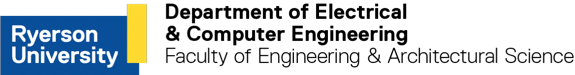
****

|  |  |
| --- | --- |
| **Course Title:** | Signals and Systems I |
| **Course Number:** | ELE 532 |
| **Semester/Year (e.g. F2017)** | F2018 |

|  |  |
| --- | --- |
| **Instructor** | Dimitri Androutsos |

|  |  |
| --- | --- |
| ***Assignment/Lab Number:*** | Lab Assignment 1 |
| ***Assignment/Lab Title:*** | Lab Assignment 1 |

|  |  |
| --- | --- |
| ***Submission Date:*** | September 30, 2018 |
| ***Due Date:*** | September 30, 2018 |

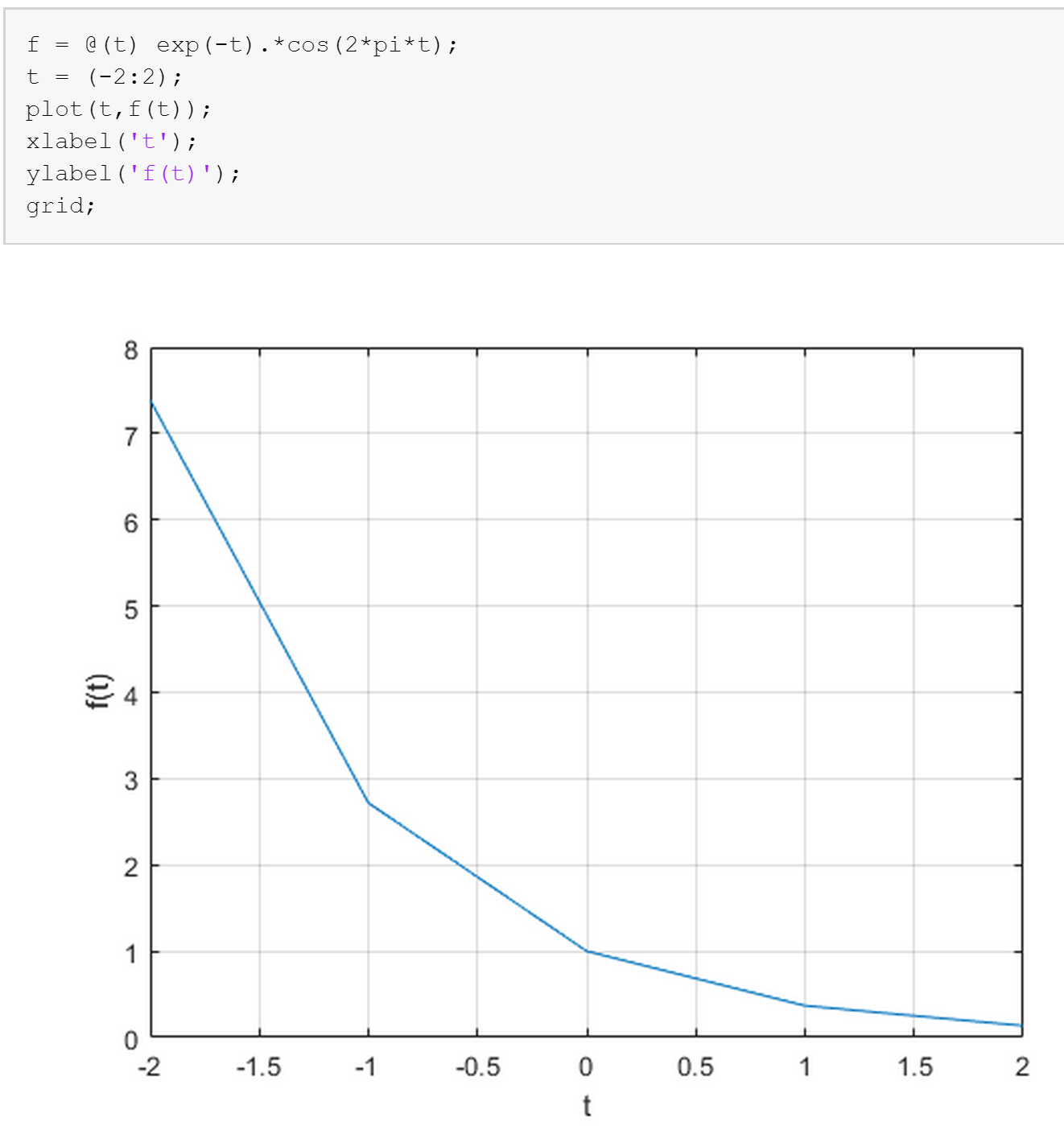
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Student LAST Name** | **Student FIRST Name** | **Student Number** | **Section** | **Signature\*** |
| Patel | Parth | 500542681 | 03 |  |
| Shreekant | Vatsal | 500771363 | 03 |  |

\*By signing above you attest that you have contributed to this written lab report and confirm that all work you have contributed to this lab report is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a “0” on the work, 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: <http://www.ryerson.ca/senate/current/pol60.pdf>

