Lab 7: Laplace Workshop

Summer 2022

Part A: Definition of the Laplace Transform.

Given a function $f(t)$ in the time domain, its one-sided Laplace Transform is defined by the following integral:

$$L\{f(t)\} = \int_0^\infty e^{-st} f(t)dt$$

Pierre-Simon Laplace (1749-1827)

The function e^{-st} is called the **kernel** or **nucleus** of the transform. There are many other useful integral transforms including the Fourier transform, Fourier sine and cosine transforms, Hartley transform, Mellin transform, Weierstrass transform, Hankel transform, Abel transform and the Hilbert transform; all defined using different kernels.

In this first section, we find a few Laplace transforms by directly evaluating the above integral.

Example: Show that the Laplace transform of the constant function $f(t) = 1$ is: $L\{1\} = \frac{1}{s}$
Directly evaluate the above integral. Tip: Use **inf** to denote infinity in the limits of integration.

```
syms s t;  assume(real(s)>0)
f = 1;
L = int( exp(-s*t) * f, 0, inf)
```

L = 1/s

You can see the answer agrees completely with the provided Table of Laplace Transforms.
You can **clear** the assumptions about a variable using: >> assume(s,'clear') % Clear any assumptions about s.

Question 1: Use the same approach, to find each of the following Laplace transforms.
Do each problem in its own section in your MATLAB file for this week's lab.

a. $f(t) = t$

$L\{t\} =$

b. $f(t) = t^2$

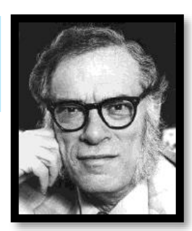
$L\{t^2\} =$

c. $f(t) = 3e^{5t}$

$L\{3e^{5t}\} =$

Record answers in the boxes above.

"Your assumptions are your windows on the world. Scrub them off every once in a while, or the light won't come in." — Isaac Asimov



You probably noticed that the answer to 1c, looks odd and does not match your Table of Laplace Transforms but instead expresses the answer using a **piecewise** function. That is because we have not told MATLAB enough information about the variable *s*. Above, we only said it was a "symbol", and that real(s) was positive.
Add the following assumption after declaring s and t to be symbols. >> `assume(s>5)` or `assume(real(s)>5)`
Repeat 1c and use this improved response as your answer in the above box.
Read the help on the assume command.
>> help assume

Now **clear your assumption** and verify the original "ugly" answer reappears.
>> assume(s,'clear')
A good reference on MATLAB assumptions is here.
http://www.mathworks.com/help/symbolic/assumptions-for-symbolic-objects.html#brvhirb-1

Question 2: Find each of the following Laplace transforms. Each question will require suitable assumptions about the variables to reproduce the form given in the Table of Laplace Transforms. Be sure to clear the variables and avoid any previous assumptions corrupting each new problem. Do each problem in its own section. Note each problem also has a new symbol such as *n* or *a*.

2a. $f(t) = t^n$

2b. $f(t) = \sin(at)$

2c. $f(t) = e^{at}$

$L\{t^n\} =$	$L\{\sin(at)\} =$	$L\{e^{at}\} =$

Record answers in the boxes above. Also include any assumptions you needed to present the answer in the "clean" form seen in the Tables. **Tip:** For positive integer arguments: $\text{gamma}(n+1) = n!$

Question 3: Built-in laplace() command

The good news is that MATLAB has a built-in command named laplace(), so you won't need to find these transforms using the defining integral, which is best used for demonstrating the fundamental properties of the transform. But be sure you can also use the tables.

Read the help info for the **laplace** command.
>> help laplace

Examples:

```
>> syms t; laplace(t^2)        ans = 2/s^3

>> syms n; assume(n>-1)
   laplace(t^n)                ans = gamma(n + 1)/s^(n + 1)
```

Find each of the following Laplace transforms using the built-in laplace() command.
Below, $u(t)$ denotes the unit step function.
Tip for 3b: You can enter the unit step function $u(t)$ as heaviside(t) or even better, declare it as an **anonymous function** using: >> u = @ (t) heaviside(t)

3a. $f(t) = 3\cosh 5t$

3b. $f(t) = (t-3)^2 \cdot u(t-3)$

3c. $f(t) = \sqrt{t}$

$L\{3\cosh 5t\} =$	$L\{(t-3)^2 \cdot u(t-3)\} =$	$L\{\sqrt{t}\} =$

Warning: You can't apply laplace() to a constant. >> laplace(1) → **Error!** The function expects a symbolic expression.
Try this instead. >> F = laplace(sym(1))

Question 4: Built-in ilaplace() command

Just as important as the Laplace transform, is its **inverse** transform.
In MATLAB, this is found using the ilaplace() command.

