Table of Contents

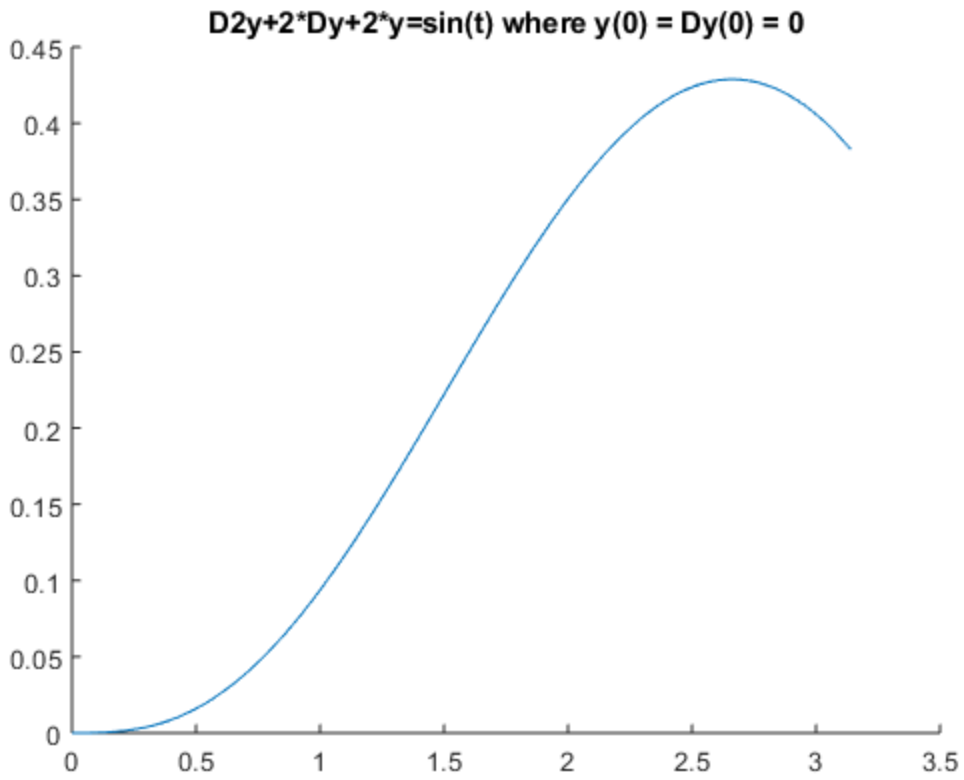

```
% Some Laplace transforms
```

QUESTION 1

12 (a)

```
syms t y
hold on
sol = dsolve('D2y = -2*Dy-2*y+sin(t)', 'y(0)=0', 'Dy(0)=0', 't')
fplot(sol, [0, pi])
title ('D2y+2*Dy+2*y=sin(t) where y(0) = Dy(0) = 0')
```

```
sol =

(2*exp(-t)*cos(t))/5 + (exp(-t)*sin(t))/5 - sin(t)*(cos(2*t)/5 -
 sin(2*t)/10) + exp(-t)*cos(t)*((cos(2*t)*exp(t))/10 - exp(t)/2 +
 (sin(2*t)*exp(t))/5)
```

