In [1]:
```
from sympy import *
```

NOTE: In addition to solving the IVPs, we will also graph the results we obtain to illustrate the behavior as a result of the discontinuous forcing functions.

And since we consider down as positive, I will multiply each solution by -1 when graphing it to help visualize the behavior

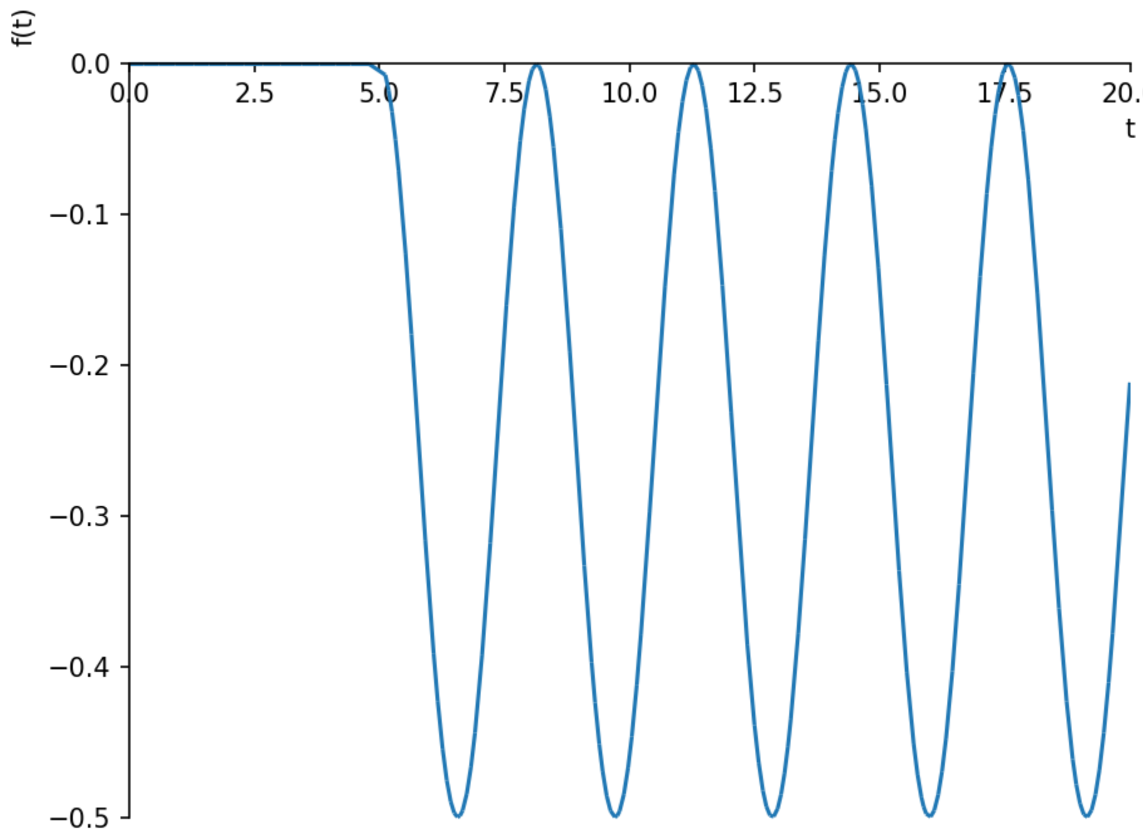
Example 1: y'' + 4y = u_5(t), y(0)=y'(0)=0

In [2]:
```
t=symbols('t')
y=Function('y')
g=Heaviside(t-5)
deq=diff(y(t),t,2)+4*y(t)-g
ysoln=dsolve(deq,y(t),ics={y(0):0,diff(y(t),t).subs(t,0):0})
print('The solution is',ysoln)
```

```
The solution is Eq(y(t), -cos(2*t - 10)*Heaviside(t - 5)/4 + Heaviside(t -
5)/4)
```

