

PHYS 331 - pulse propagation

October 21, 2021

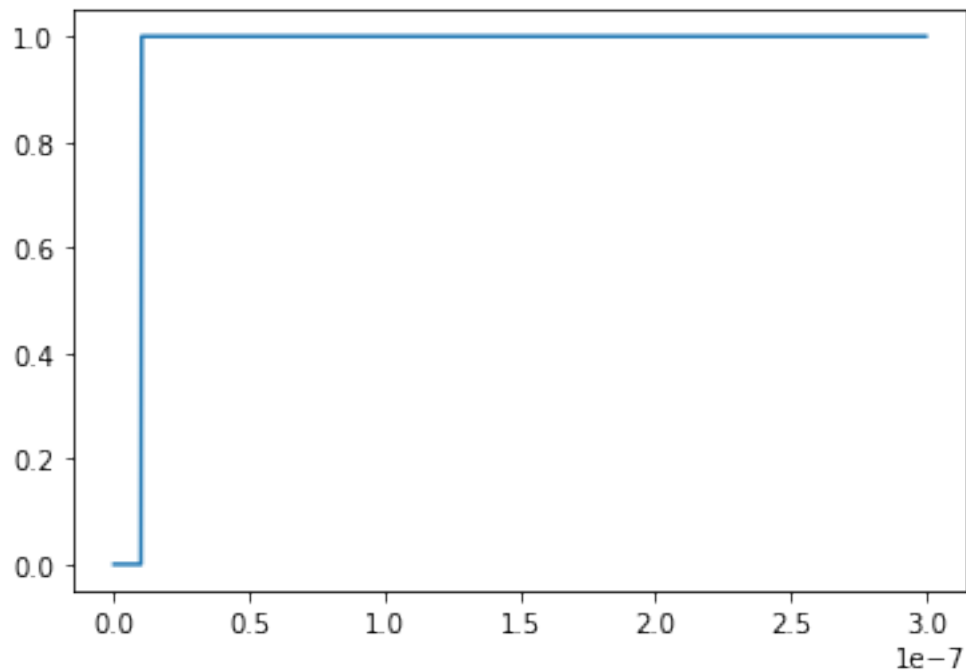
Plotting pulse propagation along a transmission - November 4, 2021

First, import required modules

```
[40]: import numpy as np
import matplotlib.pyplot as plt
```

Can make use of heaviside function to construct a square pulse. First, create a list of times and then plot the heaviside function.

```
[41]: tt = np.arange(0, 300e-9, 0.1e-9) # seconds
t0 = 10e-9 # seconds (transition time)
plt.plot(tt, np.heaviside(tt - t0, 0.5));
```

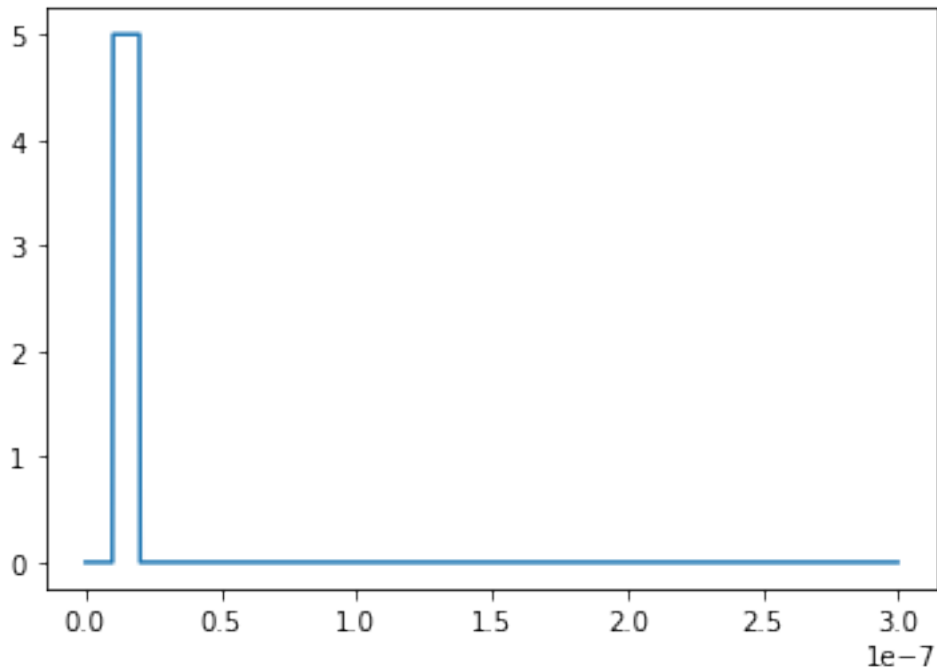


Define a function to construct a square pulse.

```
[42]: def pulse(tt, t0, width, height):
        return height*(np.heaviside(tt - t0, 0.5) - np.heaviside(tt - t0 - width, 0.5))
```

Call our function and plot the result.

```
[43]: w = 10e-9 # seconds (pulse width)
        h = 5 # volts (pulse height)
        plt.plot(tt, pulse(tt, t0, w, h));
```

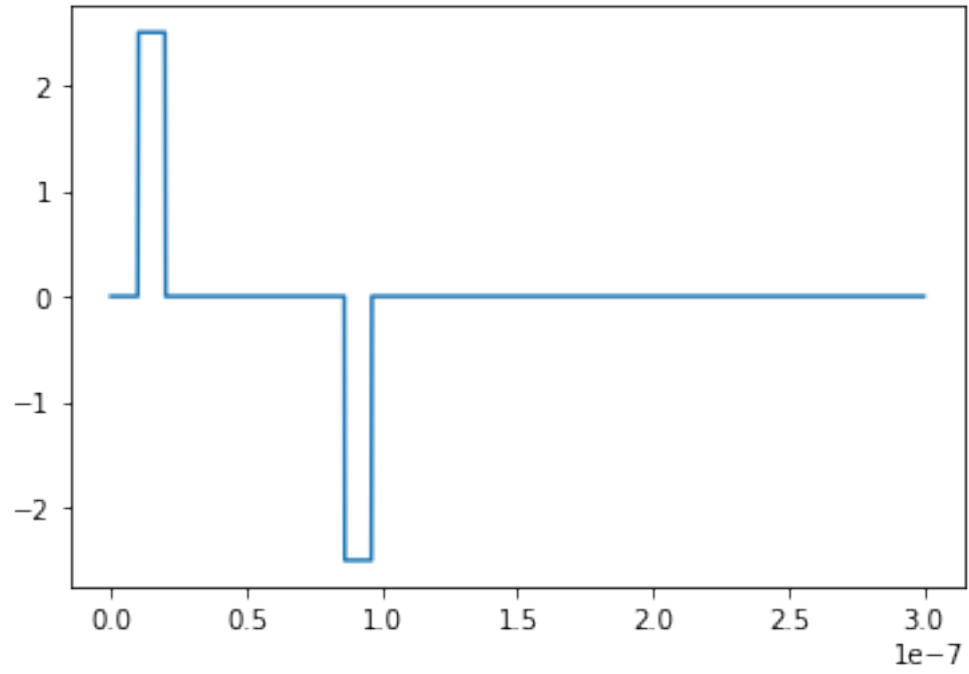


Prepare some values needed to construct the voltage at the input of the transmission line.

```
[44]: ZO = 50 # ohms (characteristics impedance of transmission line)
        ZL = 0 # Short circuit
        #ZL = 999999 # Open circuit
        #ZL = ZO # impedance matching
        #ZL = 80 # ohms
        Gamma = (ZL - ZO)/(ZL + ZO) # Reflection coefficient
        ell = 8 # m (length of transmission line)
        c = 3e8 # m/s (speed of light)
        s = 0.7*c # 70% the speed of light
```

Finally, calculate and plot v_{in} .

```
[45]: vin = pulse(tt, t0, w, h)/2 + (Gamma/2)*pulse(tt, t0 + 2*ell/s, w, h)
        plt.plot(tt, vin);
```



[]: