040170049
Mohammed
ERKMEN

# EMD HW2

(1) $\vec{H} = 0{,}2\cos(10^9 t - kx + k\sqrt{8}z)\,\vec{e_y}$ air boundary $z=0$. $\varepsilon = 9\varepsilon_0$ $\mu = \mu_0$

(a) $\arctan\left(\frac{1}{\sqrt{8}}\right) = \boxed{\theta_i \Rightarrow \approx 19{,}4710^\circ} = \theta_r = \theta_i$ $(\varepsilon_{r_2} < \varepsilon_{r_1})$

$\sin\theta_t = \sin\theta_i \sqrt{\frac{\varepsilon_{r_1}}{\varepsilon_{r_2}}} \Rightarrow \sin\theta_t = \sin\theta_i . 3$ $\boxed{\theta_t \cong 89^\circ}$

$\hookrightarrow 0{,}333$

(b) $\beta = \frac{w}{c}.\sqrt{\frac{\varepsilon_{r_1}}{\varepsilon_{r_0}}} = \frac{10^9}{3.10^8}.\sqrt{9} = 10.$

$\beta = \sqrt{k^2 + 8k^2}$

$\beta = k\sqrt{9}$

$10 = 3k$, $\boxed{k = \frac{10}{3}}$

(c) $\lambda = \frac{w}{\beta . f} = \frac{10^9}{10 . \frac{10^9}{2\pi}} = \frac{2\pi}{10} = \frac{\pi}{5} = 0{,}6283\,m$

(d) $\vec{E} = \eta(\vec{H} \times \vec{n})$

$\eta = \eta_0 . \sqrt{\frac{\mu_r}{\varepsilon_r}} = \frac{120\pi}{3} = 40\pi$

$40\pi . \left(0{,}2\cos(10^9 t - kx - k\sqrt{8}z).\vec{e_y} \times \frac{1}{10}\frac{(\vec{e_x} + \sqrt{8}\vec{e_z})}{3}\right)$

$\vec{E} = \frac{40\pi}{3}.\left(\left[0{,}2\cos(10^9 t - kx - k\sqrt{8}z)\right](-\vec{e_z} + \vec{e_x})\right] = \frac{40\pi}{3}\cos\left(10^9 t - \frac{10x}{3} - \frac{10\sqrt{8}}{3}z\right)(\vec{e_z} + \vec{e_x})$

(e) $\frac{2\cos\theta_i \sin\theta_t}{\sin(\theta_r + \theta_i).\cos(\theta_r - \theta_i)} = \frac{\cancel{2}.\cos\theta_i\,\cancel{3}.\sin\theta_i}{\cancel{\sin(2\theta_i)}}$ $2\sin\theta_i\cos\theta_i$ $= 2\sin\theta_i \cong 0{,}666$

(f) $\eta_1 . \cos\theta_t = \eta_2 . \cos\theta_{BW}$ $\qquad \Gamma_\parallel = 0$ $\qquad \mu_2 = \mu_1$, so brewster possible

$n_1 . \sin\theta_{BW} = n_2 . \sin\theta_t$

$$\eta_2 . \underbrace{\cos\theta_t}_{-0{,}15} - \eta_1 . \underbrace{\cos\theta_i}_{0{,}942} = 0$$

$$-0{,}15\,\eta_2 = 0{,}942\,\eta_1$$

$$\frac{\eta_1}{\eta_2} = \frac{-0{,}15}{0{,}942} = 0{,}159235$$

$$\theta_{BW} = \arctan(0{,}159235) \cong 9{,}047^\circ$$

(2) $\vec{E} = E_0(\cos\theta_i \vec{e_x} - \sin\theta_i \vec{e_z}) . e^{-jk_1(x\sin\theta_i + z\cos\theta_i)}$ V/m (Parallel polarized)

(a) $$\vec{H_i} = \frac{\vec{n_i} \times \vec{E_i}}{\eta_1} = \vec{e_y} . \frac{E_{i0}}{\eta_1} . e^{-jk_1(x\sin\theta_i + z\cos\theta_i)}$$

$$\vec{E_r} = E_{r0} . (\cos\theta_r . \vec{e_x} + \sin\theta_r \vec{e_z}) . e^{-jk_1(x . \sin\theta_r - z . \cos\theta_r)}$$

$$\vec{H_r} = -e_y . \frac{E_{r0}}{\eta_1} . e^{-jk_1(x\sin\theta_r - z\cos\theta_r)}$$

$$\vec{E_t} = (\vec{e_x}\cos\theta_t - \vec{e_z}\sin\theta_t) E_{t0} . e^{-jk_2(x\sin\theta_t + z\cos\theta_t)}$$

$$\vec{H_t} = \vec{e_y} . \frac{E_{t0}}{\eta_2} . e^{-jk_2 . (x\sin\theta_t + z\cos\theta_t)}$$

(b) $z=0 \rightarrow E_{ix}(x,0) + E_{rx}(x,0) = E_{tx}(x,0)$

$\hookrightarrow H_{iy}(x,0) + H_{ry}(x,0) = H_{ty}(x,0)$

(2)

