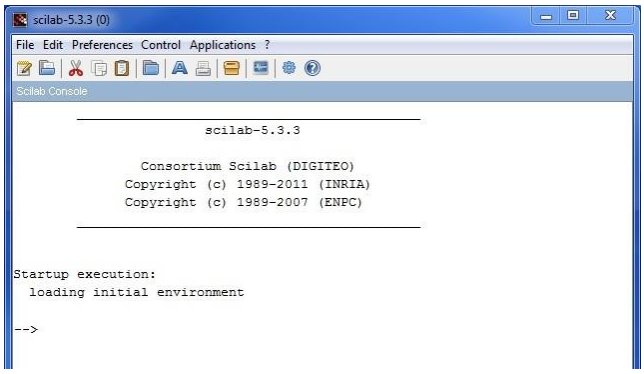
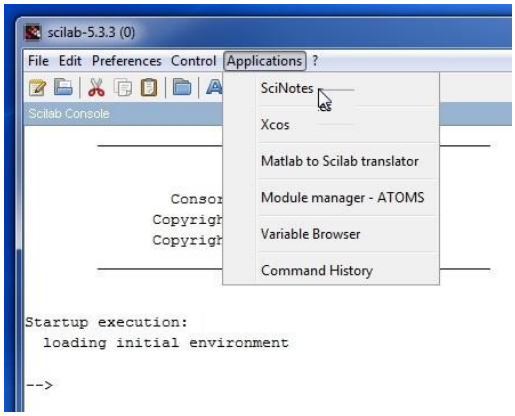
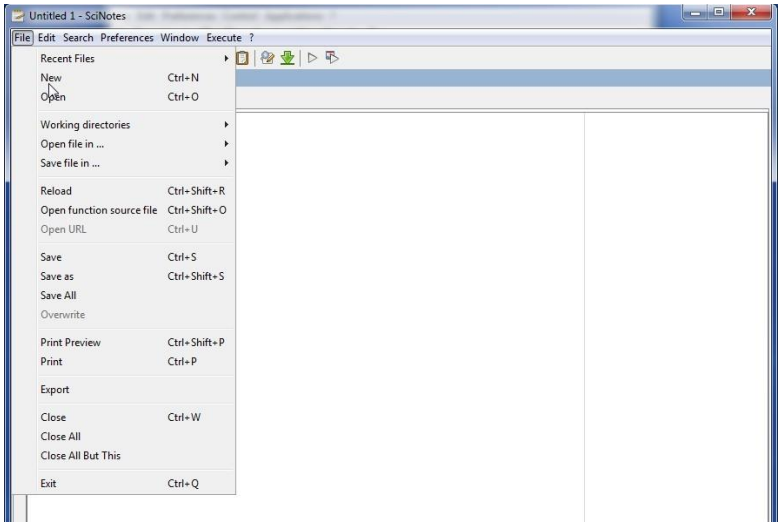
Dynamics of Machines Week 1 – 1st, 2nd and 3rd Exercises





