A Matlab code follows that gives analytical solutions to arbitrary systems of two autonomous linear differential equations

For example the system from the exercise 9 (c) is defined by the command:

```
eqns = [diff(y,t)==3*y+z, diff(z,t)==-y+3*z];
```

```
syms y(t) z(t)
eqns = [diff(y,t)==3*y+z, diff(z,t)==-y+3*z];
sol = dsolve(eqns);
soly(t) = sol.y
solz(t) = sol.z
```

A Matlab code follows that lets to draw interactively a phase portrait to arbitrary systems of autonomous linear differential equations in plane .

Diagonal elements in the matrix `D` give eigenvalues, matrix `V` gives egenvectors.

```
t0 = 0; % starttid
tend = 20; % sluttid
A=[ 3 , 1  ;
    -1  , 3 ]; %  matris
[V,D]=eig(A)%  D - egenvärden, V- egenvektorer
button=1;
        xlabel('x1');
        ylabel('x2');
        axis equal
        axis([-5,5, -5, 5])

        hold on;
        L=10;
        plot([-L*V(1,1);L*V(1,1)],[-L*V(2,1),L*V(2,1)],'r');
         plot([-L*V(1,2);L*V(1,2)],[-L*V(2,2),L*V(2,2)],'r');
while button==1;
    [i,j,button]=ginput(1);
        [~, y] = ode45(@(t,y)A*y, [t0 tend], [i; j]); %
startpunkt[i,j]
        plot(y(:,1),y(:,2), 'b');
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

A Matlab code follows that lets to draw interactively a phase portrait to arbitrary systems of autonomous linear differential equations in R^3.

```
%%
t0 = 0; % starttime
L=10;

A=[ 3,-3, 1;
    3,-2, 2;
    -1, 2, 0 ];
[V,D]=eig(A)
tend = 20; % finish time
         xlabel('x');
         ylabel('y');
         zlabel('z');
         axis equal
        axis([-L,L, -L, L, -L,L]); %command fixing the domain
         % for drawinguseful when trajectories go to infinity
         hold on;
         V=2*L*V; %scaling the eigenvector V
         plot3([V(1,1);-V(1,1)],[V(2,1);-V(2,1)],[V(3,1);-
V(3,1)],'-r');
         %Draw the first eigenvector
         %plot3([V(1,2);-V(1,2)],[V(2,2);-V(2,2)],[V(3,2);-
V(3,2)],'-r');
          %Draw the second eigenvector
         options = odeset('RelTol',1e-5); %fixing relative
tolerance

  for i=1:100;
y0=randn(3,1); y0=2/norm(y0)*y0;% choose initial points
randomly distributed on a sphere of radius 2

   [~, y] = ode45(@(t,y)A*y, [t0 tend],y0, options);
        plot3(y(:,1),y(:,2), y(:,3), 'b');
end
```