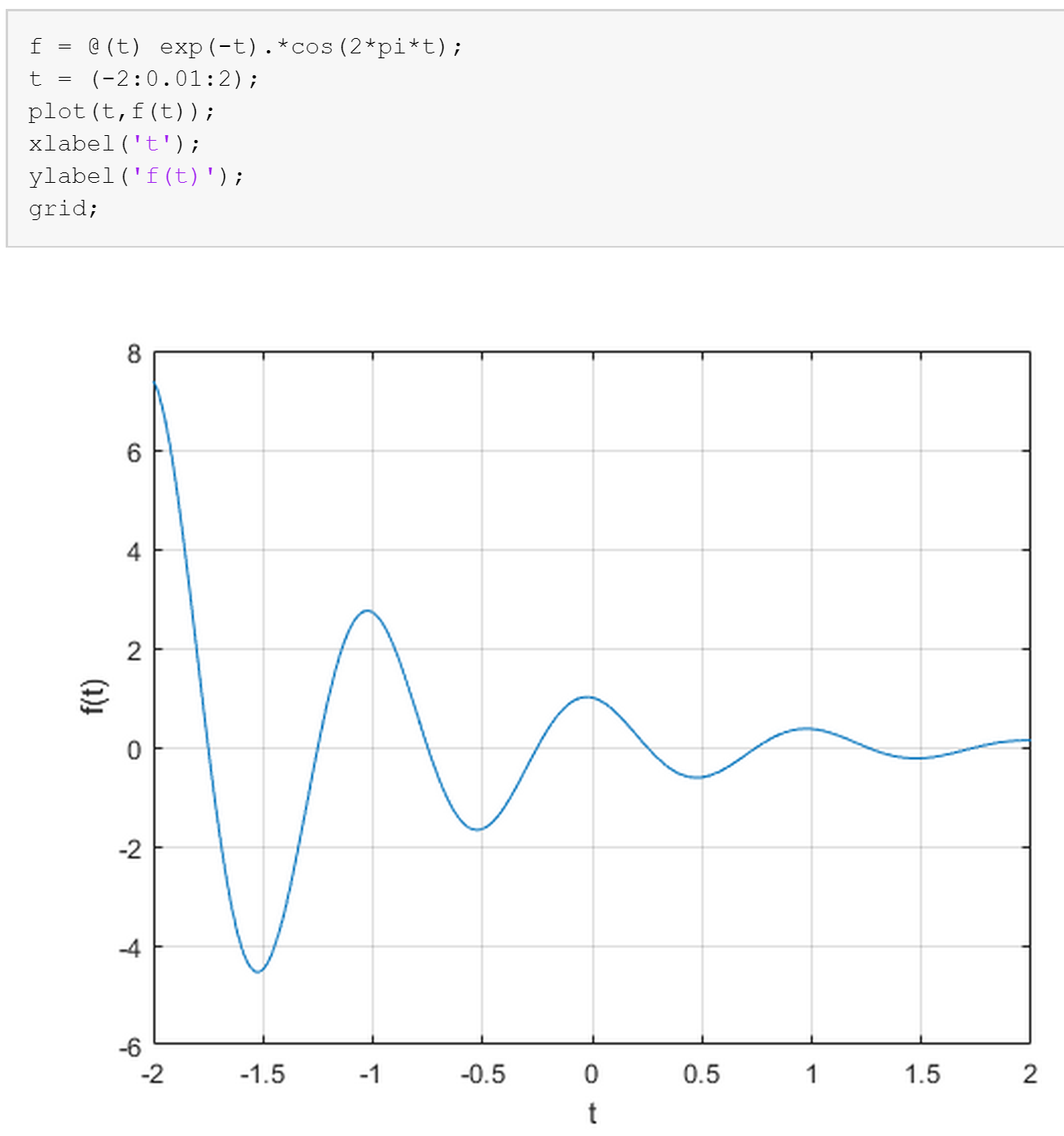
## Question A

## A1

Section 1.11-1 Anonymous Functions, Lathi, 3rd Ed., page 126. Generate and plot the graphs as shown in Figures 1.46 and 1.47 on page 127



