Week 3 - dsolve() with Initial Conditions

MAT330: Differential Equations

Student Name JoAnn Olney

Date 5/21/2025

Introduction:

This MATLAB assignment continues working with the dsolve() function. In the previous assignment, you solved differential equations without initial conditions. This resulted in a family of solutions to the differential equation.

In this MATLAB assignment, you'll learn how to add inital condition constraints to the dsolve() solution. The solution provided by dsolve() will be a single unique solution to the differential equation for those initial conditions.

Review the code and comments provided in the "Examples" section below, and then use this information to complete the problems listed in the "Problems" section.

Make sure to run your code so all relevant computations/results are displayed, delete the "Introduction" and "Examples" sections, and then export your work as a PDF file for submission (your submission only needs to contain the "Problems" section that you completed).

Examples:

```
% Example 1 Code
% This example solves the differential equation y'' +3y' = 2 - 6t with two
% different sets of initial conditions and plots their solution on a single
% figure:
% a) initial conditions y(1) = 1 and y'(0) = 4
% b) initial conditions y(1) =-2 and y'(0) = 5

%define the functions/variables of interest
syms y(t) t;

%compute derivatives of y(t) with respect to t using diff() function
Dy(t)  = diff(y(t));
Dy2(t) = diff(y(t),2);  %the 2 as second argument indicates 2nd order
derivative

%define the differential equation
```

```
ode = Dy2(t) + 3*Dy(t) == 2-6*t
```

ode =

$$\frac{\partial^2}{\partial t^2}\, y(t) + 3\frac{\partial}{\partial t}\, y(t) = 2 - 6\,t$$

```
%define the first set of initial conditions
%Note: It's best practice to define a cond1, cond2, variable for each
%condition. Then put all conditions together as a vector of conditions.
%It's also common to order the conditions in increasing order of
%derivative.
cond1 = y(1)  == 1;
cond2 = Dy(0) == 4;
cond = [cond1; cond2];

%solve the differential equation using dsolve()
%the second input to dsolve is the "cond" variable with specifies all the
%initial conditions/constraints
ySolnA(t) = dsolve(ode,cond)
```

ySolnA(t) =

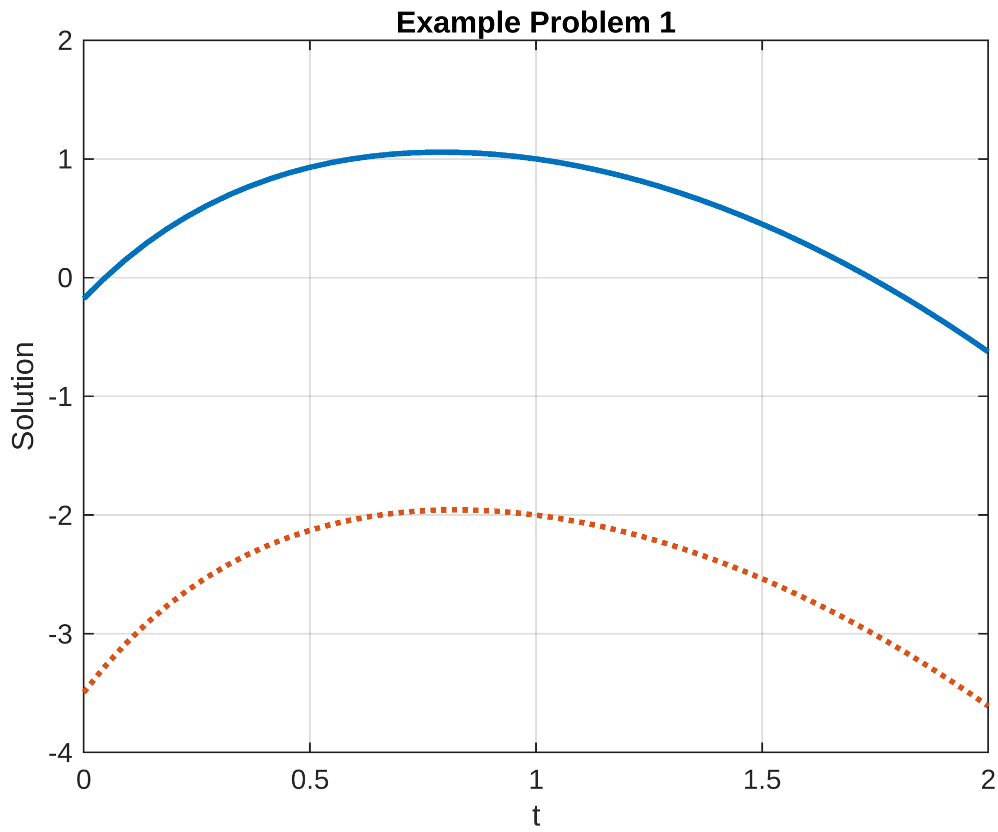
$$\frac{4\,t}{3} - \frac{8\,\mathrm{e}^{-3\,t}}{9} - t^2 + \frac{2\,\mathrm{e}^{-3}\,(5\,\mathrm{e}^3 + 4)}{9} - \frac{4}{9}$$

```
%set other conditions and solve again
cond1 = y(1)  == -2;
cond2 = Dy(0) == 5;
cond = [cond1; cond2];
ySolnB(t) = dsolve(ode,cond)
```

