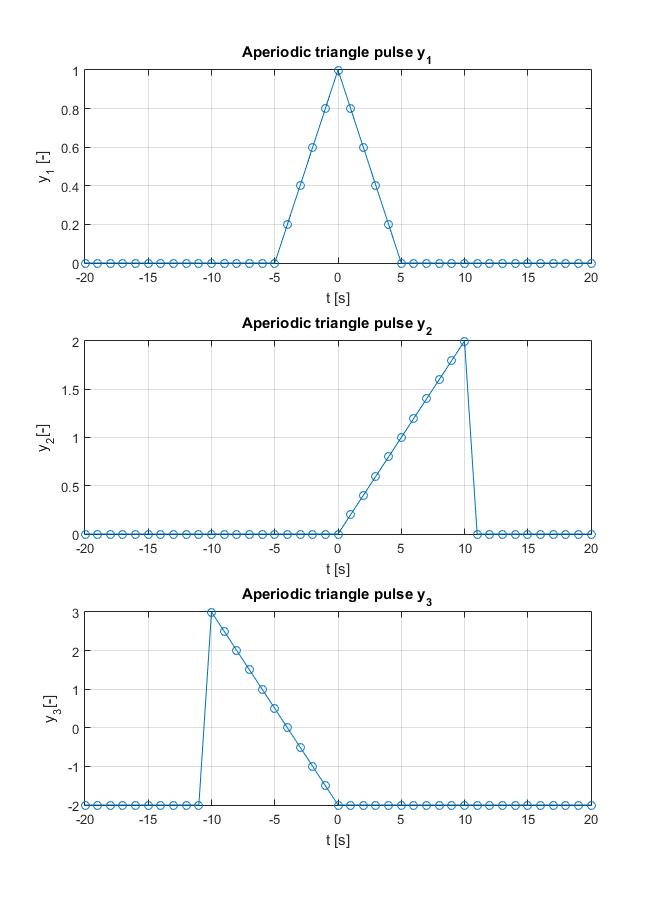
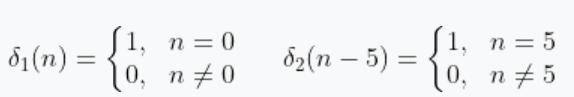
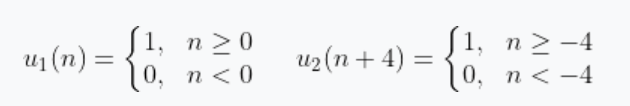
**Generate and properly show aperiodic triangle signals y1, y2 a y3 which are defined for time t=<-20; 20>s, TS=1 s, see picture below. Use function called "tripuls".**

****

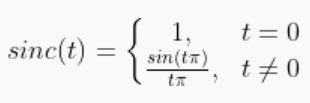
**Generate and properly show unit impulses δ1(n) and δ2(n), which are defined for samples n=<-10; 10> as follows:**