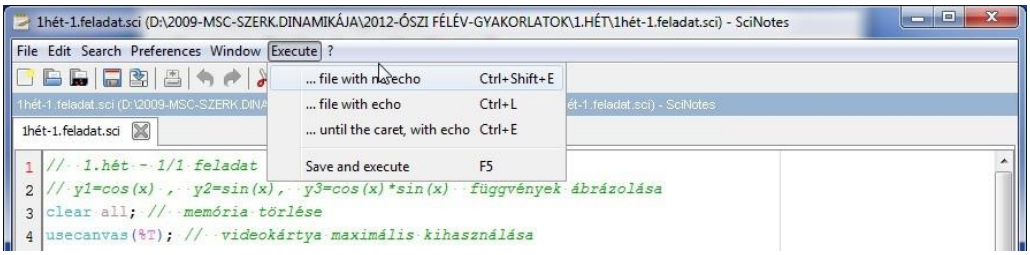


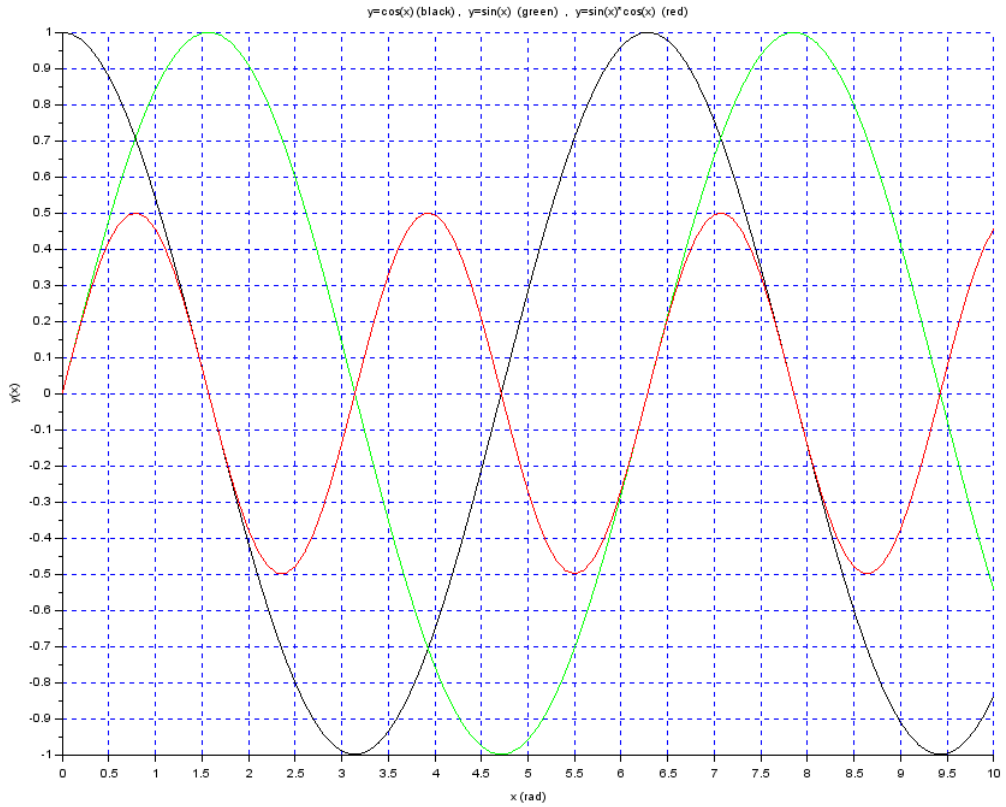
1st exercise

```
// 1st week - 1/1 exercise
// Plot the following functions y1=cos(x) ,  y2=sin(x),  y3=cos(x)*sin(x)
clear;          //  clear the memory
                //usecanvas(%T);
// Function defining
function y1=f1(x)
   y1=cos(x)
endfunction

function y2=f2(x)
   y2=sin(x)
endfunction

function y3=f3(x)
   y3=sin(x).*cos(x)
endfunction

// Discretization
x=0:0.1:10;              // domain of the functions

// Plotting Results
plot2d(x,f1(x),1)
plot2d(x,f2(x),3)
plot2d(x,f3(x),5)
xtitle(" y=cos(x) (black) ,  y=sin(x)  (green)  ,  y=sin(x)*cos(x)  (red)","x (rad)","y(x)") // Title of the
graph, Label of x axis, Label of y axis
xgrid(2)       // plot a grid in the background
```

>>>>>> Save the script.