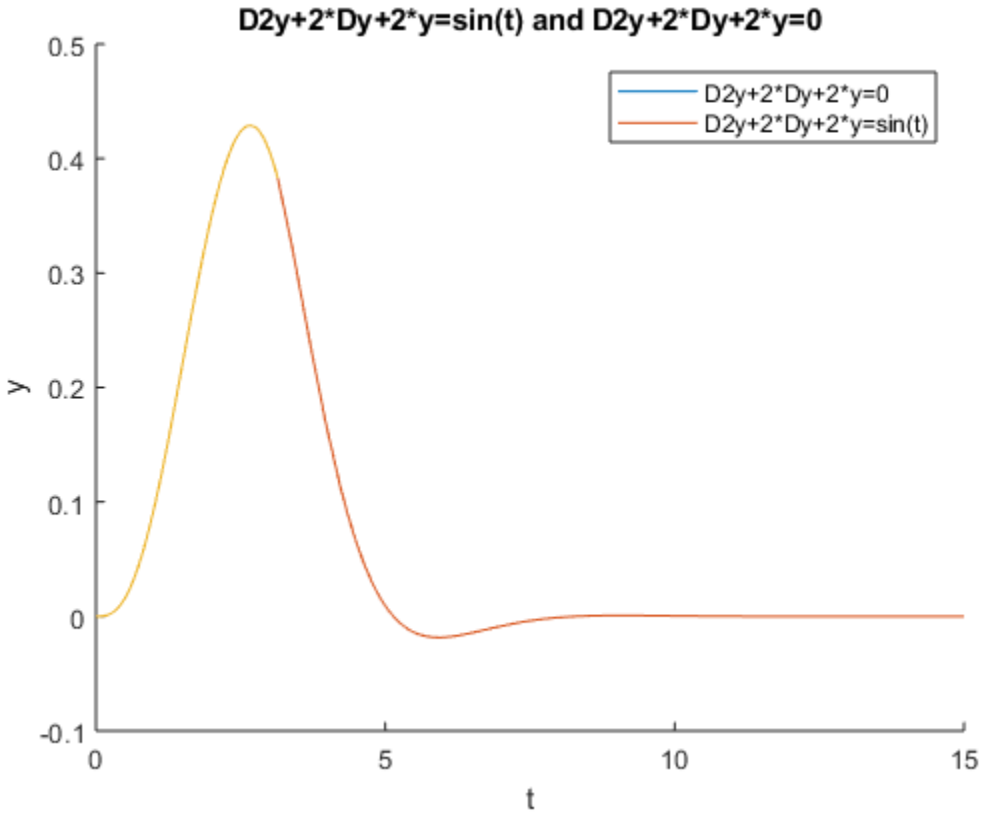


12 (b)

```
i1 = subs(sol,pi)
i2 = subs(diff(sol),pi)
sol2= dsolve('D2y+2*Dy+2*y=0','y(pi)=2/5-(2*exp(-pi))/5','Dy(pi)=-
exp(-pi)/5-1/5')
hold on
fplot(sol2,[pi 15])
fplot(sol, [0, pi])
axis([0 15 -.1 .5])
title('D2y+2*Dy+2*y=sin(t) and D2y+2*Dy+2*y=0')
xlabel 't'; ylabel 'y';
hold off
legend ('D2y+2*Dy+2*y=0','D2y+2*Dy+2*y=sin(t)')
```

```
i1 =

2/5 - (2*exp(-pi))/5


i2 =

exp(-pi)/5 - 1/5
```

```
sol2 =

- exp(-t)*cos(t)*((2*exp(pi))/5 - 2/5) - exp(-t)*sin(t)*(exp(pi)/5 -
 3/5)
```