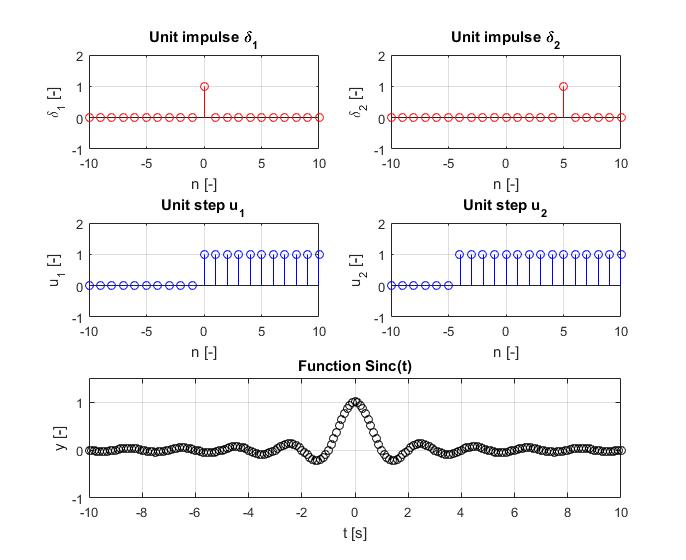


**Generate and properly show unit step u1(n) and u2(n), which are defined for samples n=<-20; 20> as follows:**

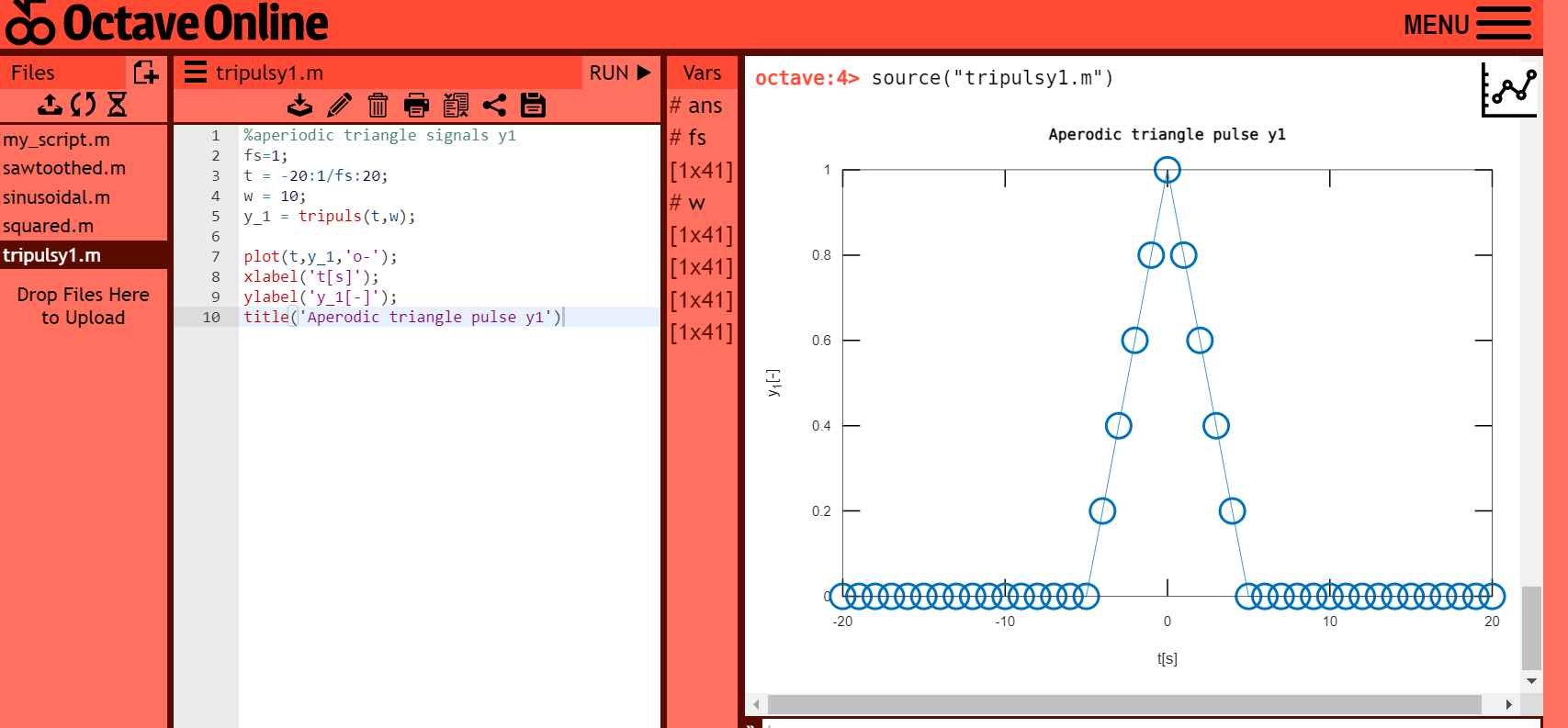


**Generate and properly show function called y(t)=sinc(t) for interval t=<-10; 10> s.**





1. **Aperodic triangle pulse y1**



**Source code for octave online**

%aperiodic triangle signals y1

fs=1;

t = -20:1/fs:20;

w = 10;

y\_1 = tripuls(t,w);

plot(t,y\_1,'o-');

