```python
import numpy as np
import matplotlib.pyplot as plt
import scipy.optimize as opt

# b)
def a_over_b(gammab):
    b = 1
    fun = -(np.tan(gammab)-gammab)/(gammab*b)
    return fun

def gammab_fun(gammab):
    return gammab


b=1
t = np.linspace(-10,10,10000)

fig, ax = plt.subplots()
ax.plot(t, a_over_b(t))
ax.set_ylim(-25,25)
ax.set_title("Problem 11 b): limit k->0 of tan(delta_0)/k")
ax.set_ylabel("a/b")
ax.set_xlabel("gamma/b")
ax.grid()
plt.savefig("Problem11b")


# c)
# Ramsauer-Townsend effect. tan(delta0)/k = 0 => delta0 = n*pi =>
# => f(theta) = 0 for these energies

min_gammab = opt.minimize(gammab_fun, x0=0.1, constraints={"type": "eq", "fun":
a_over_b})

#(a_over_b, bracket=[-3, 1], method='brentq')

print(min_gammab)

#zero_crossing = np.where(np.diff(np.sign(a_over_b(t))))[0][0]
#print(t[zero_crossing])

ax.plot(1.57, a_over_b(1.58), '*')

# d) infinites for gammab = n*pi (n=0,1,2) and gammab=0

# e)

# f) Plot wavefunc


def wave_rlb(r, gamma):
    return np.sin(gamma*r)


def wave_rgb(r, gamma, b, gammab):
    print(gammab)
    fun = (gammab*np.cos(gammab)*(r/b-1) + np.sin(gammab))
    return fun

fig, ax = plt.subplots()
b = 1
gammabs=[0, np.pi/4, np.pi/2, np.pi]
styles = ['b', 'r', 'k', 'g']

```

```python
roverb1 = np.linspace(0,b,100)
roverb2 = np.linspace(b,2,100)
for i, gammab in enumerate(gammabs):
    gamma = gammab/b
    ax.plot(roverb1, wave_rlb(roverb1, gamma),  styles[i], label=f'gammab={np.round(gamma/np.pi,2)} pi')
    ax.plot(roverb2, wave_rgb(roverb2, gamma, b, gammab), styles[i])
    ax.axvline(b)
    ax.set_title("Problem 11 f): u(r/b) for different gamma*b")
    ax.set_xlabel("r/b")
    ax.set_ylabel("u(r/b)")
ax.legend()
ax.grid()
plt.savefig("Problem11f")
plt.show()
```