**Portfolio**

**for Physics and Computer Modeling**



Made by the student of AUCA

Toktomambetov Erlan

**Program #1: “House”**

The program draws a house that consists of blocks and have X number of floors and Y number of rows. Code consists of commands that draws combination of squares using lines and the result would be a block. Also by using smaller lines I drew a window. Using cycles (for) I drew 3 rows of blocks drawn by 5. The final result is a 5-floor house.

**Code:**

%basic parameters

figure('units', 'norm', 'position', [0.1 0.1 0.8 0.7], 'color', [1 1 1])

axes('position', [0.1 0.1 0.6 0.6], 'color', [0.5 1 1], 'xlim', [0 60], 'ylim', [-2 30], 'zlim', [0 50])

view(25,15);

grid on;

%starting coordinates

x0=0;

y0=0;

z0=0;

L=10;

%start of a cycle to draw a block

for x0=0:L:20

for z0=0:L:50

%--------------------------------------------------

%drawing a window

x=[x0+2, x0+2];

y=[y0, y0];

z=[z0+2, z0+L-2];

line(x,y,z, 'linewidth', 3);

x=[x0+2, x0+L-2];

y=[y0, y0];

z=[z0+2, z0+2];

line(x,y,z, 'linewidth', 3);

x=[x0+L-2, x0+L-2];

y=[y0, y0];

z=[z0+2, z0+L-2];

line(x,y,z, 'linewidth', 3);

x=[x0+2, x0+L-2];

y=[y0, y0];

z=[z0+L-2, z0+L-2];

line(x,y,z, 'linewidth', 3);

%--------------------------------------------------

%drawing square

x=[x0, x0+L];

y=[y0, y0];

z=[z0, z0];

line(x,y,z, 'linewidth', 3);

x=[x0, x0];

y=[y0, y0+L];

z=[z0, z0];

line(x,y,z, 'linewidth', 3);

x=[x0+L, x0+L];

y=[y0, y0+L];

z=[z0, z0];

line(x,y,z, 'linewidth', 3);

x=[x0, x0+L];

y=[y0+L, y0+L];

z=[z0, z0];

line(x,y,z, 'linewidth', 3);

%--------------------------------------------------

%drawing square

x=[x0, x0+L];

y=[y0, y0];

z=[z0+L, z0+L];

line(x,y,z, 'linewidth', 3);

x=[x0, x0];

y=[y0, y0+L];

z=[z0+L, z0+L];

line(x,y,z, 'linewidth', 3);

x=[x0+L, x0+L];

y=[y0, y0+L];

z=[z0+L, z0+L];

line(x,y,z, 'linewidth', 3);

x=[x0, x0+L];

y=[y0+L, y0+L];

z=[z0+L, z0+L];

line(x,y,z, 'linewidth', 3);

%--------------------------------------------------

%drawing square

x=[x0, x0];

y=[y0, y0];

z=[z0, z0+L];

line(x,y,z, 'linewidth', 3);

x=[x0, x0];

y=[y0+L, y0+L];

z=[z0, z0+L];

line(x,y,z, 'linewidth', 3);

x=[x0+L, x0+L];

y=[y0+L, y0+L];

z=[z0, z0+L];

line(x,y,z, 'linewidth', 3);

x=[x0+L, x0+L];

y=[y0, y0];

z=[z0, z0+L];

line(x,y,z, 'linewidth', 3);

%--------------------------------------------------

%==================================================

%block is ready

%1st row of blocks is drawn

drawnow;

%==================================================

end

end

%setting new coordinates

x0=0;

y0=10;

z0=0;

L=10;

%new cycle for a 2nd row of blocks

for x0=0:L:20

for z0=0:L:50

%--------------------------------------------------

x=[x0, x0+L];

y=[y0, y0];

z=[z0, z0];

line(x,y,z, 'linewidth', 3);

x=[x0, x0];

y=[y0, y0+L];

z=[z0, z0];

line(x,y,z, 'linewidth', 3);

x=[x0+L, x0+L];

y=[y0, y0+L];

z=[z0, z0];

line(x,y,z, 'linewidth', 3);

x=[x0, x0+L];

y=[y0+L, y0+L];

z=[z0, z0];

line(x,y,z, 'linewidth', 3);

%--------------------------------------------------

x=[x0, x0+L];

y=[y0, y0];

z=[z0+L, z0+L];

line(x,y,z, 'linewidth', 3);

x=[x0, x0];

y=[y0, y0+L];

z=[z0+L, z0+L];

line(x,y,z, 'linewidth', 3);

x=[x0+L, x0+L];

y=[y0, y0+L];

z=[z0+L, z0+L];

line(x,y,z, 'linewidth', 3);

x=[x0, x0+L];

y=[y0+L, y0+L];

z=[z0+L, z0+L];

line(x,y,z, 'linewidth', 3);

%--------------------------------------------------

x=[x0, x0];

y=[y0, y0];

z=[z0, z0+L];

line(x,y,z, 'linewidth', 3);

x=[x0, x0];

y=[y0+L, y0+L];

z=[z0, z0+L];

line(x,y,z, 'linewidth', 3);

x=[x0+L, x0+L];

y=[y0+L, y0+L];

z=[z0, z0+L];

line(x,y,z, 'linewidth', 3);

x=[x0+L, x0+L];

y=[y0, y0];

z=[z0, z0+L];

line(x,y,z, 'linewidth', 3);

%--------------------------------------------------

%==================================================

drawnow;

%==================================================

end

end

%setting new coordinates

x0=0;

y0=20;

z0=0;

L=10;

%final cycle for the 3rd row of blocks

for x0=0:L:20

for z0=0:L:50

%--------------------------------------------------

x=[x0, x0+L];

y=[y0, y0];

z=[z0, z0];

line(x,y,z, 'linewidth', 3);

x=[x0, x0];

y=[y0, y0+L];

z=[z0, z0];

line(x,y,z, 'linewidth', 3);

x=[x0+L, x0+L];

y=[y0, y0+L];

z=[z0, z0];

line(x,y,z, 'linewidth', 3);

x=[x0, x0+L];

y=[y0+L, y0+L];

z=[z0, z0];

line(x,y,z, 'linewidth', 3);

%--------------------------------------------------

x=[x0, x0+L];

y=[y0, y0];

z=[z0+L, z0+L];

line(x,y,z, 'linewidth', 3);

x=[x0, x0];

y=[y0, y0+L];

z=[z0+L, z0+L];

line(x,y,z, 'linewidth', 3);

x=[x0+L, x0+L];

y=[y0, y0+L];

z=[z0+L, z0+L];

line(x,y,z, 'linewidth', 3);

x=[x0, x0+L];

y=[y0+L, y0+L];

z=[z0+L, z0+L];

line(x,y,z, 'linewidth', 3);

%--------------------------------------------------

x=[x0, x0];

y=[y0, y0];

z=[z0, z0+L];

line(x,y,z, 'linewidth', 3);

x=[x0, x0];

y=[y0+L, y0+L];

z=[z0, z0+L];

line(x,y,z, 'linewidth', 3);

x=[x0+L, x0+L];

y=[y0+L, y0+L];

z=[z0, z0+L];

line(x,y,z, 'linewidth', 3);

x=[x0+L, x0+L];

y=[y0, y0];

z=[z0, z0+L];

line(x,y,z, 'linewidth', 3);

%--------------------------------------------------

%==================================================

drawnow;

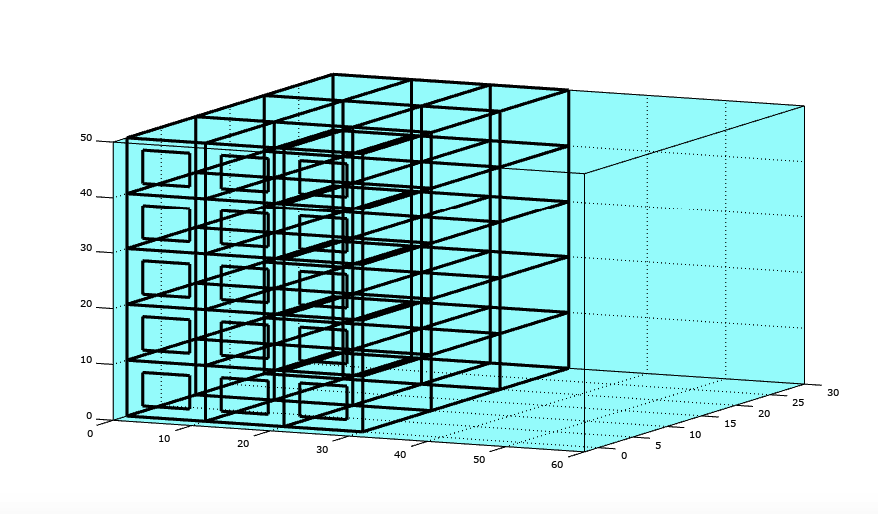
%==================================================

end

end

%the house is ready

%house consists of 5 floors and 3 rows of blocks

**Result:**

**Program #2: “Solar System”**

The program creates a 3-dimensional model of Solar System. The Sun is in center lane and moves up with the rest of the planets. There are 8 different planets that have different colors, sizes and velocities. Planets also leave trails after them and after reaching the top of the field, the program stops.

**Code:**

%basic parameters

figure('units', 'norm', 'position', [0.1 0.1 0.8 0.7]);

%turning system in 3-dimentional

axes('position', [0.1 0.1 0.8 0.7], 'xlim', [-8 8], 'ylim', [-8 8], 'zlim', [0 20], 'dataaspectratio', [1 1 1], 'color', [0.2 0.2 0.3]);

grid on;

view(30,30);

%starting coordinates

x=0; y=0; t=5;

%setting planets and its parameters

h\_Mercury=line(x,y,t, 'marker', 'O', 'markersize', 10, 'markerfacecolor', [1 0.7 0]);

h\_Venus=line(x,y,t, 'marker', 'O', 'markersize', 15, 'markerfacecolor', [1 1 0.8]);

h\_Mars=line(x,y,t, 'marker', 'O', 'markersize', 10, 'markerfacecolor', [1 0 0]);

h\_Jupiter=line(x,y,t, 'marker', 'O', 'markersize', 25, 'markerfacecolor', [1 0.3 0]);

h\_Saturn=line(x,y,t, 'marker', 'O', 'markersize', 20, 'markerfacecolor', [1 0.3 1]);

h\_Uranus=line(x,y,t, 'marker', 'O', 'markersize', 15, 'markerfacecolor', [0.5 0.5 0.5]);

h\_Neptune=line(x,y,t, 'marker', 'O', 'markersize', 20, 'markerfacecolor', [0 0.5 1]);

h\_Earth=line(x,y,t, 'marker', 'O', 'markersize', 15, 'markerfacecolor', [0 0 1]);

h\_Sun=line(x,y,t, 'marker', 'O', 'markersize', 30, 'markerfacecolor', [1 1 0]);

%setting coordinates of planets

X=0;

Y=0;

T=5;

X\_Sun=0;

Y\_Sun=0;

X\_Mercury=0;

Y\_Mercury=0;

X\_Venus=0;

Y\_Venus=0;

X\_Mars=0;

Y\_Mars=0;

X\_Jupiter=0;

Y\_Jupiter=0;

X\_Uranus=0;

Y\_Uranus=0;

X\_Saturn=0;

Y\_Saturn=0;

X\_Neptune=0;

Y\_Neptune=0;

k=1;

%start of a cycle to draw planetary movement

for fi=0:pi/30:5\*pi

x=2.25\*cos(2\*fi);

y=2.25\*sin(2\*fi);

t=5+fi;

x\_Mercury=1.2\*cos(6\*fi);

y\_Mercury=1.2\*sin(6\*fi);

x\_Venus=1.7\*cos(3\*fi);

y\_Venus=1.7\*sin(3\*fi);

x\_Mars=2.75\*cos(1.5\*fi);

y\_Mars=2.75\*sin(1.5\*fi);

x\_Jupiter=3.5\*cos(fi);

y\_Jupiter=3.5\*sin(fi);

x\_Uranus=4.5\*cos(0.8\*fi);

y\_Uranus=4.5\*sin(0.8\*fi);

x\_Saturn=5.5\*cos(0.5\*fi);

y\_Saturn=5.5\*sin(0.5\*fi);

x\_Neptune=6.5\*cos(0.3\*fi);

y\_Neptune=6.5\*sin(0.3\*fi);

%The Sun is in center of the system

x\_Sun=0;

y\_Sun=0;

%setting starting point higher so all planets are visible

T(k)=5+fi;

%setting coordinates

set(h\_Sun, 'xdata', x\_Sun, 'ydata', y\_Sun, 'zdata', t)

set(h\_Mercury, 'xdata', x\_Mercury, 'ydata', y\_Mercury, 'zdata', t);

set(h\_Earth, 'xdata', x, 'ydata', y, 'zdata', t);

set(h\_Mars, 'xdata', x\_Mars, 'ydata', y\_Mars, 'zdata', t);

set(h\_Jupiter, 'xdata', x\_Jupiter, 'ydata', y\_Jupiter, 'zdata', t);

set(h\_Saturn, 'xdata', x\_Saturn, 'ydata', y\_Saturn, 'zdata', t);

set(h\_Uranus, 'xdata', x\_Uranus, 'ydata', y\_Uranus, 'zdata', t);

set(h\_Neptune, 'xdata', x\_Neptune, 'ydata', y\_Neptune, 'zdata', t);

set(h\_Venus, 'xdata', x\_Venus, 'ydata', y\_Venus, 'zdata', t);

%applying coordinates to all planets

X(k)=x;

Y(k)=y;

X\_Sun(k)=x\_Sun;

Y\_Sun(k)=y\_Sun;

X\_Mercury(k)=x\_Mercury;

Y\_Mercury(k)=y\_Mercury;

X\_Venus(k)=x\_Venus;

Y\_Venus(k)=y\_Venus;

X\_Mars(k)=x\_Mars;

Y\_Mars(k)=y\_Mars;

X\_Jupiter(k)=x\_Jupiter;

Y\_Jupiter(k)=y\_Jupiter;

X\_Uranus(k)=x\_Uranus;

Y\_Uranus(k)=y\_Uranus;

X\_Saturn(k)=x\_Saturn;

Y\_Saturn(k)=y\_Saturn;

X\_Neptune(k)=x\_Neptune;

Y\_Neptune(k)=y\_Neptune;

%drawing lines of different colors which represent trail planets leave

line(X, Y, T, 'linestyle', '--', 'linewidth', 0.1, 'color', [0.7 1 0]);

line(X\_Mercury, Y\_Mercury, T, 'linestyle', '--', 'linewidth', 0.1, 'color', [1 0 0]);

line(X\_Venus, Y\_Venus, T, 'linestyle', '--', 'linewidth', 0.1, 'color', [1 0.7 0]);

line(X\_Mars, Y\_Mars, T, 'linestyle', '--', 'linewidth', 0.1, 'color', [0 1 0]);

line(X\_Jupiter, Y\_Jupiter, T, 'linestyle', '--', 'linewidth', 0.1, 'color', [0.5 0.5 1]);

line(X\_Uranus, Y\_Uranus, T, 'linestyle', '--', 'linewidth', 0.1, 'color', [0 0 1]);

line(X\_Saturn, Y\_Saturn, T, 'linestyle', '--', 'linewidth', 0.1, 'color', [1 0 1]);

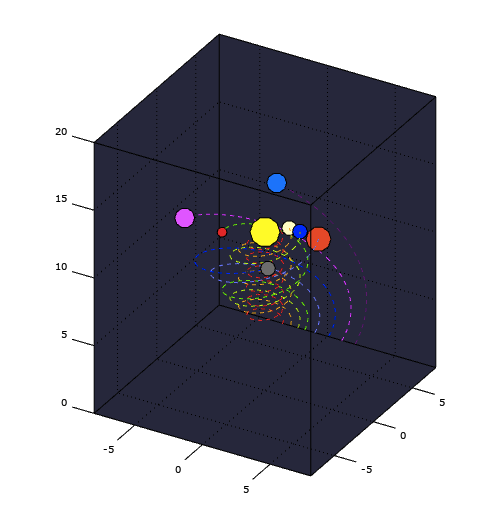
line(X\_Neptune, Y\_Neptune, T, 'linestyle', '--', 'linewidth', 0.1, 'color', [0.5 0 0.5]);

k=k+1;

drawnow;

end

**Result:**



**Program #3: “Geocentric System”**

The program creates 2-dimensional Geocentric System. The center of system is Earth and it may seem like every other planet moves around it, when actually all the planets still move around Sun. But this program shows planetary movement from the Earth’s perspective.

