Department of Electrical, Computer & Biomedical Engineering
Faculty of Engineering & Architectural Science

Course Number	ELE 532
Course Title	Signal and Systems 1
Semester/Year	Winter 2024
Instructor	Dr. Alagan Anpalagan

Lab/Tutorial Report No.	**1**

Report Title	Working with Matlab Functions, Visualization of Signals

Submission Date	2024-01-31
Due Date	2024-02-01

Name	**Student ID**	**Signature***
Magsud Allahyarov	xxxx84937	Magsud

(Note: remove the first 4 digits from your student ID)

**By signing above you attest that you have contributed to this submission and confirm that all work you have contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work <u>will result in an investigation of Academic Misconduct and may result in a "0" on the work,</u> an "F" in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Student Code of Academic Conduct, which can be found online at:*

https://www.torontomu.ca/content/dam/senate/policies/pol60.pdf

Table of Contents:

1. **Anonymous functions and plotting continuous functions - pg.3-5**
2. **Time shifting and time scaling - pg.6-11**
3. **Visualizing operations on the independent variable and algorithm vectorization - pg.11-14**
4. **Array indexing - pg.14-17**

Anonymous functions and plotting continuous functions

Problem A.1: Generate and plot Figures 1.46 and 1.47 on page 127.

Figure 1.46

Code:

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

plot(t,f(t));
title('Figure 1.46');
xlabel('t');
ylabel('f(t)');
```

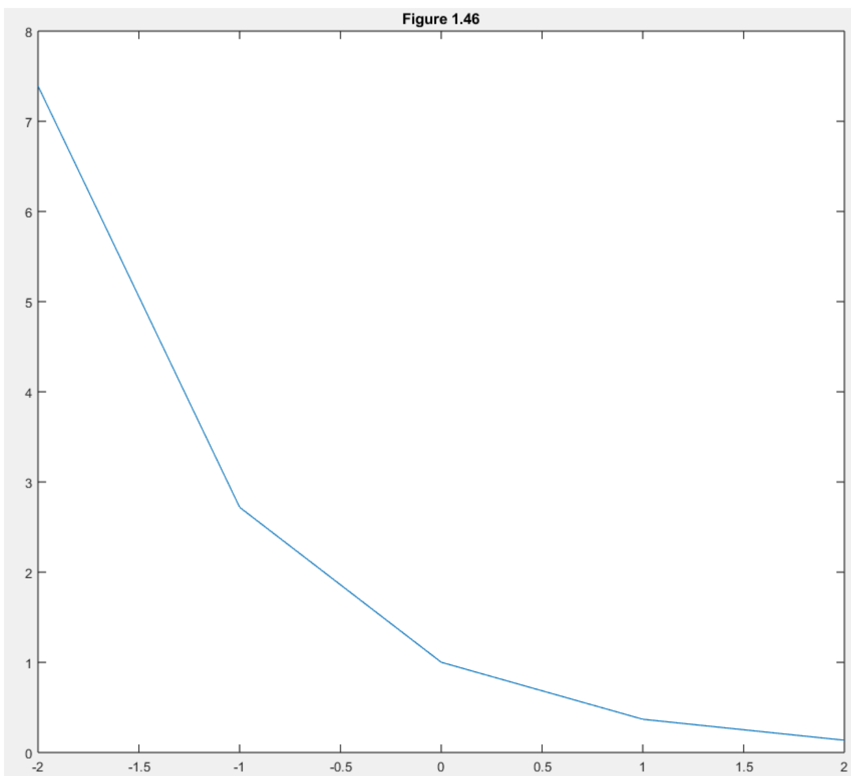
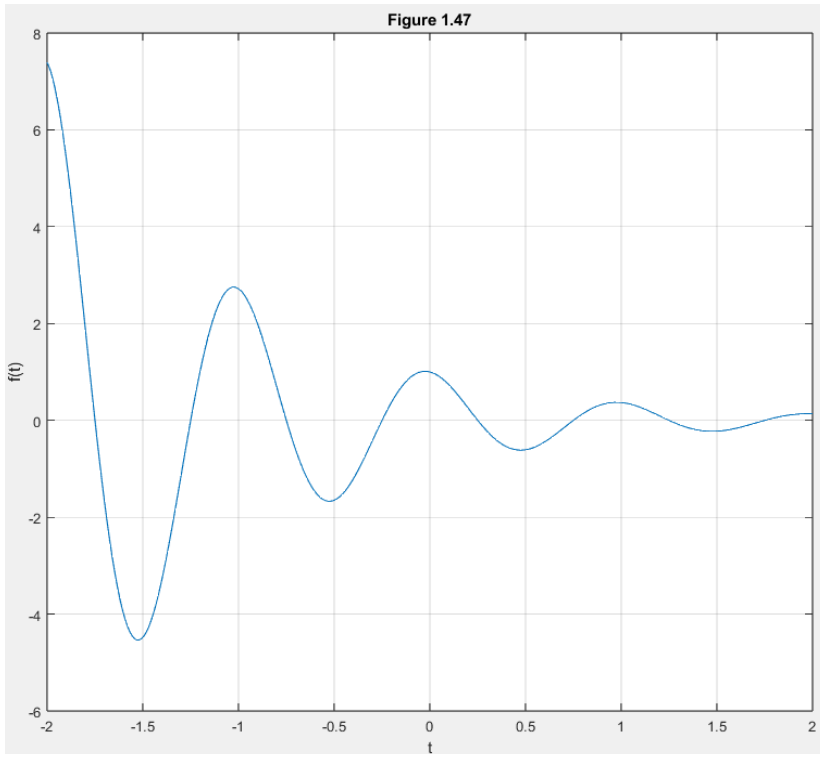


Figure 1.47

Code:

