Name: David, Ruzzel S.
Section: CPE-401
Date Assigned: August 9,2023
Date Submitted: August 11,2023

EXPERIMENT 2
Basic Signal Sequences and Other Signal Waveforms

OBJECTIVE

To generate the following basic signal sequences: unit sample or unit impulse, unit step, unit ramp, real exponential and complex exponential, and sinusoidal sequence.

MATERIALS AND EQUIPMENT

Computer installed with Matlab 2018 or Octave 5.2.0

INTRODUCTION

A discrete-time signal is represented as a sequence of numbers, called samples. A sample value of a typical discrete-time signal or sequence is denoted as **x[n]** with the argument n being an integer in the range from $-\infty$ to $+\infty$.

The discrete-time signal may be a finite length or an infinite length sequence. Length pertains to the number of samples in a sequence. If x[n]= {-4, 1.5, 2.25}, then, the length is 3. A finite length (also called finite duration or finite extent) sequence is defined only for a finite time interval:

$$N1 \leq n \leq N2$$

The length or duration N of the finite length sequence is $N = N2 - N1 + 1$.

For example, if n is from –10 to 20, the length of the sequence is calculated as $N = 20 - (-10) + 1 = 31$.

Another example, if n is from 1 to 10, then the length is $N = 10 - 1 + 1 = 10$.

Basic Sequences

1) The **unit sample sequence**, often called the discrete time impulse or the unit impulse, denoted by δ[n], is defined by

$$\delta[n] = \begin{cases} 1, & for\ n = 0 \\ 0, & for\ n \neq 0 \end{cases}$$

A unit sample sequence, d[n], of length N can be generated using the Matlab/Octave command

d = [1 zeros(1,N -1)]

2) The **unit step sequence**, denoted by u[n], is defined by

$$u[n] = \begin{cases} 1, & for\ n \geq 0 \\ 0, & for\ n < 0 \end{cases}$$

A unit step sequence u[n] of length N can be generated using the Matlab/Octave command

u = [ones(1,N)]

3) The **unit ramp sequence**, denoted by r[n], is defined by

$$r[n] = \begin{cases} n, & for\ n \geq 0 \\ 0, & for\ n < 0 \end{cases}$$

A unit ramp sequence r[n] of length N can be generated using the Matlab/Octave command

r = [1 : N]

4) The **exponential sequence** is given by

$$x[n] = Ac^n$$

If the parameter c is real, then x[n] is a real exponential signal. If c is complex, then x[n] is **complex-valued exponential signal**.

5) The **real sinusoidal sequence** with a constant amplitude is of the form

$$\mathrm{x[n]} = \mathrm{Acos}(2\pi fn + \phi)$$

The parameters A, f, and ϕ are called, respectively, the amplitude, the normalized frequency, and the initial phase of the sinusoidal sequence x[n].

PROCEDURES

GENERATION OF SEQUENCES

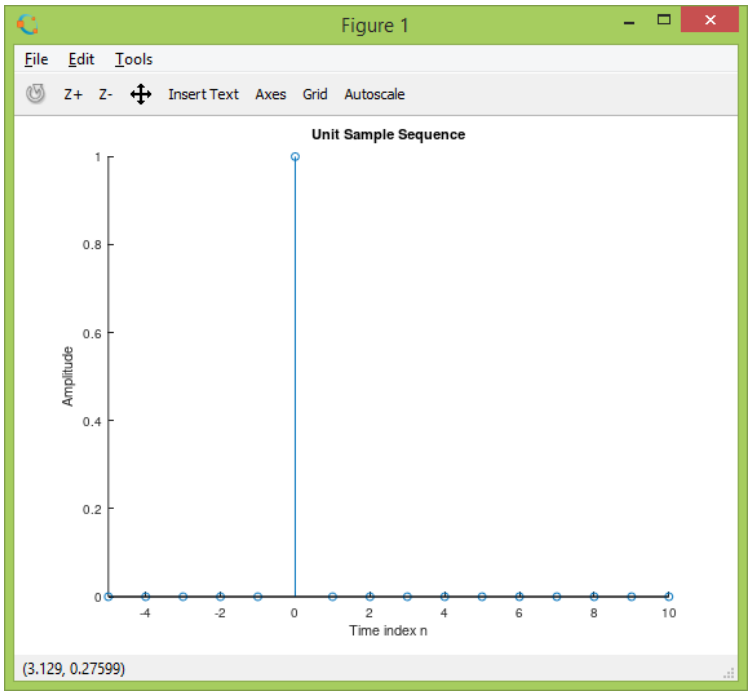
Unit Sample or Unit Impulse Sequence

The following program can be used to generate and plot a unit sample sequence:

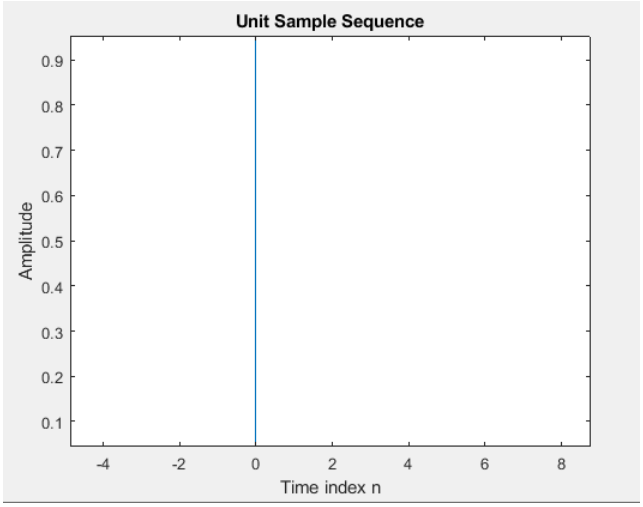
```
% Expt2_1
% Generation of a Unit Sample Sequence
clf;
clear;
% Generate a vector from -5 to 10
% The length of n is 16
n = -5:10;

% Generate the unit sample sequence
% The length of d is 16
d = [zeros(1,5) 1 zeros(1,10)];

% Plot the unit sample sequence
% The x-axis are the n values, the y-axis are the values of d
stem(n,d);
xlabel('Time index n'), ylabel('Amplitude');
title('Unit Sample Sequence');
axis([-5, 10, 0, 1]);
```



STEP 1 Create an m-file for the program above and then execute.



The purpose of **clf** command is clear current figure window.

The purpose of **clear** command is remove items from workspace, freeing up system memory.

The purpose of **axis** command is set axis limits and axis ratios.

What is the effect when the **axis** command is not used? The matlab will determine the axis limit and ensures that the data is visible.

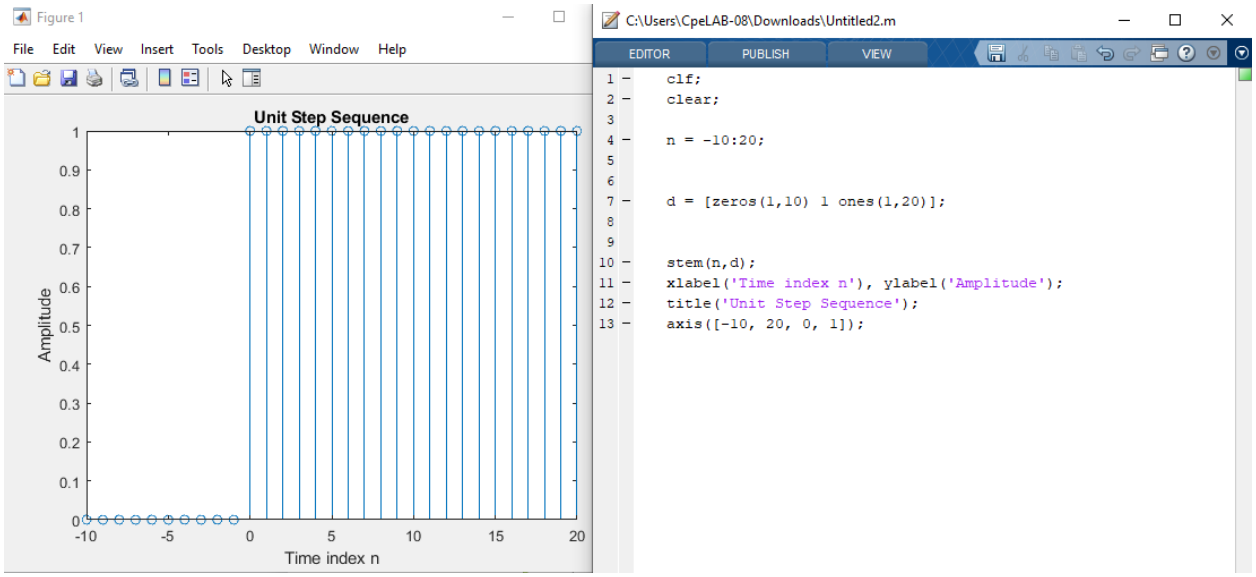
When n is changed to n=-10 to 10, what other parameters must be changed to avoid error? Write the changes. n = -10:10;

d = [zeros(1,10) 1 zeros(1,10)];

axis([-10, 10, 0, 1]);

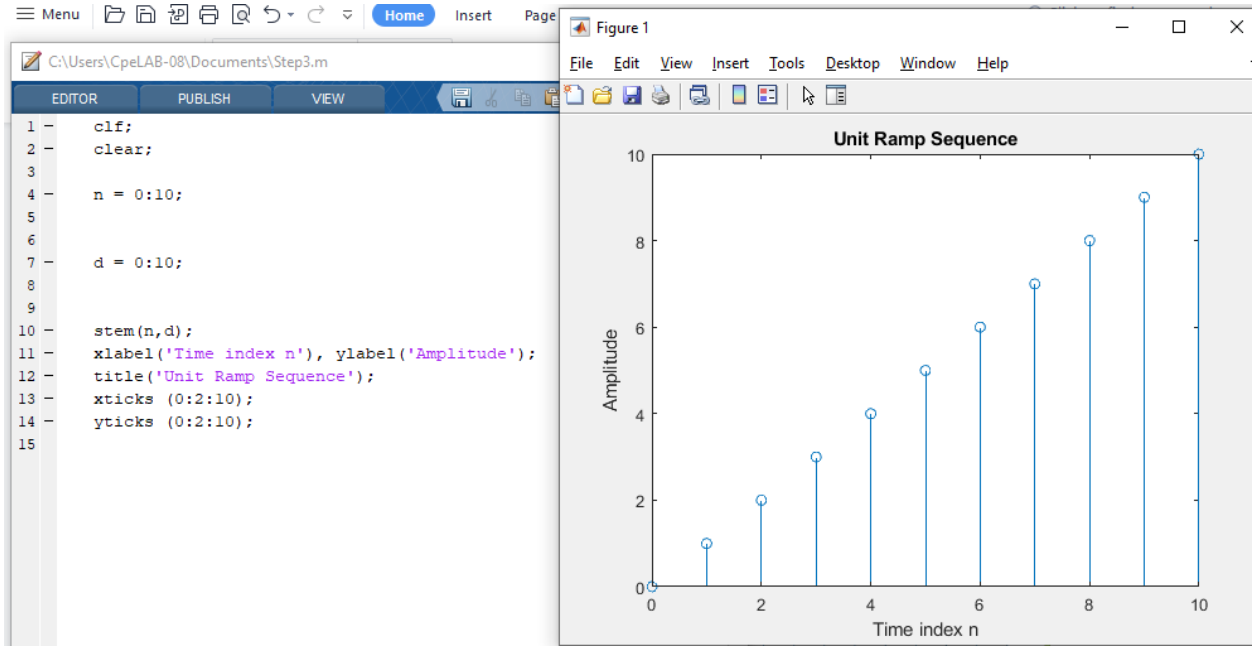
Unit Step Sequence

STEP 2 Write a program to generate a **unit step sequence** similar to the plot shown on the figure below. Write/paste your code on the space provided.



Unit Ramp Sequence

STEP 3 Write a program to generate a **unit ramp sequence** similar to the plot shown on the figure below. Write/paste your code and the figure generated on the space provided.

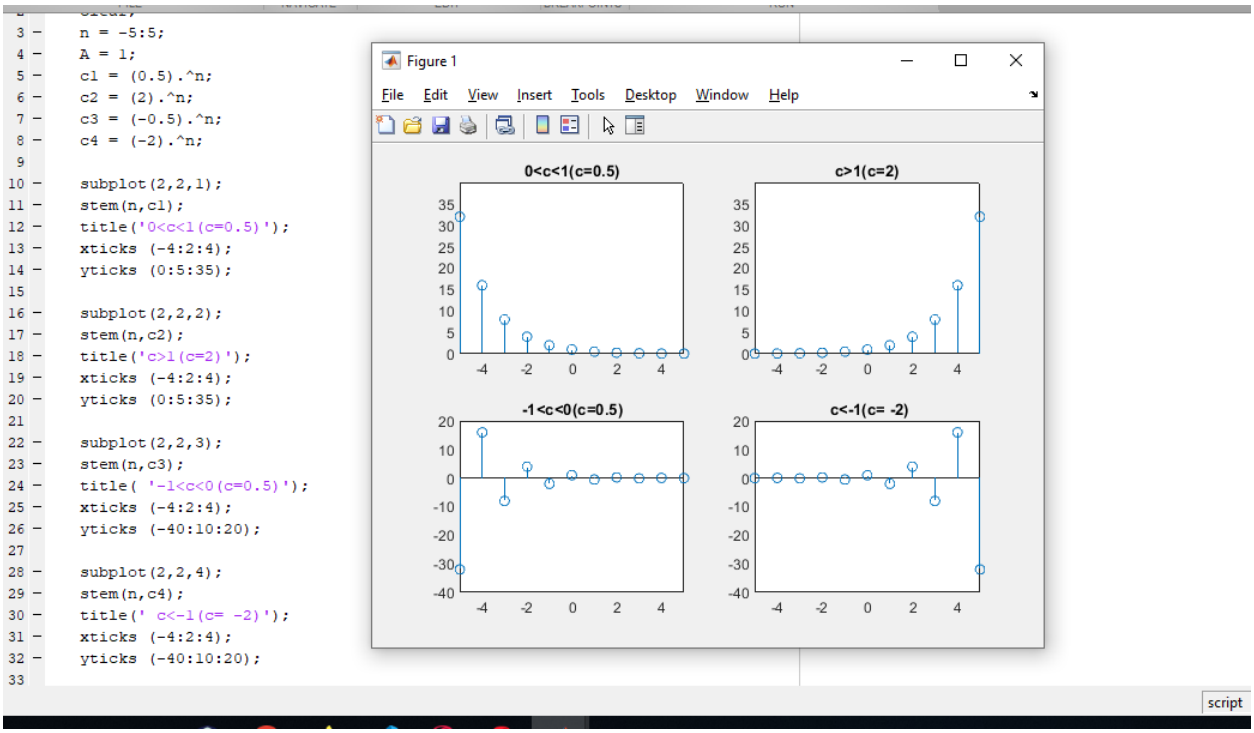


Real Exponential Sequence

STEP 4 Exponential sequence is defined by the equation, $\boldsymbol{x[n] = A(c)^n}$. Write a program to generate a subplot of four different real-valued exponential sequences with different rate of growth and decay. The time index n is from -5 to 5, amplitude A=1, and c (controls the rate of growth and decay) with four different values:

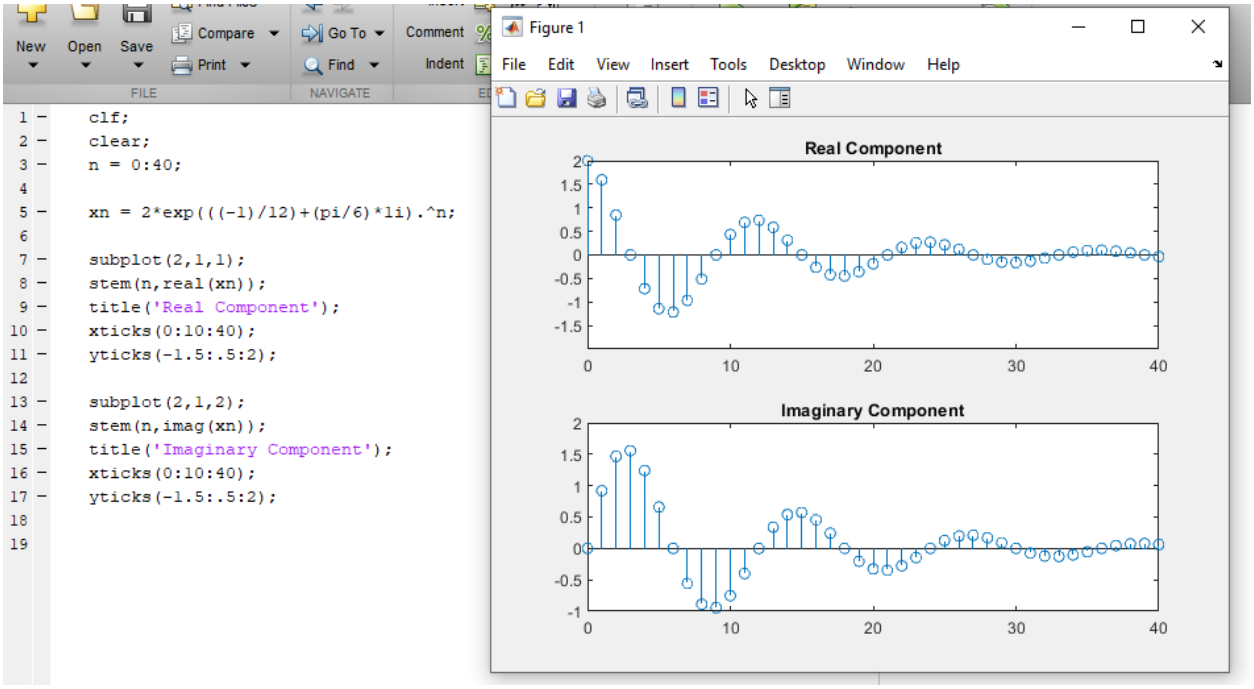
1) 0<c<1, say, c = 0.5
2) c>1, say, c = 2
3) -1<c<0, say, c = - 0.5
4) c<-1, say, c = - 2

You will be able to generate real exponential sequence similar to the figure shown. Write/paste your code and the figure generated.



Complex Valued Exponential Sequence

STEP 5 Write a program to generate a complex valued exponential sequence defined by the equation $x[n] = 2e^{(-\frac{1}{12}+\frac{\pi}{6}j)n}$. Make a subplot similar to the figure below. Write/paste your code and the figure generated.



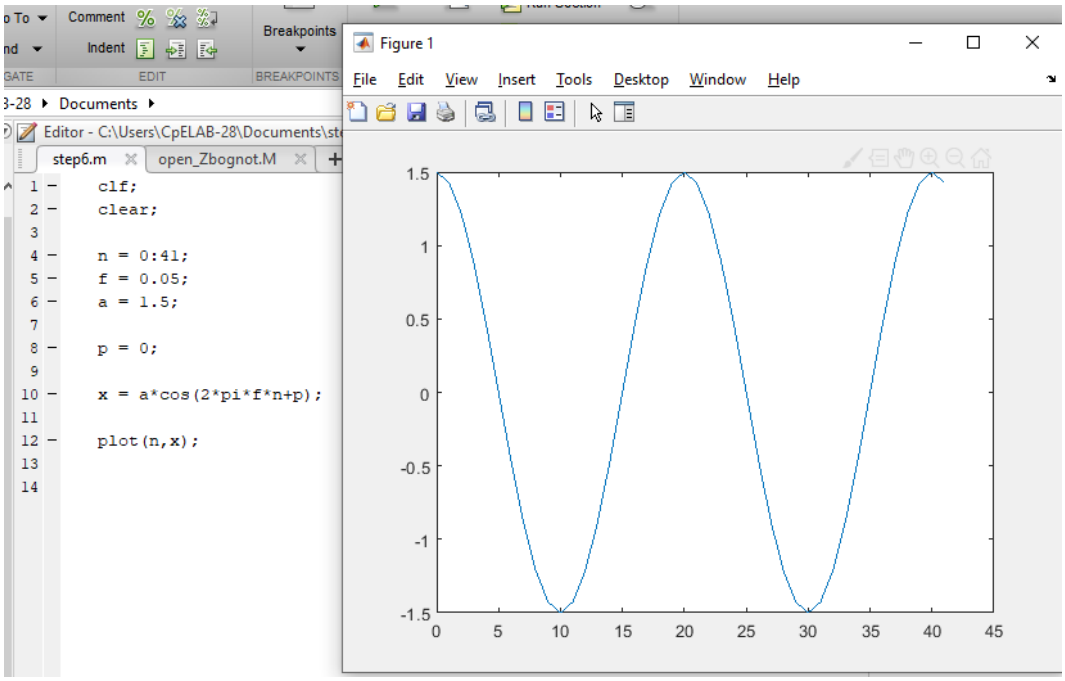
The command stem(n,x) will produce real component of x[n].

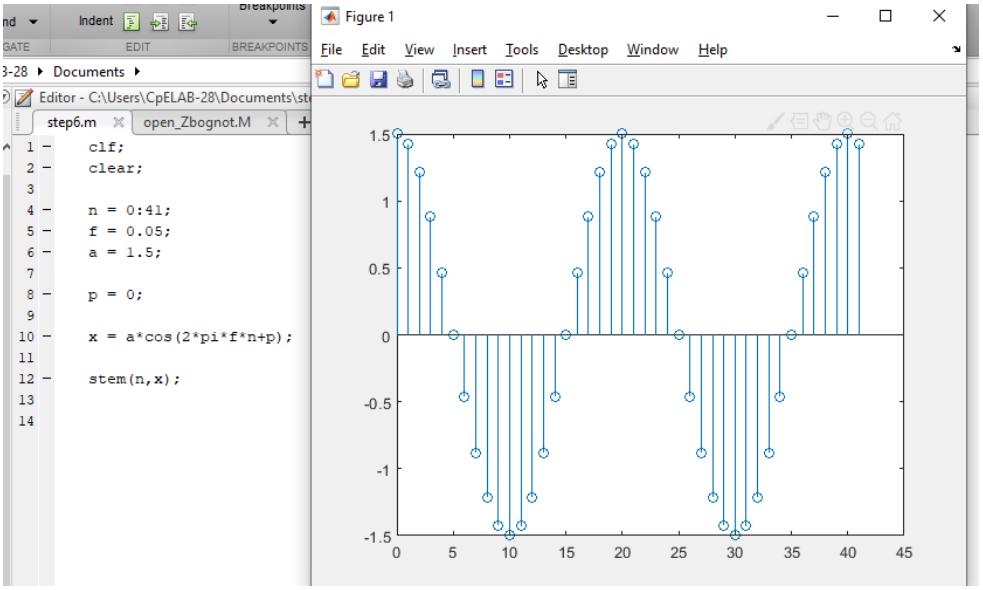
To see the values of x, you may double click the variable x on the workspace window. What are the values of the first and the last element of x? x[0] = 2 , x[40] = - 0.03567

Sinusoidal Sequence

The sinusoidal sequence can be generated using the trigonometric operators cos and sin. This sequence is generally represented as $x[n] = A\cos(\omega n + \phi) = A\cos(2\pi f n + \emptyset)$. The parameters A, ω, f, and ϕ are called, respectively, the amplitude, the angular frequency, the normalized frequency, and the phase of the sinusoidal sequence x[n].

STEP 6 Generate a sinusoidal sequence of length 41 , normalized frequency of 0.05 cycle/sample, amplitude of 1.5, and phase shift of 0. Copy and paste below the figure that you were able to produce. Write/paste your code and the figure generated.





The period of this sequence is 20.

What happens if the phase is changed to $-\pi/2$? The waveform will now start at negative instead of positive

What happens when the **stem** command is replaced with the **plot** command? Do you consider it as a continuous time signal? The stem plot will be changed into continuous line plot. The plot command will connect the points in the stem plot resulting a continuous waveform.

Differentiate **stem** and **plot**
Stem command create a stem plot, you can visualize discrete data or sequences. Each data point is represented by vertical line. Plot command is used to create line plot, it connects the data points resulting a continuous waveform.

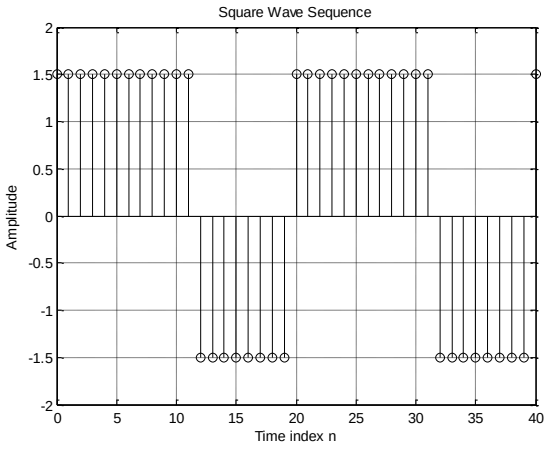
Other Signal Waveforms: Square and Sawtooth Waveforms

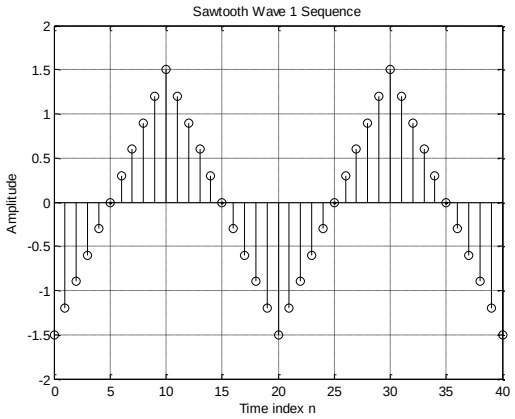
The Matlab/Octave functions **square** and **sawtooth** can be used to generate sequences of the types shown below. Consider again the signal $\mathrm{x[n]} = \mathrm{Acos}(\omega n) = Acos(2\pi fn)$.

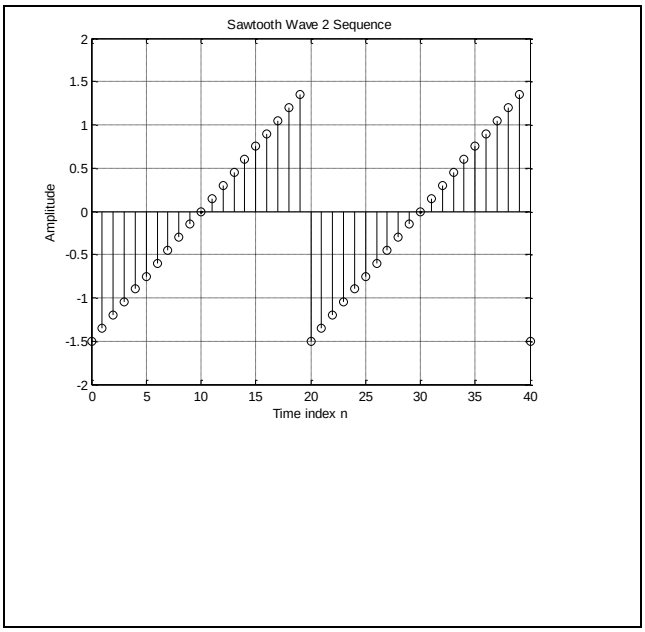
The command **square**(ωn) is like **sin**(ωn) or **cos**(ωn) , only it creates a square wave instead of a sine wave. **square**(ωn,DUTY) generates a square wave with specified duty cycle. The duty cycle, DUTY, is the percent of one cycle in which the signal is positive. A duty cycle of 25% is inputted as 25. The default duty cycle is 50% in case you do not include it within the command.

The command **sawtooth**(ωn) is like **sin**(ωn) or **cos**(ωn) , only it creates a sawtooth wave instead of a sine wave. **sawtooth**(ωn,WIDTH) generates a modified triangle wave where WIDTH, **a scalar parameter between 0 and 1**, determines the percent at which the maximum occurs in one cycle. For example, WIDTH = 0.5 gives you a triangle wave, that is maximum at the middle or 50% of one cycle. The default duty cycle is 100% which is equivalent to WIDTH=1.

STEP 7 Create the necessary Matlab/Octave Codes to generate the square and sawtooth waveforms shown by modifying the code you created in step 6. Those waveforms are similar to those shown below. Write/paste your code and the figure generated.


Square Wave Sequence
Amplitude
Time index n


Sawtooth Wave 1 Sequence
Amplitude
Time index n

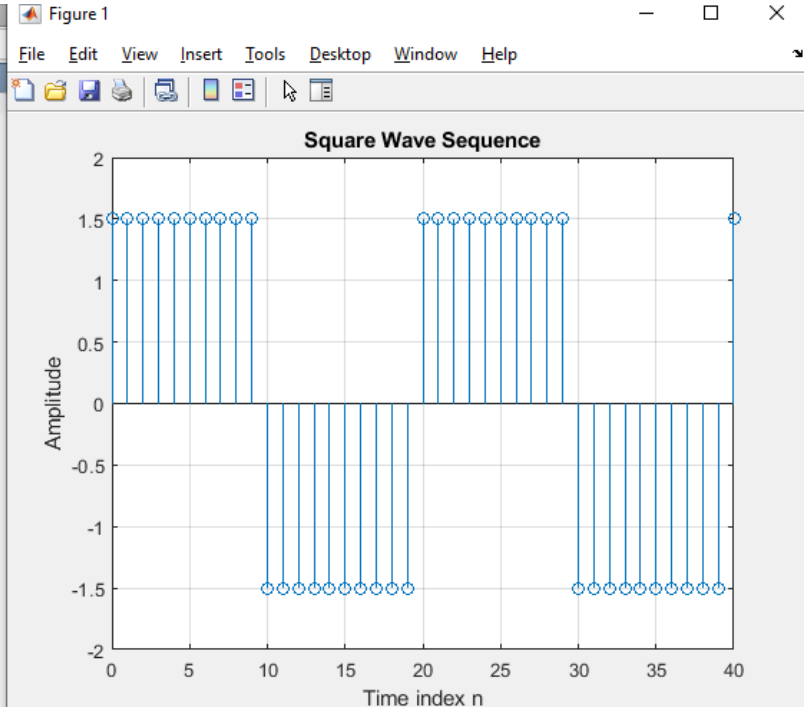



```
clf;
clear;

n = 0:40;
f = 0.05;
a = 1.5;
p = 0;

x = a*square(2*pi*f*n+p);

stem(n,x);
xlabel('Time index n');
ylabel('Amplitude');
title('Square Wave Sequence');
axis([0,40,-2,2]);
grid on;
```

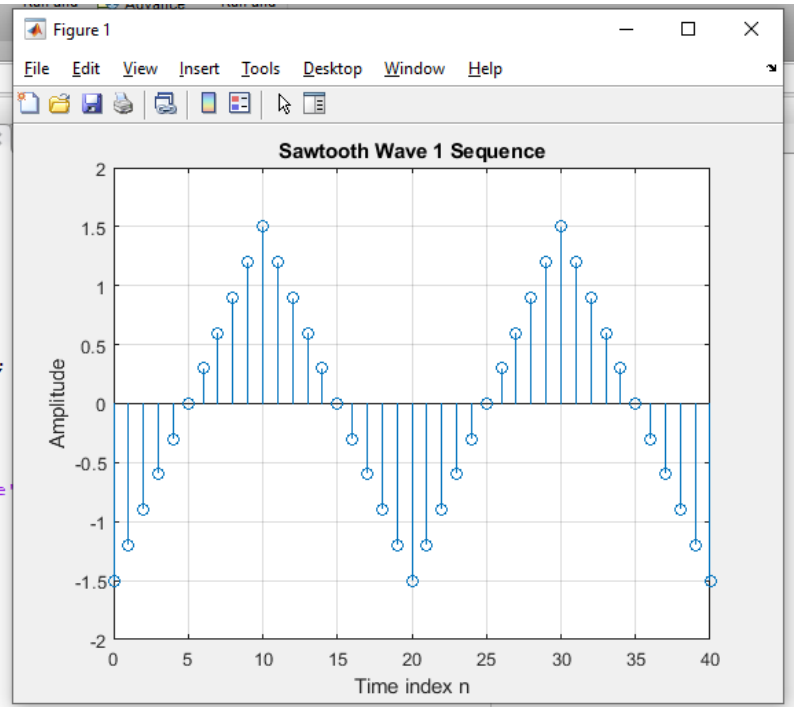


```
clf;
clear;

n = 0:40;
f = 0.05;
a = 1.5;
p = 0;

x = a*sawtooth(2*pi*f*n+p,0.5);

stem(n,x);
xlabel('Time index n');
ylabel('Amplitude');
title('Sawtooth Wave 1 Sequence');
axis([0,40,-2,2]);
grid on;
```




```
clf;
clear;

n = 0:40;
f = 0.05;
a = 1.5;
p = 0;

x = a*sawtooth(2*pi*f*n+p);

stem(n,x);
xlabel('Time index n');
ylabel('Amplitude');
title('Sawtooth Wave 2 Sequence');
axis([0,40,-2,2]);
grid on;
```