**Code:**

%basic parameters

figure('units', 'norm', 'position', [0.1 0.1 0.8 0.7]);

%the system is 2-dimensional

axes('position', [0.1 0.1 0.8 0.7], 'xlim', [-10 10], 'ylim', [-10 10], 'dataaspectratio', [1 1 1], 'color', [0.8 0.8 1]);

grid on;

%starting coordinates

x=0; y=0;

%setting planets and its parameters

h\_Mercury=line(x,y, 'marker', 'O', 'markersize', 10, 'markerfacecolor', [1 0.7 0]);

h\_Venus=line(x,y, 'marker', 'O', 'markersize', 15, 'markerfacecolor', [1 1 0.8]);

h\_Mars=line(x,y, 'marker', 'O', 'markersize', 10, 'markerfacecolor', [1 0 0]);

h\_Jupiter=line(x,y, 'marker', 'O', 'markersize', 25, 'markerfacecolor', [1 0.3 0]);

h\_Saturn=line(x,y, 'marker', 'O', 'markersize', 20, 'markerfacecolor', [1 0.3 1]);

h\_Uranus=line(x,y, 'marker', 'O', 'markersize', 15, 'markerfacecolor', [0.5 0.5 0.5]);

h\_Neptune=line(x,y, 'marker', 'O', 'markersize', 20, 'markerfacecolor', [0 0.5 1]);

h\_Earth=line(x,y, 'marker', 'O', 'markersize', 15, 'markerfacecolor', [0 0 1]);

h\_Sun=line(x,y, 'marker', 'O', 'markersize', 30, 'markerfacecolor', [1 1 0]);

%setting coordinates of planets

X\_Sun=0;

Y\_Sun=0;

X=0;

Y=0;

X\_Mercury=0;

Y\_Mercury=0;

X\_Venus=0;

Y\_Venus=0;

X\_Mars=0;

Y\_Mars=0;

X\_Jupiter=0;

Y\_Jupiter=0;

X\_Uranus=0;

Y\_Uranus=0;

X\_Saturn=0;

Y\_Saturn=0;

X\_Neptune=0;

Y\_Neptune=0;

k=1;

%start of a cycle to draw planetary movement

for fi=0:pi/25:5\*pi

%The Earth is in center of the system

x=2.25\*cos(2\*fi);

y=2.25\*sin(2\*fi);

x\_Sun=0;

y\_Sun=0;

x\_Mercury=cos(6\*fi);

y\_Mercury=sin(6\*fi);

x\_Venus=1.5\*cos(3\*fi);

y\_Venus=1.5\*sin(3\*fi);

x\_Mars=2.75\*cos(1.5\*fi);

y\_Mars=2.75\*sin(1.5\*fi);

x\_Jupiter=3.5\*cos(fi);

y\_Jupiter=3.5\*sin(fi);

x\_Uranus=4.25\*cos(0.8\*fi);

y\_Uranus=4.25\*sin(0.8\*fi);

x\_Saturn=5\*cos(0.5\*fi);

y\_Saturn=5\*sin(0.5\*fi);

x\_Neptune=6\*cos(0.3\*fi);

y\_Neptune=6\*sin(0.3\*fi);

%setting coordinates

set(h\_Mercury, 'xdata', x\_Mercury-x, 'ydata', y\_Mercury-y);

set(h\_Earth, 'xdata', x-x, 'ydata', y-y);

set(h\_Mars, 'xdata', x\_Mars-x, 'ydata', y\_Mars-y);

set(h\_Jupiter, 'xdata', x\_Jupiter-x, 'ydata', y\_Jupiter-y);

set(h\_Saturn, 'xdata', x\_Saturn-x, 'ydata', y\_Saturn-y);

set(h\_Uranus, 'xdata', x\_Uranus-x, 'ydata', y\_Uranus-y);

set(h\_Neptune, 'xdata', x\_Neptune-x, 'ydata', y\_Neptune-y);

set(h\_Venus, 'xdata', x\_Venus-x, 'ydata', y\_Venus-y);

set(h\_Sun, 'xdata', x\_Sun-x, 'ydata', y\_Sun-y);

%applying coordinates to all planets

X(k)=x;

Y(k)=y;

X\_Sun(k)=x\_Sun;

Y\_Sun(k)=y\_Sun;

X\_Mercury(k)=x\_Mercury;

Y\_Mercury(k)=y\_Mercury;

X\_Venus(k)=x\_Venus;

Y\_Venus(k)=y\_Venus;

X\_Mars(k)=x\_Mars;

Y\_Mars(k)=y\_Mars;

X\_Jupiter(k)=x\_Jupiter;

Y\_Jupiter(k)=y\_Jupiter;

X\_Uranus(k)=x\_Uranus;

Y\_Uranus(k)=y\_Uranus;

X\_Saturn(k)=x\_Saturn;

Y\_Saturn(k)=y\_Saturn;

X\_Neptune(k)=x\_Neptune;

Y\_Neptune(k)=y\_Neptune;

%drawing lines which represent trail planets leave

line(X-X, Y-Y, 'linestyle', '--', 'linewidth', 0.1, 'color', [0 0 0]);

line(X\_Mercury-X, Y\_Mercury-Y, 'linestyle', '--', 'linewidth', 0.1, 'color', [0 0 0]);

line(X\_Venus-X, Y\_Venus-Y, 'linestyle', '--', 'linewidth', 0.1, 'color', [0 0 0]);

line(X\_Mars-X, Y\_Mars-Y, 'linestyle', '--', 'linewidth', 0.1, 'color', [0 0 0]);

line(X\_Jupiter-X, Y\_Jupiter-Y, 'linestyle', '--', 'linewidth', 0.1, 'color', [0 0 0]);

line(X\_Uranus-X, Y\_Uranus-Y, 'linestyle', '--', 'linewidth', 0.1, 'color', [0 0 0]);

line(X\_Saturn-X, Y\_Saturn-Y, 'linestyle', '--', 'linewidth', 0.1, 'color', [0 0 0]);

line(X\_Neptune-X, Y\_Neptune-Y, 'linestyle', '--', 'linewidth', 0.1, 'color', [0 0 0]);

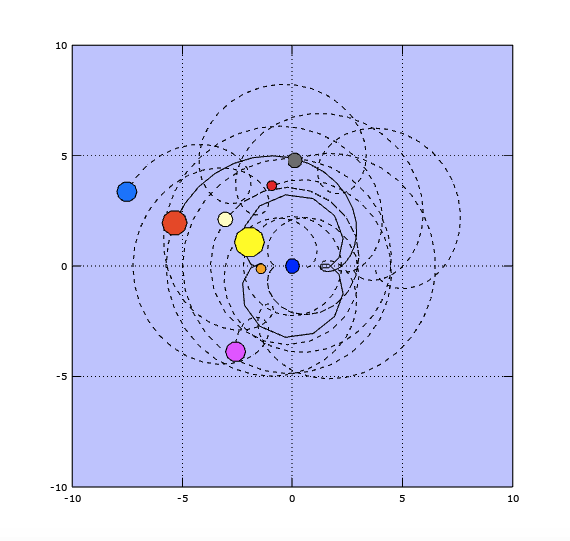
line(X\_Sun-X, Y\_Sun-Y, 'linestyle', '--', 'linewidth', 0.1, 'color', [0 0 0]);

k=k+1;

drawnow;

end

**Result:**

****

**Program #4: “Oscillation”**

The program draws Oscillation by changing directions and values of angles. It uses different coefficients that is why the final result consists of 6 different lines.