```
f = @(t) exp(-t).*cos(2*pi*t);
t=(-2:0.01:2);

plot(t,f(t));
```

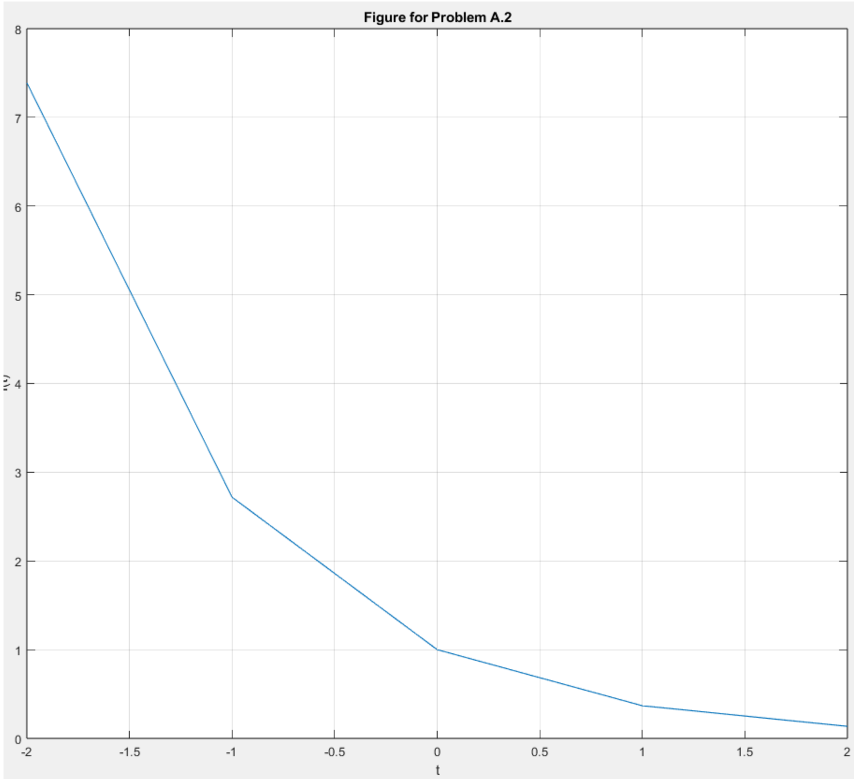
title('Figure 1.47');
xlabel('t');
ylabel('f(t)');



Problem A.2: Plot the function e −t for five points [-2,-1,0,1,2] using t=[-2:2].

Code:
f = @(t) exp(-t);
t = (-2:2)

plot (t,f(t));
title('Figure for Problem A.2');
xlabel('t');
ylabel('f(t)');



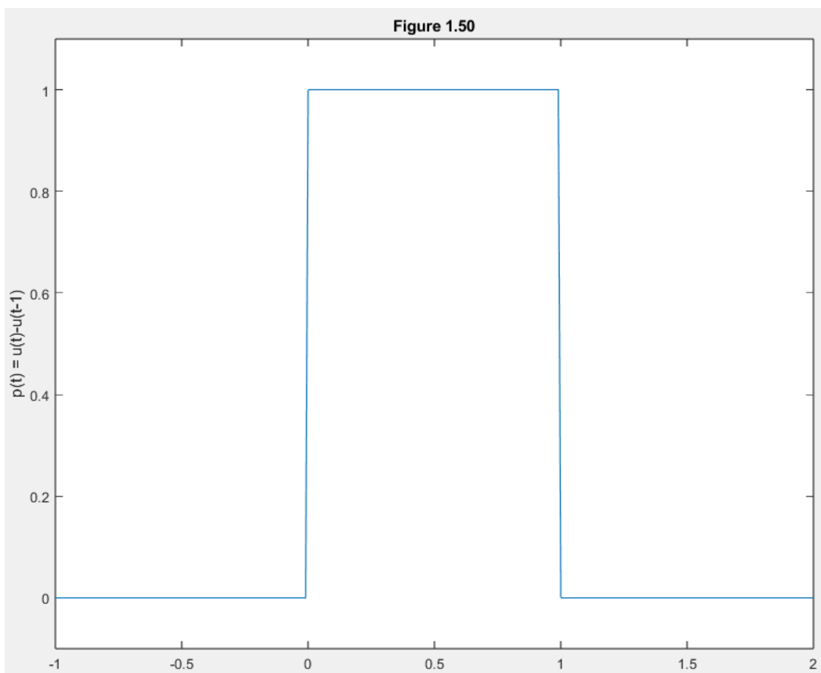
<u>Problem A.3</u>: Compare your result in Problem A.2 with Figure 1.46 in Problem A.1.

The results from both graphs appear to be identical. This is because in figure 1.46 there is a multiplication factor of cos(2πt) which results in 1 in all time intervals specified. All these values equal to 1 so basically the function plotted is e^{-1t} which is the same as the function in Problem A2.

Time shifting and time scaling

Problem B.1: Generate and plot p(t) as shown in Figure 1.50 on page 129.

```
t = (-1:0.01:2);
p = @(t) 1.0.*((t>=0)&(t<1));

plot(t,p(t));
xlabel('t');
ylabel('p(t) = u(t)-u(t-1)');
axis([-1 2 -.1 1.1]);
title('Figure 1.50');
```

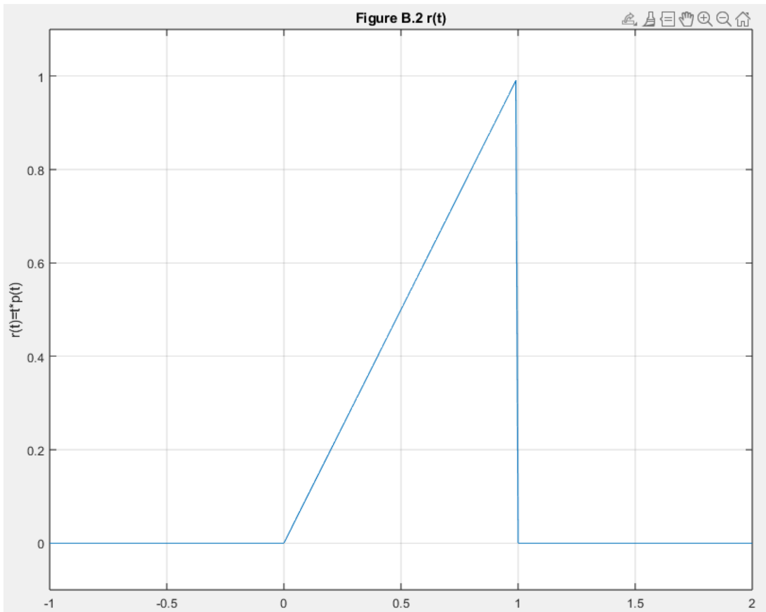


Problem B.2: Use p(t) to generate and plot functions $r(t) = tp(t)$ and $n(t) = r(t) + r(-t + 2)$.

Code:

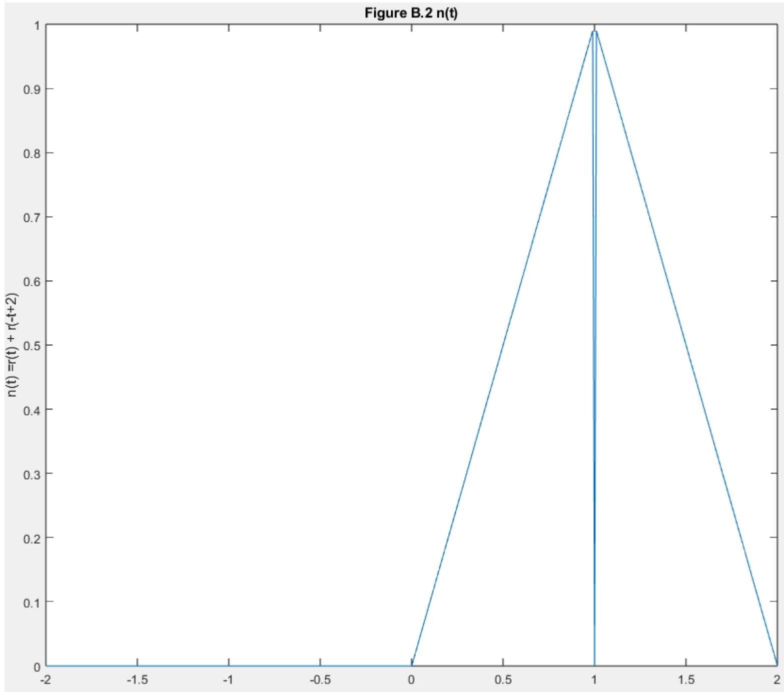
```
t=(-2:0.01:2);
u=@(t) 1.0.*(t>=0);
p=@(t) u(t)-u(t-1);
r=@(t) t.*p(t);
n=@(t) r(t)+r(-t+2);

plot(t,r(t));
```

```
xlabel('t');
ylabel('r(t)=t*p(t)');
title('Figure B.2 r(t)');
axis([-1 2 -0.1 1.1]);
```



Code:

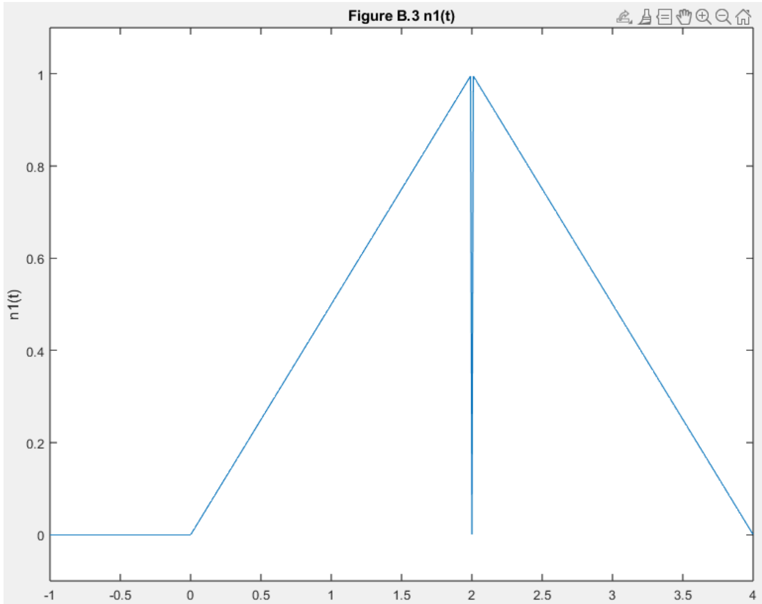
```
t=(-2:0.01:2);
u=@(t) 1.0.*(t>=0);
p=@(t) u(t)-u(t-1);
r=@(t) t.*p(t);
n=@(t) r(t)+r(-t+2);

plot(t, n(t));
xlabel("t");
ylabel("n(t) =r(t) + r(-t+2)");
title('Figure B.2 n(t)');
```



Problem B.3:Plot the following two signals: n1(t) = n(1/2 t), n2(t) = n1(t + 1/2).

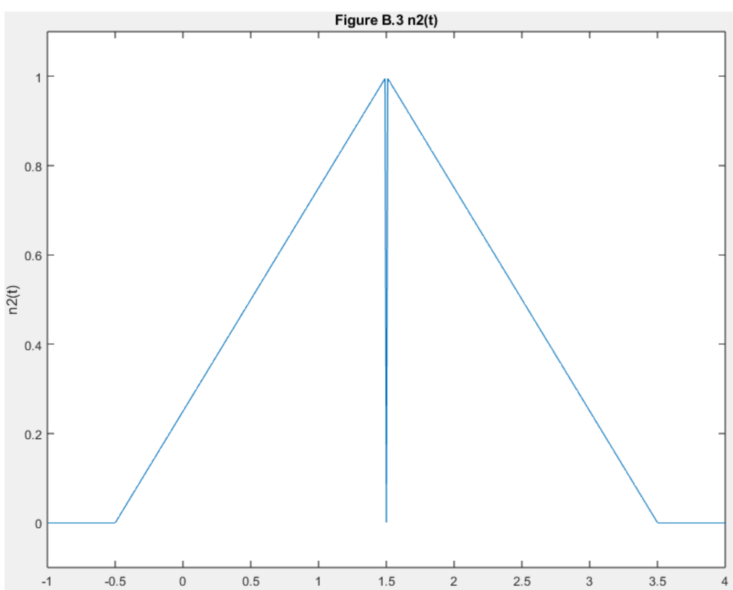
Code:

```
t = (-4:0.01:4);
n1=@(t) n(0.5*t);

plot(t,n1(t));
xlabel('t');
ylabel('n1(t)');
title('Figure B.3 n1(t)');
axis([-1 4 -0.1 1.1]);
```



Code:

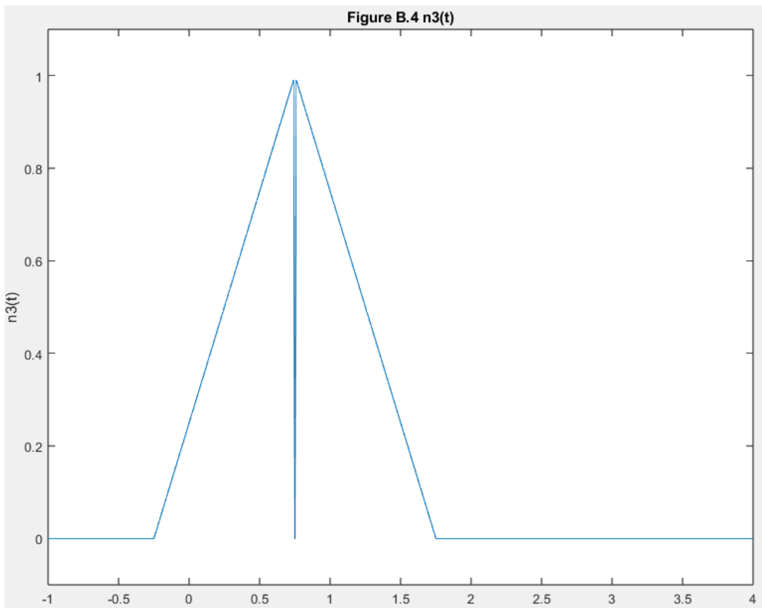
```
t = (-4:0.01:4);
n1=@(t) n(0.5*t);
n2=@(t) n1(t+0.5);

plot(t,n2(t));
xlabel('t');
ylabel('n2(t)');
title('Figure B.3 n2(t)');
axis([-1 4 -0.1 1.1]);
```



Problem B.4: Plot the following two signals: n3(t) = n(t + 1/4), n4(t) = n3(1/2t).

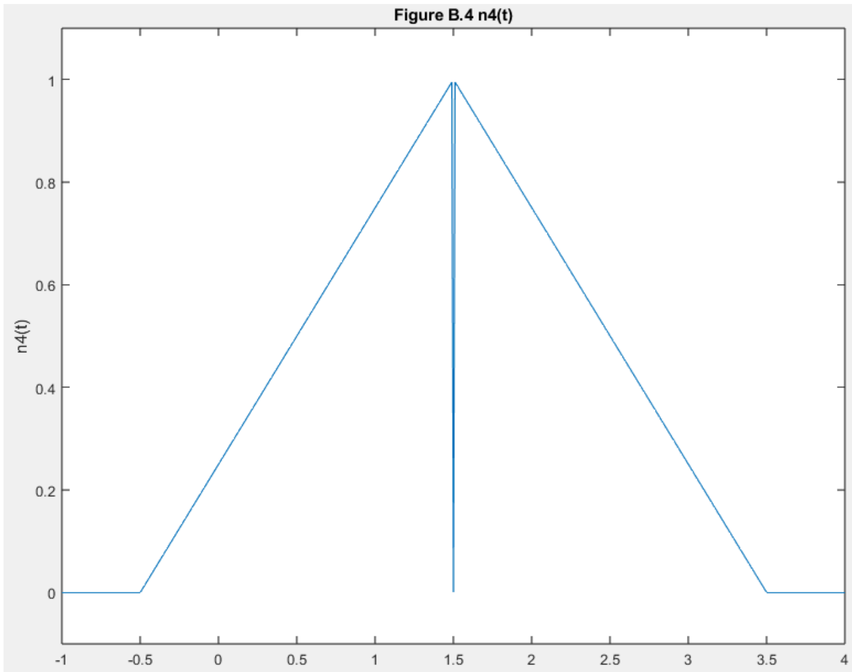
Code:

```
t = (-4:0.01:4);
n1=@(t) n(0.5*t);

plot(t,n1(t));
xlabel('t');
ylabel('n1(t)');
title('Figure B.3 n1(t)');
axis([-1 4 -0.1 1.1]);
```



Code:

```
t = (-4:0.01:4);
n1=@(t) n(0.5*t);
n2=@(t) n1(t+0.5);

plot(t,n2(t));
xlabel('t');
ylabel('n2(t)');
title('Figure B.3 n2(t)');
axis([-1 4 -0.1 1.1]);
```



Problem B.5: Compare n4(t) and n2(t) and explain their differences and/or similarities.

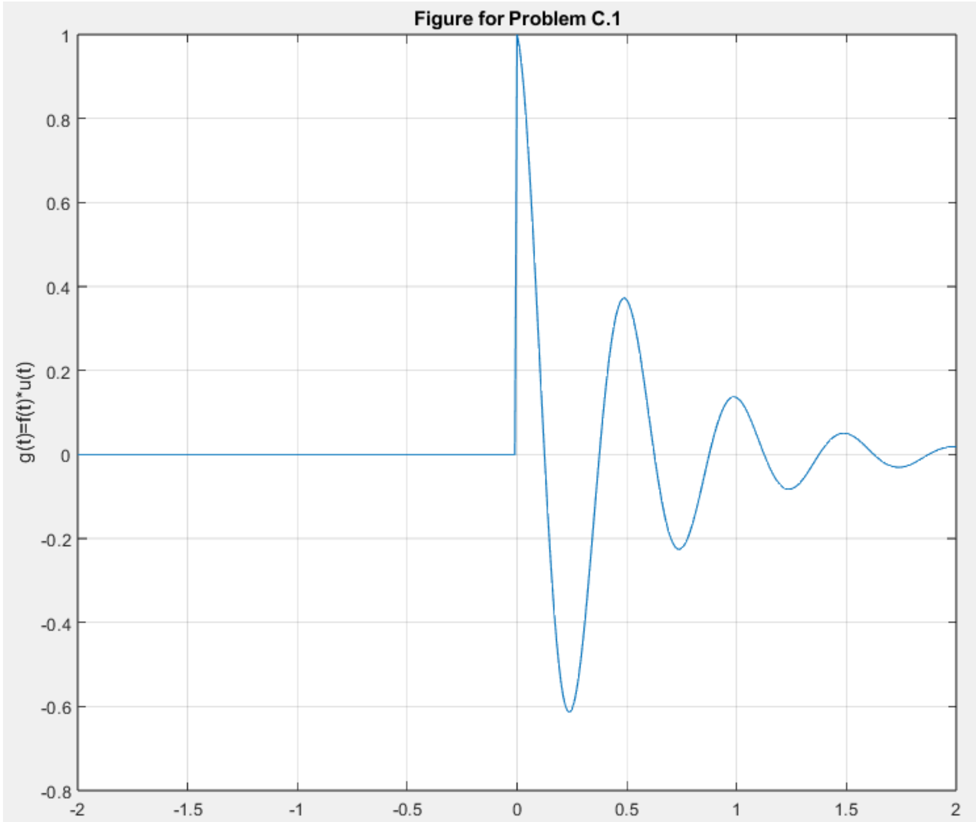
Observing both n4(t) and n2(t) we can see that they produce the exact same graph.

Visualizing operations on the independent variable and algorithm vectorization

Problem C.1: Follow the steps in this exercise, but to instead generate g(t) = f(t)u(t) where f(t) = e −2t cos(4πt).

Code:

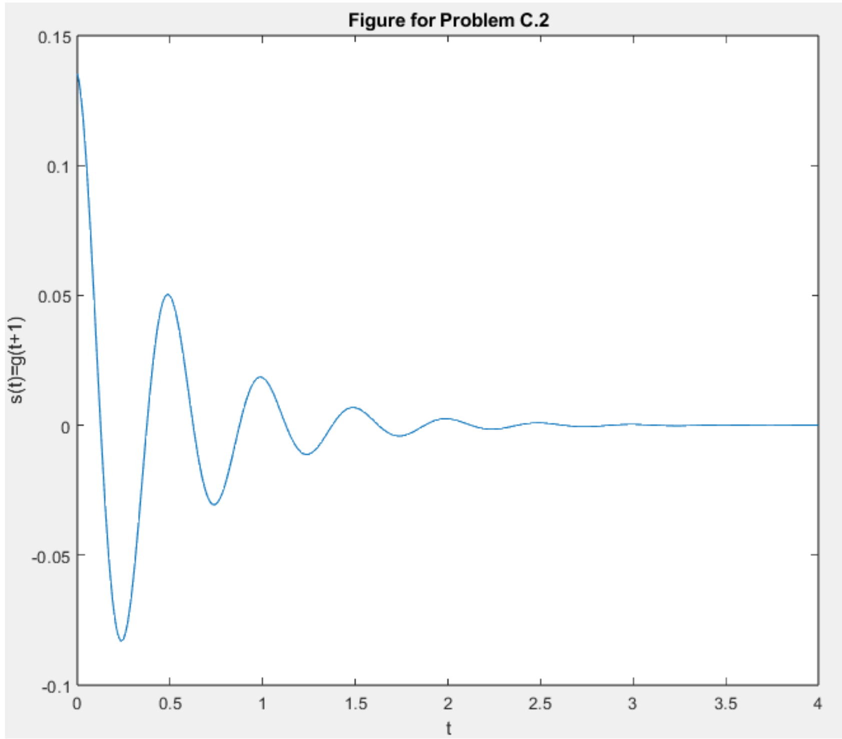
```
t=(-2:0.01:2);
f=@(t) exp(-2*t).*cos(4*pi*t);

u=@(t) 1.0.*(t>=0);
axis([-2 2 -0.1 1.1]);

g=@(t) f(t).*u(t);

plot(t,g(t));
xlabel('t');
ylabel('g(t)=f(t)*u(t)');
title('Figure for Problem C.1');
```



<u>Problem C.2</u>: Using g(t) in C.1, generate and plot s(t) = g(t+1) = e −2 e −2t cos(4πt)u(t+ 1) for t=[-2:0.01:4].

Code:

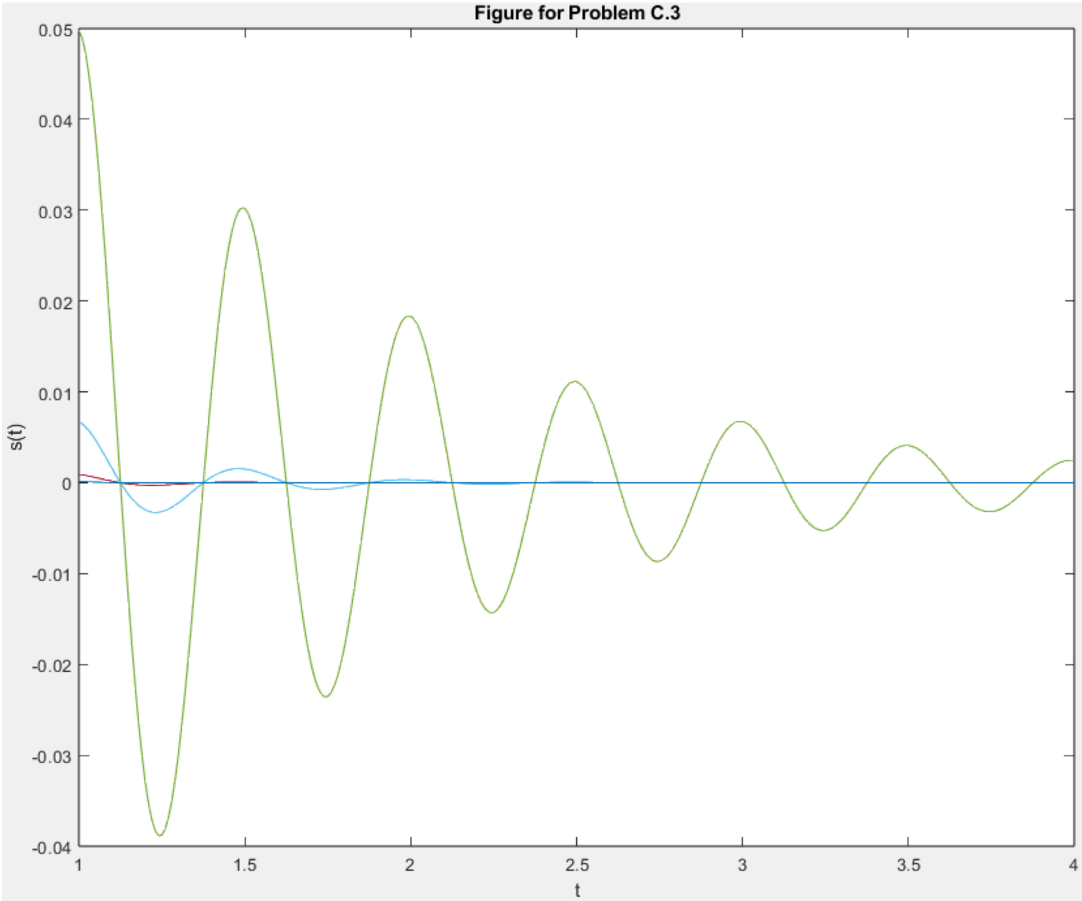
```
f=@(t) exp(-2*t).*cos(4*pi*t);
t=(-2:0.01:2);

u=@(t) 1.0*(t>=0);
axis([-2 2 -0.1 1.1])

g=@(t) f(t).*u(t);
t=(-2:0.01:2);

s=@(t) g(t+1);
t=(0:0.01:4);

plot(t,s(t));
xlabel("t");
ylabel("s(t)=g(t+1)");
title('Figure for Problem C.2');
```



<u>Problem C.3</u>: Plot sα(t) = e −2 e −αt cos(4πt)u(t) for α = [1, 3, 5, 7] in one figure for t=[0:0.01:4].

Code:

```
u=@(t) 1.0.*(t>=0);
t=(1:0.01:4);

for alpha = 1:2:7
	s=@(t) exp(-2).*exp(-alpha*t).*cos(4*pi*t).*u(t);

	plot(t,s(t));
	xlabel('t');
	ylabel('s(t)');
	title('Figure for Problem C.3');

	hold on;
end
```



Problem C.4: Determine the size of the matrix you generated in C.3.

When using the command size(s) the result that MATLAB provides is that it is a [1,1] matrix.

Array Indexing

Problem D.1: Let A be a 5 × 4 matrix array with real-valued elements: For the matrix A in Equation (1) implement the following operations:

a. A(:)

This operation lists all the elements in the matrix in one column.

```
 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

b. A([2 4 7])

Lists the second, fourth, and seventh elements of the array.

1.8339   0.8622   -0.4336

c. [A >= 0.2]

Creates a logical matrix by checking if each element in the array is larger or equal to 0.2, and providing a 1 or a 0 accordingly.

5×4 logical array

1	0	0	0
1	0	1	0
0	1	1	1
1	1	0	1
1	1	1	1

d. A([A >= 0.2])

Lists all of the elements in the array that are greater or equal to 0.2.

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

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

Finds all the elements in matrix A that are greater or equal to 0.2 and replaces them with 0. The operation then displays the new matrix.

A =

0	-1.3077	-1.3499	-0.2050
0	-0.4336	0	-0.124
-2.2588	0	0	0
0	0	-0.0631	0
0	0	0	0

Problem D.2: Let B be a 1024 × 100 data matrix representing 100 blocks of non-overlapping 1024-element input samples from a particular data source.

a. Write a simple Matlab program using two nested for loops that will set all elements of the data matrix B with magnitude values below 0.01 to zero: B(i, j) = 0, if |B(i, j)| < 0.01, (2) where B(i, j) is an element of the data matrix B in i-th row and j-th column.

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

D2_a=B;
toc
```

b. Repeat part (a) using Matlab's indexing features as described in Problem D.1.

```
tic;
B([ B >= 0.01 ]) = 0;
D2_b=B;
toc;
```

c. Use the Matlab commands tic and toc to compare the execution time of the code you wrote in parts (a) and (b).

Execution time for code in part a:

Elapsed time is 0.009291 seconds.

Execution time for code in part b:

Elapsed time is 0.007445 seconds.

Problem D.3: Let x audio be a 20,000 sample-long row vector representing 2.5 sec of an audio signal sampled at 8 kHz. A simple data compression algorithm can be implemented by setting all elements of the data array x audio with magnitude values below a threshold to zero. Write such a data compression algorithm and listen to the processed audio file.

```
x_audio_copy = x_audio;

count=0;
threshold=0.01;

for i = 1:20000
    if x_audio_copy(i,1) < threshold
        x_audio_copy(i,1) = 0;
        count=count+1;
    end
end
disp(count);
sound(x_audio_copy);
```

Message in processed audio file:
Members of a public service alliance of Canada