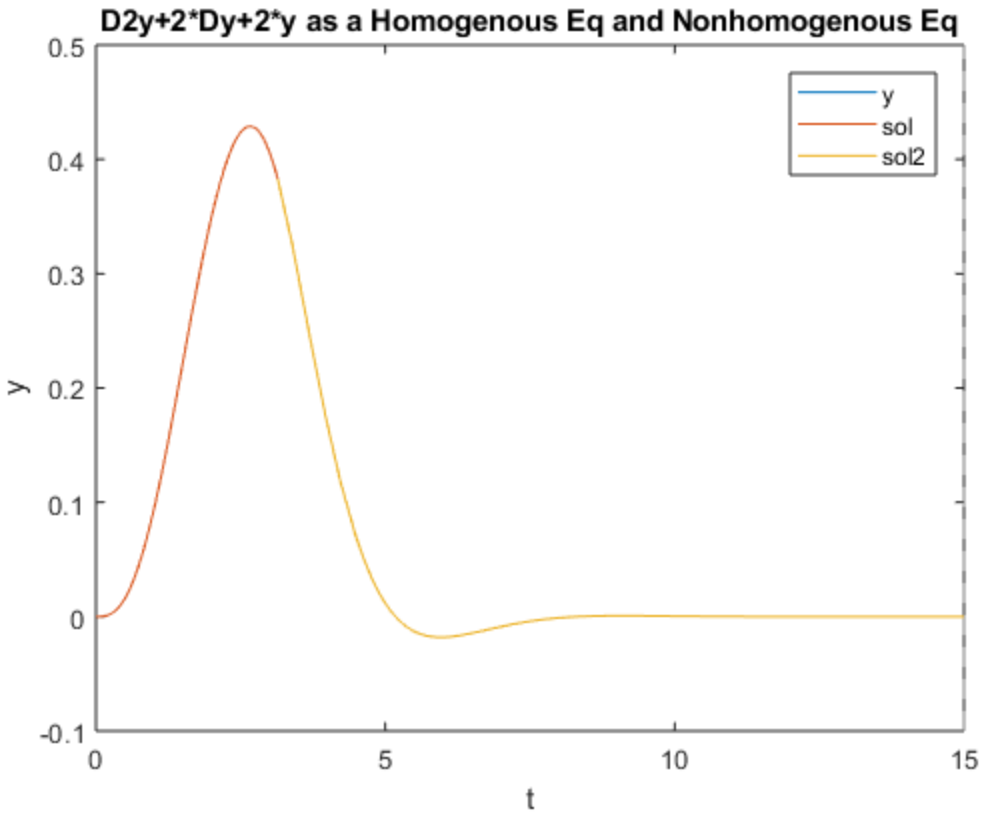


12 (c)

```
syms t s Y;
f = ['heaviside(t)*sin(t)+heaviside(t-pi)*(-sin(t))'];
eqn = sym(['D(D(y))(t)+2*D(y)(t)+2*y(t)=' f]);
lteqn = laplace(eqn, t ,s);
neweqn = subs(lteqn, {'laplace(y(t),t,s)','y(0)',subs(diff(y,t),t,0)},
{Y,0,0});
ytrans = solve(neweqn, Y);
y = ilaplace(ytrans, s, t)

% (c) contd..
y = ilaplace(ytrans, s, t);
sol = dsolve('D2y = -2*Dy-2*y+sin(t)', 'y(0)=0', 'Dy(0)=0', 't');
sol2= dsolve('D2y+2*Dy+2*y=0','y(pi)=0.3827','Dy(pi)=-0.1914');
fplot(y,[0 15])
hold on
fplot(sol,[0 pi])
hold on
fplot(sol2,[pi 15])
```

```
hold on
axis([0 15 -.1 0.5])
title('D2y+2*Dy+2*y as a Homogenous Eq and Nonhomogenous Eq')
xlabel 't'; ylabel 'y';
legend ('y','sol','sol2')
hold off
% the Laplace transform method gives the same solution and the same
 graph as from part b
% so Laplace gives the same result with less work than splitting the
 problem into two IVPs
```

```
Warning: Support of character vectors will be removed in a future
release. Character vectors can be used only for variable names and
numbers. Instead, to create symbolic expressions first create symbolic
variables using 'syms'. To evaluate character vectors and strings
representing symbolic expressions, use 'str2sym'.
Warning: Support of character vectors will be removed in a future
release. Character vectors can be used only for variable names and
numbers. Instead, to create symbolic expressions first create symbolic
variables using 'syms'. To evaluate character vectors and strings
representing symbolic expressions, use 'str2sym'.
Warning: Support of character vectors will be removed in a future
release. Character vectors can be used only for variable names and
numbers. Instead, to create symbolic expressions first create symbolic
variables using 'syms'. To evaluate character vectors and strings
representing symbolic expressions, use 'str2sym'.

y =

sin(t)/5 - (2*cos(t))/5 + (2*exp(-t)*(cos(t) + sin(t)/2))/5 -
 heaviside(t - pi)*(sin(t)/5 - (2*cos(t))/5 + (2*exp(pi - t)*(cos(t) +
 sin(t)/2))/5) + exp(-t)*sin(t)*subs(diff(y(t), t), t, 0)

Warning: Finite sets ('DOM_SET') not supported. Using element 't = 0'
instead.
```



12 (d)

