```python
import numpy as np
import matplotlib.pyplot as plt

def calc_q(k, alpha):
    return 2*k*np.sin(alpha/2)


def calc_beta(k,R):
    beta = 1/k*(6/k**4+np.exp(-1j*k*R)*(1j*R**3/k - 3*R**2/k**2 - 1j*6*R/k**3 - 6/k**4)) + 2/k**2*(np.sin(k*R)-R**2/k*np.cos(k*R))*(np.exp(1j*k*R)*(-1j*R**2/k + R/k**2 + 2j/k**3)-2j/k**3) - 2/k**2*(2*R/k**2*np.sin(k*R)-R**2/k*np.cos(k*R)+2/k**3*np.cos(k*R) + 2/k**3)
    #beta = 1/k*(6/k**4+np.exp(-1j*k*R))
    return beta

def calc_sigma(k, R, alpha):
    q = calc_q(k, alpha)
    term1 = 1/q**6 * (np.sin(q*R))**2
    term2 = 4/(q**8 * R**2)*(1-np.cos(q*R))**2
    term3 = -4/(q**7 * R)*np.sin(q*R)*(1-np.cos(q*R))
    return term1+term2+term3



# constants - in SI
eV = 1.6e-19  # J
c = 3e8  # m/s
hbar = 1.054e-34  # ev s
R = 1e-10  # 1 ångström
v0 = 1*eV  # 1 ev
E = 10*eV # 1 ev
m = 938e6*eV/c**2  # 938 MeV/c^2

k = np.sqrt(2*m*E)/hbar
eps = 1e-2
alpha = np.linspace(-np.pi,np.pi,1000)

print(R*k)

dsig_prob6 = [calc_sigma(k, R, a) for a in alpha]
dsig_prob6 = np.array(dsig_prob6)
constant_factor = 4*m**2*v0**2/hbar**4
print(constant_factor)


fig, ax = plt.subplots(2)
fig.suptitle("Problem 6: Diff. cross. sec. to first order")
ax[0].plot(alpha/np.pi, constant_factor*dsig_prob6, 'b')
ax[0].set_xlabel("alpha, from -pi to pi")
ax[0].set_ylabel("differential cross section, m^2")
ax[0].grid()
ax[1].set_title("Zoomed in")
ax[1].plot(alpha/np.pi, constant_factor*dsig_prob6, 'b')
ax[1].set_xlim([-0.1, 0.1])
ax[1].set_xlabel("alpha, from -pi to pi")
ax[1].set_ylabel("differential cross section, m^2")
ax[1].grid()
plt.tight_layout()
plt.savefig("problem6.png")



plt.show()
```