Execute: Execute >>>> …. file with no echo





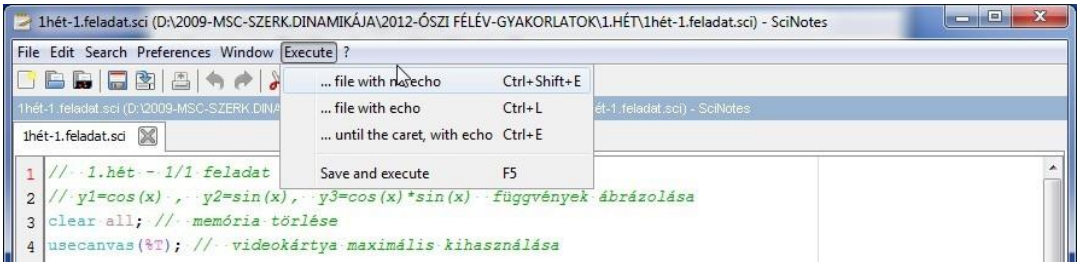
2nd exercise

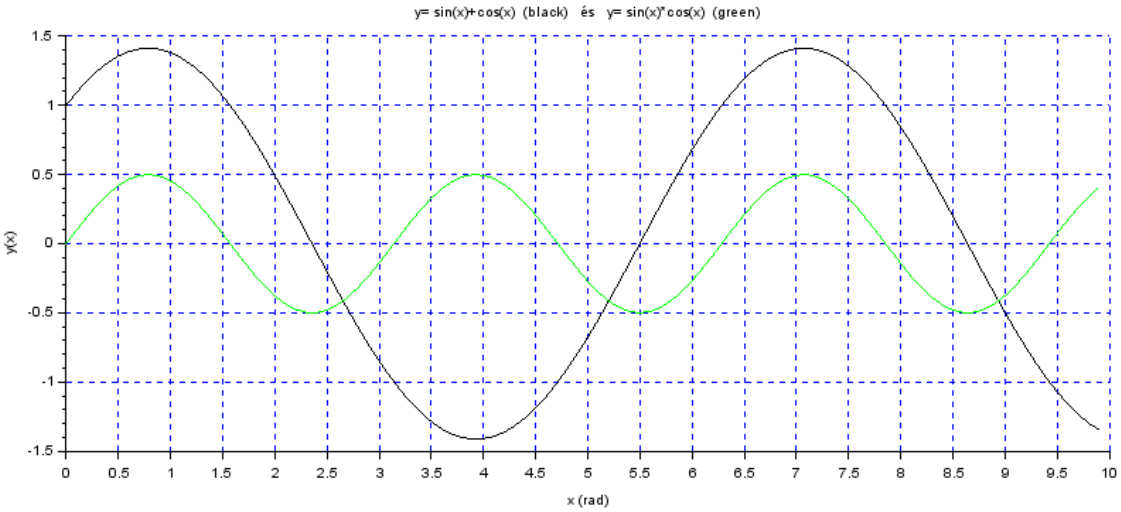
```
// 1st week - 1/2 exercise
// Plot the following functions
// a) function  y=sin(x)+cos(x) (ya=ya(x))
// b) function  y=sin(x)*cos(x) (yb=yb(x))
// c) function  y=cos(x)*e^(-0.05x)  (yc=yc(x))

clear;          // clear data from memory

// Calculate with for loop
xmax=10;    // final value of the calculation [radian]
dx=0.1;     // increment [radian]
n=int(xmax/dx);         // number of steps - integer
x=(1:n);       ya=(1:n);
yb=(1:n);
yc=(1:n);
x0=-dx;     // initial value
for i=1:n
  x(i)=x0+dx
  ya(i)=sin(x(i))+cos(x(i))                  // ya function
  yb(i)=sin(x(i)).*cos(x(i))                 // yb function - * is matrix multiplication!!!
  yc(i)=cos(x(i))*((%e)^(-0.05*x(i)))        // yc function
  x0=x(i)                                    // Variable value exchange
end
```

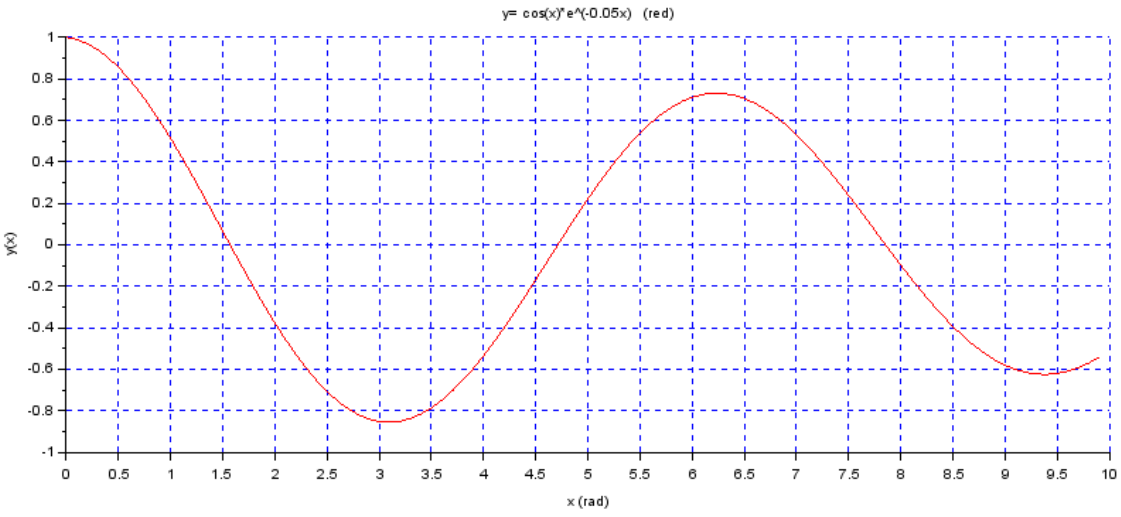
```
// Plotting results ----------------------------------------------------------------------------------------
subplot(2,1,1) //Divide the graphic window into 2x1 matrix of sub-windows with subplot command
plot2d(x,ya,1)
plot2d(x,yb,3)
xtitle(" y= sin(x)+cos(x)  (black)   és   y= sin(x)*cos(x)  (green) "," x (rad)","y(x)") // Title of the graph, Label of x axis, Label of y axis
xgrid(2)       // plot a grid in the background
subplot(2,1,2)
plot2d(x,yc,5)
xtitle(" y= cos(x)*e^(-0.05x)   (red)","x (rad)","y(x)")
xgrid(2)
```

>>>>>> Save the script.

Execute: Execute >>>> …. file with no echo





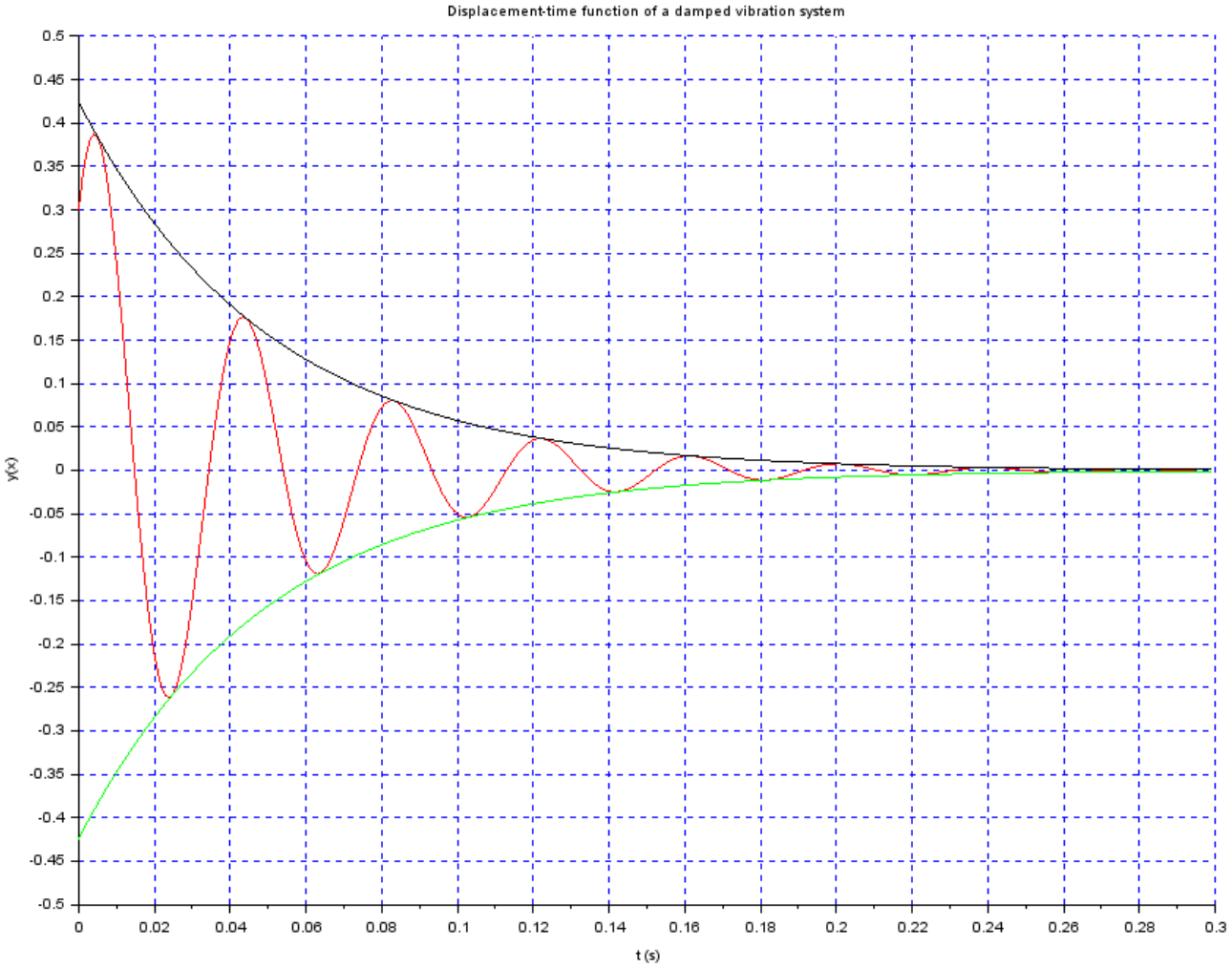


3rd exercise

```
// 1st week - 1/3 exercise

// Plot the displacement-time function of a single degree of freedom vibration system without excitation
// y=(0.3+0.3i)*e^((-20+160i)*t) function
clear;          //  clear memory

// Basic variables
A1=(0.30+(%i)*0.3);  // complex amplitude
A1a=sqrt(0.3^2+0.3^2)  // length of the complex amplitude
kit=(-20.0+(%i)*160.0);          // e^(kit)  exponent
tmax=0.3;                // final time of the calculation – [s]
dt=0.001;                // increment – [s]
n=int(tmax/dt);          // number of steps - integer
t0=-dt;                  // initial value

// for loop
for i=1:n
        t(i)=t0+dt;
        y(i)=A1*(%e)^(kit*(t(i)));          // Complex function
        yupenv(i)=(A1a)*(%e)^(-20*(t(i)));          // upper envelope
        ylowenv(i)=(-A1a)*(%e)^(-20*(t(i)));       // lower envelope
        t0=t(i);                                   // variable value exchange
        yim(i)=imag(y(i));       // imaginary part of the y complex function
end
// Plot the functions
plot2d(t,yim,5)
plot2d(t,yupenv,1)
plot2d(t,ylowenv,3)
xtitle("Displacement-time function of a damped vibration system","t (s)","y(x)")    // Title of the graph,
Label of x axis, Label of y axis
xgrid(2)   // plot a grid in the background
```

>>>>>> Save the script.

Execute: Execute >>>> …. file with no echo


Displacement-time function of a damped vibration system
y(x)
t (s)
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
-0.25
-0.3
-0.35
-0.4
-0.45
-0.5
0
0.02
0.04
0.06
0.08
0.1
0.12
0.14
0.16
0.18
0.2
0.22
0.24
0.26
0.28
0.3