f(t) = e−t cos(2πt) with Δt = 1 s

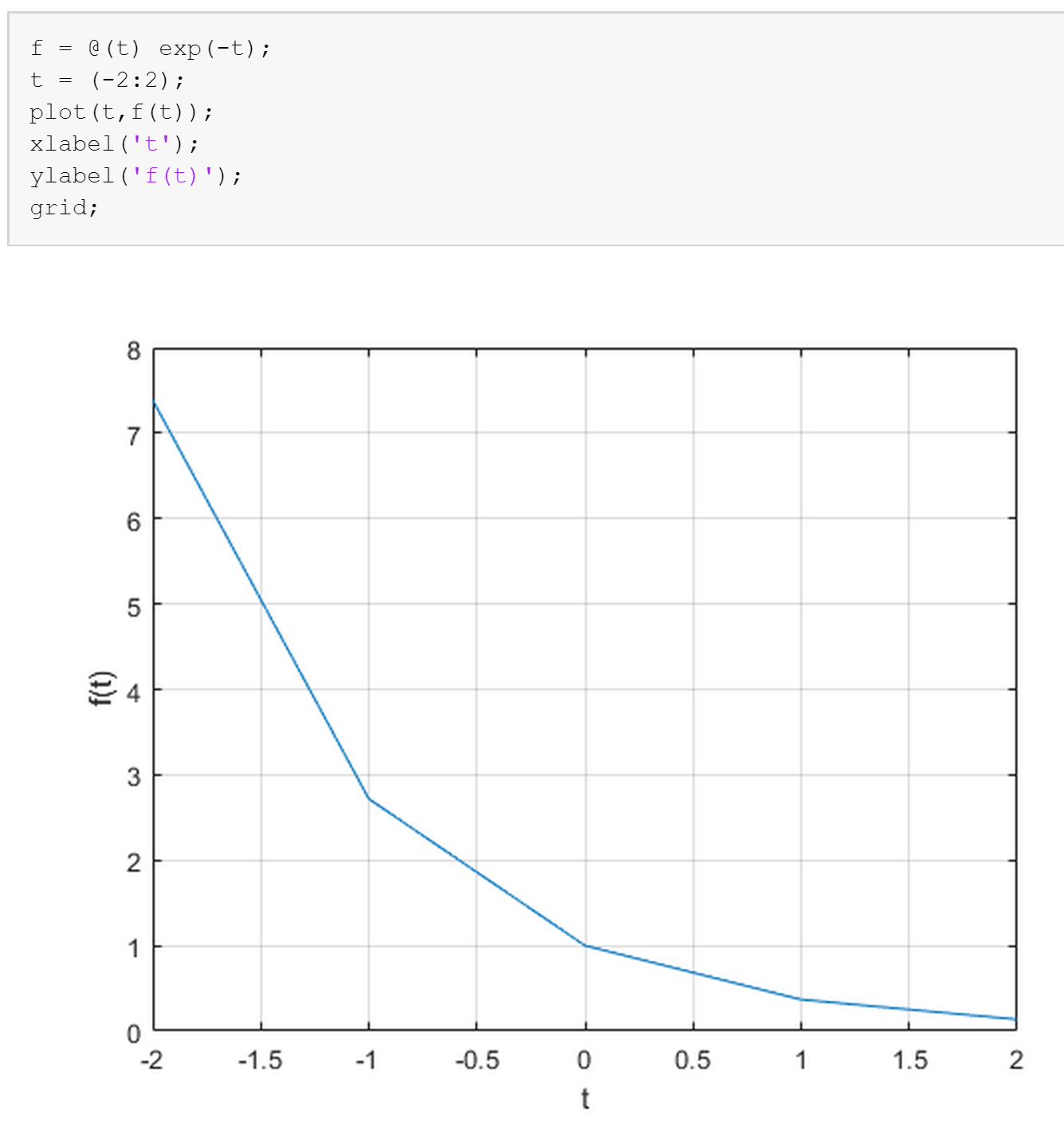


f(t) = e−t cos(2πt) with Δt = 10-2 s

## 

## A2

Plot the function e −t for t taking on integer values contained in −2 ≤ t ≤ 2; you can generate these values using the MATLAB command tt=[-2:2].

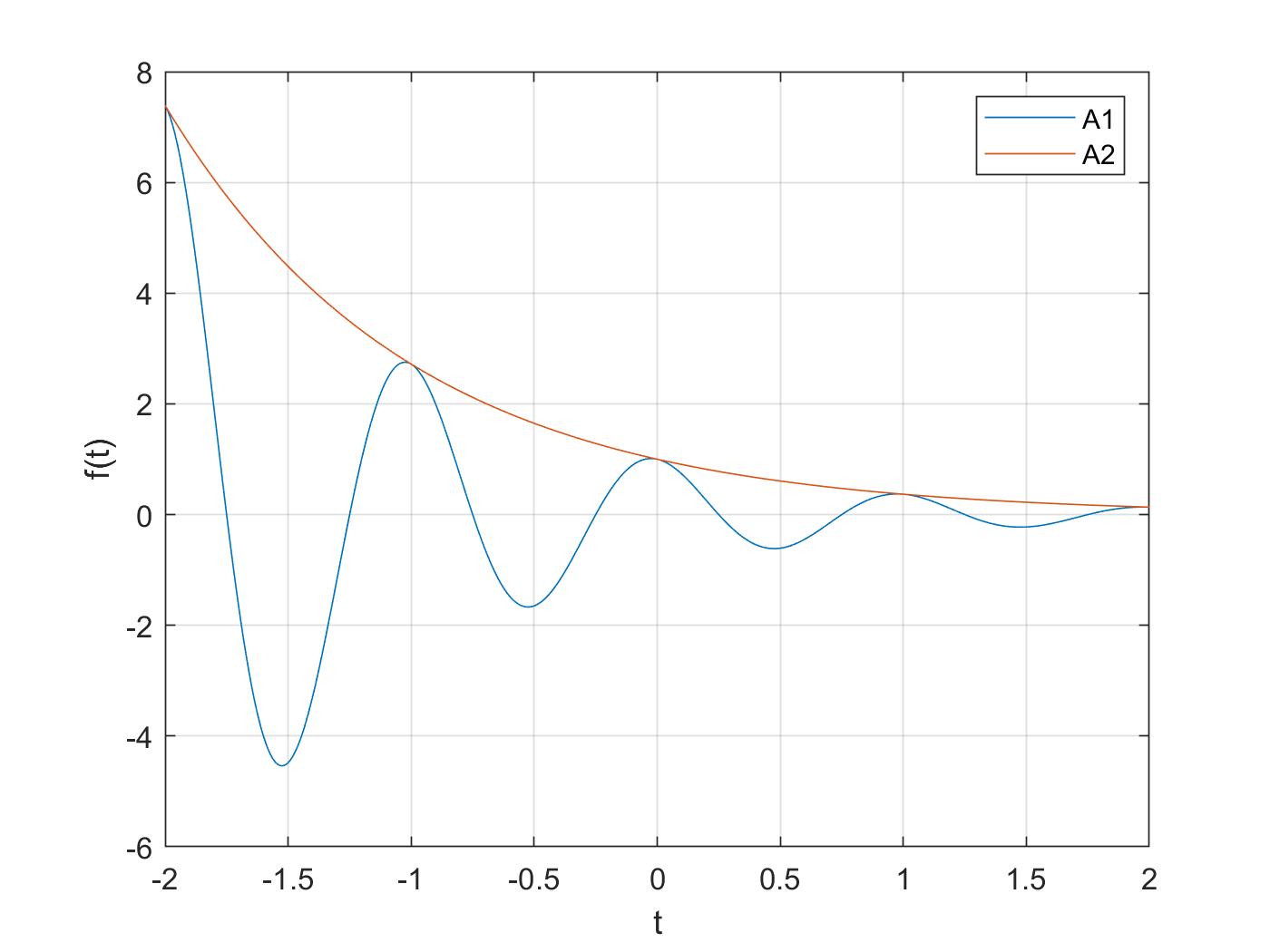


f(t) = e−t with Δt = 1 s

## A3

Compare the results of Problem A.2 with Figure 1.46 in Problem A.1.

The results of problem A.2 with figure 1.46 in problem A.1 are in contrast with each other. It is obvious that the two given functions are different in terms of the formula however they produce strikingly similar graphs. Figure 1.46 is a dampening plot in which an inverse exponential function is multiplied with a cos function to produce the dampening effect. Whereas, the figure in A.2 is simply an inverse exponential function. This can be observed graphically by the plots below.



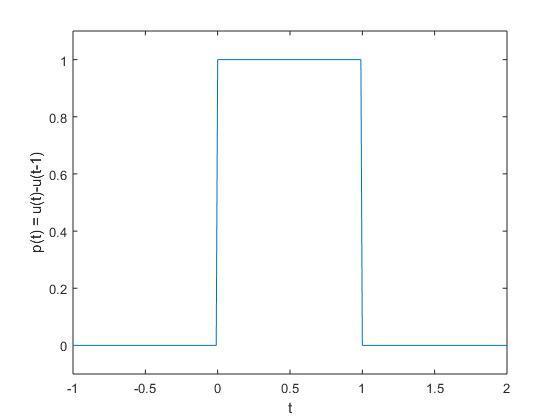
## 

## Question B

## B1

Section 1.11-2 Lathi, 3rd Ed., page 128. Generate and plot p(t) as shown in Figure 1.50 on page 129.

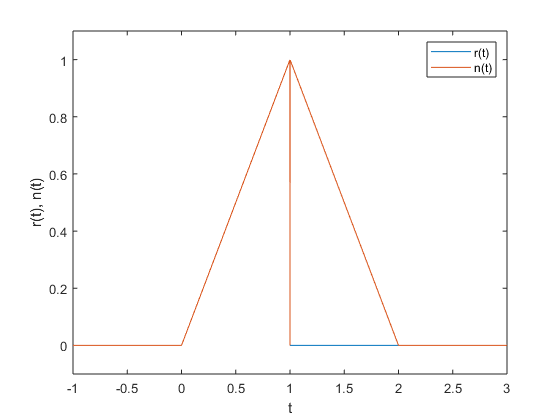
p = @(t) 1.0.\*((t>=0)&(t<1));  
t = (-1:0.001:5);  
  
plot(t,p(t));  
xlabel('t');  
ylabel('p(t) = u(t)-u(t-1)');  
axis([-1 2 -.1 1.1]);  
legend("p(t)")



## B2

Use p(t) de?ned in Problem B.1 to generate and plot functions r(t) = tp(t) and n(t) = r(t) + r(?t + 2).

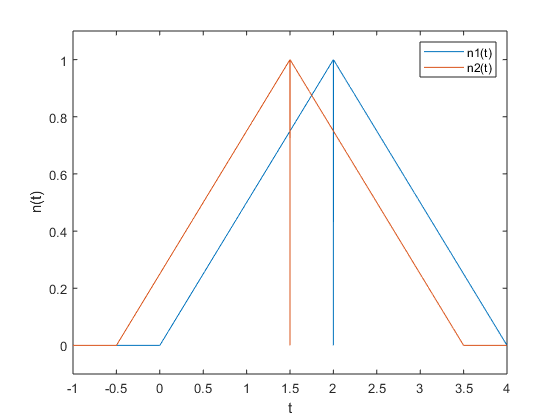
r = @(t) (t.\*p(t));  
n = @(t) (r(t) + r(-t + 2));  
  
plot(t,r(t),t,n(t));  
xlabel('t');  
ylabel('r(t), n(t)');  
axis([-1 3 -.1 1.1]);  
legend("r(t)", "n(t)")



