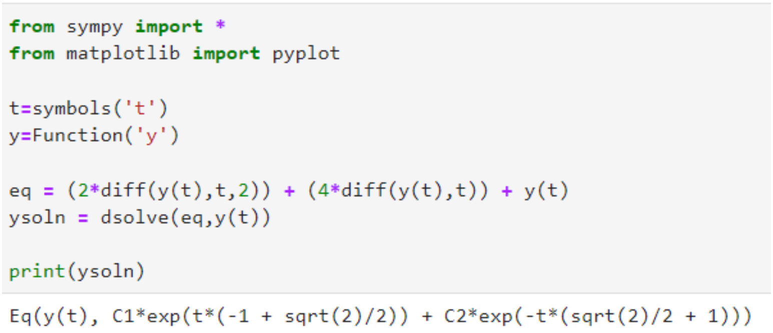
Programming Assignment 2 Solutions

Problem 1: Use the dsolve command in Python to solve

$$2y'' + 4y' + y = 0.$$

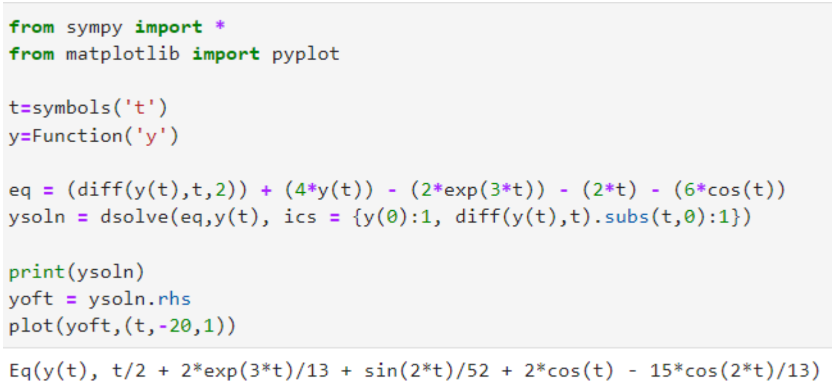


```
from sympy import *
from matplotlib import pyplot

t=symbols('t')
y=Function('y')

eq = (2*diff(y(t),t,2)) + (4*diff(y(t),t)) + y(t)
ysoln = dsolve(eq,y(t))

print(ysoln)
```

```
Eq(y(t), C1*exp(t*(-1 + sqrt(2)/2)) + C2*exp(-t*(sqrt(2)/2 + 1)))
```

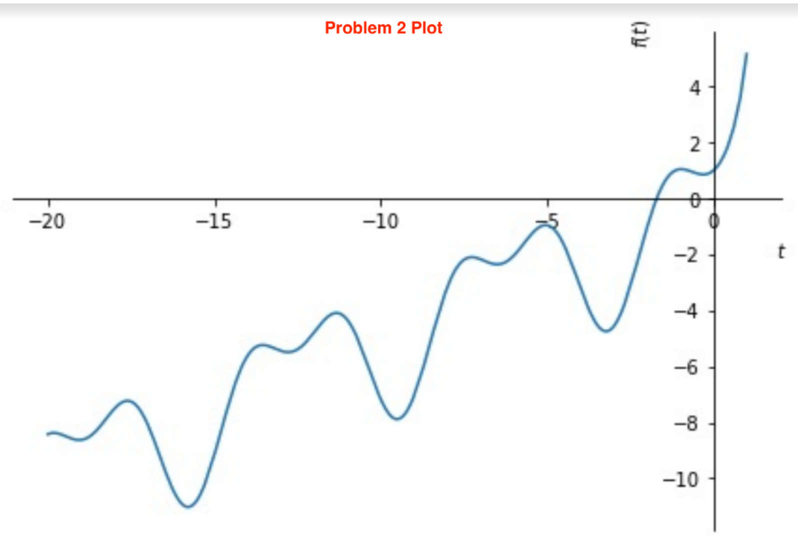
Problem 2: Use the dsolve command in Python to solve and plot for $-20 \leq t \leq 1$

$$\begin{cases} y'' + 4y = 2e^{3t} + 2t + 6\cos(t) \\ y(0) = 1 \\ y'(0) = 1. \end{cases}$$



