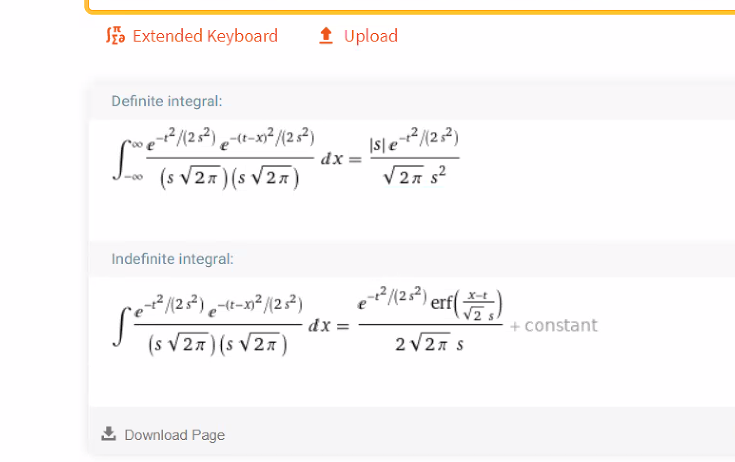
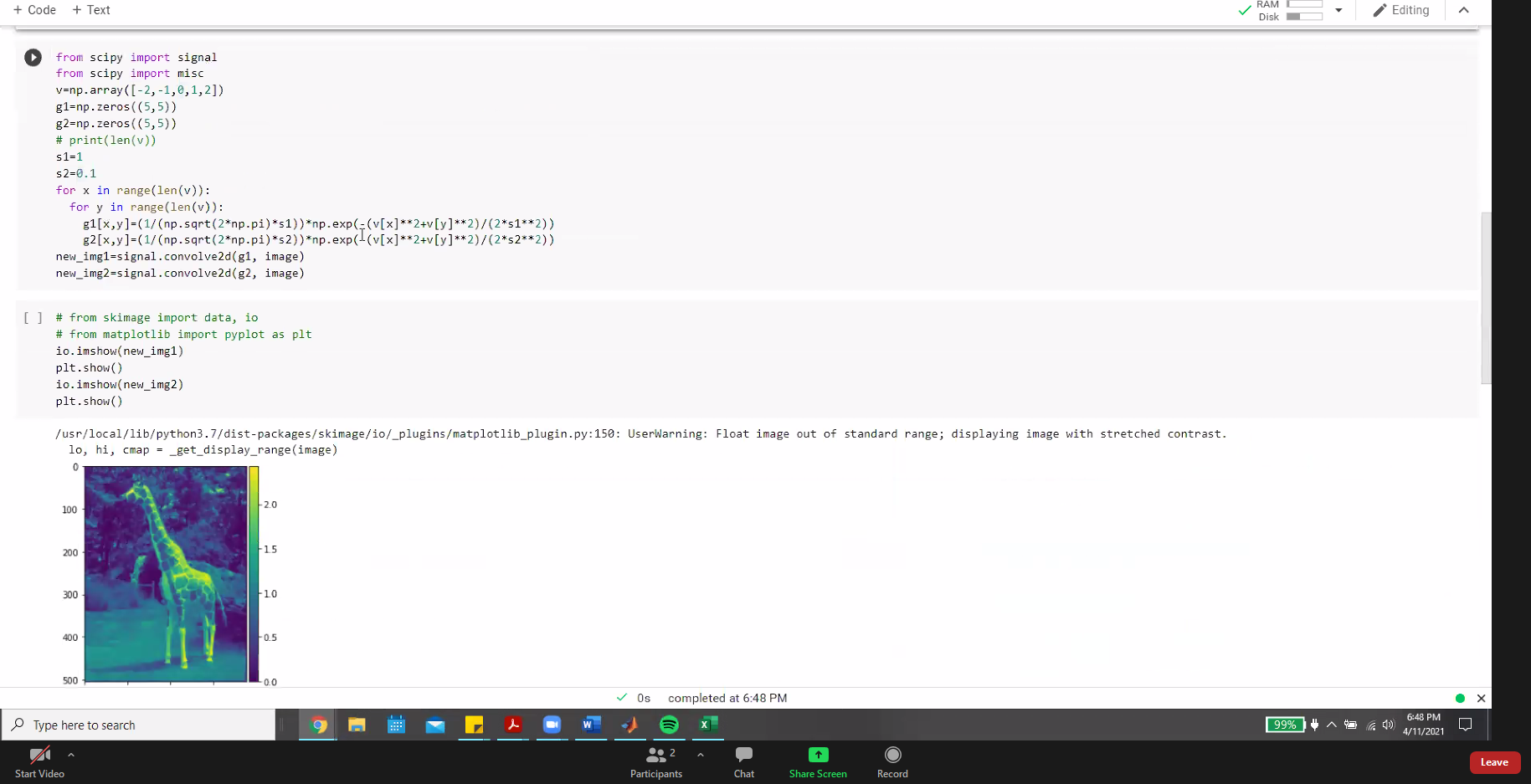
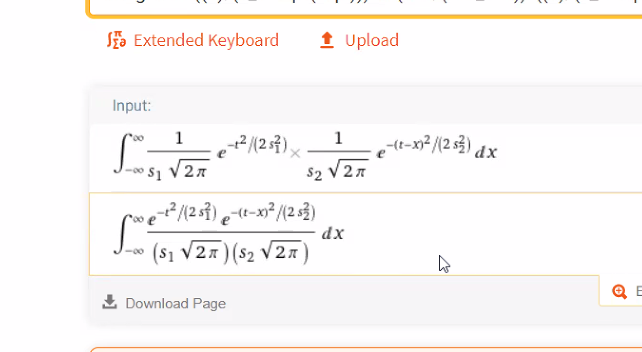
