% This program will solve a second order linear equation with given

% initial conditions. \*\*This program can only handle the case with

% **complex roots** of the characteristic equation. \*\*

clc; clear; close all; % Clear old values of variables

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

% Prompt user for coeffiecients ODE to solve

%

% MATLAB note: polynomials are stored as row vectors of

% coefficients; the size determines the degree.

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

%input coefficients for m^2+m+c=0

charpoly=input('Enter the characteristic polynomial as a vector [a b c] =>');

disp(sprintf('We will solve the homogeneous equation: '));

disp(sprintf(' %d y'''' + %d y'' + %d y = 0.\n',charpoly(1),charpoly(2),charpoly(3)));

t0=0;

y0=input('Enter the value of y(0) => ');

yp0 = input('Enter the value of y''(0) => ');

tmin = input('Enter the value of tmin =>' );

tmax = input('Enter the value of tmax =>' );

r=roots(charpoly); %find roots of characteristic equation

%error handling for checking roots

if (r(1) == r(2)) % == means Determine equality

disp(sprintf('This program won''t work for repeated roots of the characteristic equation.'));

disp(sprintf('Exiting program.'));

return

elseif (imag(r(1))==0) %need to check this

disp(sprintf('This program won''t work for non-complex roots of the characteristic equation.'));

disp(sprintf('Exiting program.'));

return

else

disp(sprintf('This program can solve the ODE'))

end

%Find mu and lambda

mu = imag(r(1));

lambda = real(r(1));

% Define the fundamental set of solutions as strings

y1spec = "exp(%8.2g\*t).\*cos(%8.2g\*t)";

y2spec = "exp(%8.2g\*t).\*sin(%8.2g\*t)";

y1str=sprintf(y1spec,lambda, mu); %y1 = e^(lambda\*t)\*cos(mu\*t)

y2str=sprintf(y2spec,lambda, mu);

% Convert fundamental set of solutions to functions

y1=inline(y1str,'t');

y2=inline(y2str,'t');

%solving system of y and y' to find C1 and C2

A=[1 0;lambda mu];

b=[y0;yp0];

x=A\b; %x is vector [c1,c2]

stry=sprintf('%.9g\*%s + %.9g\*%s',x(1),y1str,x(2),y2str); % general solution

%Plotting gen solution

disp(sprintf('\nThe solution to the IVP is y(t) = %s.',stry));

y=inline(stry,'t');

figure;

hold on;

tpts=linspace(-10,10,1000);

ypts=feval(y,tpts);

plot(tpts,ypts);

[ymin,j]=min(ypts); % Find smallest y-value so we can construct

miny=min(round(ymin-1),0); % nice window dimensions

axis([-10 10 miny 10]); % Set window size

gphtitle=sprintf('Solution to %dy'''' + %dy'' + %dy = 0, y(%.2g) = %.8g, y''(%.2g)=%.8g\n',charpoly(1),charpoly(2),charpoly(3),t0,y0,t0,yp0);

title(gphtitle); % add title to graph.

%Output

Enter the characteristic polynomial as a vector [a b c] =>[5 2 7]

We will solve the homogeneous equation:

5 y'' + 2 y' + 7 y = 0.

Enter the value of y(0) => 0

Enter the value of y'(0) => 1

Enter the value of tmin =>0

Enter the value of tmax =>20

This program can solve the ODE

The solution to the IVP is y(t) = 0\*exp( -0.2\*t).\*cos( 1.2\*t) + 0.857492926\*exp( -0.2\*t).\*sin( 1.2\*t).Chart

Description automatically generated

% This program will solve a second order linear equation with given

% initial conditions. \*\*This program can only handle the case with

% **real repeated roots** of the characteristic equation. \*\*

% We assume that t0 = 0.

% (Section 3.1 of text)

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

clc; clear; close all; % Clear old values of variables

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

% Prompt user for coeffiecients ODE to solve

%

% MATLAB note: polynomials are stored as row vectors of

% coefficients; the size determines the degree.

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

%input coefficients for m^2+m+c=0

charpoly=input('Enter the characteristic polynomial as a vector [a b c] =>');

disp(sprintf('We will solve the homogeneous equation: '));

disp(sprintf(' %d y'''' + %d y'' + %d y = 0.\n',charpoly(1),charpoly(2),charpoly(3)));

t0=0;

y0=input('Enter the value of y(0) => '); %input firt inital condition

yp0 = input('Enter the value of y''(0) => '); %input second inital condition

tmin = input('Enter the value of tmin =>' );

tmax = input('Enter the value of tmax =>' );

r=roots(charpoly); %find roots of characteristic equation

%error handling for checking roots (only handles m1!=m2)

if (r(1) ~= r(2)) % == means Determine equality

disp(sprintf('This program won''t work for distinct roots of the characteristic equation.'));

disp(sprintf('Exiting program.'));

return

elseif (imag(r(1))~=0)

disp(sprintf('This program won''t work for complex roots of the characteristic equation.'));

disp(sprintf('Exiting program.'));

return

else

disp(sprintf('This program can solve the ODE'))

end

% Define the fundamental set of solutions as strings

%sprintf function concacts string and var

y1spec = "exp(%8.2g\*t)";

y2spec = "t.\*exp(%8.2g\*t)";

y1str=sprintf(y1spec,r(1)); %y1 = e^(r1\*t)

y2str=sprintf(y2spec,r(1)); %y1 = e^(r1\*t)

% Convert fundamental set of solutions to functions

y1=inline(y1str,'t');

y2=inline(y2str,'t');

%solving system of y and y' to find C1 and C2

%y'=root\*y1 ie y=e^mt so y'=me^mt

%A=

A=[1 0; r(1) 1]; % This is the coefficient matrix

b=[y0;yp0];

x=A\b; % Solves Ax=b by x=Inv(A)\*b, x is vector [c1,c2]

stry=sprintf('%.9g\*%s + %.9g\*%s',x(1),y1str,x(2),y2str); % general solution

%Plotting gen solution

disp(sprintf('\nThe solution to the IVP is y(t) = %s.',stry));

y=inline(stry,'t');

figure;

hold on;

tpts=linspace(-10,10,1000);

ypts=feval(y,tpts);

plot(tpts,ypts);

[ymin,j]=min(ypts); % Find smallest y-value so we can construct

miny=min(round(ymin-1),0); % nice window dimensions

axis([-10 10 miny 10]); % Set window size

gphtitle=sprintf('Solution to %dy'''' + %dy'' + %dy = 0, y(%.2g) = %.8g, y''(%.2g)=%.8g\n',charpoly(1),charpoly(2),charpoly(3),t0,y0,t0,yp0);

title(gphtitle); % add title to graph.

%Output

Enter the characteristic polynomial as a vector [a b c] =>[4 12 9]

We will solve the homogeneous equation:

4 y'' + 12 y' + 9 y = 0.

Enter the value of y(0) => 1

Enter the value of y'(0) => -4

Enter the value of tmin =>0

Enter the value of tmax =>5

This program can solve the ODE

The solution to the IVP is y(t) = 1\*exp( -1.5\*t) + -2.5\*t.\*exp( -1.5\*t).

Graphical user interface

Description automatically generated