Table of Contents


```
%%%% Differential Equations Homework 4 - Romeo Perlstein %%%%

%%%% Useful notation:
%%%% heaviside function u(t), u_c(t-c)
close all
close all force
```

Question E14

a)

y'' + 4y = (1 - u(t-2pi))sin(t), y(0) = 0, y'(0)=0

```
syms sA yA(tA) YA
og_eq_A = diff(yA, 2) + 4*yA == (1-heaviside(tA-(2*pi)))*sin(tA) % The OG
 equation
laplace_eq_A = laplace(og_eq_A, tA, sA) % The equation, transformed using the
 Laplace transformation

better_laplace_eq_A = subs(laplace_eq_A, [yA(0), subs(diff(yA(tA), tA), tA,
 0), laplace(yA(tA), tA, sA)], [0, 0, YA])
% uhhh not going to lie this is just what the textbook suggested. It
% apparently makes it easier to solve for y? my understanding is that,
% since eq1 had a bunch of subs values in it, this new format allows for
% us to simply do all that subbing in one line.

solved_laplace_eq_A = simplify(solve(better_laplace_eq_A, YA)) % solve for da
 equation
og_vspace_eq_A = ilaplace(solved_laplace_eq_A) % back to the original vector
 space and solve for it

%%%% Time to do this several more times!
```

b)

y'' + 6y' + 8y = h(t)

```
syms hB yB(tB) YB sB
```

```
hB = heaviside(tB-5) + heaviside(tB-10)*(1-0) + heaviside(tB-10)*(-1-2) % the
 piecewise equation
og_eq_B = diff(yB, 2) + 6*diff(yB, 1) + 8*yB == hB % the OG equation
laplace_eq_B = laplace(og_eq_B, tB, sB) % Laplace transform of eq B

% even though I believe the previous step was unecessary, I will be leaving
% it in
better_laplace_eq_B = subs(laplace_eq_B, [laplace(yB(tB), tB, sB), yB(0),
 subs(diff(yB(tB), tB), tB, 0)], [YB, 0, 2]) % different from equation A!
solved_laplace_eq_B = (simplify(solve(better_laplace_eq_B, YB)))
og_vspace_eq_B = ilaplace(solved_laplace_eq_B)
```

c)

y'' + 4y = dirac(t-3pi), y(0)=1, y'(0)=0

```
syms yC(tC) YC sC
og_eq_C = diff(yC, 2) + 4*yC == dirac(tC - (3*pi)) %the original equation C
laplace_eq_C = laplace(og_eq_C, tC, sC) % laplace transform of eq C

better_laplace_eq_C = subs(laplace_eq_C, [laplace(yC(tC), tC, sC), yC(0),
 subs(diff(yC(tC), tC), tC, 0)], [YC, 1, 0])
solved_laplace_eq_C = simplify(solve(better_laplace_eq_C, YC))
og_vspace_eq_C = ilaplace(solved_laplace_eq_C)
```

d)