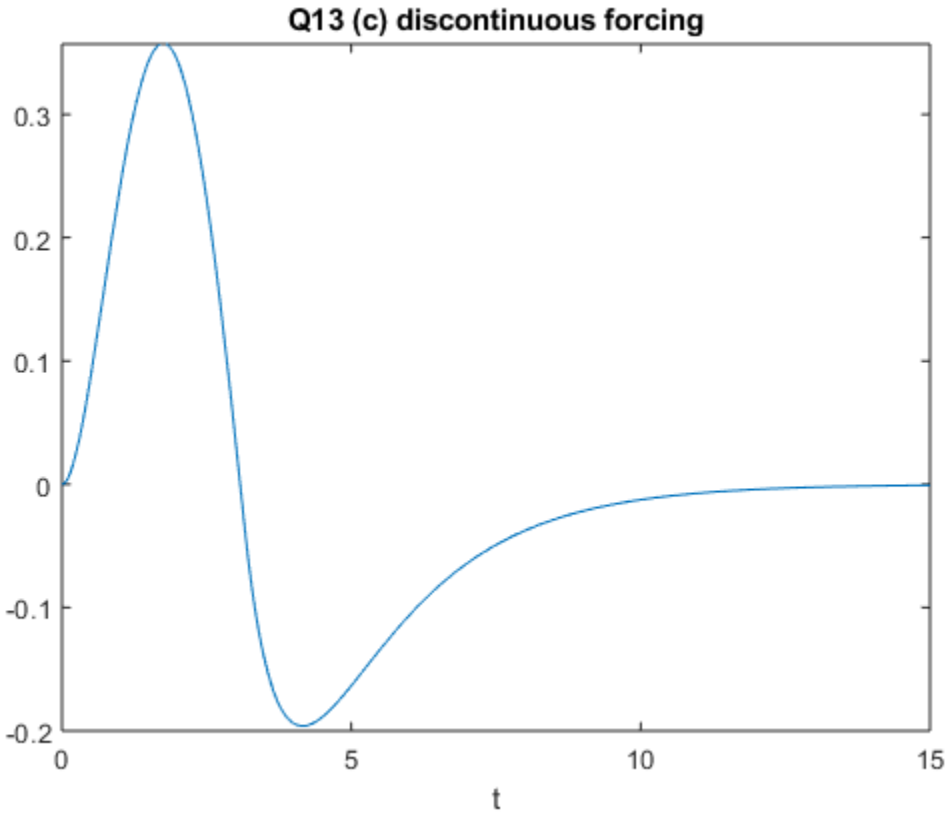
```
dsolve('D2y+2*Dy+2*y=0')
% As t approaches infinity, the behavior of solutions is that they
 approach 0.
% Since the characteristic roots are complex with negative real parts,
 the
% homogeneous equation will decay to zero in an oscillatory manner.
% The inhomogeneous equation will remain at 0 for its longterm
 behavior.


ans =

C12*exp(-t)*cos(t) + C13*exp(-t)*sin(t)
```

QUESTION 2

13 (c)

```
syms s t Y;
g= cos(t)+ (0-cos(t))*heaviside(t-pi);
G=laplace(g,t,s);
Y1=s*Y-0;
```

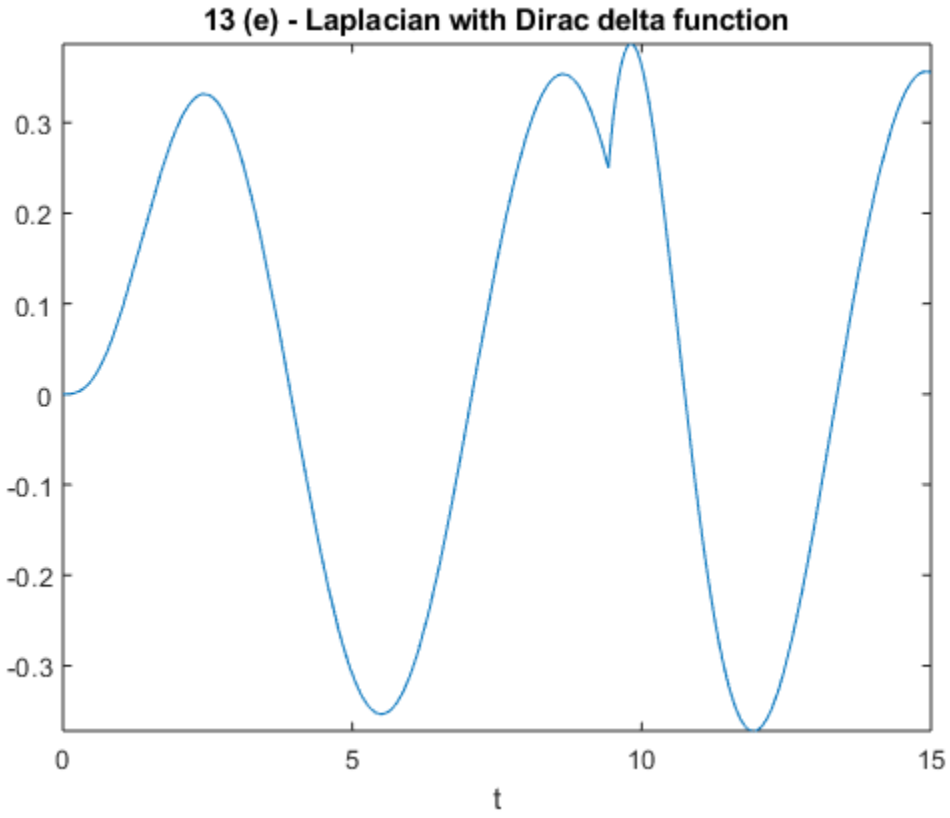
```
Y2=s*Y1-0;
EQN=solve(Y2+2*Y1+(4/5)*Y-G,Y);
sol=ilaplace(EQN,s,t);
fplot(sol,[0,15]); xlabel('t')
title('Q13 (c) discontinuous forcing');
% Due to the forcing, the solutions are switched on and off at a given
 particular time.
% Between 0 to pi, the solution looks like a cosine function. After t
 > pi,
% the solution is forced to zero.
```



13 (e)

```
figure;
syms s t Y
f= sin(t)+ dirac(t-3*pi);
F=laplace(f,t,s);
Y1=s*Y-0;
Y2=s*Y1-0;
EQN= solve(Y2+2*Y1+3*Y-F,Y);
sol=ilaplace(EQN,s,t);
fplot(sol,[0,15]); xlabel('t')
title('13 (e) - Laplacian with Dirac delta function');
% Due to the forcing, the solution looks like a sine function till t =
 3Pi.
% At that instant there is an impulse function that is applied. That
```

```
% impulse causes the amplitude to increase just after t>3pi. After t >
% 3pi,the amplitude has increased slightly more as compared to the
 original
% amplitude.
```



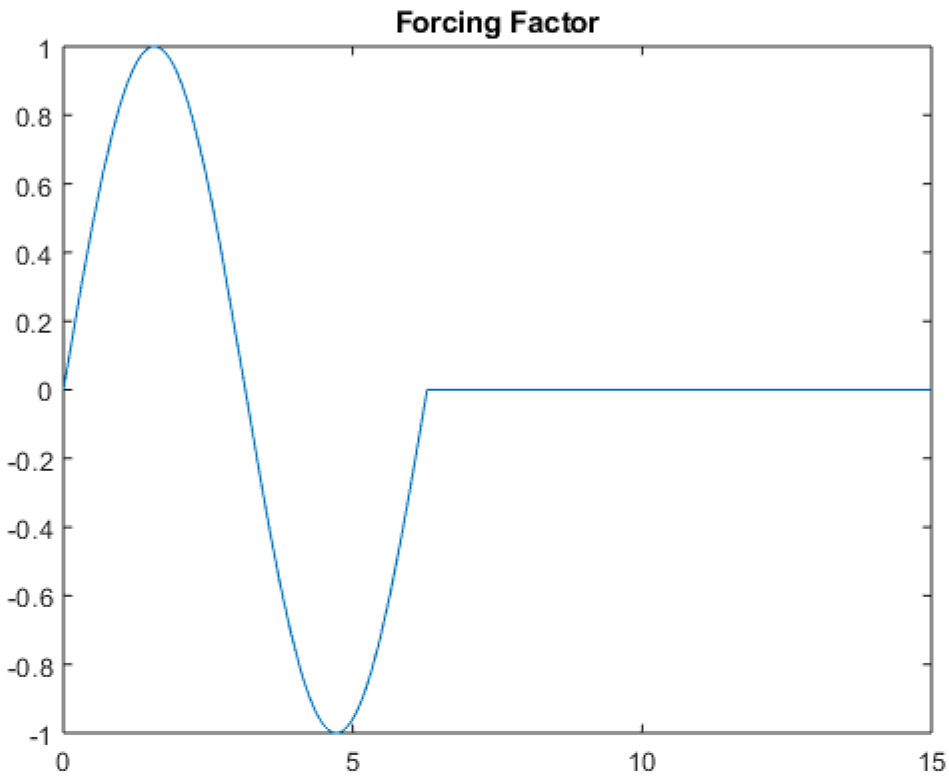
QUESTION 3

14 (a)

```
syms s t Y
f = ['sin(t) - heaviside(t - 2*pi)*sin(t - 2*pi)'];
equation = sym(['D(D(y))(t) + 4*y(t) = ' f]);
ltequation = laplace(equation, t, s);
newequation = subs(ltequation,
 {'laplace(y(t),t,s)', 'y(0)',subs(diff(y,t),t,0)},{Y, 0, 0});
ytrans = solve(newequation, Y);
y = ilaplace(ytrans, s ,t);
fplot(y, [0 15])
title 'Solution', axis auto
fplot(f, [0 15])
title 'Forcing Factor', axis auto
% Due to the forcing, the solution looks like a sine function till t =
 2Pi.
% the step function is turns on when t = 2pi and sets the solution to
 zero
```

```
% for t>2pi. The unit step function turns the sine function off when t
 =
% 2pi.
```

Warning: Support of character vectors will be removed in a future release. Character vectors can be used only for variable names and numbers. Instead, to create symbolic expressions first create symbolic variables using 'syms'. To evaluate character vectors and strings representing symbolic expressions, use 'str2sym'.
Warning: Support of character vectors will be removed in a future release. Character vectors can be used only for variable names and numbers. Instead, to create symbolic expressions first create symbolic variables using 'syms'. To evaluate character vectors and strings representing symbolic expressions, use 'str2sym'.
Warning: Support of character vectors will be removed in a future release. Character vectors can be used only for variable names and numbers. Instead, to create symbolic expressions first create symbolic variables using 'syms'. To evaluate character vectors and strings representing symbolic expressions, use 'str2sym'.
*Warning: Char input to fplot will be removed in a future release. Use fplot(@(t)sin(t)-heaviside(t-2.*pi).*sin(t-2.*pi)) instead.*



14 (c) attached towards the end

QUESTION 4

17 (a)

```
tic
syms y t
y='D2y+Dy+y=(t+1)^(3)*(exp(-t))*(cos(t))*(sin(3*t))';
solna =dsolve(y, 'y(0)=1', 'Dy(0)=0');
%sprintf('\n')
toc
```

```
Elapsed time is 12.742109 seconds.
```

17 (b)

```
tic
syms s t Y
eqn=sym('D(D(y))(t)+D(y)(t)+y(t)= (t+1)^(3)*(exp(-
t))*cos(t)*sin(3*t)');
lteqn=laplace(eqn,t,s);
neweqn=subs(lteqn,{'laplace(y(t),t,s)'...
'y(0)','D(y)(0)'},{Y,1,0});
ytrans=simplify(solve(neweqn,Y));
sprintf('\n')
y2=ilaplace(ytrans,s,t);
toc
```

```
Warning: Support of character vectors will be removed in a future
release. Character vectors can be used only for variable names and
numbers. Instead, to create symbolic expressions first create symbolic
variables using 'syms'. To evaluate character vectors and strings
representing symbolic expressions, use 'str2sym'.
Warning: Support of character vectors will be removed in a future
release. Character vectors can be used only for variable names and
numbers. Instead, to create symbolic expressions first create symbolic
variables using 'syms'. To evaluate character vectors and strings
representing symbolic expressions, use 'str2sym'.
Warning: Support of character vectors will be removed in a future
release. Character vectors can be used only for variable names and
numbers. Instead, to create symbolic expressions first create symbolic
variables using 'syms'. To evaluate character vectors and strings
representing symbolic expressions, use 'str2sym'.
Warning: Support of character vectors will be removed in a future
release. Character vectors can be used only for variable names and
numbers. Instead, to create symbolic expressions first create symbolic
variables using 'syms'. To evaluate character vectors and strings
representing symbolic expressions, use 'str2sym'.

ans =
```

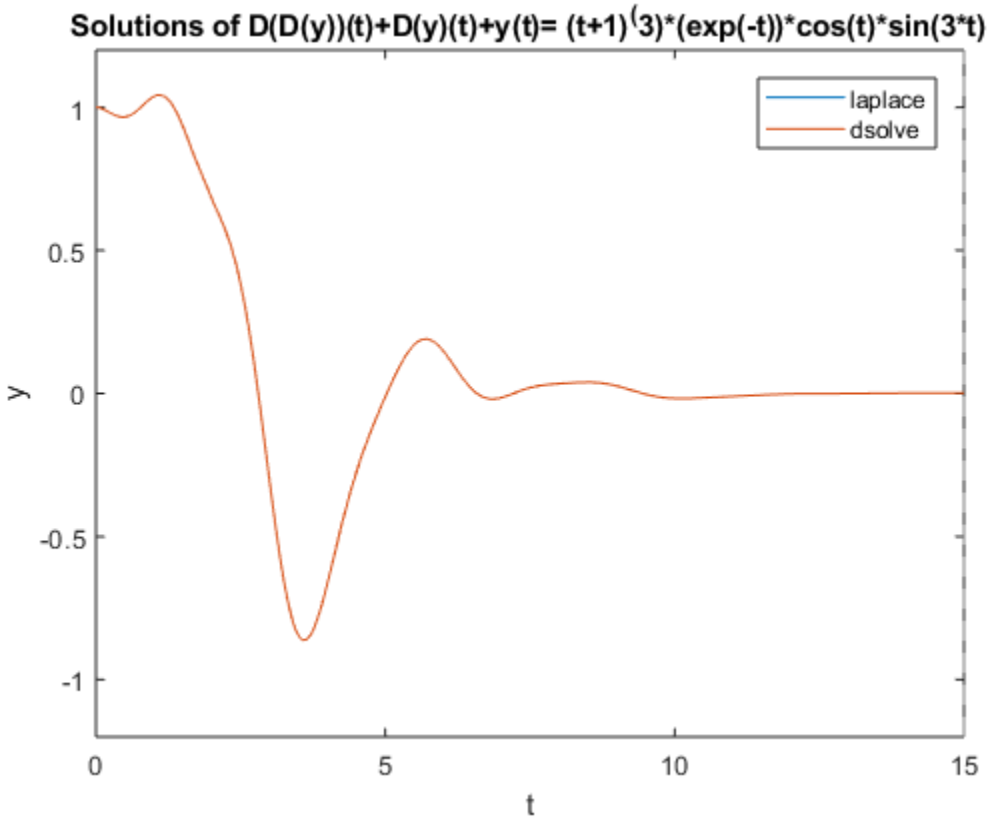
```
    '
     '
```

Elapsed time is 1.405361 seconds.

17 (c)

```
figure
fplot(y2,[0 15])
hold on
fplot(solna,[0 15])
hold off
title('Solutions of D(D(y))(t)+D(y)(t)+y(t)= (t+1)^(3)*(exp(-
t))*cos(t)*sin(3*t)')
xlabel 't'
ylabel 'y'
axis([0 15 -1.2 1.2])
legend ('laplace','dsolve')
% Both the graphs are the same
```

Warning: Finite sets ('DOM_SET') not supported. Using element 't = 0' instead.



Published with MATLAB® R2017b