**Code:**

%basic parameters

figure('units', 'norm', 'position', [0.1 0.1 0.8 0.7]);

axes('position', [0.1 0.1 0.8 0.7], 'xlim', [0 10], 'ylim', [-3 3], 'color', [0.9 0.5 0.7]);

grid on;

%setting ball and its parameters to track movement of the line

h\_Ball=line(0, 0, 'marker', 'o');

%setting coordinates and coefficients

A=1;

W=1;

k=1;

T=0;

X=0;

%starting a cycle to draw an ocsillation

for t=0:0.1:10

x=A\*sin(W\*t);

X(k)=x;

T(k)=t;

set(h\_Ball, 'xdata', t, 'ydata', x);

line(T, X, 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end

%changing coordinates and coefficients for a new ocsillation

A=-1;

W=1;

k=1;

T=0;

X=0;

%starting cycle to draw new ocsillation

for t=0:0.1:10

x=A\*sin(W\*t);

X(k)=x;

T(k)=t;

set(h\_Ball, 'xdata', t, 'ydata', x);

line(T, X, 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end

%changing coordinates and coefficients for a new ocsillation

A=2;

W=1;

k=1;

T=0;

X=0;

%starting cycle to draw new ocsillation

for t=0:0.1:10

x=A\*sin(W\*t);

X(k)=x;

T(k)=t;

set(h\_Ball, 'xdata', t, 'ydata', x);

line(T, X, 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end

%changing coordinates and coefficients for a new ocsillation

A=-2;

W=1;

k=1;

T=0;

X=0;

%starting cycle to draw new ocsillation

for t=0:0.1:10

x=A\*sin(W\*t);

X(k)=x;

T(k)=t;

set(h\_Ball, 'xdata', t, 'ydata', x);

line(T, X, 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end

%changing coordinates and coefficients for a new ocsillation

A=3;

W=1;

k=1;

T=0;

X=0;

%starting cycle to draw new ocsillation

for t=0:0.1:10

x=A\*sin(W\*t);

X(k)=x;

T(k)=t;

set(h\_Ball, 'xdata', t, 'ydata', x);

line(T, X, 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end

%changing coordinates and coefficients for a new ocsillation

A=-3;

W=1;

k=1;

T=0;

X=0;

%starting cycle to draw new ocsillation

for t=0:0.1:10

x=A\*sin(W\*t);

X(k)=x;

T(k)=t;

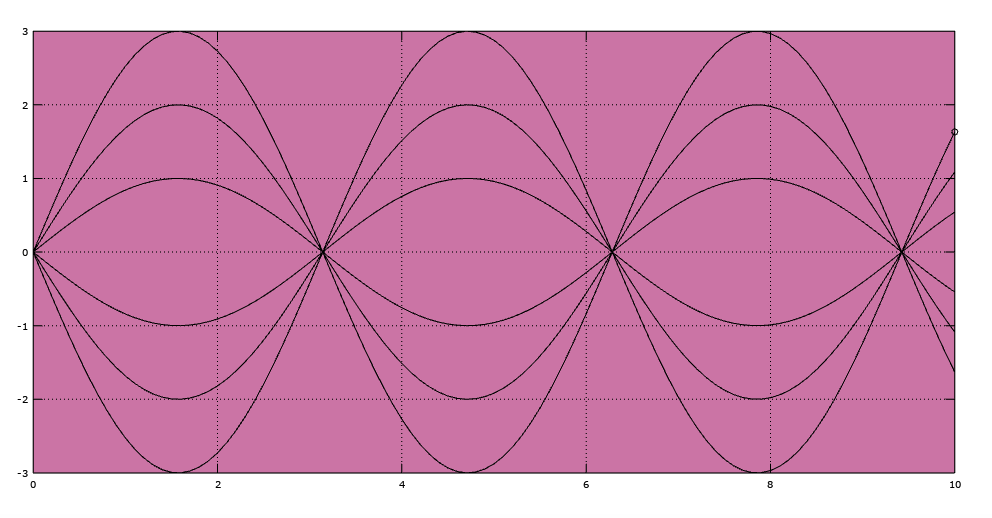
set(h\_Ball, 'xdata', t, 'ydata', x);

line(T, X, 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end

**Result:**

**Program #5: “Oscillation 2”**

Second oscillation which uses different function and values that is why the final result consists of several lines.

**Code:**

%basic parameters

figure('units', 'norm', 'position', [0.1 0.1 0.8 0.7]);

axes('position', [0.1 0.1 0.8 0.7], 'xlim', [0 10], 'ylim', [-2 2], 'color', [1 0.8 1]);

grid on;

%setting ball and its parameters to track movement of the line

h\_Ball=line(0, 0, 'marker', 'o');

%setting coordinates and coefficients

A=1;

W=1;

k=1;

T=0;

X=0;

%starting a cycle to draw an ocsillation

for t=0:0.1:10

x=A\*sin(W\*t);

X(k)=x;

T(k)=t;

set(h\_Ball, 'xdata', t, 'ydata', x);

line(T, X, 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end

%changing coordinates and coefficients for a new ocsillation

A=1;

W=-1;

k=1;

T=0;

X=0;

%starting cycle to draw new ocsillation

for t=0:0.1:10

x=A\*sin(W\*t);

X(k)=x;

T(k)=t;

set(h\_Ball, 'xdata', t, 'ydata', x);

line(T, X, 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end

%changing coordinates and coefficients for a new ocsillation

A=1;

W=2;

k=1;

T=0;

X=0;

%starting cycle to draw new ocsillation

for t=0:0.1:10

x=A\*sin(W\*t);

X(k)=x;

T(k)=t;

set(h\_Ball, 'xdata', t, 'ydata', x);

line(T, X, 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end

%changing coordinates and coefficients for a new ocsillation

A=1;

W=-2;

k=1;

T=0;

X=0;

%starting cycle to draw new ocsillation

for t=0:0.1:10

x=A\*sin(W\*t);

X(k)=x;

T(k)=t;

set(h\_Ball, 'xdata', t, 'ydata', x);

line(T, X, 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end

%changing coordinates and coefficients for a new ocsillation

A=1;

W=3;

k=1;

T=0;

X=0;

%starting cycle to draw new ocsillation

for t=0:0.1:10

x=A\*sin(W\*t);

X(k)=x;

T(k)=t;

set(h\_Ball, 'xdata', t, 'ydata', x);

line(T, X, 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end

%changing coordinates and coefficients for a new ocsillation

A=1;

W=-3;

k=1;

T=0;

X=0;

%starting cycle to draw new ocsillation

for t=0:0.1:10

x=A\*sin(W\*t);

X(k)=x;

T(k)=t;

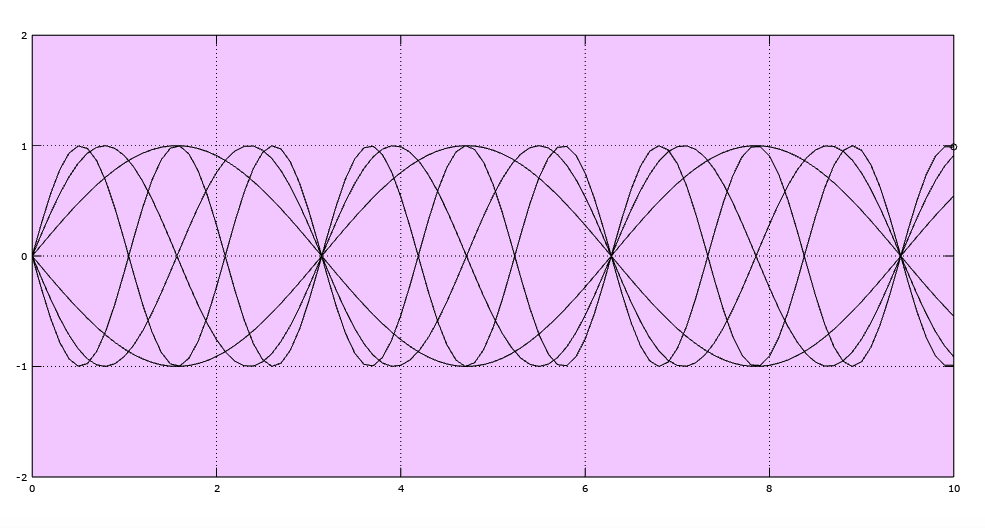
set(h\_Ball, 'xdata', t, 'ydata', x);

line(T, X, 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end

**Result:**

**Program #6: “Oscillation 3”**

This program contains more complex oscillation since there are different starting points that is why lines have a similar amplitude but still end their way at different points.

**Code:**

%basic parameters

figure('units', 'norm', 'position', [0.1 0.1 0.8 0.7]);

axes('position', [0.1 0.1 0.8 0.7], 'xlim', [0 10], 'ylim', [-2 2], 'color', [0.75 1 0.75]);

grid on;

%setting ball and its parameters to track movement of the line

h\_Ball=line(0, 0, 'marker', 'o');

%setting coordinates and coefficients

A=1;

W=5;

k=1;

T=0;

X=0;

A0=1;

W1=1;

i=0;

%starting a cycle to draw an ocsillation

for t=0:0.05:10

x=A\*sin(W\*t+i);

X(k)=x;

T(k)=t;

%applying coordinates and drawing a line with a ball

set(h\_Ball, 'xdata', t, 'ydata', x);

line(T, X, 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end

%setting new ball and its parameters to track movement of the different line

h\_Ball=line(0, 0, 'marker', 'o');

%changing coordinates and coefficients for a new ocsillation

A=1;

W=5;

k=1;

T=0;

X=0;

A0=1;

W1=1;

i=pi/4;

%starting cycle to draw new ocsillation

for t=0:0.05:10

x=A\*sin(W\*t+i);

X(k)=x;

T(k)=t;

set(h\_Ball, 'xdata', t, 'ydata', x);

line(T, X, 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end

%setting new ball and its parameters to track movement of the different line

h\_Ball=line(0, 0, 'marker', 'o');

%changing coordinates and coefficients for a new ocsillation

A=1;

W=5;

k=1;

T=0;

X=0;

A0=1;

W1=1;

i=-pi/4;

%starting cycle to draw new ocsillation

for t=0:0.05:10

x=A\*sin(W\*t+i);

X(k)=x;

T(k)=t;

set(h\_Ball, 'xdata', t, 'ydata', x);

line(T, X, 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end

%setting new ball and its parameters to track movement of the different line

h\_Ball=line(0, 0, 'marker', 'o');

%changing coordinates and coefficients for a new ocsillation

A=1;

W=5;

k=1;

T=0;

X=0;

A0=1;

W1=1;

i=pi/2;

%starting cycle to draw new ocsillation

for t=0:0.05:10

x=A\*sin(W\*t+i);

X(k)=x;

T(k)=t;

set(h\_Ball, 'xdata', t, 'ydata', x);

line(T, X, 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end

%setting new ball and its parameters to track movement of the different line

h\_Ball=line(0, 0, 'marker', 'o');

%changing coordinates and coefficients for a new ocsillation

A=1;

W=5;

k=1;

T=0;

X=0;

A0=1;

W1=1;

i=-pi/2;

%starting cycle to draw new ocsillation

for t=0:0.05:10

x=A\*sin(W\*t+i);

X(k)=x;

T(k)=t;

set(h\_Ball, 'xdata', t, 'ydata', x);

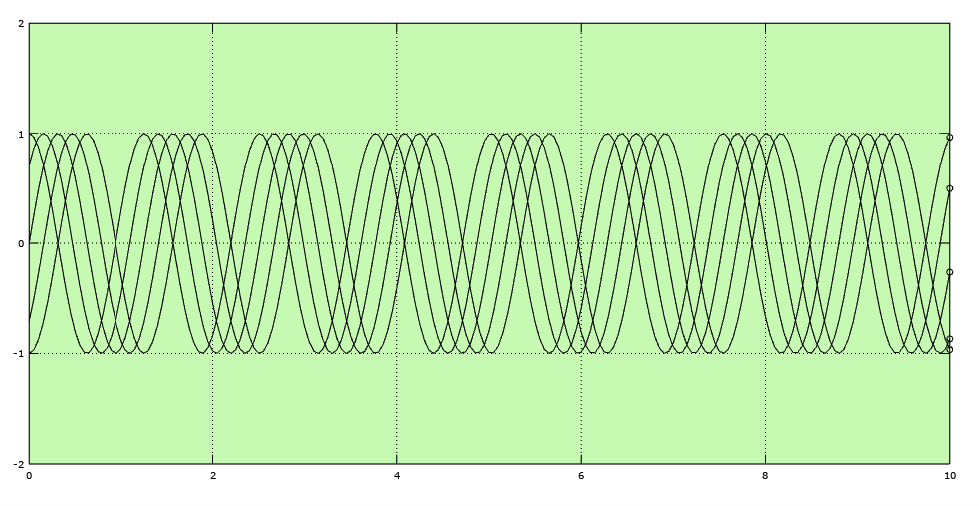
line(T, X, 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end

**Result:**



**Program #7: “Complex Oscillations”**

The program creates more complex oscillations which can be similar to waves and systematically changes its amplitude.

**Code:**

%basic parameters

figure('units', 'norm', 'position', [0.1 0.1 0.8 0.7]);

axes('position', [0.1 0.1 0.8 0.7], 'xlim', [0 10], 'ylim', [-2 2], 'color', [0.8 0.8 0.8]);

grid on;

%setting ball and its parameters to track movement of the line

h\_Ball=line(0, 0, 'marker', 'o');

%setting coordinates and coefficients

A=1;

W=20;

k=1;

T=0;

X=0;

A0=1;

W1=1;

i=0;

%starting a cycle to draw an ocsillation

for t=0:0.02:10

%the new function adds a new equation which causes oscillations to systematically change and return back

A=A0\*sin(W1\*t);

x=A\*sin(W\*t);

X(k)=x;

T(k)=t;

%applying coordinates and drawing a line with a ball

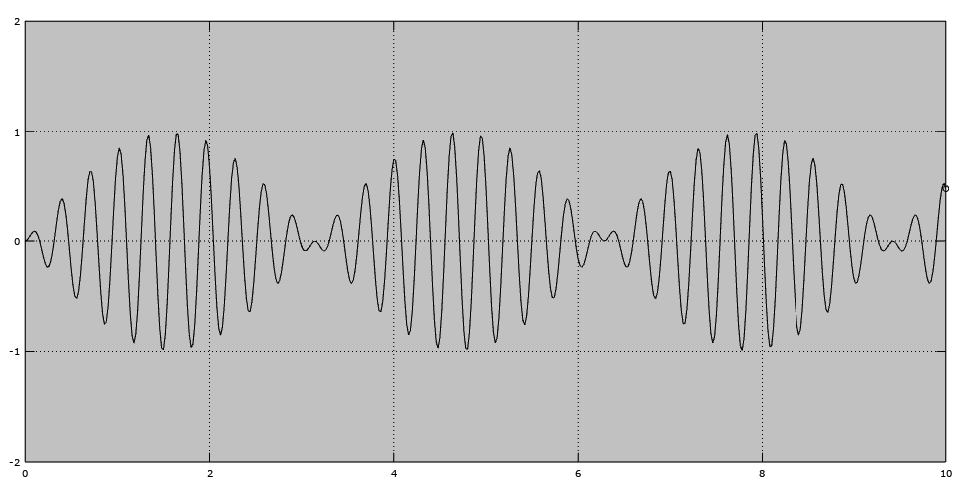
set(h\_Ball, 'xdata', t, 'ydata', x);

line(T, X, 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end

**Result:**

**Program #8: “Complex Oscillations 2”**

The program creates complex oscillations which systematically change its amplitude and represent certain kind of progression since it does not return to previous value and continue changing values until it stops.

**Code:**

%basic parameters

figure('units', 'norm', 'position', [0.1 0.1 0.8 0.7]);

axes('position', [0.1 0.1 0.8 0.7], 'xlim', [0 10], 'ylim', [-2 2], 'color', [0.8 0.8 0.8]);

grid on;

%setting ball and its parameters to track movement of the line

h\_Ball=line(0, 0, 'marker', 'o');

%setting coordinates and coefficients

A=1;

W=20;

W0=5;

k=1;

T=0;

X=0;

A0=1;

W1=1;

i=0;

%starting a cycle to draw an ocsillation

for t=0:0.025:10

%the new function adds a new equation which causes oscillations to systematically change and saves them

W=W0\*sin(W1\*t);

x=A\*sin(W\*t);

X(k)=x;

T(k)=t;

%applying coordinates and drawing a line with a ball

set(h\_Ball, 'xdata', t, 'ydata', x);

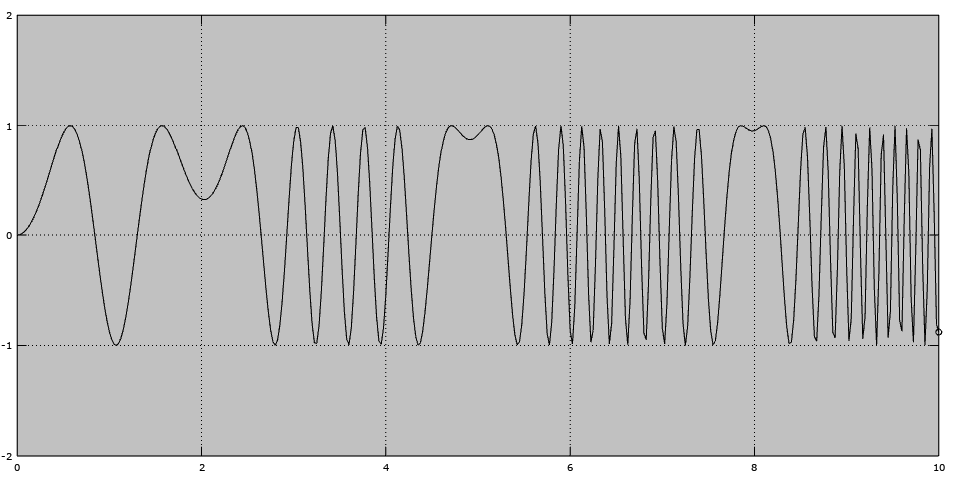
line(T, X, 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end

**Result:**



**Program #9: “Lissajous”**

The program creates Lissajous curve by drawing oscillations which return back to the starting point after ending certain type of function.

**Code:**

%basic parameters

figure('units', 'norm', 'position', [0.1 0.1 0.8 0.7]);

axes('position', [0.1 0.1 0.8 0.7], 'xlim', [-2 2], 'ylim', [-2 2], 'color', [1 0.8 0.8]);

grid on;

%setting ball and its parameters to track movement of the line

h\_Ball=line(0, 0, 'marker', 'o');

%setting coordinates and coefficients

A=1;

W=20;

k=1;

T=0;

X=0;

A0=1;

W1=1;

i=0;

Y=0;

B=1;

W2=3;

%starting a cycle to draw an ocsillation

for t=0:0.05:10

%because coordinates are not in the beginning of the graph and functions are systematic, the movement is cyclic and ends in the same place

x=A\*cos(W1\*t);

W1=5;

y=B\*sin(W2\*t);

W2=3;

b=1;

a=1;

X(k)=x;

Y(k)=y;

T(k)=t;

%applying coordinates and drawing a line with a ball

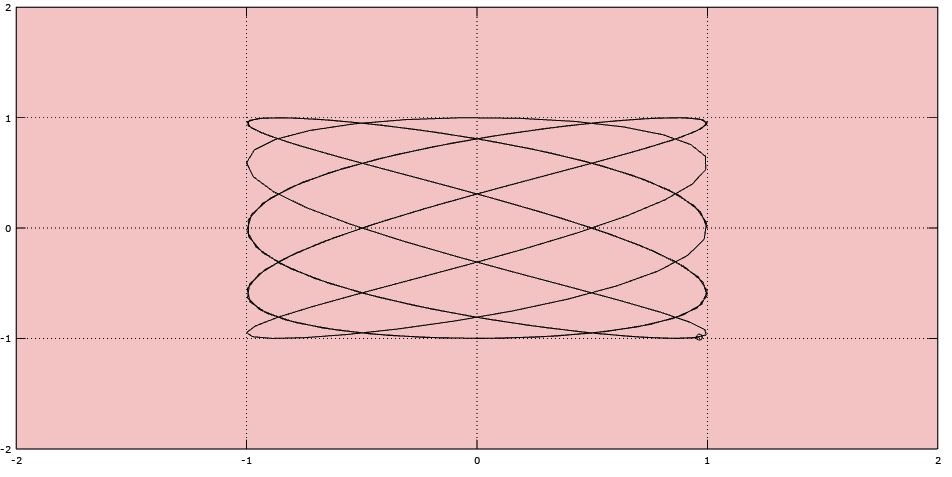
set(h\_Ball, 'xdata', x, 'ydata', y);

line(X, Y, 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end



**Result:**

**Program #10: “Electromagnetic field”**

Program draws 3-dimentional graph of electromagnetic field drawing them in form of lines of 2 colors. As we can see they change simultaneously in a harmonic order. Red is electric field and blue is magnetic.

**Code:**

%basic parameters

figure('units', 'norm', 'position', [0.1 0.1 0.8 0.7]);

axes('position', [0.1 0.1 0.8 0.7], 'xlim', [0 10], 'ylim', [-3 3], 'zlim', [-2 2], 'color', [0.8 1 0.7]);

grid on;

view(30, 30)

%setting coordinates and coefficients

t=0;

x=0;

y=0;

z=0;

X=0;

Y=0;

Z=0;

%starting a cycle to draw electromagnetic field

for t=0:0.05:10

%coordinates for an electric field

x=t;

y=0;

z=sin(t);

%drawing electric field using different lines of chosen color

X=[x, x];

Y=[0, 0];

Z=[0, z];

line(X, Y, Z, 'color', [1 0 0]);

drawnow

%coordinates for an electric field

x=t;

y=-sin(t);

z=0;

%drawing magnetic field using different lines of chosen color

X=[x, x];

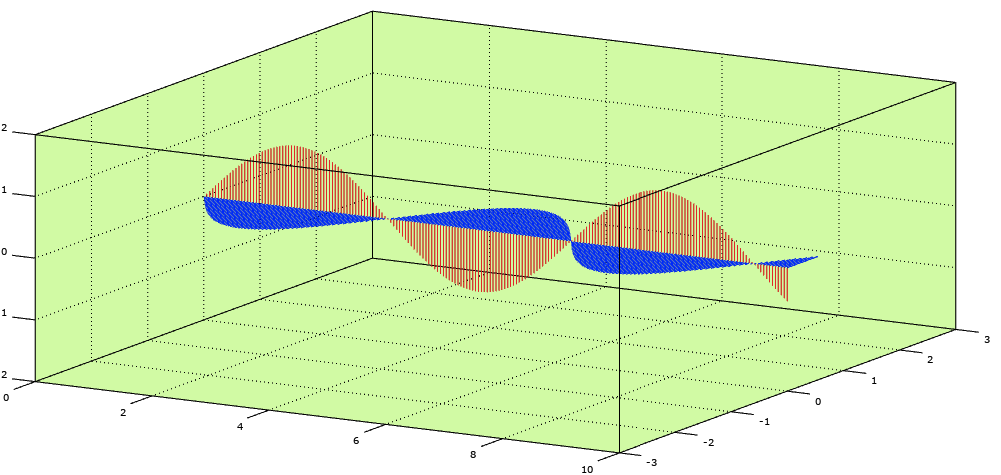
Y=[0, y];

Z=[0, 0];

line(X, Y, Z, 'color', [0 0 1]);

drawnow

end

**Result:**

**Program #11: “Projectile Movement”**

The program consists of a ball (cannon ball) that is shot under an angle A and after some time it slows and starts to fall. It has its certain trajectory and during its path the ball goes through 3 missiles that also move in their directions.

**Code:**

%basic parameters

figure('units', 'norm', 'position', [0.1 0.1 0.8 0.7]);

axes('position', [0.1 0.1 0.8 0.7], 'xlim', [0 50], 'ylim', [0 20], 'color', [0.5 0.8 1]);

grid on;

%setting coordinates and coefficients for Projectile movement and its missles

v0=20;

v1=10;

x1=0;

z1=10;

x2=0;

z2=15;

x3=50;

z3=13;

vx=0;

vy=0;

x=0;

z=0;

t=0;

g=9.8;

fi=pi/3;

X=0;

Z=0;

X1=0;

Z1=0;

X2=0;

Z2=0;

%setting ball and its parameters to track movement of the line

h\_Projectile=line(x,z, 'marker', 'O', 'markersize', 10, 'markerfacecolor', [0 0 0]);

%setting missles that projectile will hit

h\_Missle=line(x1,z1, 'marker', 'O', 'markersize', 20, 'markerfacecolor', [1 1 1]);

h\_Missle2=line(x2,z2, 'marker', 'O', 'markersize', 20, 'markerfacecolor', [1 1 1]);

h\_Missle3=line(x3,z3, 'marker', 'O', 'markersize', 20, 'markerfacecolor', [1 1 1]);

%starting a cycle to draw projectile movement and missles

for t=0:0.1:10

vx=v0\*cos(fi);

vy=v0\*sin(fi);

x=vx\*t;

z=vy\*t-1/2\*g\*t\*t;

x1=v1\*t;

x2=v1\*t;

x3=50-v1\*t;

X(k)=x;

T(k)=t;

Z(k)=z;

X1(k)=x1;

Z1(k)=z1;

X2(k)=x2;

Z2(k)=z2;

X3(k)=x3;

Z3(k)=z3;

%applying coordinates and drawing a line with a ball

set(h\_Projectile, 'xdata', x, 'ydata', z);

set(h\_Missle, 'xdata', x1, 'ydata', z1);

set(h\_Missle2, 'xdata', x2, 'ydata', z2);

set(h\_Missle3, 'xdata', x3, 'ydata', z3);

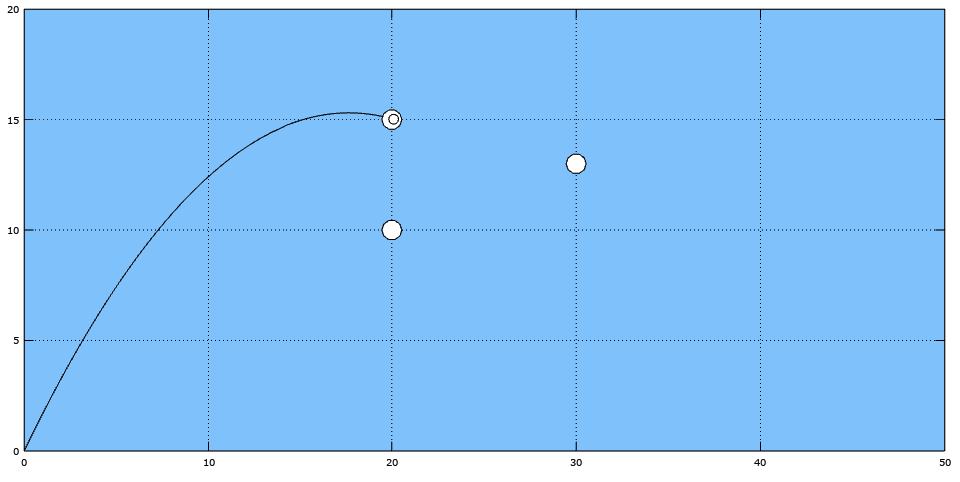
%drawing line which represent trail bullet leaves

line(X, Z, 'linestyle', '--', 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

end

**Result:**

**Program #12: “Firework”**

Program creates an animation of a firework. First it shots 3 balls of different colors in different directions. Then balls leave a trail after them and in one moment they explode and separate into 6 different colored sparkles which have shape of diamonds (rhombus). At the end sparkles spread and fall down until disappear.

**Code:**

%basic parameters

figure('position', [50 100 1200 800], 'color', [1 1 1]);

axes('position', [0.05 0.05 0.9 0.9], 'xlim', [-10 10], 'ylim', [-10 10], 'zlim', [0 4], 'color', [0.6 1 0.6]);

grid on;

view(45, 25);

%setting necessary variables

g=9.81;

v0=10;

alfa=pi/3;

%setting balls which will fly in different directions and separate in 6 stars each

h\_Ball = line(0,0,0, 'marker', 'o', 'markerfacecolor', [1 0 0]);

h\_Ball1 = line(0,0,0, 'marker', 'o', 'markerfacecolor', [0 1 0]);

h\_Ball2 = line(0,0,0, 'marker', 'o', 'markerfacecolor', [0 0 1]);

%setting colors that will be applyed to stars of salut after explosion in a certain order

colors = [[1 0 0]; [0 1 0]; [0 0 1]; [1 1 0]; [1 0 1]; [0 1 1]];

colors(1, :)

%setting coordinates and coefficients

x=0;

y=0;

z=0;

x1=0;

y1=0;

z1=0;

x2=0;

y2=0;

z2=0;

k=1;

p=1;

%starting a cycle to draw a ball movement

for i=1:6

%using 3 different functions to move balls in different diractions

h\_Salut(i) = line(x,y,z, 'marker', 'd', 'markersize', 20, 'markerfacecolor', colors(i,:), 'markeredgecolor', [0 0 0]);

h\_Salut1(i) = line(x1,y1,z1, 'marker', 'd', 'markersize', 20, 'markerfacecolor', colors(i,:), 'markeredgecolor', [0 0 0]);

h\_Salut2(i) = line(x2,y2,z2, 'marker', 'd', 'markersize', 20, 'markerfacecolor', colors(i,:), 'markeredgecolor', [0 0 0]);

end

for t= 0:0.01:0.4

%using functions to create the movement of the ball

x = v0\*cos(alfa)\*t;

y = v0\*cos(alfa)\*t;

z = v0\*sin(alfa)\*t - (1/2)\*g\*t^2;

x1 = v0\*(-cos(alfa))\*t;

y1 = v0\*cos(alfa)\*t;

z1 = v0\*sin(alfa)\*t - (1/2)\*g\*t^2;

x2 = v0\*(-cos(alfa))\*t;

y2 = v0\*(-cos(alfa))\*t;

z2 = v0\*sin(alfa)\*t - (1/2)\*g\*t^2;

%applying new coordinates to balls

set(h\_Ball, 'xdata', x, 'ydata', y, 'zdata', z);

set(h\_Ball1, 'xdata', x1, 'ydata', y1, 'zdata', z1);

set(h\_Ball2, 'xdata', x2, 'ydata', y2, 'zdata', z2);

%applying coordinates

X(k)=x;

Y(k)=y;

Z(k)=z;

X1(k)=x1;

Y1(k)=y1;

Z1(k)=z1;

X2(k)=x2;

Y2(k)=y2;

Z2(k)=z2;

%drawing a line trail for balls

line(X, Y, Z, 'linestyle', '--', 'color', [0 0 0], 'linewidth', 1);

line(X1, Y1, Z1, 'linestyle', '--', 'color', [0 0 0], 'linewidth', 1);

line(X2, Y2, Z2, 'linestyle', '--', 'color', [0 0 0], 'linewidth', 1);

k=k+1;

drawnow;

pause(0.05)

if(z < 0)

break;

endif;

end

set(h\_Ball, 'visible', 'off')

set(h\_Ball1, 'visible', 'off')

set(h\_Ball2, 'visible', 'off')

t0=t;

vS=5;

%starting a cycle to draw explosion

for t=0:0.01:1.5

%using functions to create the movement of the stars

x = v0\*cos(alfa)\*(t+t0);

y = v0\*cos(alfa)\*(t+t0);

z = v0\*sin(alfa)\*(t+t0) - (1/2)\*g\*(t+t0)^2;

x1 = v0\*(-cos(alfa))\*(t+t0);

y1 = v0\*cos(alfa)\*(t+t0);

z1 = v0\*sin(alfa)\*(t+t0) - (1/2)\*g\*(t+t0)^2;

x2 = v0\*(-cos(alfa))\*(t+t0);

y2 = v0\*(-cos(alfa))\*(t+t0);

z2 = v0\*sin(alfa)\*(t+t0) - (1/2)\*g\*(t+t0)^2;

%starting a cycle to draw movement of 6 stars after exposion

for i = 1:6

xS = x + vS\*cos(i\*2\*pi/6)\*t;

yS = y + vS\*sin(i\*2\*pi/6)\*t;

zS = z;

xS1 = x1 + vS\*cos(i\*2\*pi/6)\*t;

yS1 = y1 + vS\*sin(i\*2\*pi/6)\*t;

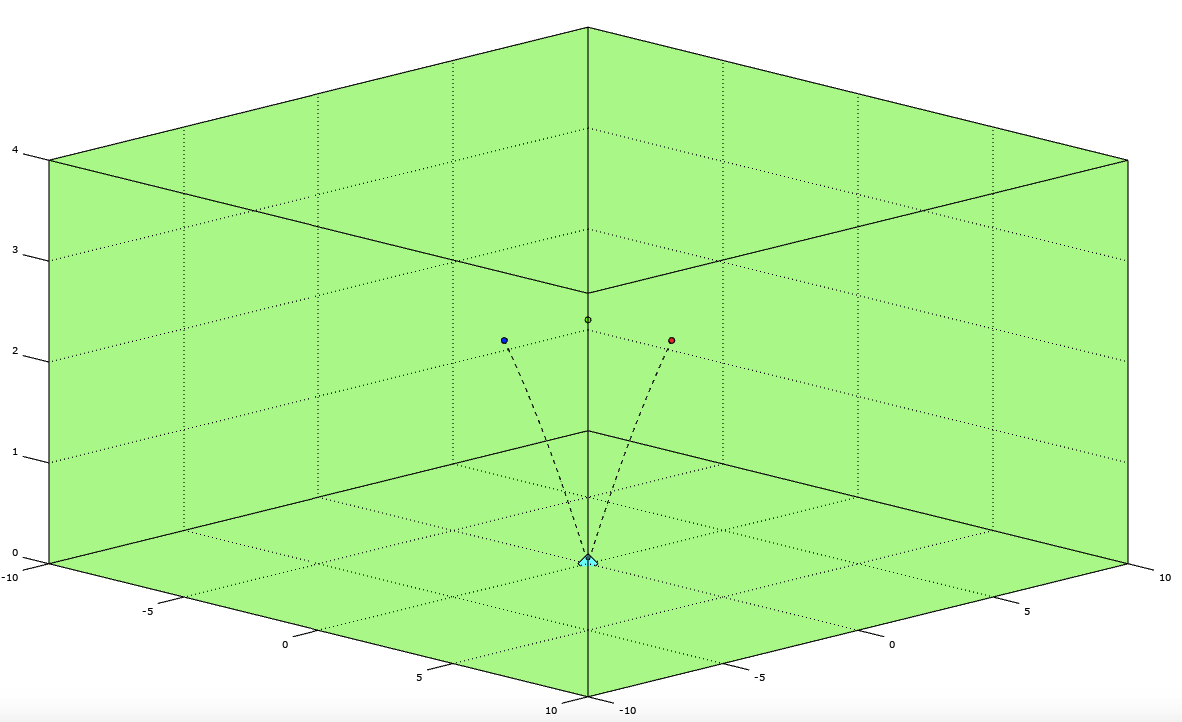
zS1 = z1;

xS2 = x2 + vS\*cos(i\*2\*pi/6)\*t;

yS2 = y2 + vS\*sin(i\*2\*pi/6)\*t;

zS2 = z2;

%applying new coordinates to stars

set(h\_Salut(i), 'xdata', xS, 'ydata', yS, 'zdata', zS);

set(h\_Salut1(i), 'xdata', xS1, 'ydata', yS1, 'zdata', zS1);

set(h\_Salut2(i), 'xdata', xS2, 'ydata', yS2, 'zdata', zS2);

end

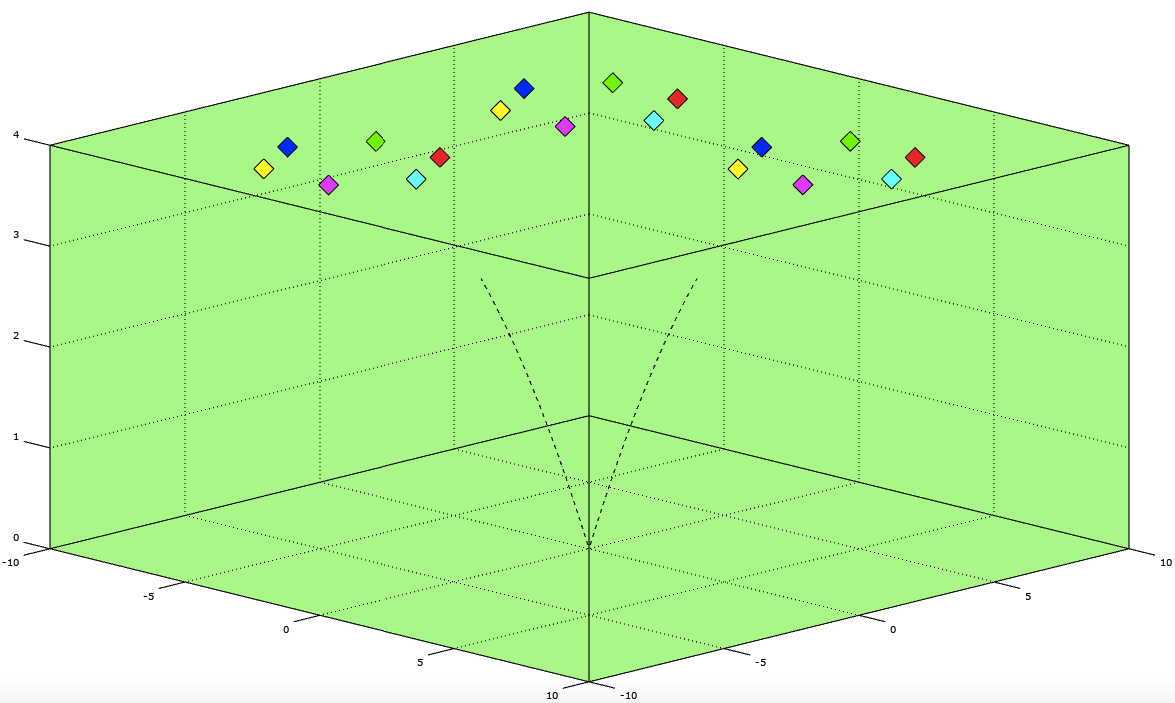
drawnow;

pause(0.05)

end

**Result:**

**Picture 1.**

**Pictu****re 2.**

**Program #13: “Runge-Kutta”**

Program draws a graph using Runge-Kutta method. It reminds oscillation but in this case the line consists of markers where every part have its own position and have the shape of star (\*).

**Code:**

%basic parameters

figure('units', 'norm', 'position', [0.1 0.1 0.8 0.7])

axes('xlim', [0 10], 'ylim', [-2 2], 'color', [0.6 0.6 1]);

grid on;

function ax = funVX(t, x, vx)

k=1;

m=1;

ax = - k\*x/m;

endfunction;

function Vx = funX(t, x, vx)

Vx = vx;

endfunction;

x = 0;

vx = 1;

dt = 0.05;

%starting a cycle to draw a line movement

for t= 0:dt:10

k1x = funX(t, x, vx);

k1Vx = funVX(t, x, vx);

k2x = funX(t+dt/2, x+k1x\*dt/2, vx+k1Vx\*dt/2);

k2Vx = funVX(t+dt/2, x+k1x\*dt/2, vx+k1Vx\*dt/2);

k3x = funX(t+dt/2, x+k2x\*dt/2, vx+k2Vx\*dt/2);

k3Vx = funVX(t+dt/2, x+k2x\*dt/2, vx+k2Vx\*dt/2);

k4x = funX(t+dt, x+k3x\*dt, vx+k3Vx\*dt);

k4Vx = funVX(t+dt, x+k3x\*dt, vx+k3Vx\*dt);

x = x + (k1x + 2\*k2x + 2\*k3x + k4x)\*dt/6;

vx = vx + (k1Vx + 2\*k2Vx + 2\*k3Vx + k4Vx)\*dt/6;

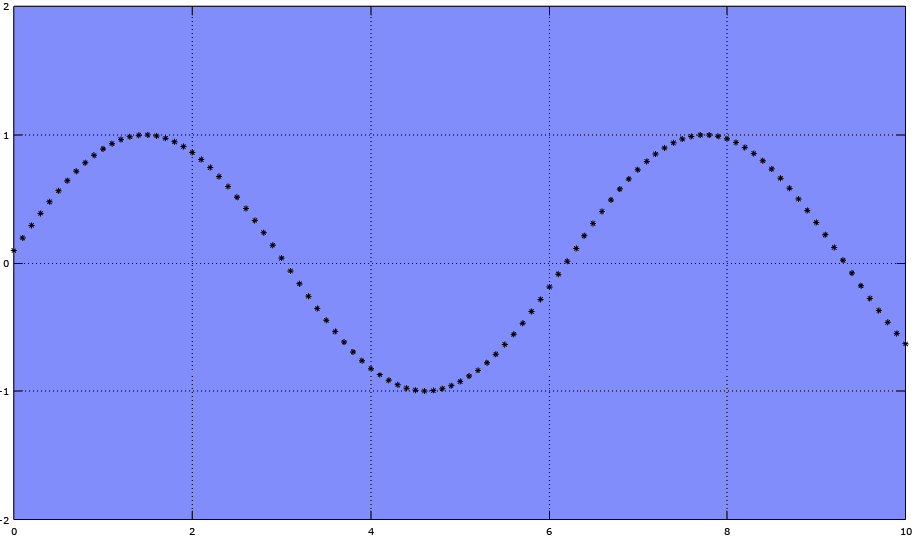
%setting a line that consists of \*

line(t,x,'marker','\*');

drawnow;

end;

**Result:**

****