y'' + y = driac(t-2) - dirac(t-8)

```
syms yD(tD) YD sD
og_eq_D = diff(yD, 2) + yD == (dirac(tD - 2)) - (dirac(tD - 8))
laplace_eq_D = laplace(og_eq_D, tD, sD)

better_laplace_eq_D = subs(laplace_eq_D, [laplace(yD(tD), tD, sD), yD(0),
 subs(diff(yD(tD), tD), tD, 0)], [YD, 0, 0])
solved_laplace_eq_D = simplify(solve(better_laplace_eq_D, YD))
og_vspace_eq_D = ilaplace(solved_laplace_eq_D)
```

```
og_eq_A(tA) =

4*yA(tA) + diff(yA(tA), tA, tA) == -sin(tA)*(heaviside(tA - 2*pi) - 1)


laplace_eq_A =

sA^2*laplace(yA(tA), tA, sA) - sA*yA(0) - subs(diff(yA(tA), tA), tA,
 0) + 4*laplace(yA(tA), tA, sA) == (exp(2*pi*sA) - 1)/(exp(2*pi*sA) +
 sA^2*exp(2*pi*sA))


better_laplace_eq_A =
```

```
YA*sA^2 + 4*YA == (exp(2*pi*sA) - 1)/(exp(2*pi*sA) + sA^2*exp(2*pi*sA))


solved_laplace_eq_A =

(exp(-2*pi*sA)*(exp(2*pi*sA) - 1))/((sA^2 + 1)*(sA^2 + 4))


og_vspace_eq_A =

sin(t)/3 - sin(2*t)/6 + heaviside(t - 2*pi)*(sin(2*t)/6 - sin(t)/3)


hB =

heaviside(tB - 5) - 2*heaviside(tB - 10)


og_eq_B(tB) =

8*yB(tB) + 6*diff(yB(tB), tB) + diff(yB(tB), tB, tB) == heaviside(tB - 5) -
 2*heaviside(tB - 10)


laplace_eq_B =

6*sB*laplace(yB(tB), tB, sB) - 6*yB(0) - sB*yB(0) + sB^2*laplace(yB(tB),
 tB, sB) - subs(diff(yB(tB), tB), tB, 0) + 8*laplace(yB(tB), tB, sB) ==
 exp(-5*sB)/sB - (2*exp(-10*sB))/sB


better_laplace_eq_B =

YB*sB^2 + 6*YB*sB + 8*YB - 2 == exp(-5*sB)/sB - (2*exp(-10*sB))/sB


solved_laplace_eq_B =

(2*sB + exp(-5*sB) - 2*exp(-10*sB))/(sB*(sB^2 + 6*sB + 8))


og_vspace_eq_B =

exp(-2*t) - exp(-4*t) + heaviside(t - 5)*(exp(20 - 4*t)/8 - exp(10 - 2*t)/4 +
 1/8) - 2*heaviside(t - 10)*(exp(40 - 4*t)/8 - exp(20 - 2*t)/4 + 1/8)


og_eq_C(tC) =

4*yC(tC) + diff(yC(tC), tC, tC) == dirac(tC - 3*pi)


laplace_eq_C =
```

```
sC^2*laplace(yC(tC), tC, sC) - sC*yC(0) - subs(diff(yC(tC), tC), tC, 0) +
 4*laplace(yC(tC), tC, sC) == exp(-3*pi*sC)
```

```
better_laplace_eq_C =

YC*sC^2 - sC + 4*YC == exp(-3*pi*sC)
```

```
solved_laplace_eq_C =

(sC + exp(-3*pi*sC))/(sC^2 + 4)
```

```
og_vspace_eq_C =

cos(2*t) + (sin(2*t)*heaviside(t - 3*pi))/2
```

```
og_eq_D(tD) =

diff(yD(tD), tD, tD) + yD(tD) == dirac(tD - 2) - dirac(tD - 8)
```

```
laplace_eq_D =

sD^2*laplace(yD(tD), tD, sD) - sD*yD(0) - subs(diff(yD(tD), tD), tD, 0) +
 laplace(yD(tD), tD, sD) == exp(-2*sD) - exp(-8*sD)
```

```
better_laplace_eq_D =

YD*sD^2 + YD == exp(-2*sD) - exp(-8*sD)
```

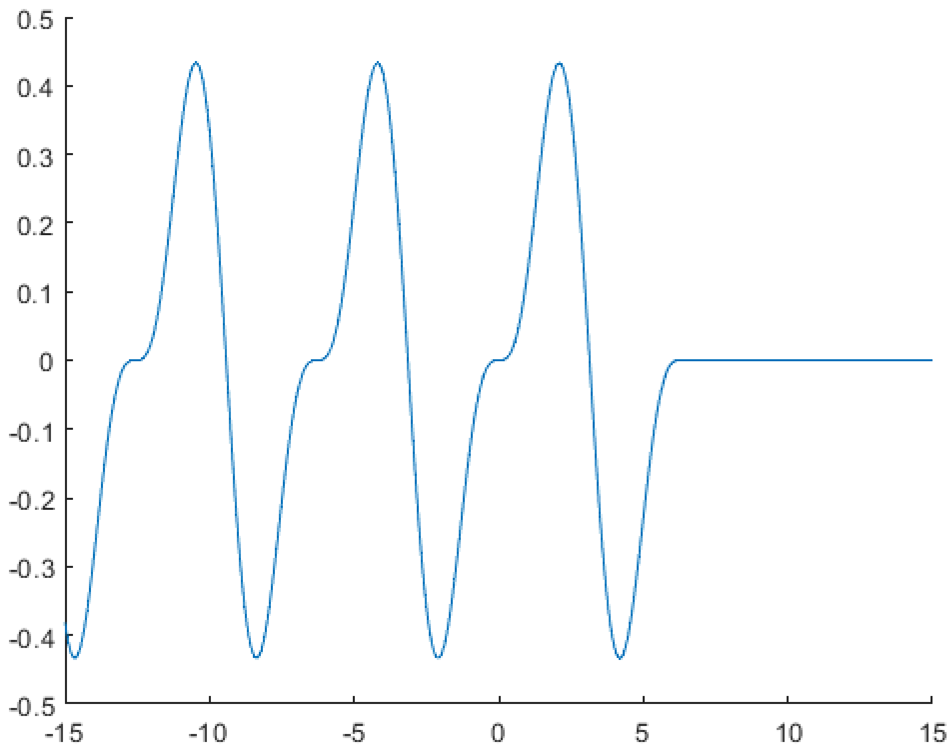
```
solved_laplace_eq_D =

(exp(-8*sD)*(exp(6*sD) - 1))/(sD^2 + 1)
```

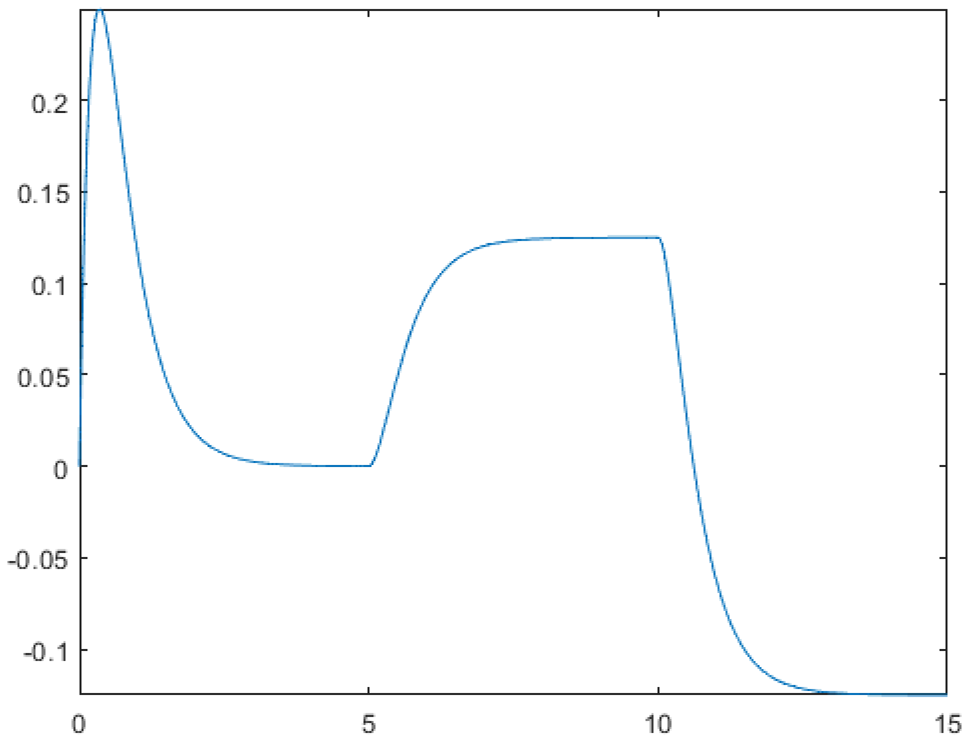
```
og_vspace_eq_D =

heaviside(t - 2)*sin(t - 2) - heaviside(t - 8)*sin(t - 8)
```

Plotting The Results

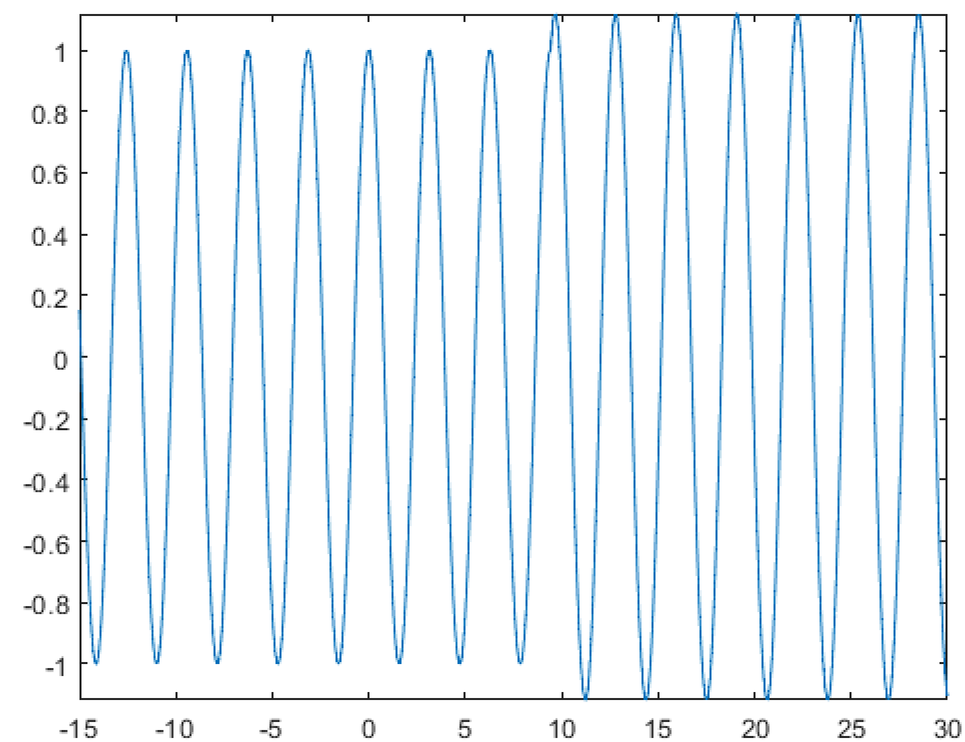
Here, I plot the results from A, B, C, and D on different intervals to show the full form of the grap

```
hold on
fplot(og_vspace_eq_A, [-15,15])
figure
fplot(og_vspace_eq_B, [0, 15])
figure
fplot(og_vspace_eq_C, [-15, 30])
figure
fplot(og_vspace_eq_D, [-15, 15])
```

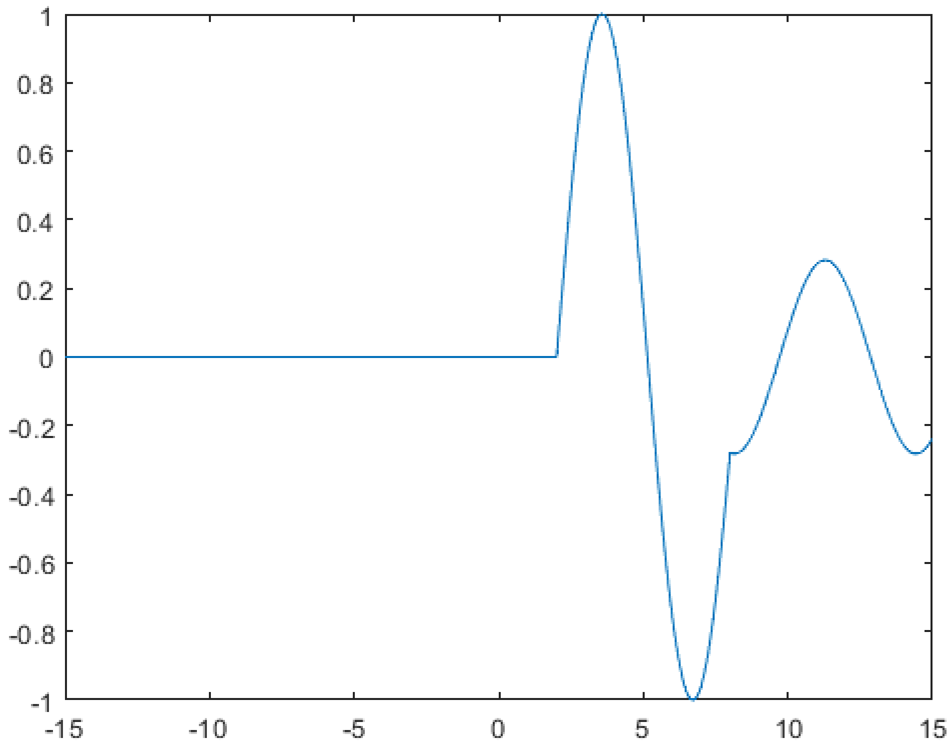




0.2
0.15
0.1
0.05
0
-0.05
-0.1
0
5
10
15



1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-15
-10
-5
0
5
10
15
20
25
30



Published with MATLAB® R2022b