## B3

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

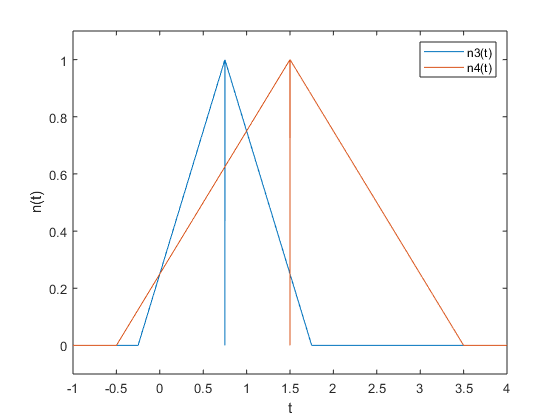
n1 = @(t) (n(0.5.\*t));  
n2 = @(t) (n1(t + 0.5));  
  
plot(t,n1(t),t,n2(t));  
xlabel('t');  
ylabel('n1(t), n2(t)');  
axis([-1 5 -.1 1.1]);  
legend("n1(t)", "n2(t)")



## B4

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

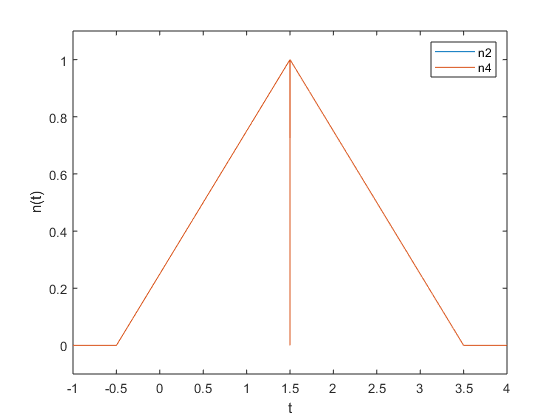
n3 = @(t) (n(t + 0.25));  
n4 = @(t) (n3(0.5.\*t));  
  
plot(t,n3(t),t,n4(t))  
xlabel('t');  
ylabel('n3(t), n4(t)');  
axis([-1 4 -.1 1.1]);  
legend("n3(t)", "n4(t)")



## B5

Compare n2(t) and n4(t)

plot(t,n2(t),t,n4(t));  
xlabel('t');  
ylabel('n(t)');  
axis([-1 4 -.1 1.1]);  
legend("n2(t)", "n4(t)");



[*Published with MATLAB® R2018a*](https://www.mathworks.com/products/matlab)

### 

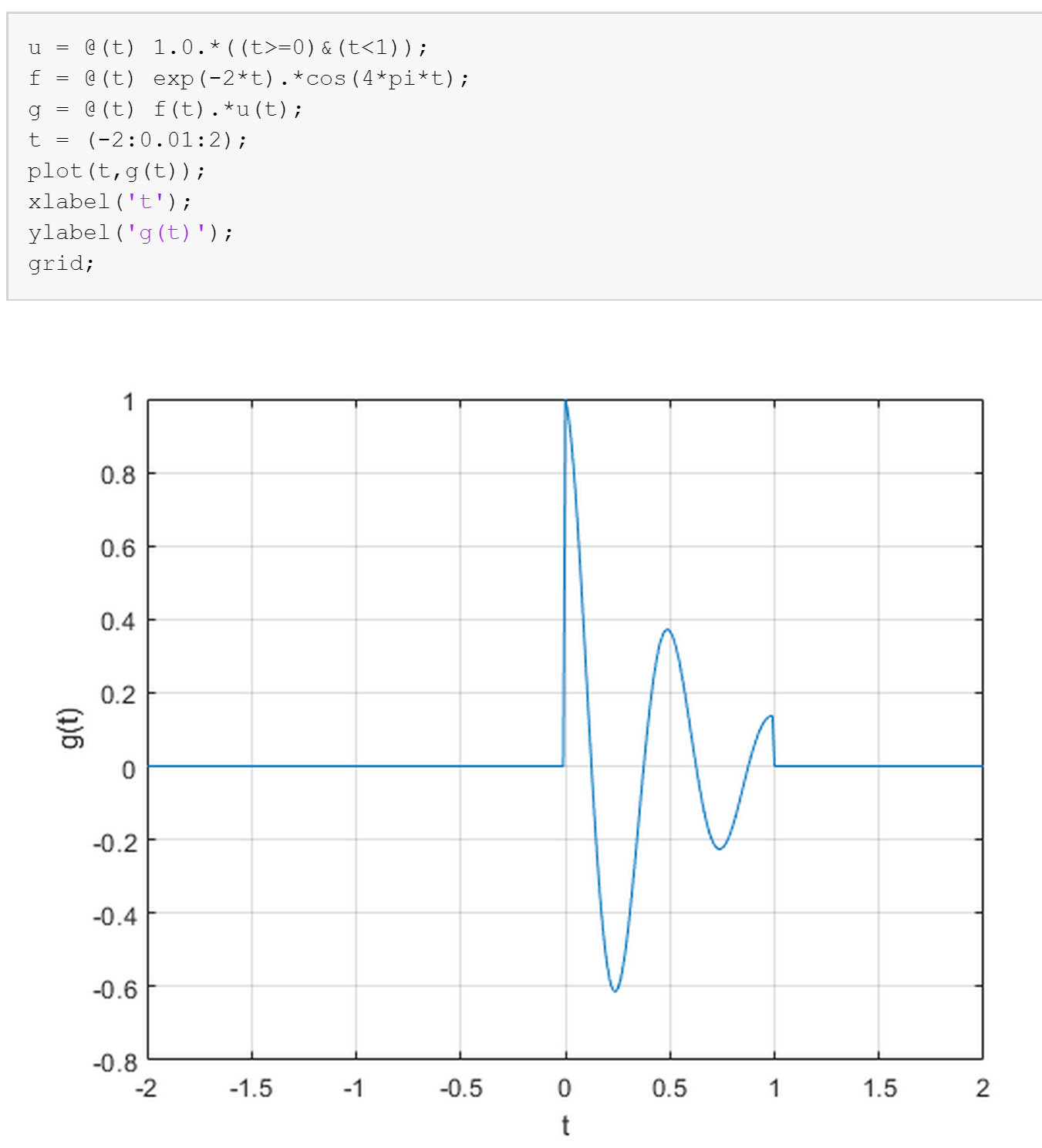
* Here n2 and n4 overlap each other, i.e. they are the same.

## 

## Question C

## C1

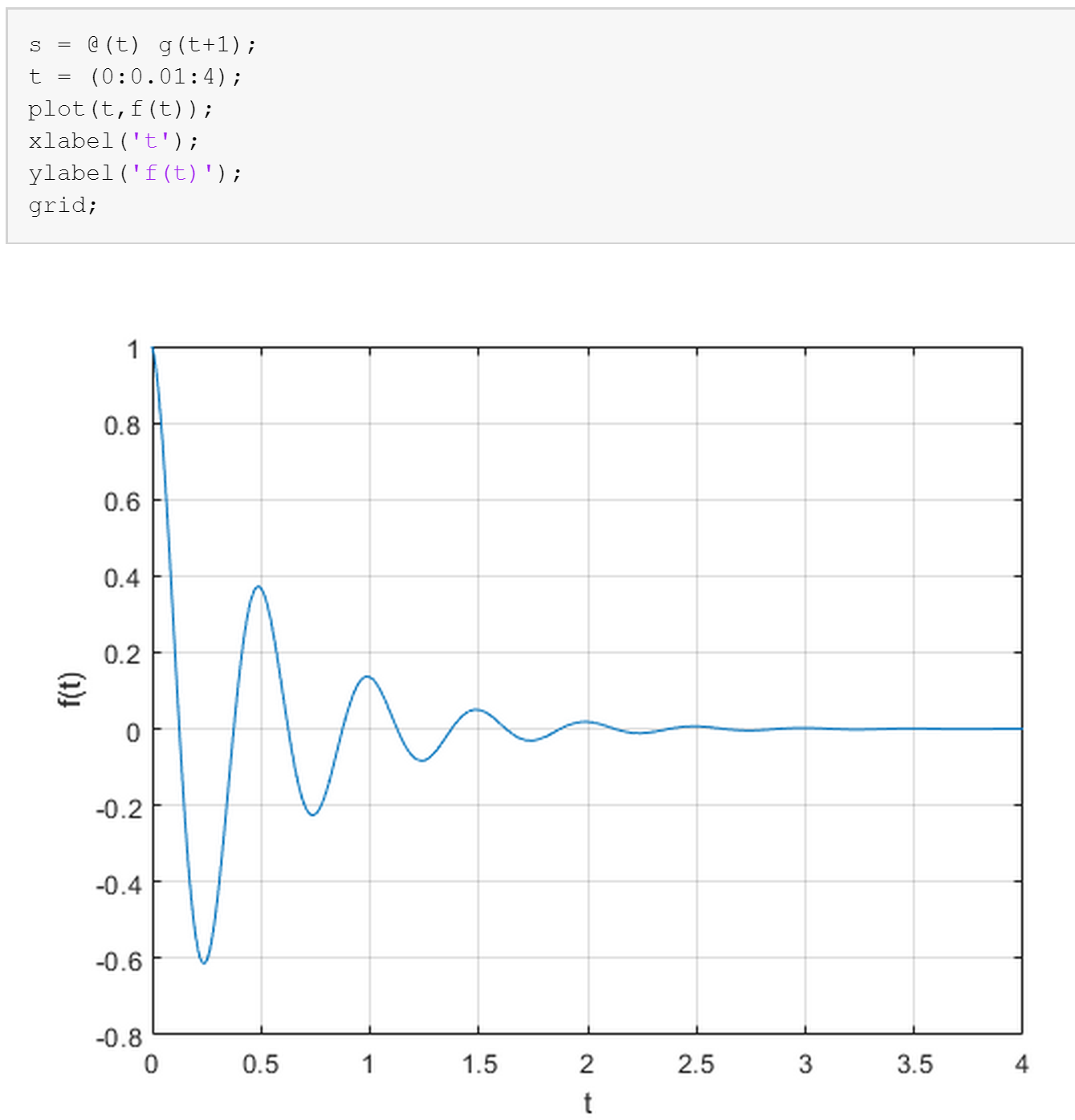
Section 1.11-3, Lathi, 3rd Ed., page 130. Follow the steps, but instead, generate g(t) = f(t)u(t) where f(t) = e −2t cos 4πt



g(t) = e−2t cos (4πt) with Δt = 10-2 s

## C2

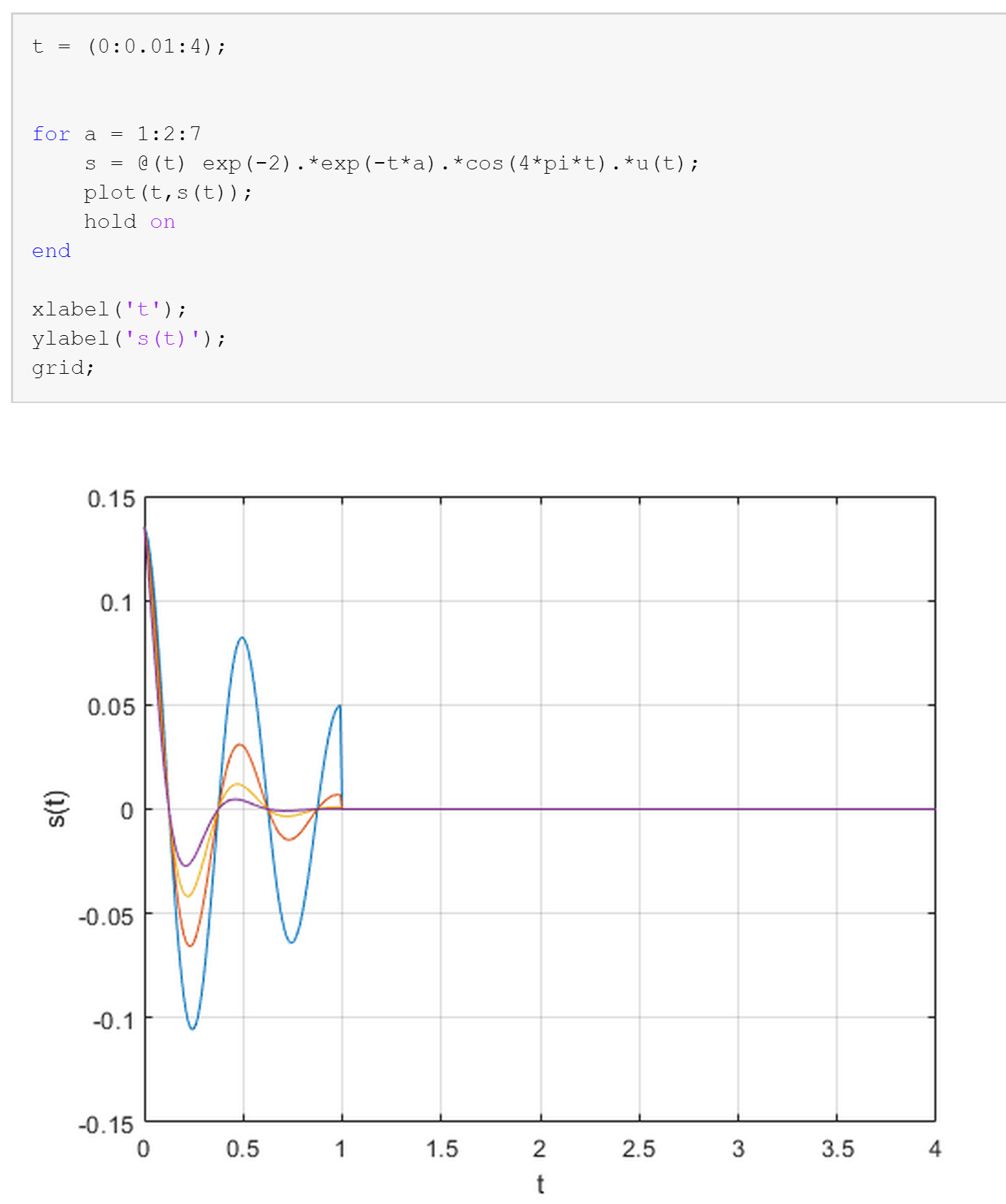
Using g(t) as described in Problem C.1, generate and plot s(t) = g(t + 1) for t = [0 : 0.01 : 4]



s(t) = g(t+1) with Δt = 10-2 s

## C3

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].

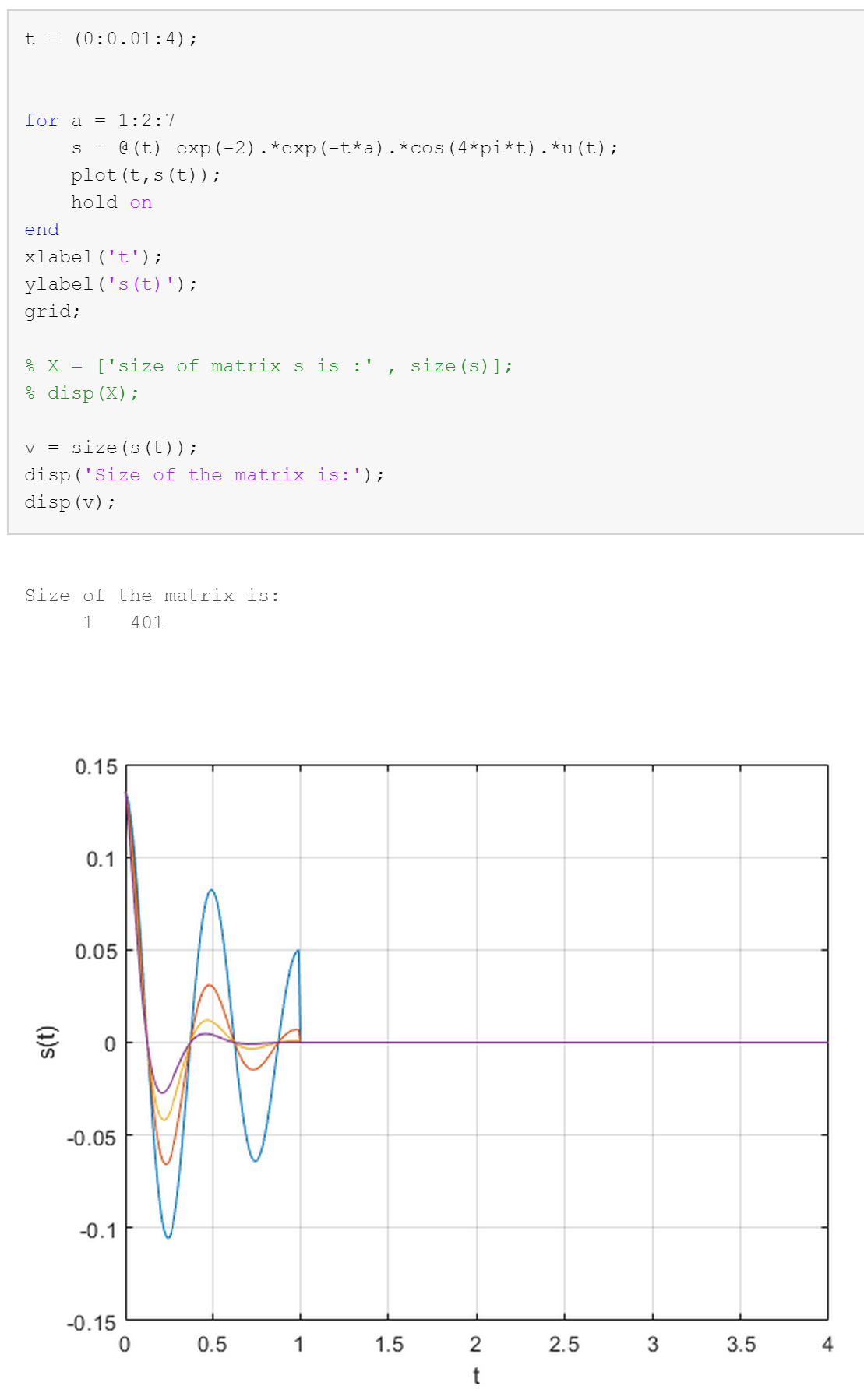


sα(t) = e-2eαtcos(4πt)u(t) with Δt = 10-2 s

## C4

Determine the size of the matrix s(t) generated in Problem C.3.

The size of the matrix s(t) generated in Problem C.3 is 401.



Question D

## D1

Perform indexing operations on matrix A

a) This operation shows the whole matrix as a list

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

b) This operation shows the requested elements from the matrix A(in the list form)

A([ 2 4 7 ])

ans =  
  
 1.8339 0.8622 -0.4336

c) This operation convertes the matrix A into a logical array of 1’s and 0’s. Any value smaller than 0.2 is converted to zero and rest are converted to one.

[ 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

#### 

d) This operation shows all the elements of matrix A that are greater than or equal to 0.2. It does this by multiplying the matrix A by its logical array, leaving only the values greater than or equal to threshold of 0.2.

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

#### 

e) This operation shows turns all the elements of the matrix A to zero whose value is greater than or equal to 0.2.

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

## D2

Write a simple MATLAB that will set all elements of the data matrix B with magnitude values below 0.01 to zero, once using a nested for loop and once using MATLAB's indexing operations. Also time both algorithms.

B1=B;  
B2=B;  
  
disp("Time spent while using double for loops: ");  
tic  
for i = 1:1024  
 for j= 1:100  
 if (abs(B1(i,j)))<0.01  
 B1(i,j) = 0;  
 end  
 end  
end  
toc  
  
disp(newline + "Time spent while using MATLAB’s indexing feature : ");  
tic  
B2((abs(B2) < 0.01))=0;  
toc

Time spent while using double for loops:   
Elapsed time is 0.008721 seconds.  
  
Time spent while using MATLAB’s indexing feature :   
Elapsed time is 0.001879 seconds.

[*Published with MATLAB® R2018a*](https://www.mathworks.com/products/matlab)

## D3

Write a simple compression algorithm to zero all the values less than the threshold value in an audio file.

x = x\_audio;  
threshold = 0.015;  
  
tic  
x((abs(x) < threshold))=0;  
zero\_count = size(find(x==0));  
toc  
  
disp("Number of elements that are zero = " + zero\_count(1));  
sound(x,8000);

Elapsed time is 0.001935 seconds.  
Number of elements that are zero = 4343

[*Published with MATLAB® R2018a*](https://www.mathworks.com/products/matlab)

### 