Table of Contents

Eric Wan - ezw23@drexel.edu - Lab7

```
%{
Solving IVP
y'' + y' - 6y = exp(-t) * cos(t)
y(0) = 1, y'(0) = 1
%}
```

Part A

```
%{
LaPlace Transform
y'' + y' - 6y = exp(-t) * cos(t)
a[ s^2Y(s) - sy(0) - y'(0) ] + b[ sY(s) - y(0) ] + c[ Y(s) ] = U(s)
s^2Y(s) - sy(0) - y'(0) + sY(s) - y(0) - 6[ Y(s) ] = U(s)
s^2Y(s) + sY(s) - 6Y(s) - sy(0) - y'(0) - y(0) = U(s)
(s^2 + s - 6)Y(s) - sy(0) - y'(0) - y(0) = U(s)
(s-2)(s+3)Y(s) - s - 1 - 1 = U(s)
F(s) = 1/(s+1) * s/(s^2 + 1)
(s-2)(s+3)Y(s) - s - 2 = 1/(s+1) * s/(s^2 + 1)
Y(s) = [1/(s+1) * s/(s^2 + 1) + s + 2] / (s-2)(s+3)
Y(s) = s/[(s+1)(s^2 + 1)] + s/[(s-2)(s+3)] + 2/[(s-2)(s+3)]
%}
```

Part B

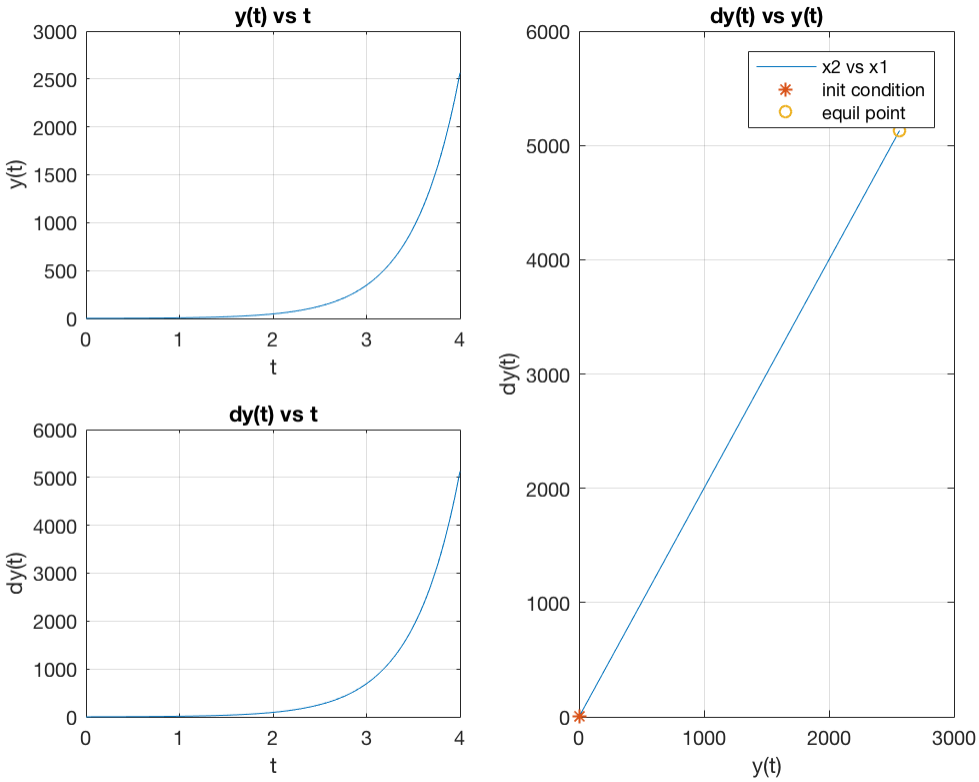
```
syms t s
u = exp(-t) * cos(t); % u(t)
U = s/((s+1)*(s^2 + 1)) + s/((s-2)*(s+3)) + 2/((s-2)*(s+3)); % Y(S)
Us = laplace(u, t, s) % U(s)
yt = ilaplace(U, s, t) % y(t)
```

```
Us =

(s + 1)/((s + 1)^2 + 1)


yt =

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

Part C

```
xI = [1; 1]; % init conditions
tSpan = 0:0.01:4; % span of T
type NumericSolution % print IVP function
[tode, yode] = ode45(@NumericSolution, tSpan, xI); % ode45
x1 = yode(:,1); % setting x1
x2 = yode(:,2); % setting x2
figure

% plotting x = y(t) vs t
subplot(2, 2, 1)
plot(tode, x1)
title('y(t) vs t')
xlabel('t')
ylabel('y(t)')
grid on
hold on

% plotting x2 vs t
subplot(2, 2, 3)
plot(tode, x2)
title('dy(t) vs t')
xlabel('t')
ylabel('dy(t)')
grid on
hold on

% plotting x2 vs x1
subplot(2, 2, [2 4])
plot(x1, x2)
title('dy(t) vs y(t)')
xlabel('y(t)')
ylabel('dy(t)')
grid on
hold on
plot(x1(1,1),x2(1,1),'*')
plot(x1(401,1),x2(401,1),'o')
legend('x2 vs x1', 'init condition', 'equil point')


function [xPrimes] = NumericSolution(t, xI)
%{
y'' + y' - 6y = exp(-t) * cos(t)
y(0) = 1, y'(0) = 1

x'(t) = Ax(t) + Bf(t)
x1 = y, x2 = y'
x1' = 0 * x1 + 1 * x2
x2' + 1 * x2 + 6 * x1 = exp(-t) * cos(t)
x2' = exp(-t) * cos(t) + 6 * x1 - 1 * x2
[x1'; x2'] = [0 1; 6 -1] * [x1; x2] + [0; exp(-t)*cos(t)]
```

```
%}

xPrimes = [0 1; 6 -1] * xI + [0; exp(-t)*cos(t)];

end
```



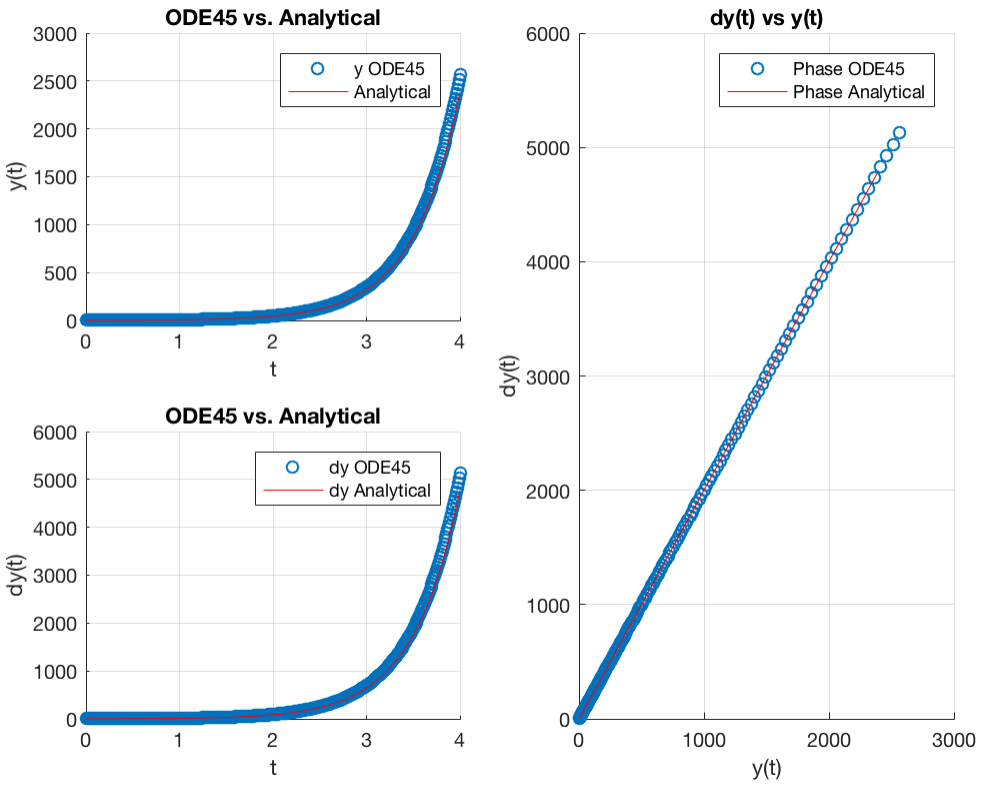
Part D

```
y = matlabFunction(yt) % matlabFunction of y(t) ilaplace of Y(s)
dy = matlabFunction(diff(yt)) % matlabFunction of dy(t)


y =

  function_handle with value:

    @(t)exp(-
t).*(-1.0./2.0)+exp(t.*2.0).*(4.0./5.0)+exp(t.*-3.0).*(1.0./5.0)+cos(t).*(1.0./2.0


dy =

  function_handle with value:
```

```
    @(t)exp(-t).*(1.0./2.0)+exp(t.*2.0).*(8.0./5.0)-
exp(t.*-3.0).*(3.0./5.0)+cos(t).*(1.0./2.0)-sin(t).*(1.0./2.0)
```

Part E

```
figure
% plotting ODE45 vs Analytical y vs t
subplot(2, 2, 1)
grid on
hold on
plot(tode, x1, 'o') % ode45
plot(tSpan, y(tSpan), 'r') % analytical
title('ODE45 vs. Analytical')
xlabel('t')
ylabel('y(t)')
legend('y ODE45', 'Analytical')

% plotting ODE45 vs Analytical dy vs t
subplot(2, 2, 3)
grid on
hold on
plot(tode, x2, 'o') % ode45
plot(tSpan, dy(tSpan), 'r') % analytical
title('ODE45 vs. Analytical')
xlabel('t')
ylabel('dy(t)')
legend('dy ODE45', 'dy Analytical')

% plotting ODE45 vs Analytical
subplot(2, 2, [2 4])
grid on
hold on
plot(x1, x2, 'o') % ode45
plot(y(tSpan), dy(tSpan), 'r') % analytical
title('dy(t) vs y(t)')
xlabel('y(t)')
ylabel('dy(t)')
legend('Phase ODE45', 'Phase Analytical')
```



Published with MATLAB® R2017a