In [3]:
```
matplotlib notebook
```

In [4]:
```
t=symbols('t')
y=ysoln.rhs
plot(-1*y,(t,0,20))
```



Out[4]: <sympy.plotting.plot.Plot at 0x73088d0>

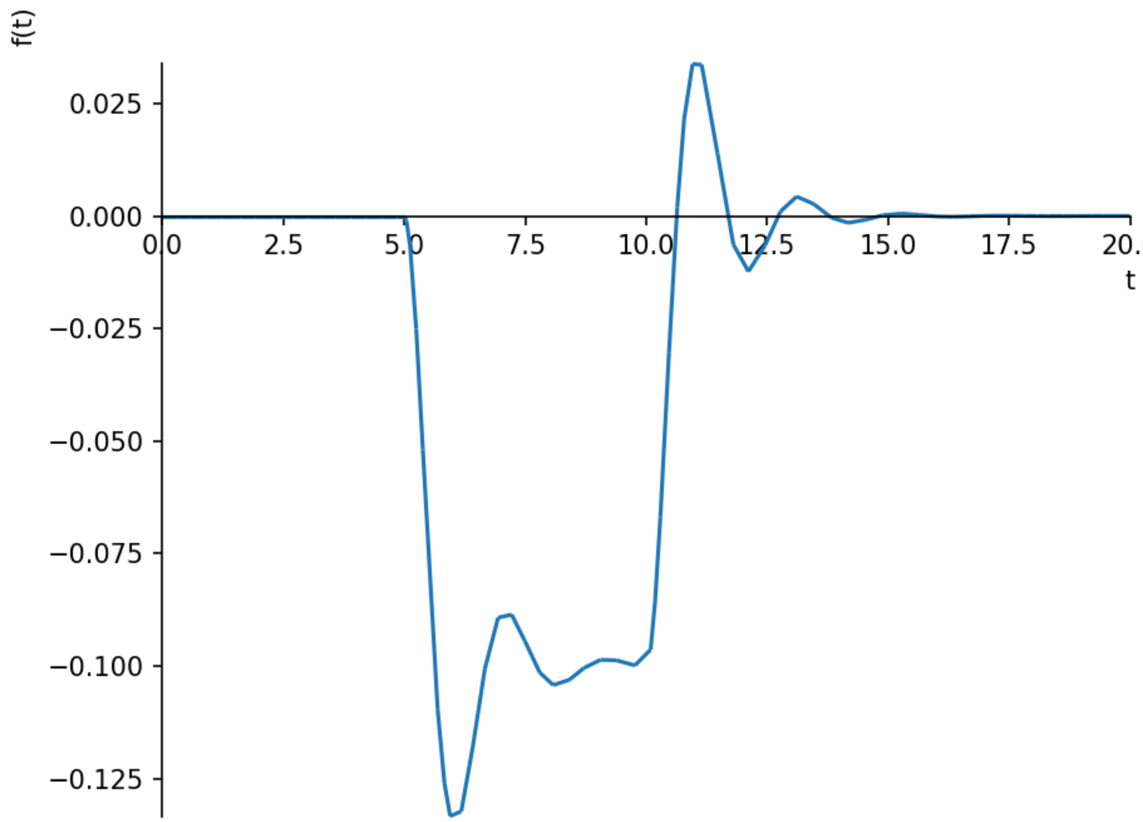
Example 2: y'' + 2y' + 10y = u_5(t) - u_10(t); y(0)=y'(0)=0

In [5]:
```
t=symbols('t')
y=Function('y')
g=Heaviside(t-5)-Heaviside(t-10)
deq=diff(y(t),t,2)+2*diff(y(t),t)+10*y(t)-g
ysoln=dsolve(deq,y(t),ics={y(0):0,diff(y(t),t).subs(t,0):0})
print('The solution is',ysoln)
```

```
The solution is Eq(y(t), (exp(10)*sin(3*t - 30)*Heaviside(t - 10)/30 - exp
(5)*sin(3*t - 15)*Heaviside(t - 5)/30 + exp(10)*cos(3*t - 30)*Heaviside(t -
10)/10 - exp(5)*cos(3*t - 15)*Heaviside(t - 5)/10)*exp(-t) - Heaviside(t -
10)/10 + Heaviside(t - 5)/10)
```

In [6]:
```
matplotlib notebook
```

In [7]:
```
t=symbols('t')
y=ysoln.rhs
plot(-1*y,(t,0,20))  #NOTE the changes in the solution at t=5 and t=10
```