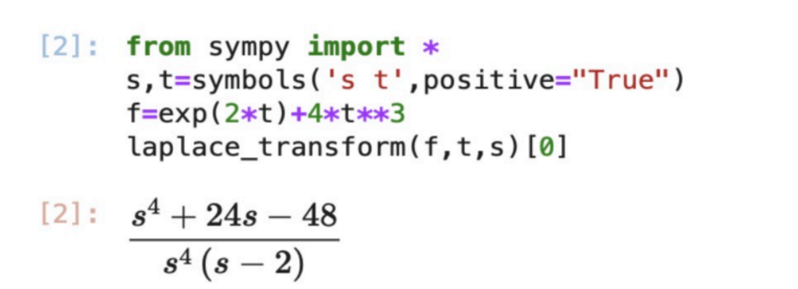
```
from sympy import *
from matplotlib import pyplot

t=symbols('t')
y=Function('y')

eq = (diff(y(t),t,2)) + (4*y(t)) - (2*exp(3*t)) - (2*t) - (6*cos(t))
ysoln = dsolve(eq,y(t), ics = {y(0):1, diff(y(t),t).subs(t,0):1})

print(ysoln)
yoft = ysoln.rhs
plot(yoft,(t,-20,1))
```

```
Eq(y(t), t/2 + 2*exp(3*t)/13 + sin(2*t)/52 + 2*cos(t) - 15*cos(2*t)/13)
```



Problem 3: Use Python to find the Laplace transform of

$$e^{2t} + 4t^3.$$

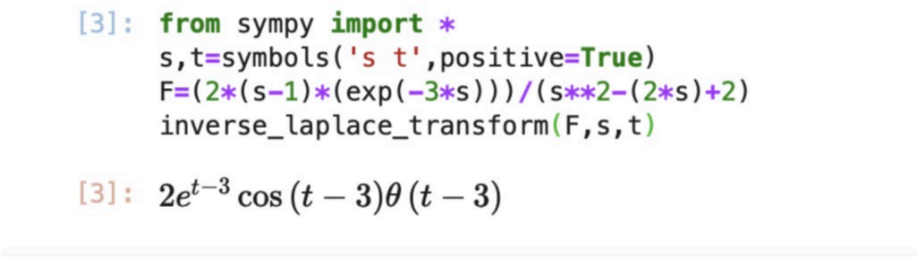


```
[2]: from sympy import *
     s,t=symbols('s t',positive="True")
     f=exp(2*t)+4*t**3
     laplace_transform(f,t,s)[0]
```

[2]: $$\frac{s^4 + 24s - 48}{s^4\,(s - 2)}$$

Problem 4: Use Python to find a function whose Laplace transform is

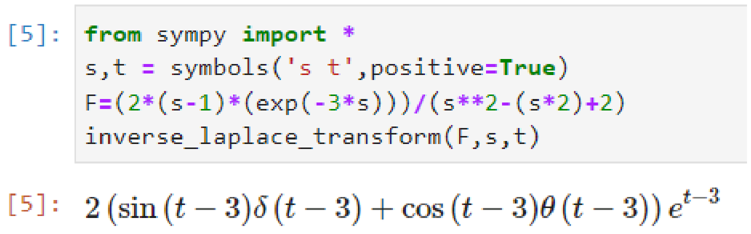
$$\frac{2(s-1)e^{-3s}}{s^2 - 2s + 2}.$$



```
[3]: from sympy import *
     s,t=symbols('s t',positive=True)
     F=(2*(s-1)*(exp(-3*s)))/(s**2-(2*s)+2)
     inverse_laplace_transform(F,s,t)
```

[3]: $$2e^{t-3}\cos\left(t-3\right)\theta\left(t-3\right)$$

ALTERNATE SOLUTION: The code may instead return the following, which is also correct:

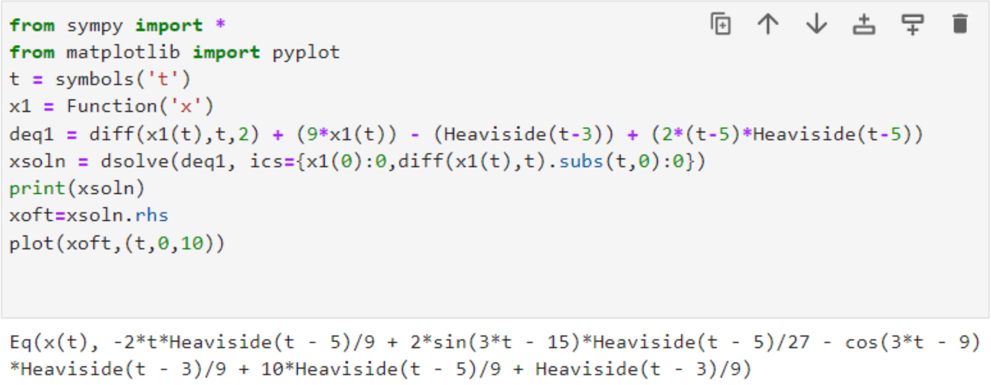


```
[5]: from sympy import *
     s,t = symbols('s t',positive=True)
     F=(2*(s-1)*(exp(-3*s)))/(s**2-(s*2)+2)
     inverse_laplace_transform(F,s,t)
```

[5]: $2\left(\sin\left(t-3\right)\delta\left(t-3\right)+\cos\left(t-3\right)\theta\left(t-3\right)\right)e^{t-3}$

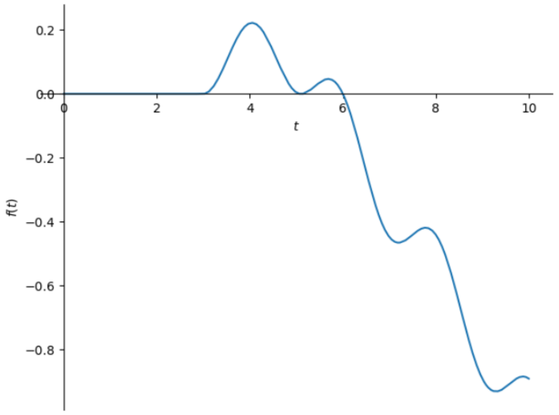
Problem 5: Use the dsolve command in Python to solve and plot for $0 \leq t \leq 10$

$$\begin{cases} y'' + 9y = u_3(t) - 2(t-5)u_5(t) \\ y(0) = 0 \\ y'(0) = 0. \end{cases}$$



```
from sympy import *
from matplotlib import pyplot
t = symbols('t')
x1 = Function('x')
deq1 = diff(x1(t),t,2) + (9*x1(t)) - (Heaviside(t-3)) + (2*(t-5)*Heaviside(t-5))
xsoln = dsolve(deq1, ics={x1(0):0,diff(x1(t),t).subs(t,0):0})
print(xsoln)
xoft=xsoln.rhs
plot(xoft,(t,0,10))
```

```
Eq(x(t), -2*t*Heaviside(t - 5)/9 + 2*sin(3*t - 15)*Heaviside(t - 5)/27 - cos(3*t - 9)
*Heaviside(t - 3)/9 + 10*Heaviside(t - 5)/9 + Heaviside(t - 3)/9)
```


0.2
0.0
−0.2
−0.4
−0.6
−0.8
f(t)
0
2
4
6
8
10
t