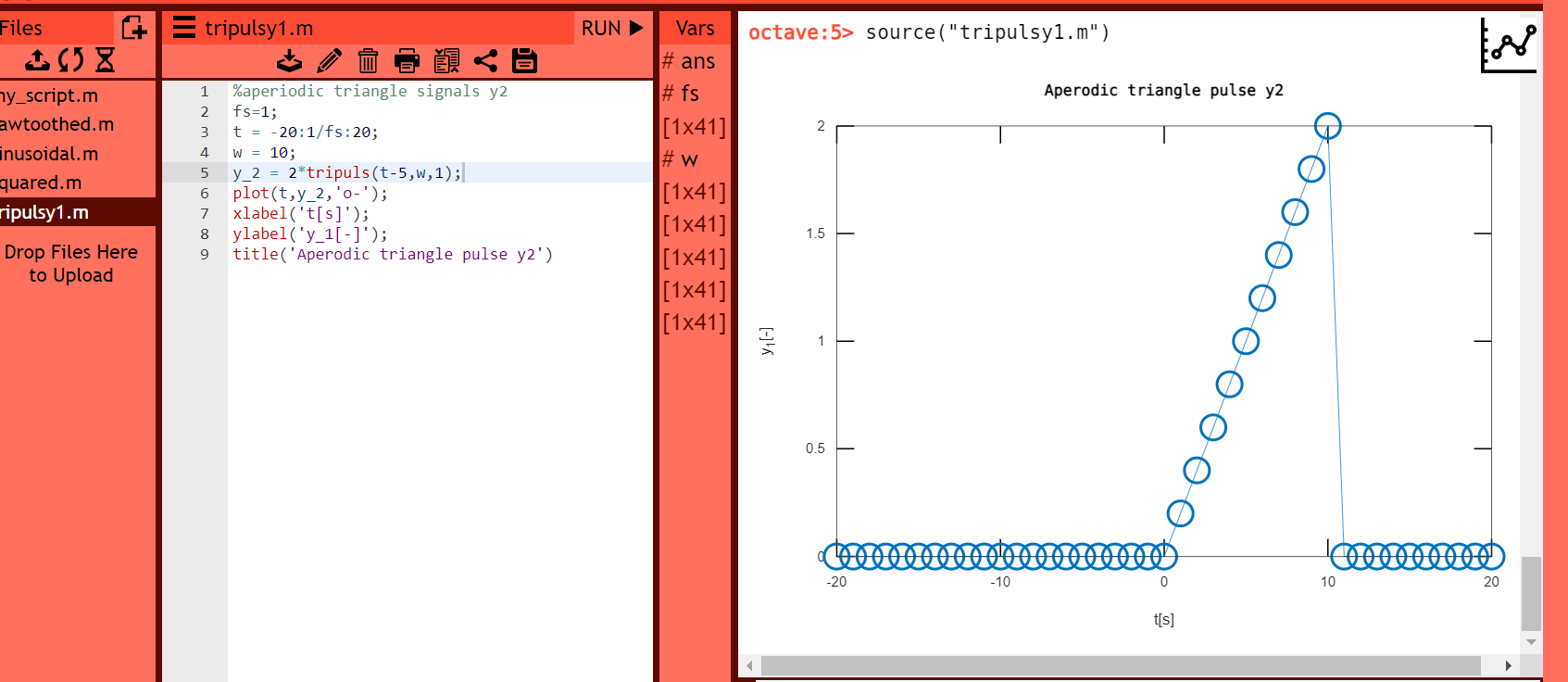
xlabel('t[s]');

ylabel('y\_1[-]');

title('Aperodic triangle pulse y1')

Comment: This code generates an aperiodic triangle signal y\_1 using the tripuls function in Octave. The signal has a frequency of fs=1, a duration of 40 seconds (from -20 to 20), and a width of 10. The plot function is used to visualize the signal.The xlabel, ylabel, and title functions are used to add labels and a title to the plot.

1. **Aperodic triangle pulse y2**



**Source code for octave online**

%aperiodic triangle signals y2

fs=1;

t = -20:1/fs:20;

w = 10;

y\_2 = 2\*tripuls(t-5,w,1);

plot(t,y\_2,'o-');

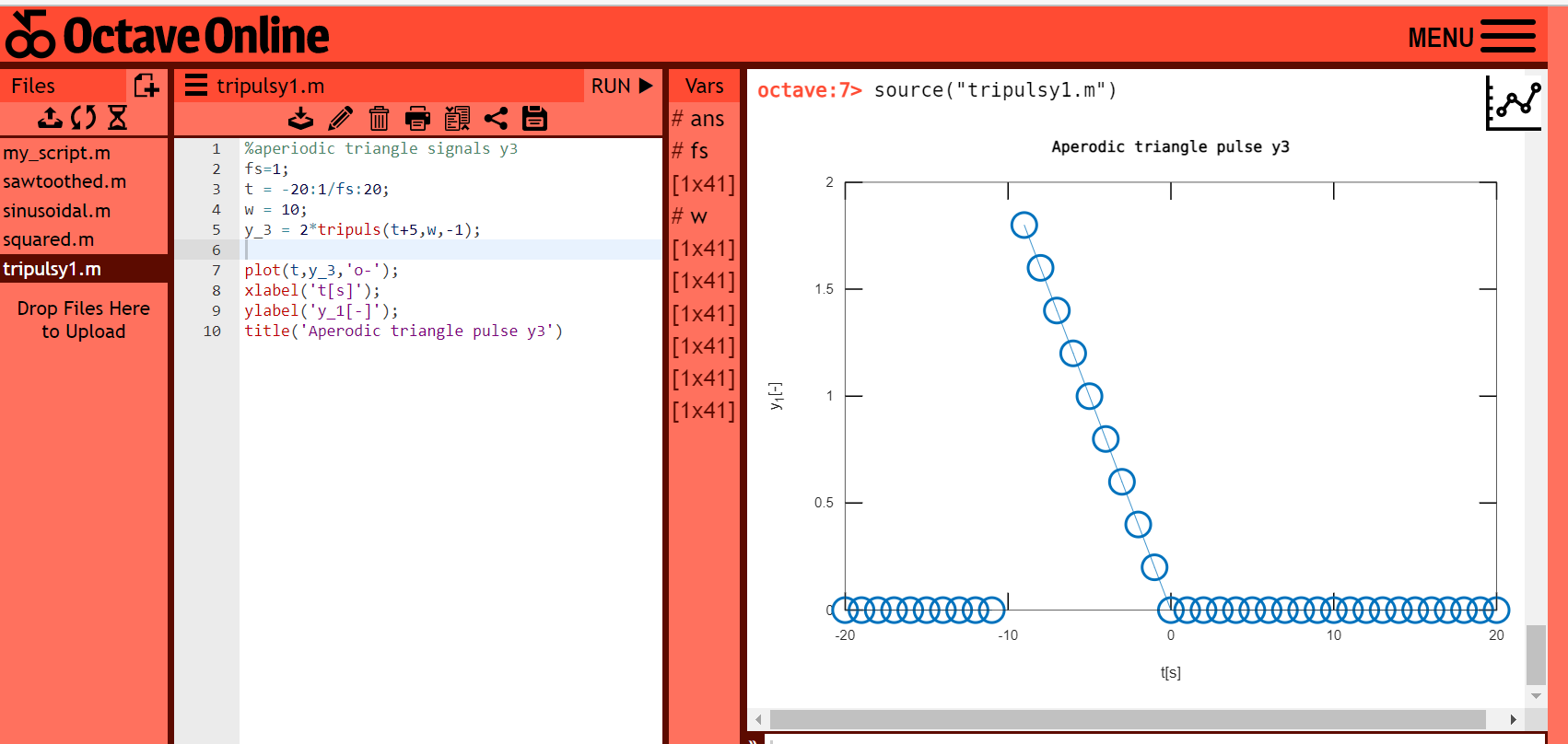
xlabel('t[s]');

ylabel('y\_1[-]');

title('Aperodic triangle pulse y2')

Comment: his code generates another aperiodic triangle signal y\_2 using the tripuls function in Octave. The signal has a frequency of fs=1, a duration of 40 seconds (from -20 to 20), a width of 10, and a height of 2. In addition, the tripuls function is modified to shift the signal to the right by 5 seconds and to set the third input parameter (norm) to 1, which normalizes the signal so that its maximum value is 1. The plot function is used to visualize the signal. The xlabel, ylabel, and title functions are used to add labels and a title to the plot.

1. **Aperodic triangle pulse y3**



**Source code for octave online**

%aperiodic triangle signals y3

fs=1;

t = -20:1/fs:20;

w = 10;

y\_3 = 2\*tripuls(t+5,w,-1);

plot(t,y\_3,'o-');

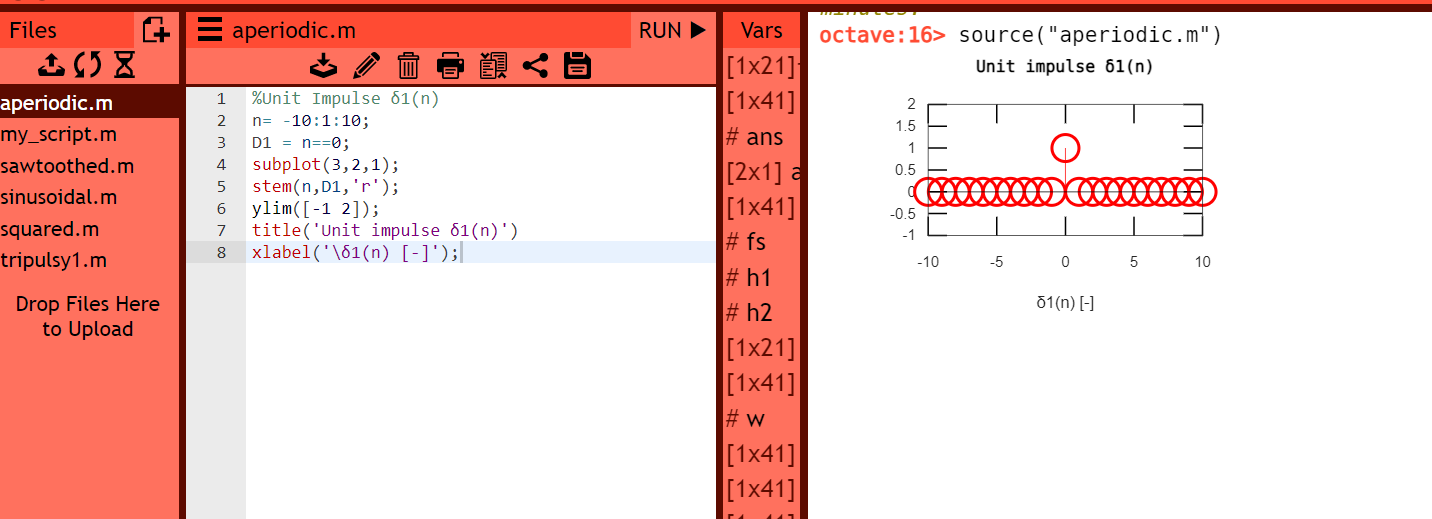
xlabel('t[s]');

ylabel('y\_1[-]');

title('Aperodic triangle pulse y3')

Comment: This code generates yet another aperiodic triangle signal y\_3 using the tripuls function in Octave. The signal has a frequency of fs=1, a duration of 40 seconds (from -20 to 20), a width of 10, and a height of 2. In addition, the tripuls function is modified to shift the signal to the left by 5 seconds and to set the third input parameter (norm) to -1, which flips the signal upside down and normalizes it so that its maximum value is -1. The plot function is used to visualize the signal. The xlabel, ylabel, and title functions are used to add labels and a title to the plot.

1. **Unit Impulse δ1(n)**



**Source code for octave online**

%Unit Impulse δ1(n)

n= -10:1:10;

D1 = n==0;

subplot(3,2,1);

stem(n,D1,'r');

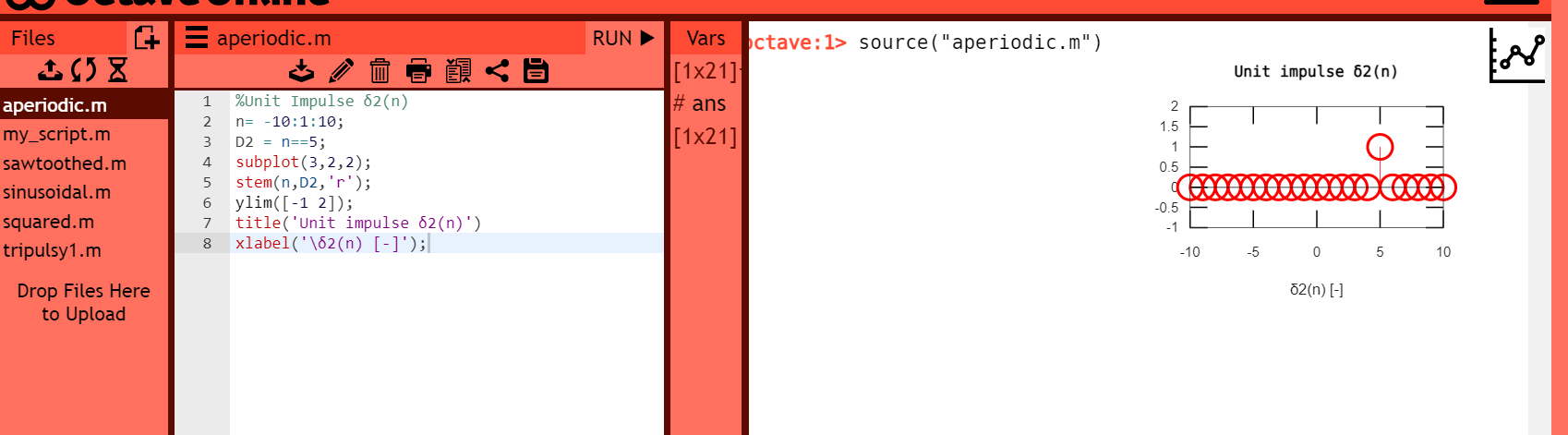
ylim([-1 2]);

title('Unit impulse δ1(n)')

xlabel('\δ1(n) [-]');

Comment: This code generates a unit impulse signal D1 using the comparison operator == in Octave. The signal has a duration of 21 samples (from -10 to 10) and is defined as 1 at n=0 and 0 everywhere else.The subplot function is used to create a subplot with 3 rows, 2 columns, and an index of 1. The stem function is used to visualize the signal as a discrete sequence of impulses. The ylim function is used to set the y-axis limits of the plot to [-1 2], and the title and xlabel functions are used to add a title and a label to the plot, respectively.

1. **Unit Impulse δ2(n)**



**Source code for octave online**

%Unit Impulse δ2(n)

n= -10:1:10;

D2 = n==5;

subplot(3,2,2);

stem(n,D2,'r');

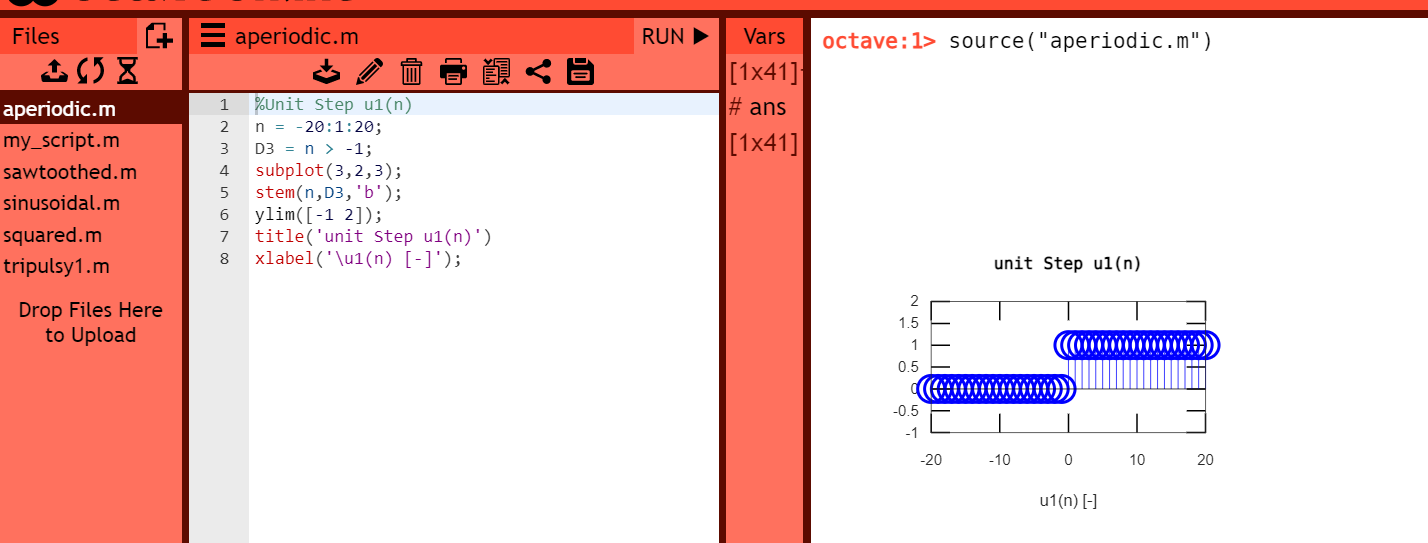
ylim([-1 2]);

title('Unit impulse δ2(n)')

xlabel('\δ2(n) [-]');

Comment: This code generates another unit impulse signal D2 using the comparison operator == in Octave. The signal has a duration of 21 samples (from -10 to 10) and is defined as 1 at n=5 and 0 everywhere else.The subplot function is used to create a subplot with 3 rows, 2 columns, and an index of 2. The stem function is used to visualize the signal as a discrete sequence of impulses. The ylim function is used to set the y-axis limits of the plot to [-1 2], and the title and xlabel functions are used to add a title and a label to the plot, respectively.

1. **Unit step u1(n)**



**Source code for octave online**

%Unit Step u1(n)

n = -20:1:20;

D3 = n > -1;

subplot(3,2,3);

stem(n,D3,'b');

ylim([-1 2]);

title('unit Step u1(n)')

xlabel('\u1(n) [-]');

Comment: This code generates a unit step signal D3 using the comparison operator > in Octave. The signal has a duration of 41 samples (from -20 to 20) and is defined as 1 for **n>=0** and 0 for n<0.The subplot function is used to create a subplot with 3 rows, 2 columns, and an index of 3. The stem function is used to visualize the signal as a discrete sequence of steps. The ylim function is used to set the y-axis limits of the plot to [-1 2], and the title and xlabel functions are used to add a title and a label to the plot, respectively.

1. **Unit Step u2(n)**



**Source code for octave online**

%aperiodic triangle signals y3

%Unit Step u2(n)

n = -20:1:20;

D4 = n > -4;

subplot(3,2,4);

stem(n,D4,'b');