Out[7]: <sympy.plotting.plot.Plot at 0x9af9f30>

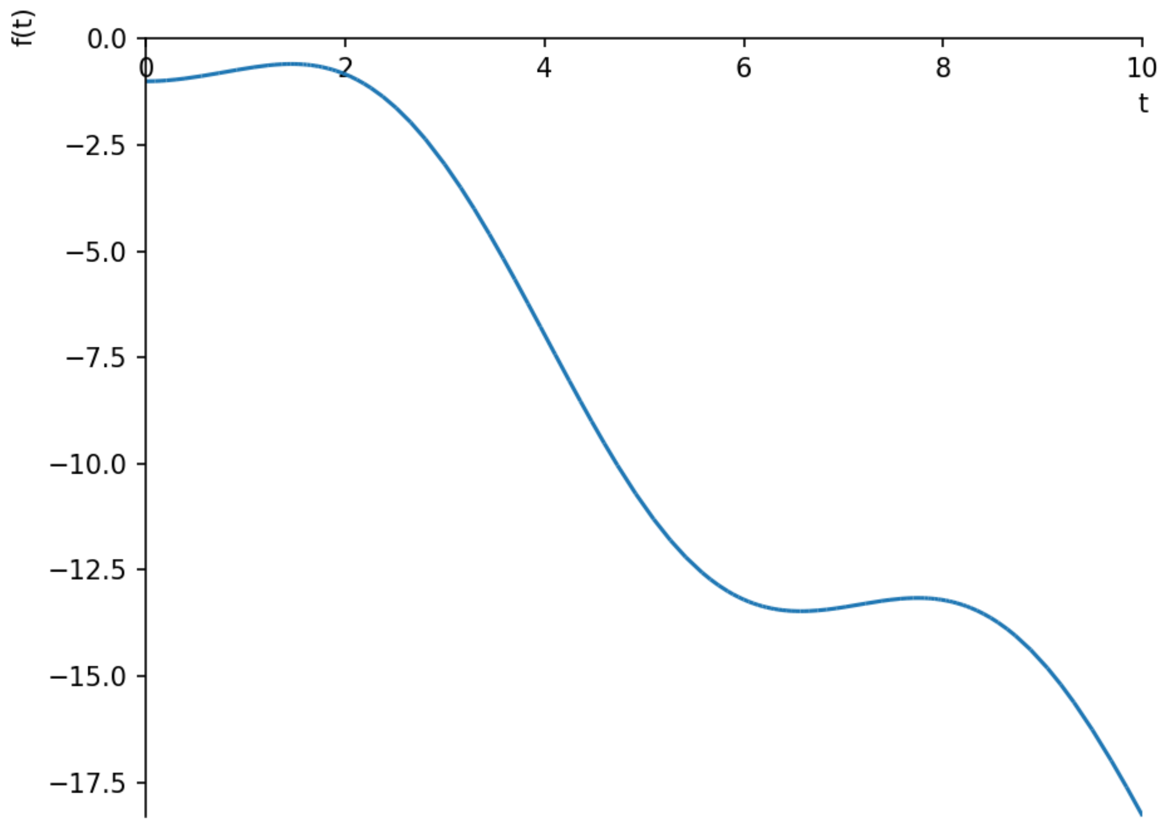
Example 3: y'' + y = t - u_1(t) * (1-t), y(0)=1, y'(0)=0

In [8]:
```
t=symbols('t')
y=Function('y')
g=t-Heaviside(t-1)*(1-t)
deq=diff(y(t),t,2)+y(t)-g
ysoln=dsolve(deq,y(t),ics={y(0):1,diff(y(t),t).subs(t,0):0})
print('The solution is',ysoln)
```

```
The solution is Eq(y(t), t*Heaviside(t - 1) + t - sin(t) - sin(t - 1)*Heavi
side(t - 1) + cos(t) - Heaviside(t - 1))
```

In [9]:
```
matplotlib notebook
```

In [10]:
```
t=symbols('t')
y=ysoln.rhs
plot(-1*y,(t,0,10))
```



Out[10]: <sympy.plotting.plot.Plot at 0x9dd7770>

In [ ]: