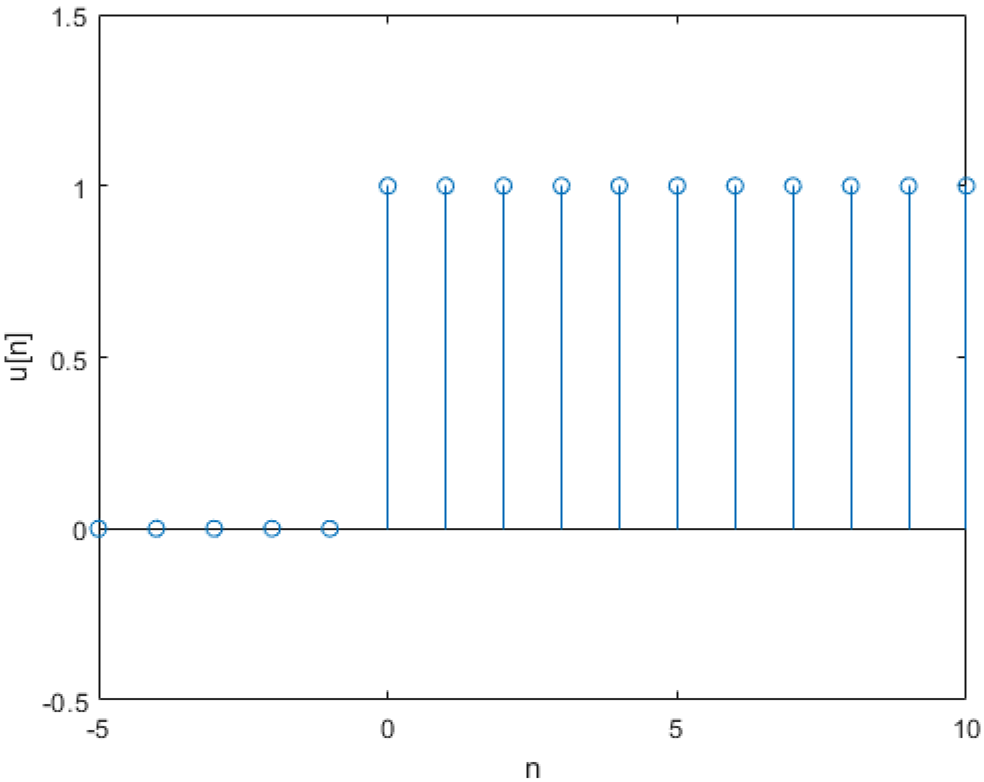
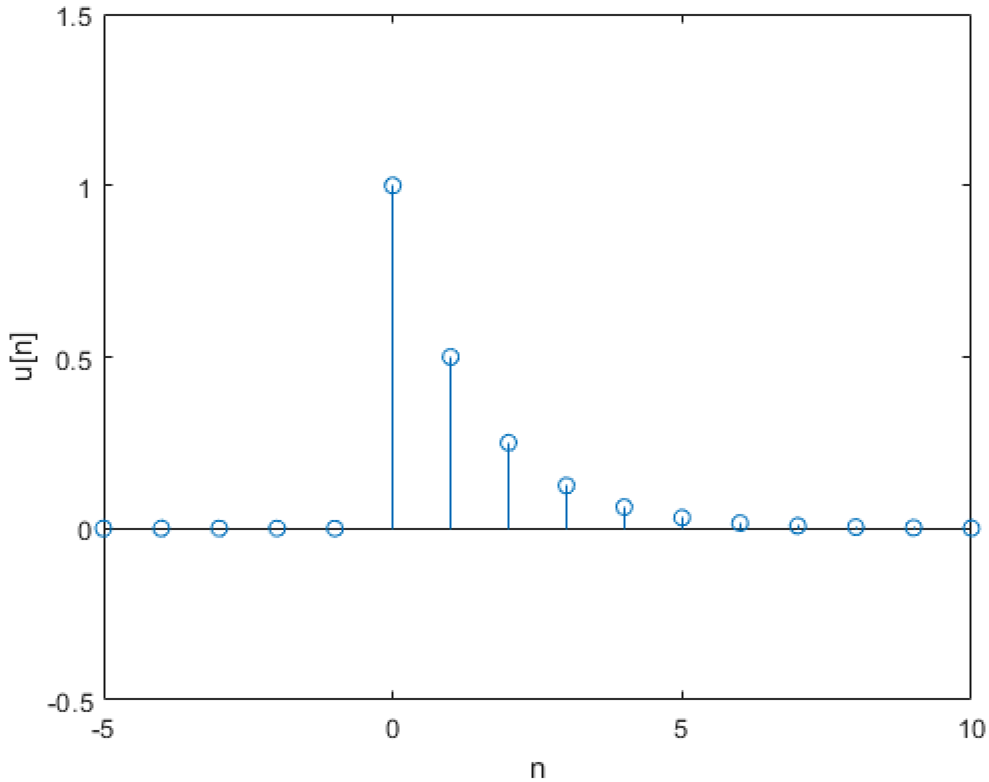
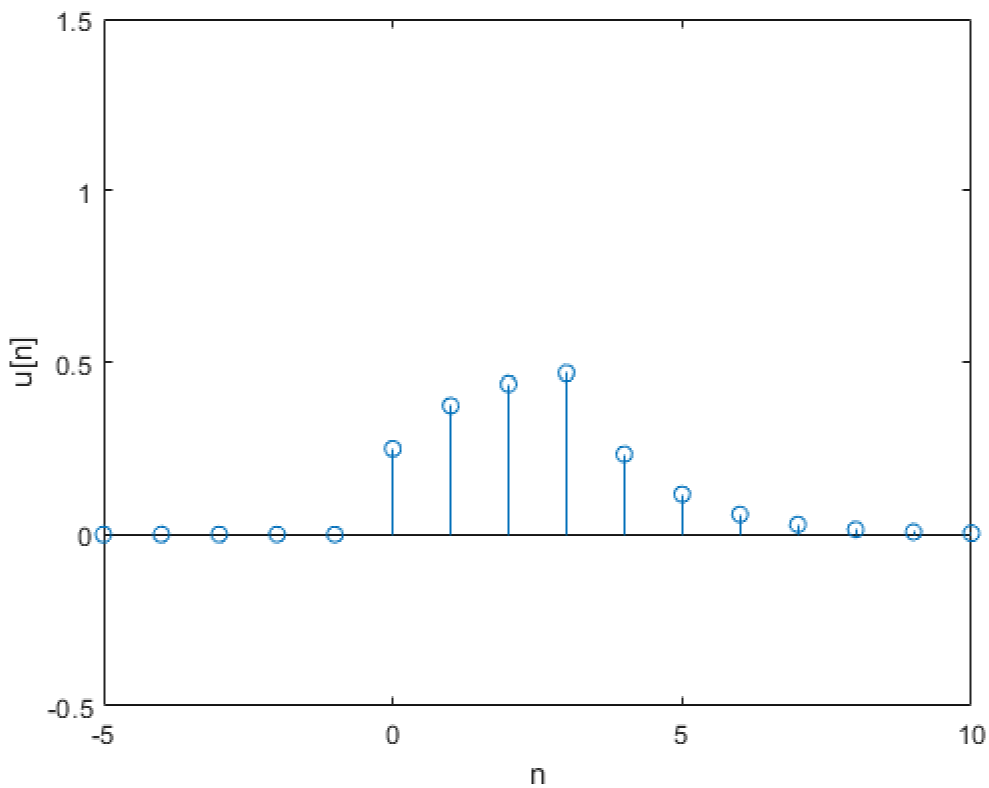
```
% Tune-Up #5

% Copy this file into a Matlab script window, add your code and answers to the
% questions as Matlab comments, hit "Publish", and upload the resulting PDF
 file
% to this page for the tune-up assignment.  Please do not submit a link to a
 file
% but instead upload the file itself.   Late penalty: 2 points per minute
 late.

% The tuneup is to solve homework problem 5.1.

% Intro.  A step function u[n] is a function that turns “on” at the origin and
% stays on.  This can model turning on a switch and leaving it on
 indefinitely.
% Mathematically, u[n] is
%       1 when n >= 0
%       0 otherwise.
% In Matlab, one can implement u[n] as ( (n) >= 0 ).  The logical operator >=
% returns 1 if true and 0 if false.

clc
close all

% Part (a).  Make a plot of u[n] for -5 <= n <= 10.  Describe what you see.

n = -5 : 10;
unitstep = (n >= 0);
figure;
stem(n, unitstep);
xlabel('n');
ylabel('u[n]');
ylim([-0.5 1.5]);

% What I see: A row of impulse signal with a magnitude of 1 starting from n>=
 0;

% Part (b).  We can use the unit-step sequence to represent other sequences
% that are zero for n < 0.  Plot x[n] = (0.5)^n * u[n] for -5 <= n <= 10.
% Describe what you see.

n = -5:10;
unitstep = (n>=0);
x = (0.5 .^ n) .* unitstep;
figure;
stem(n, x);
xlabel('n');
ylabel('u[n]');
ylim([-0.5 1.5]);

% What I see: The magnitude of u[n] decrease by half as n increasing to 10
```

```
% Part (c).  Apply a four-point averaging filter to x[n] and plot the result
% for -5 <= n <= 10.  Describe what you see.

averagingFilterCoeffs = [1/4, 1/4, 1/4, 1/4];
y = filter(averagingFilterCoeffs, 1, x);
figure;
stem(n, y);
xlabel('n');
ylabel('u[n]');
ylim([-0.5 1.5]);

% What I see: The 1st four signal n(0-3) is increasing then after n = 3, the
 signal
% decreasing over time.
```




1.5
1
0.5
0
-0.5
-5
0
5
10
u[n]
n


1.5
1
0.5
0
-0.5
-5
0
5
10
u[n]
n

Published with MATLAB® R2023a