Read the help info for the **ilaplace** command.
>> help ilaplace

Examples: >> help ilaplace

>> syms s; ilaplace(1/(s-1)) ans = exp(t)

>> syms s w ; ilaplace(s/(s^2 + w^2)) ans = cos(t*w)

>> syms f(t) ; ilaplace(laplace(f(t))) ans = f(t)

Find the inverse Laplace transform for each of the following functions defined in the *s*-domain.

4a. $F(s) = 1$ See hint below.

4b. $F(s) = \frac{5s+8}{s^2+16}$

4c. $F(s) = \frac{1}{s^{3/2}}$

$f(t) =$	$f(t) =$	$f(t) =$

Hint: You may have to enter *F*(*s*) = 1 as sym(1) for 4a. Otherwise you will see this error message.

>> ilaplace(1) % Does not work!

Undefined function 'ilaplace' for input arguments of type 'double'.

Hint: You can print an "ugly" mathematical answer f in nicer format using: >> pretty(f)

Part B: Partial fraction expansions.

Partial fraction expansions are absolutely necessary so you can find inverse Laplace transforms using the standard Laplace tables. Fortunately, MATLAB now has the built-in command partfrac(). Let's see how that works now.
For more help on partfrac see: https://www.mathworks.com/help/symbolic/partfrac.html

```
Alert!  partfrac is a fairly recent command.
Students with an older version of MATLAB may
need to use the following command instead.
feval(symengine, 'partfrac', F)
```

Examples: >> help partfrac

>> syms s; partfrac((s+2)/(s^2 - 2*s)) ans = 2/(s - 2) - 1/s

Question 5: Find the partial fraction expansion for each of the following functions in the *s*-domain. Use partfrac().

5a. $F(s) = \frac{16}{s^2-8s}$

5b. $F(s) = \frac{9s^2-52s+72}{(s-2)(s-3)(s-4)}$

5c. $F(s) = \frac{3s^2-14s+20}{(s-3)^3}$

Grading: TA will randomly pick one part from each of questions 1 – 5 above and award 1 point if correct.

Part C: Solving a Differential Equation using the Laplace Transform.

Example: This same example appeared in the recitation notes.
Using the method of Laplace Transforms, find the solution to the linear differential equation:

DE: $y'' + y = 6\sin 2t$ and initial conditions: **IC:** $y(0) = 0, \quad y'(0) = 6$

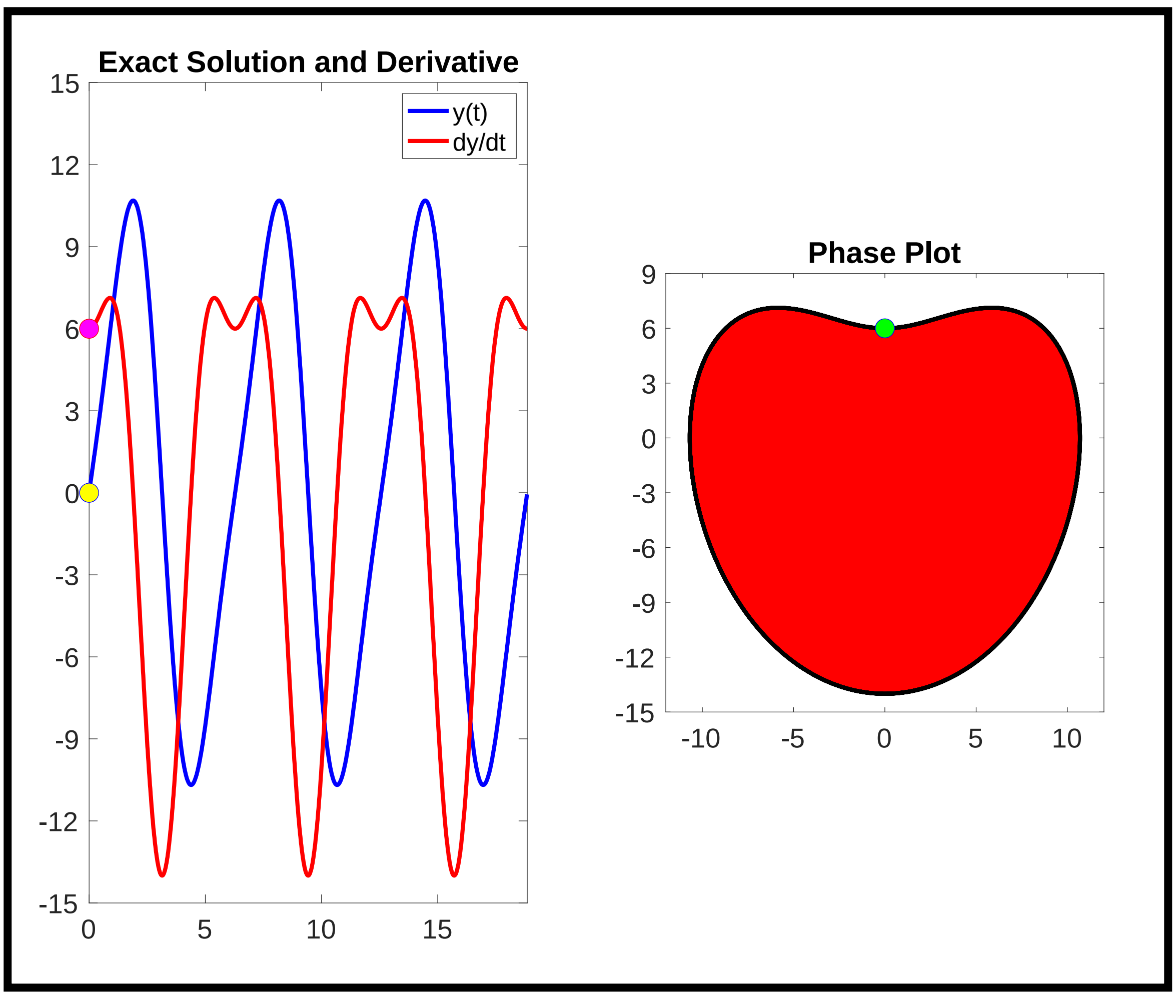
Question 6: First solve the DE exactly using dsolve.

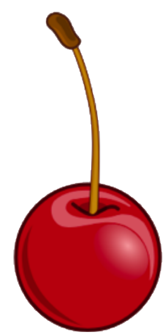
i. First enter the differential equation as usual, and find the exact solution using dsolve().
We will compare this later to the answer found using the Laplace method to confirm it gives the same answer.
Assign the exact solution to Y(t) and its derivative to DY(t) using matlabFunction(). Confirm both initial conditions are true. Record the exact solution below found using dsolve:

Question 6: The exact solution for $y(t)$ is:

$y(t) =$

Here's a component plot and the phase plot for the exact solution. (Not required, but good review for MATLAB Final). I filled the phase curve in **red** to resemble a heart or cherry or maybe an apple.





Laplace Transform Method for Solution

ii. Now let's solve the same DE using Laplace transforms: $y'' + y = 6\sin 2t$ **IC**: $y(0) = 0, \quad y'(0) = 6$
% Step-by-step solution. Study each step carefully!

```
%% Laplace Transform Method for Solution

% a. Define the necessary symbolic variables.
clear, clc
fprintf("Question 7: Solve a DE using the Laplace Transform\n")
syms s t Y  % Now Y(s) denotes the transform of the unknown function y(t).

% b. Find the Laplace transform of y'(t): Y1 = s Y - y(0)
% This is necessary, even though this term does not appear in the LHS
% of the differential equation.
y0 = 0;  dy0 = 6;  % Initial conditions
f = 6 * sin(2*t) % the forcing function

disp 'The transform of the derivative is:'
Y1 = s*Y - y0   % Add the initial value y(0)=y0 manually here.

% c. Find the Laplace transform of y''(t): Y2 = s Y1 - y'(0)
disp 'The transform of the double derivative is:'
Y2 = s*Y1 - dy0 % Add the initial value y'(0)=dy0 manually here.

% d. Find the Laplace transform F of the forcing term f(t) = 6*sin(2*t)
disp 'The transform F(s) of the forcing term f(t) is:'
F = laplace( f )

% e. Combine all the terms into the transform of the entire equation,
% which we will name LTofDE for Laplace Transform of DE.
% y'' + y = f(t)  with the initial conditions   y(0)=y0,  y'(0)=dy0
LTofDE = Y2 + Y == F

% f. Use solve to solve this algebraic equation for the unknown Y.
Sol =  solve(LTofDE, Y);
Y = matlabFunction(Sol); Y(s)
Y = partfrac(Y(s)) % express solution in partial fraction form

% g. Find the inverse Laplace transform of the solution:
sol = ilaplace(Sol,s,t);
y = matlabFunction(sol); y(t) % solution in the time domain
```

Question 7: Record **both** the solution $Y(s)$ in partial fraction form and the solution $y(t)$ in the time-domain that were just found using the Laplace technique here. Did you get the same answer for $y(t)$?

Question 7:
$Y(s) =$ ____ (must be in partial fraction form)

$y(t) =$ ____

Part D: Solve a new DE using the Laplace transform technique. (3 points)

The last three points will be earned by using code similar to that given above to solve the new differential equation:

The last three points will be earned by using code similar to that given above to solve the new differential equation:

$$\textbf{DE:}\quad y'' + y' + \tfrac{5}{4}y = 13 \cdot e^{-2t} \qquad \textbf{IC:}\ y(0) = 4,\ \ y'(0) = 2$$

Points 8-10: Solve this new DE using the Laplace technique and past these three answers below.

8. Give the result for the transformed solution $Y(s)$. If necessary, use simplifyFraction() so it is in the form of a ratio, with the numerator a quadratic and the denominator a cubic. A linear term can factor out of the cubic.

9. Find the partial fraction expansion for Y using partfrac.

10. Give the result for the time-domain solution $y(t)$.

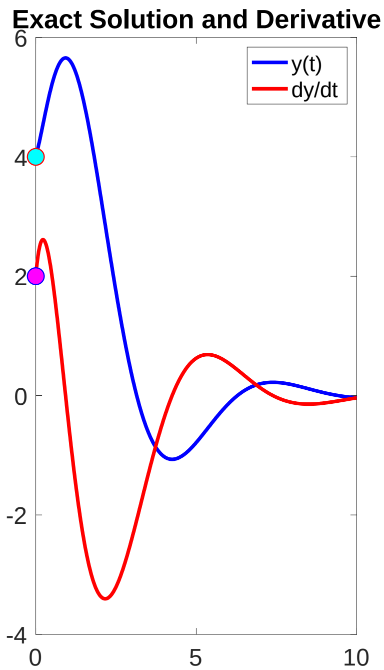
Questions 8-10:

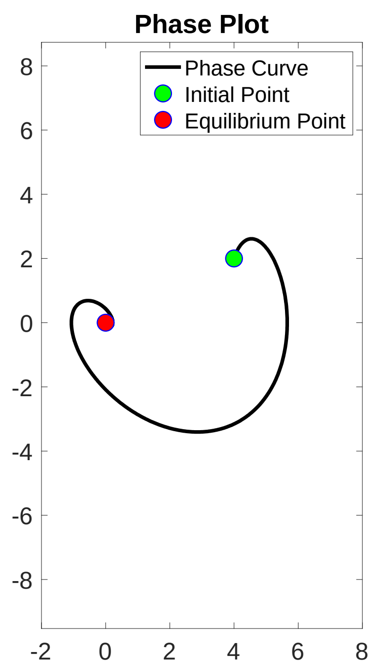
8: $Y(s) = \_\_\_$ ←Must be a quadratic over a cubic for points.

9: $Y(s)$ as a partial fraction $= \_\_\_$

10: $y(t) = \_\_\_$

Tip: You might want to confirm you are correct using dsolve() before submitting your work.
You can also improve its appearance using pretty(sol).
Component plot and phase plot are not required.





Ready to Submit?

Be sure all ten questions are answered. When your lab is complete, be sure to submit three files:

1. Your **completed Answer Template** as a PDF file
2. A copy of your **MATLAB Live Script**
3. A **PDF** copy of your **MATLAB Live Script** (Save-Export to PDF…)

The due date is the day after your lab section by **11:59pm** to receive full credit. You have one more day, to submit the lab (but with a small penalty), and then the window closes for good and your grade will be zero.