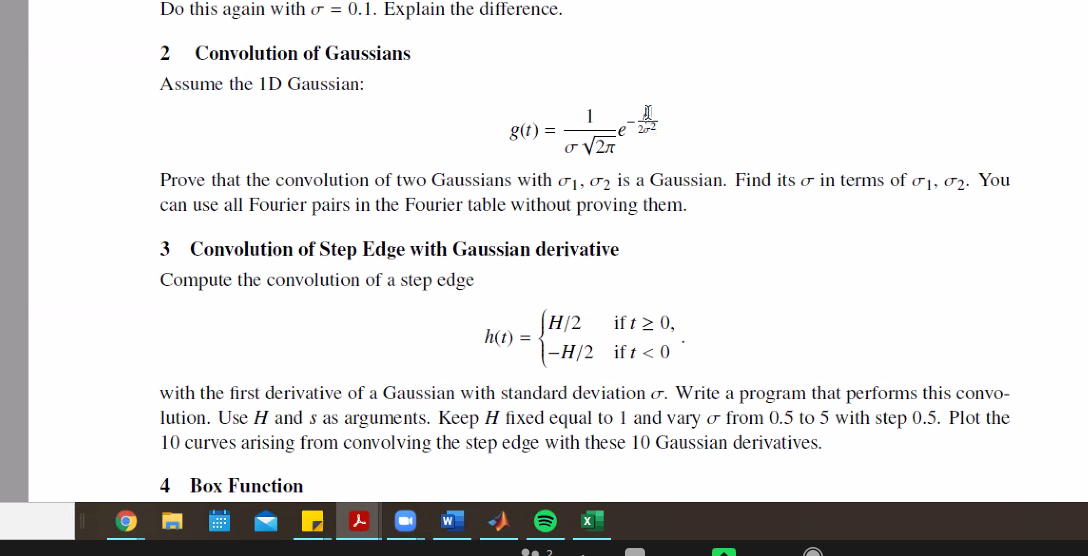
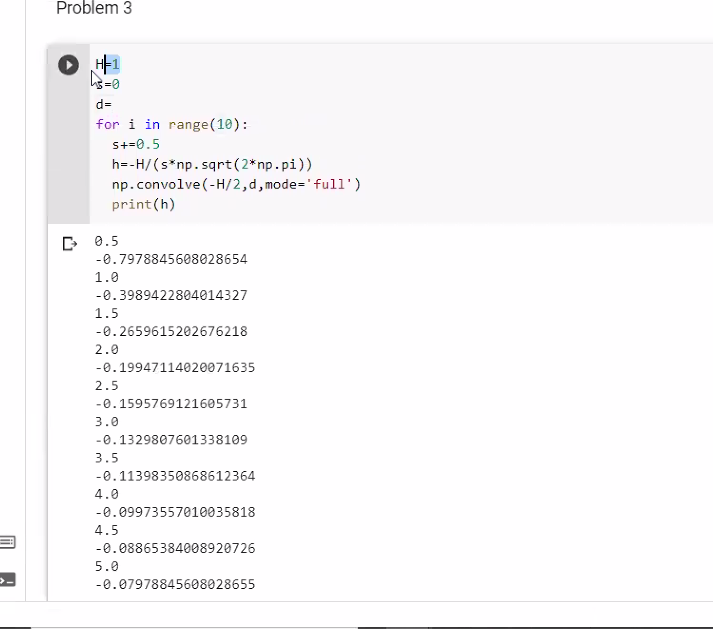
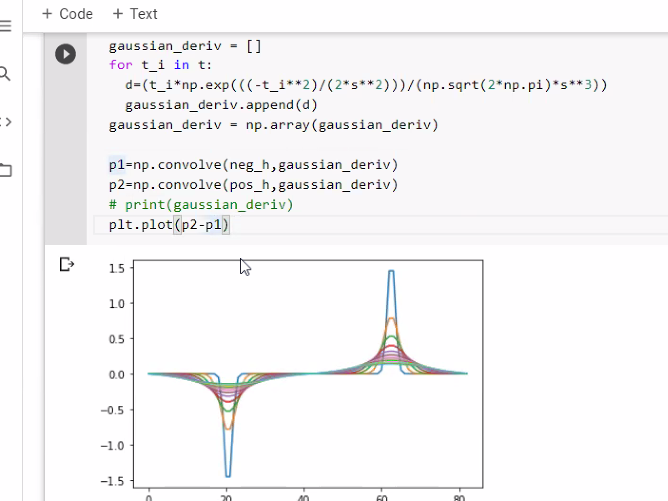
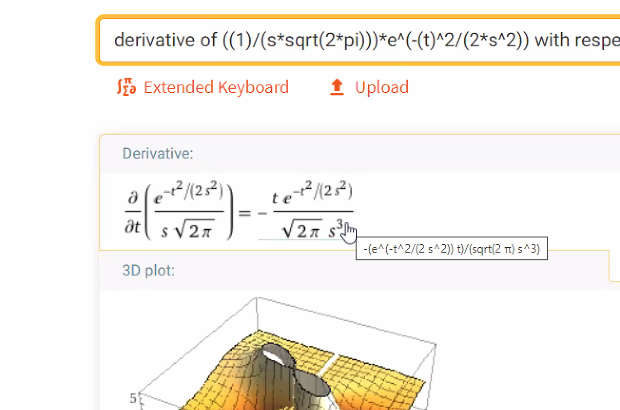
<https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.convolve2d.html>

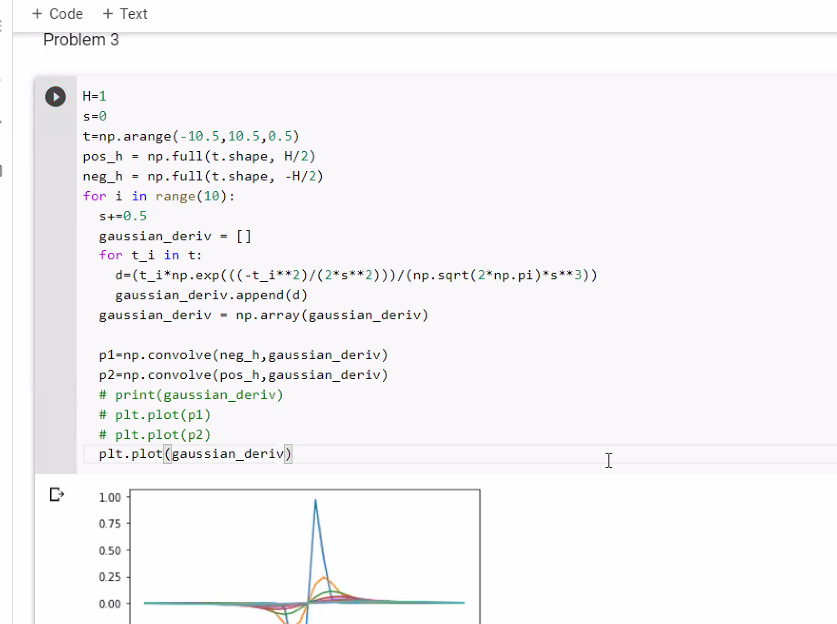




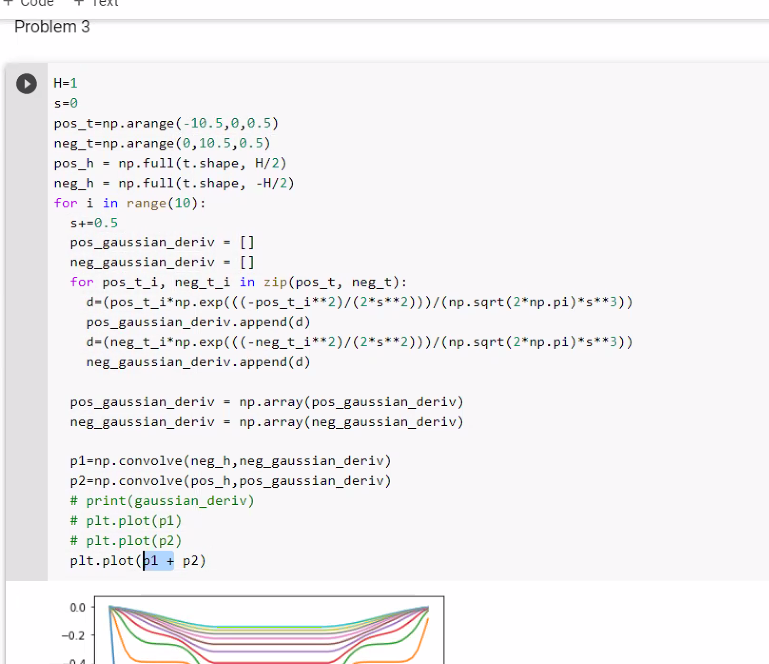
Sigma for q2 is the sum of sigmas: proved via calculus in mathematica





-

N



H=1s=0pos\_t=np.arange(-10.5,0,0.5)neg\_t=np.arange(0,10.5,0.5)pos\_h = np.full(t.shape, H/2)neg\_h = np.full(t.shape, -H/2)for i in range(10): s+=0.5 pos\_gaussian\_deriv = [] neg\_gaussian\_deriv = [] for pos\_t\_i, neg\_t\_i in zip(pos\_t, neg\_t): d=(pos\_t\_i\*np.exp(((-pos\_t\_i\*\*2)/(2\*s\*\*2)))/(np.sqrt(2\*np.pi)\*s\*\*3)) pos\_gaussian\_deriv.append(d) d=(neg\_t\_i\*np.exp(((-neg\_t\_i\*\*2)/(2\*s\*\*2)))/(np.sqrt(2\*np.pi)\*s\*\*3)) neg\_gaussian\_deriv.append(d) pos\_gaussian\_deriv = np.array(pos\_gaussian\_deriv)

neg\_gaussian\_deriv = np.array(neg\_gaussian\_deriv)

p1=np.convolve(neg\_h,neg\_gaussian\_deriv)

p2=np.convolve(pos\_h,pos\_gaussian\_deriv)

# print(gaussian\_deriv)

plt.plot(p1)

# plt.plot(p2)

# plt.plot(p1 + p2)

