0         0         0    1.1233
    0.0849    0.0985         0         0    0.1699         0         0    2.0618    1.5149    1.5070
    0.9487         0    1.4884         0         0    1.3053         0    0.0601    3.0559         0
```

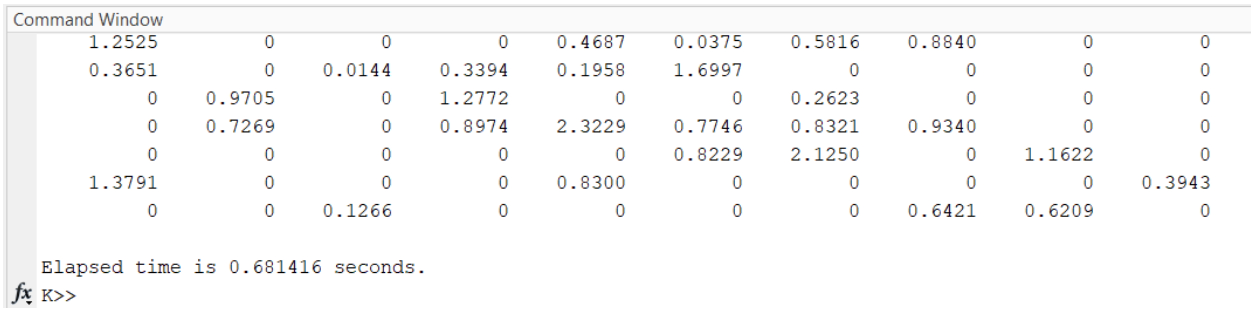
Figure 24: Results from Figure 23.

c) Use the MATLAB commands tic and toc to compare the execution code of the parts (a) and (b).



```
tic
for i = 1 :1024
    for j = 1:100
        if B(i,j) < 0.01
            B(i,j)=0
            toc
        end
    end
end
```

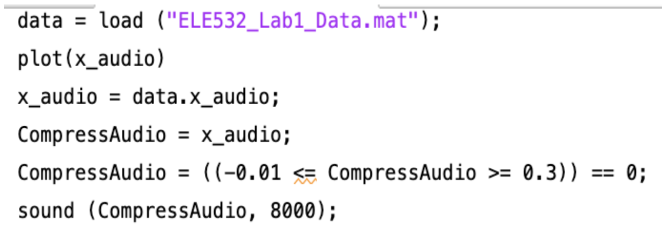
Figure 25: Matlab code by using tic and toc.



```
    1.2525         0         0         0    0.4687    0.0375    0.5816    0.8840         0         0
    0.3651         0    0.0144    0.3394    0.1958    1.6997         0         0         0         0
         0    0.9705         0    1.2772         0         0    0.2623         0         0         0
         0    0.7269         0    0.8974    2.3229    0.7746    0.8321    0.9340         0         0
         0         0         0         0         0    0.8229    2.1250         0    1.1622         0
    1.3791         0         0         0    0.8300         0         0         0         0    0.3943
         0         0    0.1266         0         0         0         0    0.6421    0.6209         0

Elapsed time is 0.681416 seconds.
K>>
```

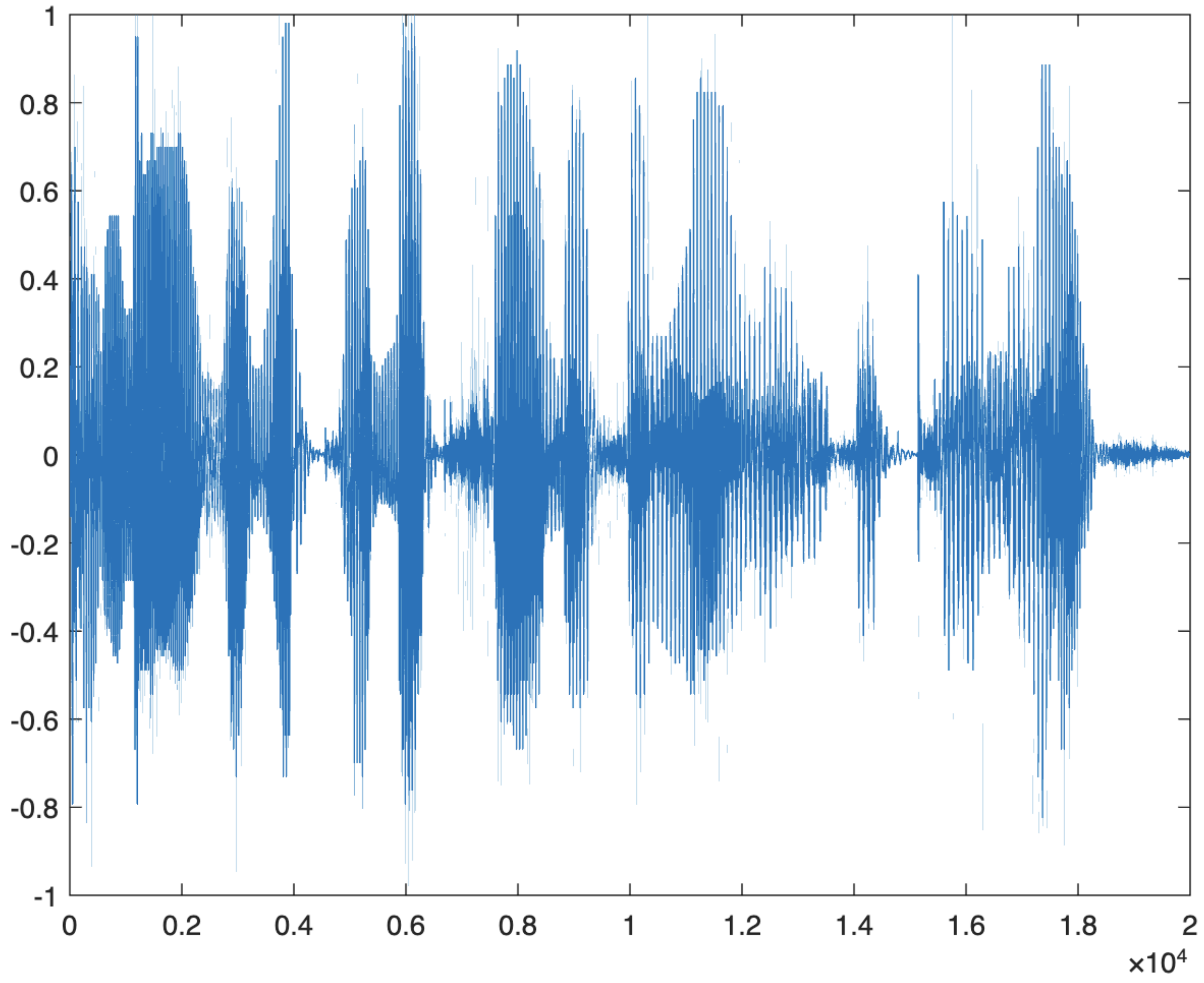
Figure 26: Results from Figure 25.

D.3



```
data = load ("ELE532_Lab1_Data.mat");
plot(x_audio)
x_audio = data.x_audio;
CompressAudio = x_audio;
CompressAudio = ((-0.01 <= CompressAudio >= 0.3)) == 0;
sound (CompressAudio, 8000);
```

Figure 27: Matlab code for the x_audio.



Graph 14: Results from Figure 27.

The above code is for the x_audio for about a sound of 8000 Hz. The ELE 532 lab 1 matlab file gives the values of x_audio and gives a small beep sound and has produced the above graph which is the form of sound frequency waves.