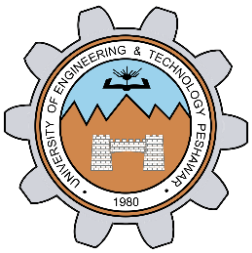
INTRODUCTION TO COMPLEX SIGNALS IN MATLAB

LAB # 05



Spring 2023

CSE301L Signals & Systems Lab

Submitted by: **Ali Asghar**

Registration No. : **21PWCSE2059**

Class Section: **C**

"On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work."

Submitted to:

Engr. Sumayyea Salahuddin

Date:

April 3, 2023

Department of Computer Systems Engineering

University of Engineering and Technology, Peshawar

Lab Objective(s):

Objectives of this Lab are;

- Gain familiarity with Complex Numbers and plot them
- Complex exponential signals
- Real exponential signals

Task # 01:

Write matlab function zprint, which takes a complex number and returns it real part, imaginary part, magnitude, phase in radians, and phase in degrees.
A sample run of program is:

```
>> zprint(z)
Z = X + jY Magnitude Phase Ph(deg)
      3     4         5         0.927   53.13
```

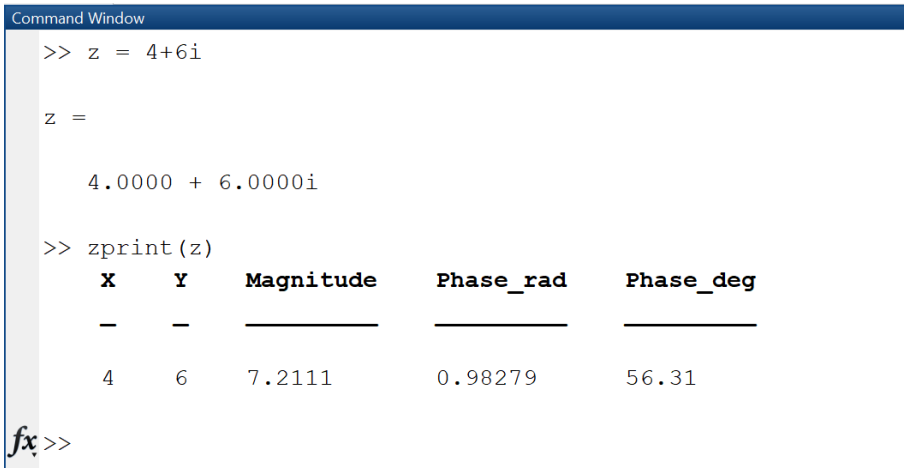
Problem Analysis:

For complex calculations, extensive analysis is required. Here, a complicated number is examined.

Algorithm:

- Define the function.
- Find the real part and store it in X variable.
- Find the imaginary part and store it in Y variable.
- Find the absolute and store it in Magnitude variable.
- Find the phase in degrees and radians and then store it in their respective variables.
- Display data using disp
- Define a complex number z = 4+6i
- Print it using zprint

Output / Graphs / Plots / Results:



Discussion and Conclusion:

Complex numbers can be analysed with MATLAB.

Task # 02:

Compute the conjugate ź (i.e. z_conj [give variable name]) and the inverse 1/z (i.e. z_inv [give variable name]) for any complex number z. Display the results numerically with zprint.

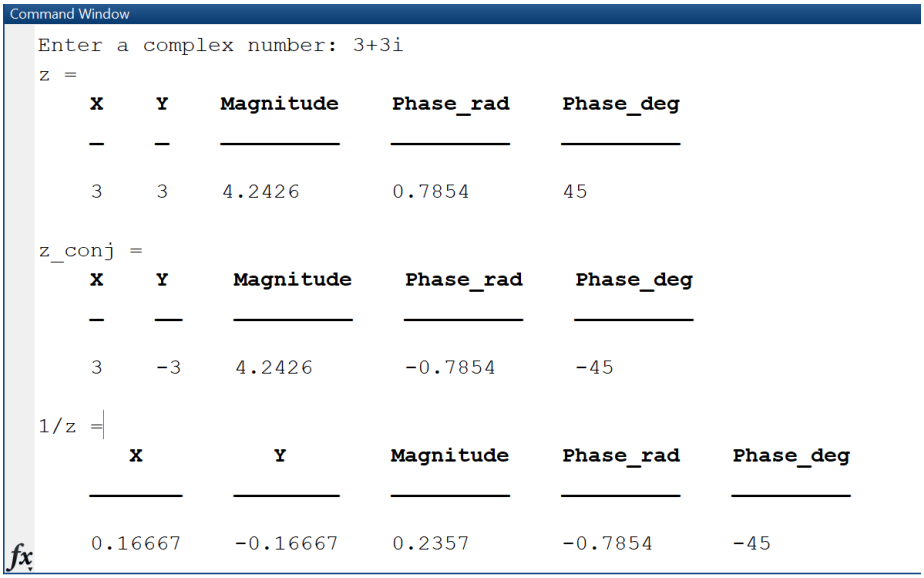
Problem Analysis:

For complex calculations, extensive analysis is required. Here, a complicated number is examined.

Algorithm:

- Prompt the user to input a complex number and store it in the variable "z".
- Compute the complex conjugate of "z" using the "conj()" function, and store it in the variable "z_conj".
- Compute the inverse of "z" by dividing 1 by "z", and store it in the variable "z_inv".
- Display the results numerically using the "zprint()" function for each complex number: "z", "z_conj", and "z_inv".
- Print using zprint

Output / Graphs / Plots / Results:



Discussion and Conclusion:

Complex numbers can be analysed with MATLAB.

Task # 03:

Take two complex number and compute z1 +z2 and display the results numerically using zprint.

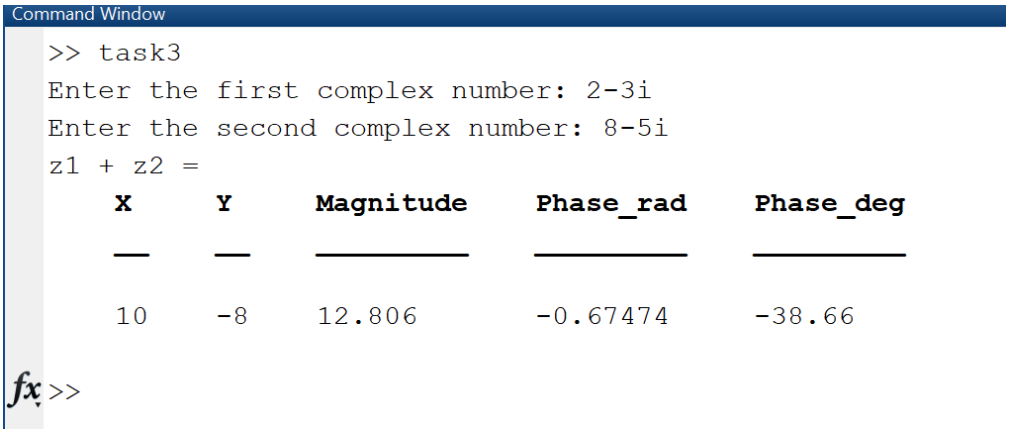
Problem Analysis:

Complex calculations require complex operations.

Algorithm:

- Prompt the user to input two complex numbers and store them in the variables "z1" and "z2".
- Compute the sum of "z1" and "z2" by adding them together and storing the result in the variable "z_sum".
- Display the numerical result of the sum of "z1" and "z2" using the "zprint()" function.

Output / Graphs / Plots / Results:



Discussion and Conclusion:

Complex numbers can be computed in MATLAB as well as analysed.

Task # 04:

Take two complex numbers and compute z1z2 and z1/z2. Use zprint to display the results numerically.

Problem Analysis:

For Complex Analysis, complex operations are required.

Algorithm:

- Prompt the user to input two complex numbers and store them in the variables "z1" and "z2".
- Compute the product of "z1" and "z2" by multiplying them together and storing the result in the variable "z_product".
- Compute the quotient of "z1" and "z2" by dividing "z1" by "z2" and storing the result in the variable "z_quotient".
- Display the numerical result of the product of "z1" and "z2" using the "zprint()" function.
- Display the numerical result of the quotient of "z1" and "z2" using the "zprint()" function.

Output / Graphs / Plots / Results:

```
>> task4
Enter the first complex number: 4+i
Enter the second complex number: 3-3i
z1 x z2 =
    X      Y      Magnitude     Phase_rad     Phase_deg
    __     __     _________     _________     _________

    15     -9     17.493        -0.54042      -30.964

z1 / z2 =
     X         Y        Magnitude     Phase_rad     Phase_deg
    ___    _______     _________     _________     _________

    0.5    0.83333     0.97183       1.0304        59.036

fx >>
```

Discussion and Conclusion:

with addition to analysis, complex numbers can be computed with MATLAB.

Task # 05:

Determine the complex conjugate of the exponential signal given in above example and plot its real and imaginary portions.

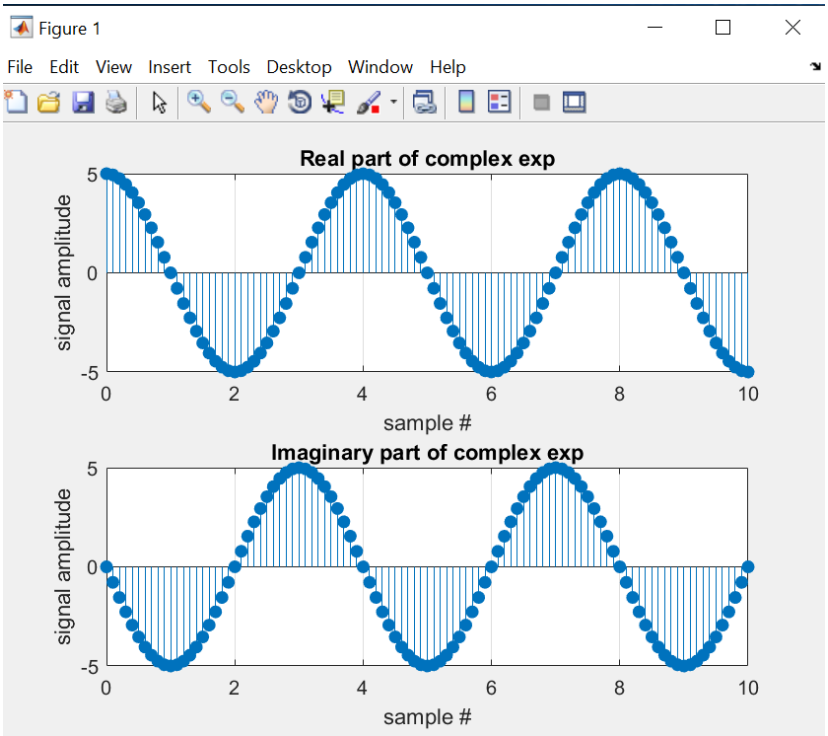
Problem Analysis:

In MATLAB, complex numbers may be computed and studied.

Algorithm:

- Define a discrete time range from 0 to 10 with a step size of 1/10, and store it in the variable "n".
- Set the values of amplitude "k" and phase angle "a" for the exponential function.
- Compute the complex exponential function using "k", "a", and "n", and store it in the variable "x".
- Compute the complex conjugate of "x" using the "conj()" function, and store it in the variable "x_conj".
- Plot the real and imaginary parts of "x_conj" using the "stem()" function, with "n" on the x-axis and the real and imaginary parts of "x_conj" on the y-axis. Add appropriate titles, x-labels, y-labels, and grids to the plots.

Output / Graphs / Plots / Results:



Discussion and Conclusion:

Complex numbers can be computed in MATLAB as well as analysed.

Task # 06:

Generate the complex valued signal and plot its magnitude, phase, the real part, and the imaginary part in separate subplots.
$y(n) = \exp(-0.2 + j0.5n)$, $-10 \leq n \leq 10$

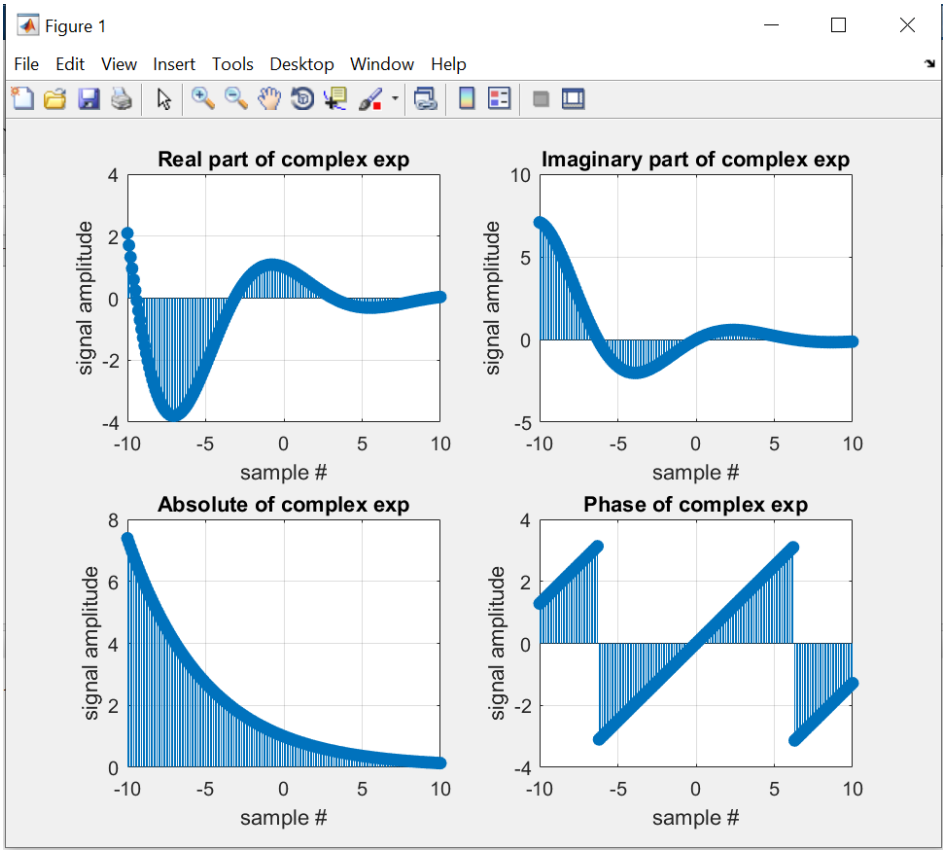
Problem Analysis:

Complex Analysis requires complex operation as well as complex visualization.

Algorithm:

- Define a discrete time range from -10 to 10 with a step size of 1/10, and store it in the variable "n".
- Compute the complex exponential function using the values of amplitude (-0.2) and phase angle (0.5i), and the values of "n", and store it in the variable "x".
- Plot the real, imaginary, absolute, and phase angle of "x" using the "stem()" function, with "n" on the x-axis and the corresponding values on the y-axis. Add titles, x-labels, y-labels, and grids to each plot using the "title()", "xlabel()", "ylabel()", and "grid()" functions.

Output / Graphs / Plots / Results:



Discussion and Conclusion:

MATLAB can analyze as well visualize a complex number.

Task # 07:

a) Generate a real-exponential x=a * exp (n) for a=0.7 and n ranging from 0-10. Find the discrete time as well as the continuous time version of this signal. Plot the two signals on same graph (holding both the graphs).
b) Repeat the same program with value of a=1.3.

Problem Analysis:

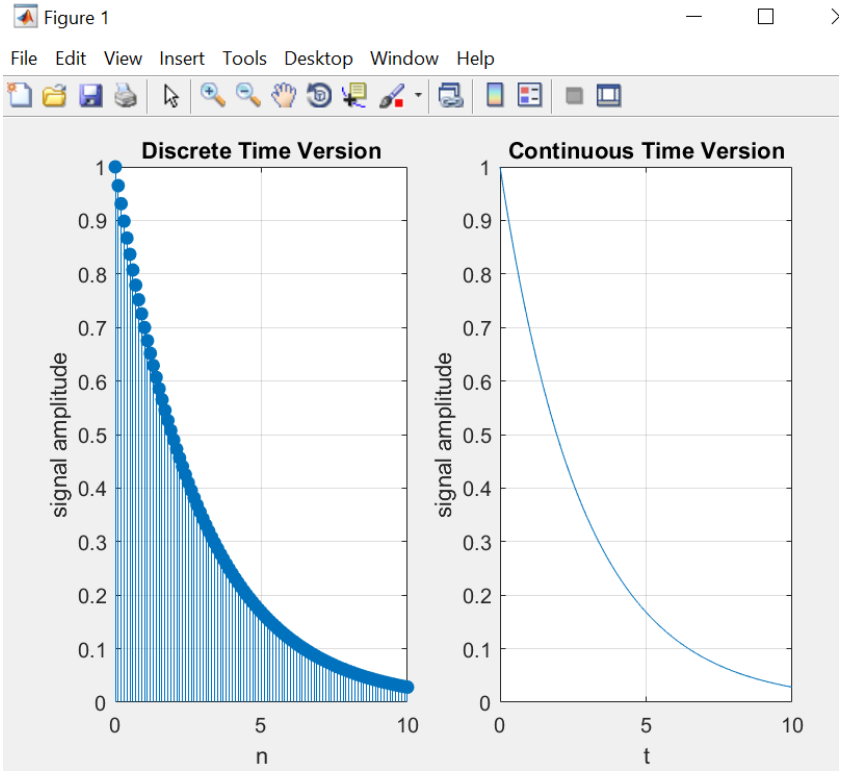
Different signal systems employ exponential signals. In MATLAB, we work with exponential signals.

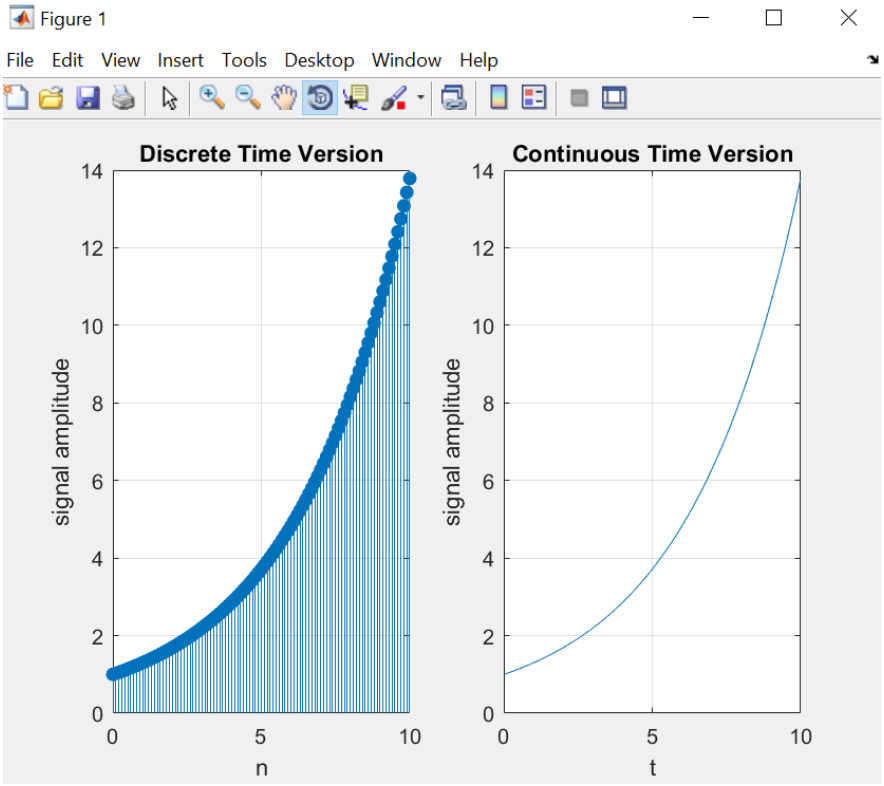
Algorithm:

- Enter a exponential signal
- Plot its discrete as well as its continuous function

Output / Graphs / Plots / Results:

a)





Discussion and Conclusion:

MATLAB can analyze as well visualize a complex number.

Task # 08:

Multiply the two discrete signals x1=5 exp(i*n*pi/4) and x2= a * exp(n) (use point-by-point multiplication of the two signals). Plot the real as well as the exponential parts for 0<a<1 and a>1.

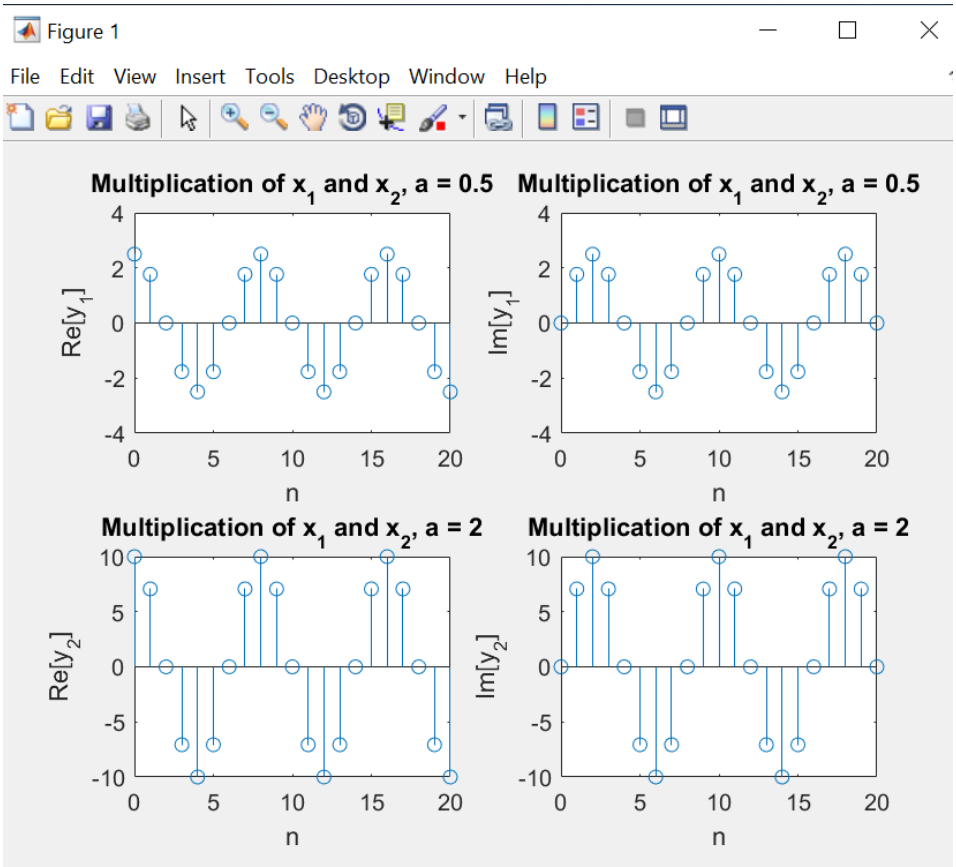
Problem Analysis:

Signal and systems computations for exponential signals are necessary. Here, we do that using MATLAB.

Algorithm:

- Enter two exponential signals
- Multiply them

Output / Graphs / Plots / Results:



Discussion and Conclusion:

MATLAB can analyze as well visualize a complex number.

Task # 09:

Plot the discrete signal x=a^|n| for n ranging from -10 to 10. Draw two subplots for 0<a<1 and a>1.

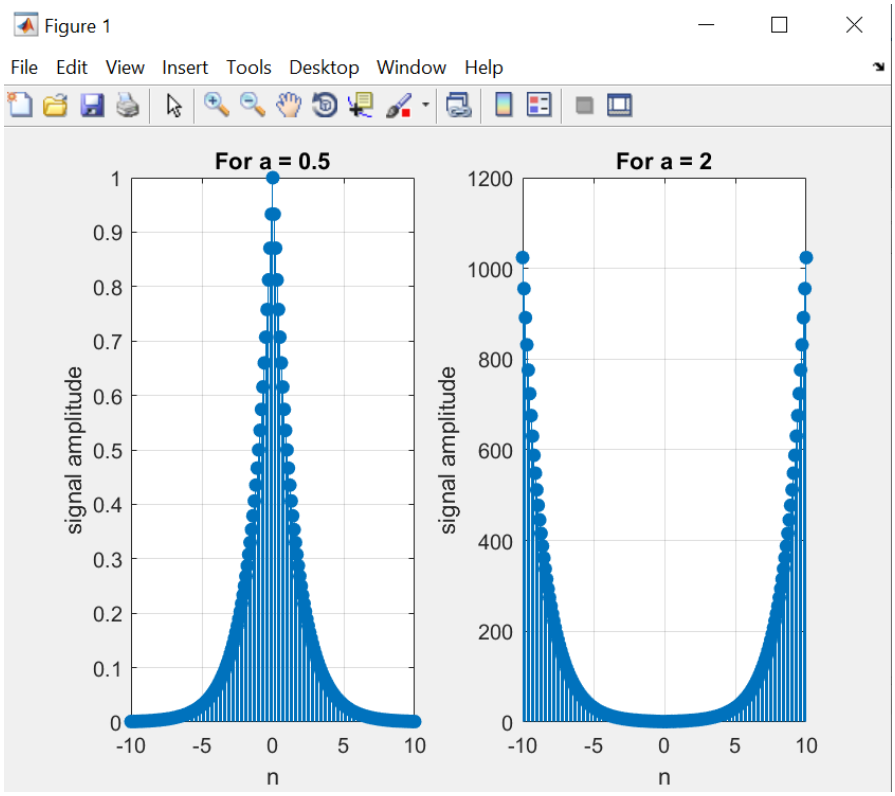
Problem Analysis:

We usually desire to plot discrete signals. That is accomplished in MATLAB.

Algorithm:

- Enter the discrete signal.
- Plot it.

Output / Graphs / Plots / Results:



Discussion and Conclusion:

MATLAB can analyze as well visualize a complex number.

Task # 10:

a) Generate the signal x(t) = Ae(jωt + π) for A = 3, π= -0.4, and ω = 2π(1250). Take a range for t that will cover 2 or 3 periods.
b) Plot the real part versus t and the imaginary part versus t. Use subplot(2,1,i) to put both plots in the same window.
c) Verify that the real and imaginary parts are sinusoids and that they have the correct frequency, phase, and amplitude.

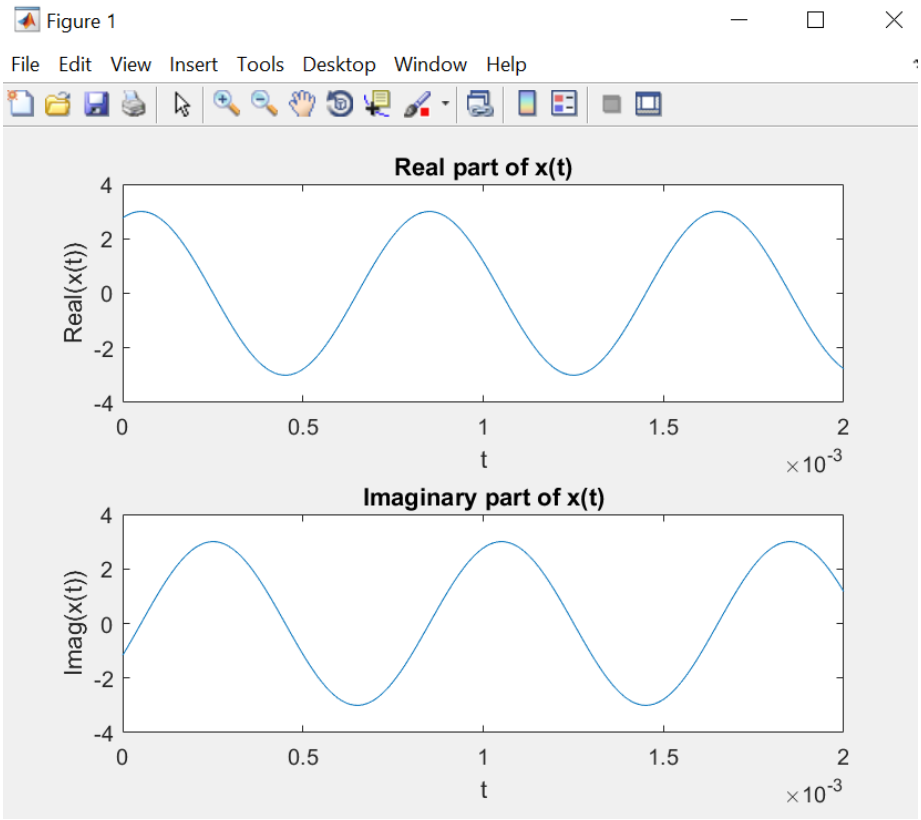
Problem Analysis:

In addition to other signals, the complex signal can also be plotted. (real and imaginary plot)

Algorithm:

- Enter/generate the complex signal
- Plot its real and imaginary parts

Output / Graphs / Plots / Results:



Discussion and Conclusion:

MATLAB can analyze as well visualize a complex number.