**11/09/2020 EXPERIMENT-8**

**AIM**-Writing brief scripts starting each script with a request for input (using input) to evaluate the function h(T) using if-else statement, where

h(T)=(T-10) for 0<T<100

=(0.45T+900) for T>100

Exercise: Testing the scripts written using

1. T=5, h=-5 and B) T=110, h=949.5

**TOOLS USED**- MATLAB 7.0

**THEORY**-When using if, else if, else statements there are few points to keep in mind:

1. An if can have zero or one else’s and it must come after any else ifs.
2. An if can have zero to many else if’s and they must come before the else.
3. Once an else if succeeds, none of the remaining else if’s or else’s will be tested.

**PROGRAMS-**

**FOR T=5**

T=input('enter the value of T for h(t)');

enter the value of T for h(t)**>** 5

if(T>0&&T<100)

display('value of h');

h=T-10;

display(h);

elseif(T>100);

display('value of h');

h=0.45\*T;

h=h+900;

display(h);

elseif(T<0)

T=0;

display('T<0');

display(T);

else

display('ERROR');

end

OUTPUT-

value of h

h = -5

**FOR T=110**

T=input('enter the value of T for h(t)');

enter the value of T for h(t)**>** 110

if(T>0&&T<100)

display('value of h');

h=T-10;

display(h);

elseif(T>100);

display('value of h');

h=0.45\*T;

h=h+900;

display(h);

elseif(T<0)

T=0;

display('T<0');

display(T);

else

display('ERROR');

end

OUTPUT-

value of h

h = 949.50

**FOR T<0**

T=input('enter the value of T for h(t)');

enter the value of T for h(t)**>** -21

if(T>0&&T<100)

display('value of h');

h=T-10;

display(h);

elseif(T>100);

display('value of h');

h=0.45\*T;

h=h+900;

display(h);

elseif(T<0)

T=0;

display('T<0');

display(T);

else

display('ERROR');

end

OUTPUT-

T<0

T = 0

**FOR T=0**

T=input('enter the value of T for h(t)');

enter the value of T for h(t)> 0

if(T>0&&T<100)

display('value of h');

h=T-10;

display(h);

elseif(T>100);

display('value of h');

h=0.45\*T;

h=h+900;

display(h);

elseif(T<0)

T=0;

display('T<0');

display(T);

else

display('ERROR');

end

OUTPUT-

ERROR

**CONCLUSION**- Written brief scripts starting each script with a request for input (using input) to evaluate the function h(T) using if-else statement.

**18/09/2020 EXPERIMENT-9**

**AIM-** Generating a square wave from sum of sine waves of certain amplitude and frequencies.

The Gibbs phenomenon involves both the fact that Fourier sums overshoot at a jump discontinuity, and that this overshoot does not die out as the frequency increases

**TOOLS USED-** MATLAB 7.0

**THEORY-**

**PROCEDURE-**

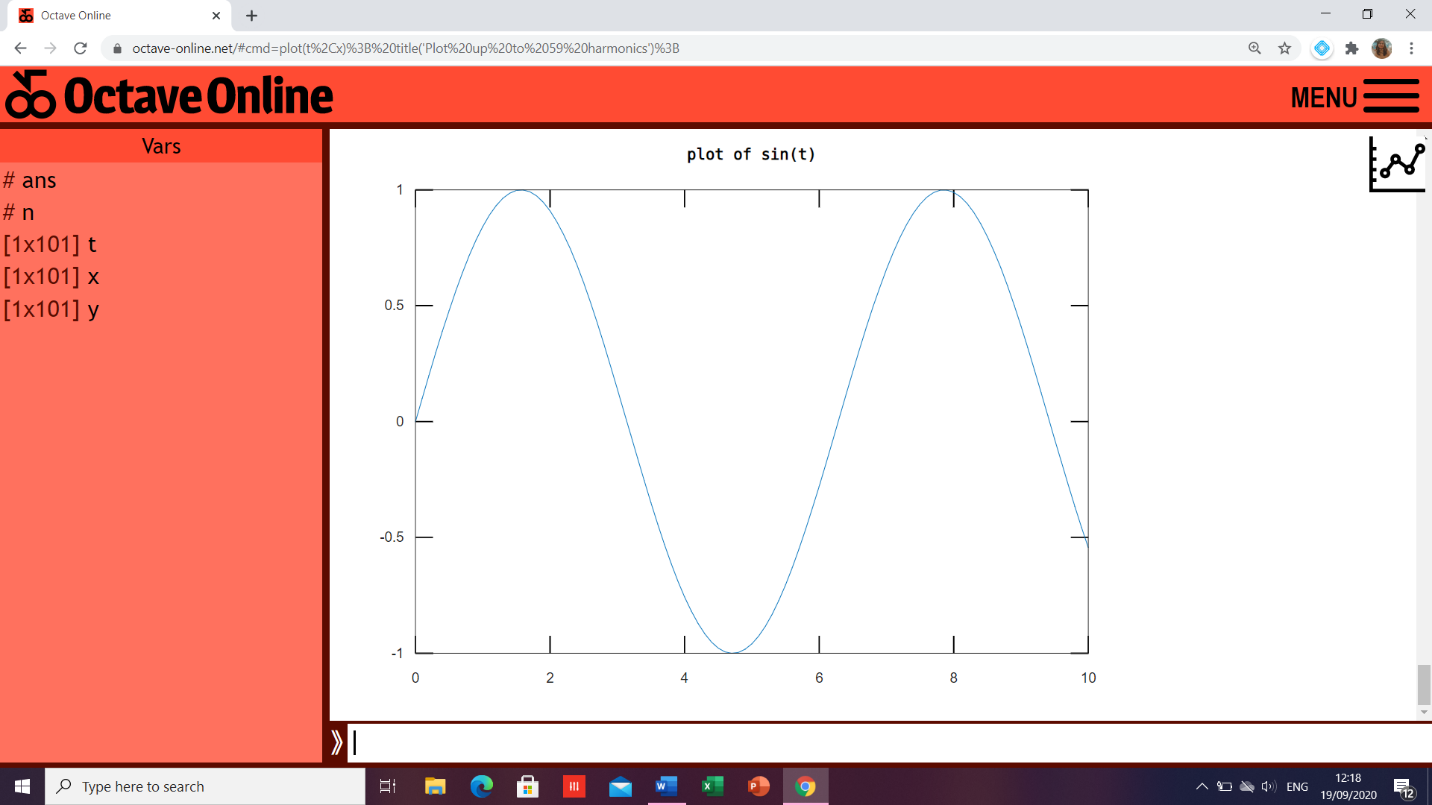
1. The Fourier series expansion for a square-wave is made up of a sum of odd harmonics.
2. We start by forming a time vector running from 0 to 10 in steps of 0.1, and take the sine of all the points.
3. Now add the third harmonic to the fundamental, and plot it.
4. Now use the first, third, fifth, seventh and ninth harmonics.
5. For a finale, we will go from the fundamental to the 19th harmonic, creating % vectors of successively more harmonics and saving all intermediate steps as % the rows of a matrix.
6. These vectors are plotted on the same figure to show evolution of the square wave. Note Gibb’s effect.

**MATLAB PROGRAMS-**

t=0:.1:10;

y=sin(t);

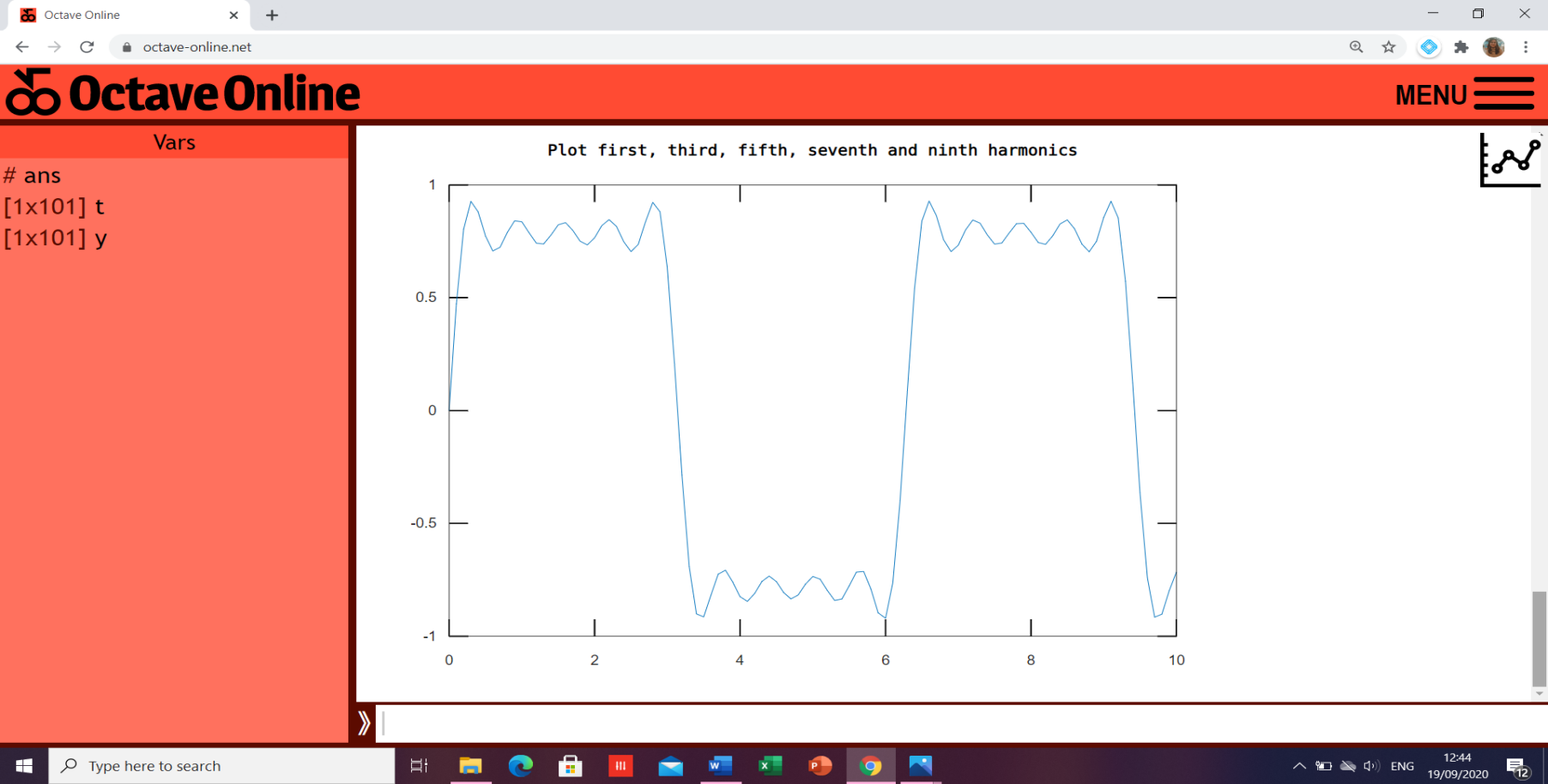
plot(t,y); title('plot of sin(t)');



y=sin(t)+sin(3\*t)/3+sin(5\*t)/5+sin(7\*t)/7+sin(9\*t)/9;

plot(t,y);

title(‘plot first, third, fifth, seventh and ninth harmonics’);



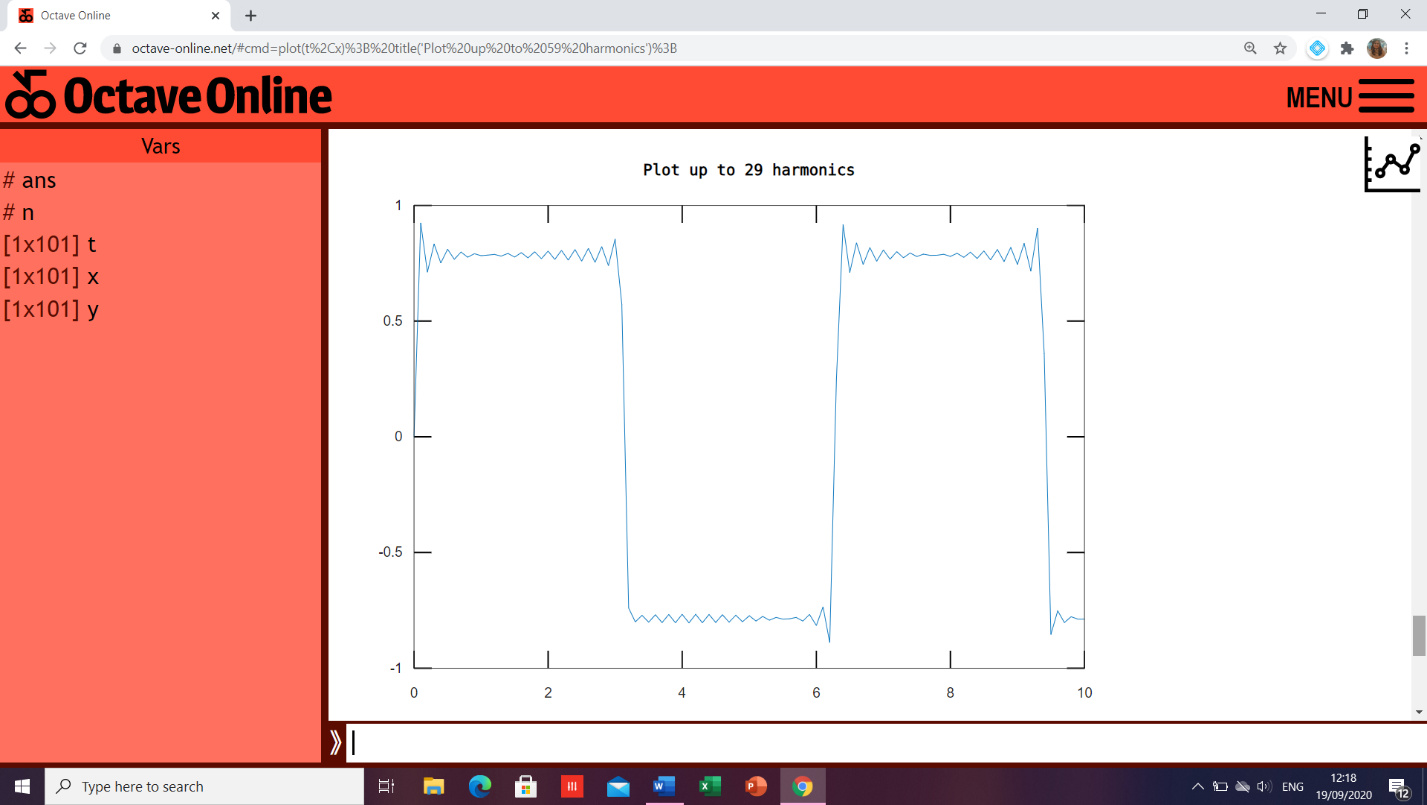
x=zeros(size(t));

for n=1:2:29

> **>** x=x+sin(n\*t)/n;

> **>** end

plot(t,x); title('Plot up to 29 harmonics');



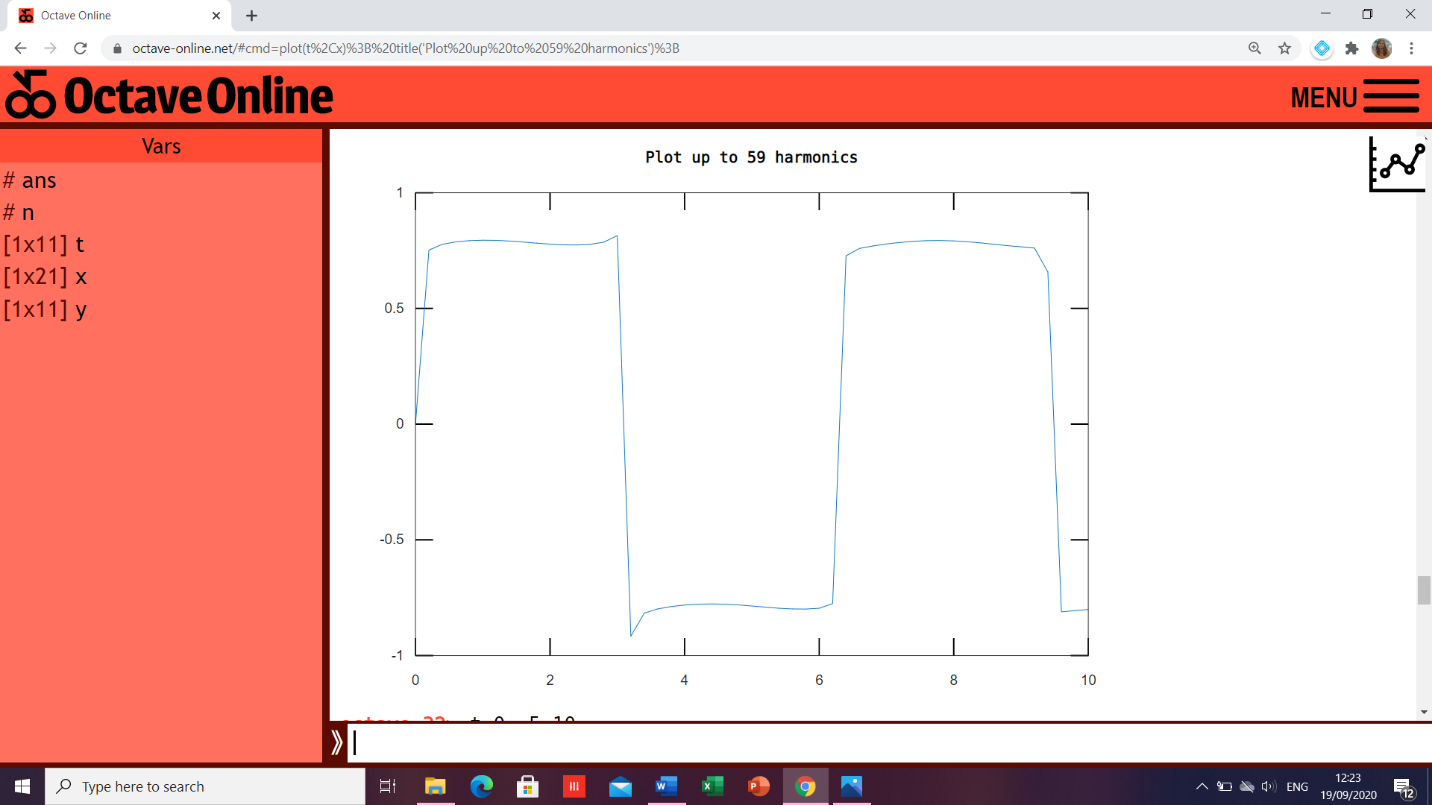
x=zeros(size(t));

for n=1:2:59

> **>** x=x+sin(n\*t)/n;

> **>** end

plot(t,x); title('Plot up to 59 harmonics');



**CONCLUSION**- A square wave from sum of sine waves of certain amplitude and frequencies has been generated