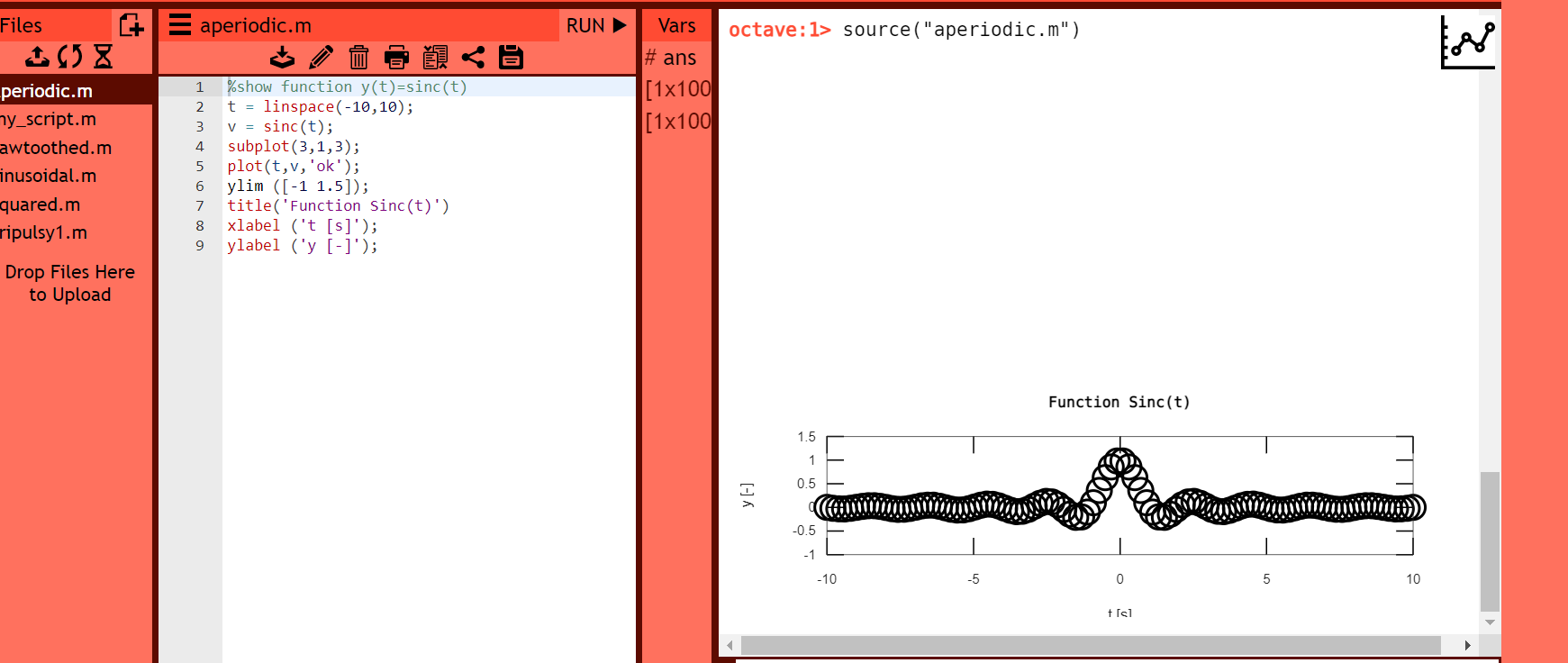
ylim([-1 2]);

title('unit Step u2(n)')

xlabel('\u2(n) [-]');

Comment: This code generates another unit step signal D4 using the comparison operator > in Octave. The signal has a duration of 41 samples (from -20 to 20) and is defined as 1 for **n>=4** and 0 for n<4. The subplot function is used to create a subplot with 3 rows, 2 columns, and an index of 4. The stem function is used to visualize the signal as a discrete sequence of steps. The ylim function is used to set the y-axis limits of the plot to [-1 2], and the title and xlabel functions are used to add a title and a label to the plot, respectively.

1. **Show function called y(t)=sinc(t)**



**Source code for octave online**

%show function y(t)=sinc(t)

t = linspace(-10,10);

v = sinc(t);

subplot(3,1,3);

plot(t,v,'ok');

ylim ([-1 1.5]);

title('Function Sinc(t)')

xlabel ('t [s]');

ylabel ('y [-]');

Comment: This code generates and visualizes the sinc function in Octave. The linspace function is used to generate a vector of 100 equally spaced points between -10 and 10, which represent the time axis t. The sinc function is then applied to this vector to generate the y values of the sinc function.The subplot function is used to create a subplot with 3 rows, 1 column, and an index of 3. The plot function is used to visualize the sinc function as a continuous curve. The ylim function is used to set the y-axis limits of the plot to [-1 1.5], and the title, xlabel, and ylabel functions are used to add a title and labels to the plot, respectively.