ySolnB(t) =

$$\frac{4\,t}{3} - \frac{11\,\mathrm{e}^{-3\,t}}{9} - t^2 - \frac{\mathrm{e}^{-3}\,(17\,\mathrm{e}^3 - 11)}{9} - \frac{4}{9}$$

```
%plot the solutions on a single plot
%Examining the plot for y(t), we see that the curves pass through the
%specified initial condition value y(1)=1 and y(1) =-2 as desired. If your
%plots for y(t) don't pass through the correct initial condition value,
%you've probably made a mistake somewhere.
figure;
fplot(ySolnA,[0 2],'-','linewidth',2);
hold on;
fplot(ySolnB,[0 2],':','linewidth',2);
xlabel('t');
ylabel('Solution');
title('Example Problem 1');
grid on;
ylim([-4 2]);
```



Problems:

Problem 1: Use the MATLAB dsolve() function to solve the differential equation $y'' - y' = 2t^2 - t - 5$ with initial conditions $y(1) = 3, y'(0) = 0$. Plot the your solution on a figure as a single solid curve for time $t = -2$ to 2. Make sure to label both axes, title your figure, and turn on the plotting legend. Set the y-axis limits to [-15 10].

```
% Problem 1 Code Here
%define the funtions/variables
syms y(t) t;

%derivatives of y(t) w/ respect to t using the diff() funct
Dy(t) = diff(y(t));
Dy2(t) = diff(y(t),2);

%define the diff equation
ode = Dy2 - Dy == 2*t^2-t-5
```

ode(t) =

$$\frac{\partial^2}{\partial t^2}\, y(t) - \frac{\partial}{\partial t}\, y(t) = 2\,t^2 - t - 5$$

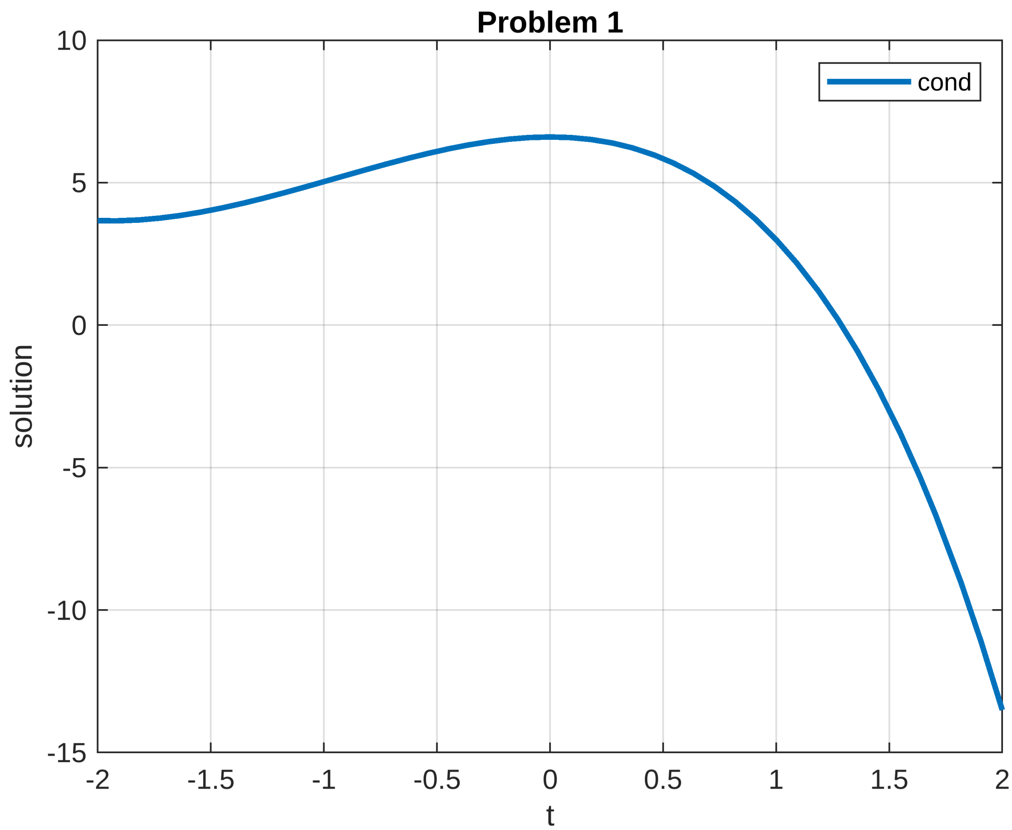
```


%define the first set of initial conditions
```

```
cond1 = y(1) == 3;
cond2 = Dy(0) == 0;
%put the conditions together
cond = [cond1; cond2];

%solve the diff equation using dsolve()
ysolnA(t) = dsolve(ode,cond)
```

ysolnA(t) =

$$2t + 2e - 2e^t - \frac{3t^2}{2} - \frac{2t^3}{3} + \frac{19}{6}$$

```


%plot the solution
%figure
figure;
fplot(ysolnA, [-2, 2], '-', 'LineWidth', 2);

%label axes
%title
xlabel('t');
ylabel('solution');
title('Problem 1');

%grid on
grid on;

%legend
legend('cond');

%y limit
ylim([-15, 10]);
```



Problem 2: Analyze your solution from Problem 1 to check that the initial conditions are satisfied. Compute $y(1)$ and $y'(0)$ and by hand to verify their values. Make sure to show your work.

Put explanation/math here...

given: y′′−y′=2t^2−t−5

So the complementary solution is: yh(t)=C1+C2e^t

Since the right-hand side is a quadratic polynomial: yp(t)=At^2 +Bt +C

yp′=2At+B

yp′′=2A

Substitute into the ODE: yp′′−yp′= 2A− (2At+B) = −2A t−B +2A

−2At −B +2A = 2t^2 −t −5

yp′′−yp′ = (6At+2B) −(3At^2 +2Bt+C)=−3At^2 + (6A−2B)t + (2B−C)

So the particular solution is: yp(t)= $-\frac{2}{3}t^3 - \frac{3}{2}t^2 + 2t + D$

The general solution is: $y(t) = \mathrm{yh} + \mathrm{yp} = \mathrm{C1} + \mathrm{C2c}^{t} + \left(-\frac{2}{3}t^3 - \frac{3}{2}t^2 + D\right)$

Compute y(0): y(0)=A+C2e^0−0−0+0=A+C2=0⇒A=−C2

Compute y(1): $\mathrm{C2}(e-1)+\left(2-\frac{2}{3}-\frac{3}{2}\right)=\mathrm{C2}(e-1)=\frac{19}{6}\Rightarrow \mathrm{C2}=\frac{19}{6(e-1)}$

$A=-\mathrm{C2}=-\frac{19}{6(e-1)}$

Final Solution:

$$y(t)=-\frac{19}{6(e-1)}+\frac{19}{6(e-1)}e^t-\frac{2}{3}t^3-\frac{3}{2}t^2+2t$$

Problem 3: Use the MATLAB dsolve() function to solve the differential equation $8y'''=\cos(20t)+\sin(2t)$ for initial conditions $y(10)=50, y'(0)=0, y''(0)=0$ and $y(10)=15, y'(0)=0, y''(0)=3$. Plot the your solutions on a single figure as a solid curve for the first set initial conditions and a dotted curve for the second set of initial conditions for time $t=-10$ to 20. Make sure to label both axes and title your figure, and turn on the plotting legend. Set the y-axis limits to [-150 200].

```
% Problem 3 Code Here
%define the funtion/ variables
syms y(t) t;

%compute the derivatives of y(t) w/ respect to t using diff() funct
Dy3 = diff(y(t),3);

%define the differential equation
ode = 8*Dy3 == cos(20*t) + sin(2*t)
```

ode =

$$8\frac{\partial^3}{\partial t^3}\,y(t)=\cos(20\,t)+\sin(2\,t)$$

```
% define the set of initial conditions
cond1 = y(10) == 50;
cond2 = Dy(0) == 0;
cond3 = Dy2(0) == 0;
%put the conditions together
cond = [cond1; cond2; cond3];

%solve the diff equation using dsolve()
ysolnA(t) = dsolve(ode, cond)
```

ysolnA(t) =

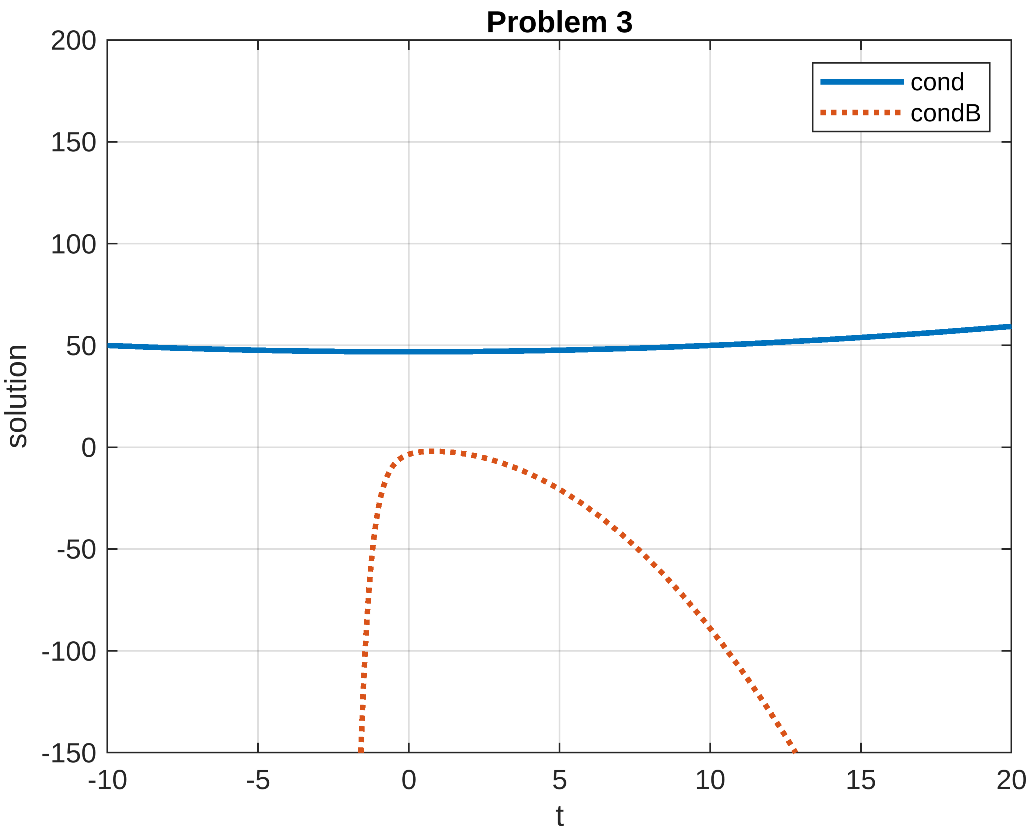
$$\frac{t}{3200}+\frac{\cos(2\,t)}{64}-\frac{\sin(20\,t)}{64000}-\frac{\cos(20)}{64}+\frac{\sin(200)}{64000}+\frac{t^2}{32}+\frac{14999}{320}$$

```
%define the condtions for B
cond4 = y(10) == 15;
cond5 = Dy(0) == 0;
cond6 = Dy2(0) == 3;
%put the conditions together for B
condB = [cond4; cond5; cond6];

%solve the diff equation B using dsolve()
ysolnB(t) = dsolve(ode, condB)
```

ysolnB(t) =

$$\frac{t}{3200}+\frac{\cos(2\,t)}{64}-\frac{\sin(20\,t)}{64000}-\frac{\cos(20)}{64}+\frac{\sin(200)}{64000}+\frac{49\,t^2}{32}-\frac{44201}{320}$$

```
%plot the solution
%figure
figure;
fplot(ysolnA, [-10, 20], '-', 'LineWidth', 2);
%hold on
hold on;
fplot(ySolnB, [-10, 20], ':', 'LineWidth', 2);
%label axes
%title
xlabel('t');
ylabel('solution');
title('Problem 3');
%grid on
grid on;
%legend
legend('cond', 'condB');
%y limits
ylim([-150, 200]);
```



Problem 4: Analyze your solution from Problem 3. How did changing the initial conditions change the solution? Which terms changed when the initial conditions changed? Explain.

Put explanation/math here...

Changing the initial condition in the problem changed the constant and the t^2 value, but the rest of the solution stayed the same. The t^2 value went from t^2/32 to 49t^2/32 and the constant went from 14999/320 to -44201/320.