(c) medium I and II is lossless and different mediums -ase)

$$\Gamma_{\parallel} = \frac{\eta_2 \cos\theta_t - \eta_1 \cos\theta_i}{\eta_2 \cos\theta_t - \eta_1 \cos\theta_i} \qquad T_{\parallel} = \frac{2\eta_2 \cdot \cos\theta_i}{\eta_2 \cdot \cos\theta_t + \eta_1 \cos\theta_i}$$

$$\Gamma_{\parallel} = \frac{\sqrt{\frac{\mu_2}{\varepsilon_2}} \cdot \cos\theta_t - \sqrt{\frac{\mu_1}{\varepsilon_1}} \cdot \cos\theta_i}{\sqrt{\frac{\mu_2}{\varepsilon_2}} \cdot \cos\theta_t - \sqrt{\frac{\mu_1}{\varepsilon_1}} \cdot \cos\theta_i} \qquad T_{\parallel} = \frac{2 \cdot \sqrt{\frac{\mu_2}{\varepsilon_2}} \cdot \cos\theta_i}{\sqrt{\frac{\mu_2}{\varepsilon_2}} \cdot \cos\theta_t + \sqrt{\frac{\mu_1}{\varepsilon_1}} \cdot \cos\theta_i}$$

(e) $\vec{E} = E_i \cdot e^{-jk_1(x\sin\varphi_i + z\cos\varphi_i)} \cdot \vec{e_y}$

(e.a) $\vec{H_i} = \dfrac{\vec{n_i} \times \vec{E_i}}{\eta_1} = \dfrac{(\sin\varphi_i \vec{e_x} + \cos\varphi_i \vec{e_z}) \times \vec{e_y}}{\eta_1} \cdot E_{io} \cdot e^{-jk_1(x\sin\varphi_i + z\cos\varphi_i)}$

$\vec{H_i} = \dfrac{(\vec{e_z}\cdot\sin\varphi_i - \cos\varphi_i\cdot\vec{e_x})}{\eta_1} \cdot E_{io} \cdot e^{-jk_1(x\sin\varphi_i + z\cos\varphi_i)}$

$\vec{E_r} = E_{io} \cdot e^{-jk_1(x\sin\varphi_r - z\cos\varphi_r)} \cdot \vec{e_y}$

$\vec{H_r} = \dfrac{\vec{n_r} \times \vec{E_r}}{\eta_1} = \dfrac{E_{ro}}{\eta_1} \cdot (\vec{e_x}\cdot\cos\varphi_r + \vec{e_z}\cdot\sin\varphi_r) \cdot e^{-jk_1(x\sin\varphi_r - z\cos\varphi_r)}$

$\vec{E_t}(x,z) = \vec{e_y} \cdot E_{to}\, e^{-jk_2(x\sin\varphi_t + z\cos\varphi_t)}$

$\vec{H_t}(x,z) = \dfrac{\vec{n_t} \times \vec{E_t}}{\eta_t} = \dfrac{E_{to}}{\eta_t} \cdot (-\vec{e_x}\cos\varphi_t + \vec{e_z}\cdot\sin\varphi_t) \cdot e^{-jk_2(x\sin\varphi_t + z\cos\varphi_t)}$

(e.b) $z=0 \rightarrow E_{iy}(x,0) + E_{ry}(x,0) = E_{ty}(x,0)$

$z=0 \rightarrow H_{ix}(x,0) + H_{rx}(x,0) = H_{tx}(x,0)$

(e.c)

$$\Gamma_\perp = \frac{\eta_2\cos\varphi_i - \eta_1\cos\varphi_t}{\eta_2\cos\varphi_i + \eta_1\cos\varphi_t} = \frac{\sqrt{\frac{\mu_2}{\varepsilon_2}}\cos\varphi_i - \sqrt{\frac{\mu_1}{\varepsilon_1}}\cos\varphi_t}{\sqrt{\frac{\mu_2}{\varepsilon_2}}\cos\varphi_i + \sqrt{\frac{\mu_1}{\varepsilon_1}}\cos\varphi_t}$$

$$T_\perp = 1 + \Gamma_\perp = \frac{2\eta_2\cos\varphi_i}{\eta_2\cos\varphi_i + \eta_1\cos\varphi_t} = \frac{2\cdot\sqrt{\frac{\mu_2}{\varepsilon_2}}\cos\varphi_i}{\sqrt{\frac{\mu_2}{\varepsilon_2}}\cos\varphi_i + \sqrt{\frac{\mu_1}{\varepsilon_1}}\cos\varphi_t}$$

EMD HW2

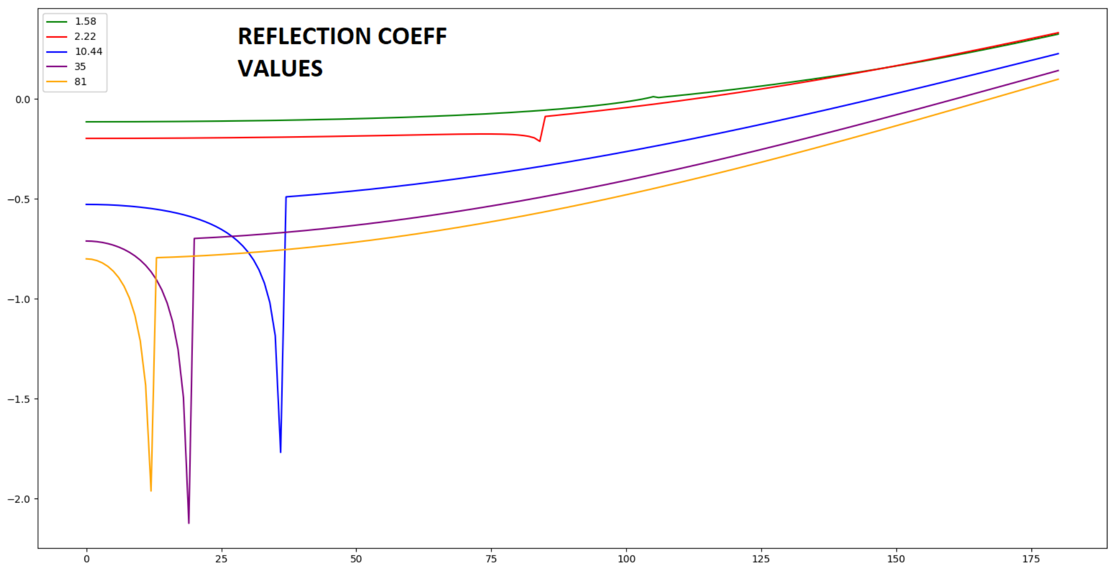
2-D)

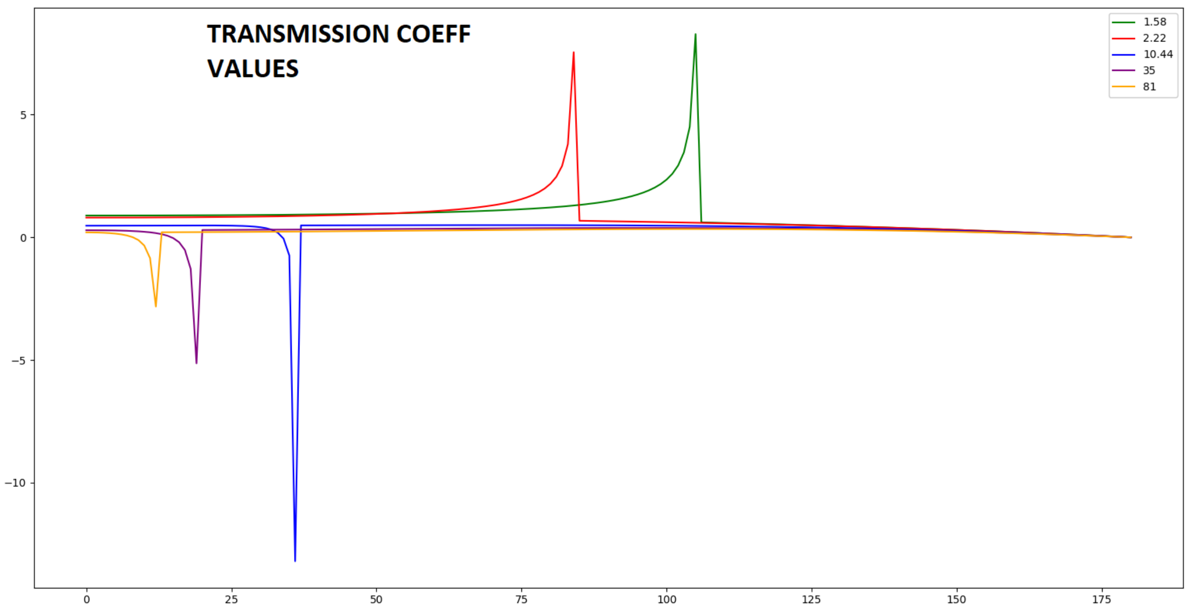
Source code:

```
import math
import matplotlib.pyplot as plt

e2e1 = [1.58,2.22,10.44,35,81]

alphas = []
for i in range (181):
    alphas.append(math.radians(i*0.5))


alphast158 = []
for i in range(len(alphas)):
    first = math.sin(alphas[i]) * math.sqrt(1.58)
    if(first<=1):
        arcsin = math.asin(first)
        alphast158.append(arcsin)
    else:
        alphast158.append(0)

alphast222 = []
for i in range(len(alphas)):
    first = math.sin(alphas[i]) * math.sqrt(2.22)
    if(first<=1):
        arcsin = math.asin(first)
        alphast222.append(arcsin)
    else:
        alphast222.append(0)

alphast1044 = []
for i in range(len(alphas)):
    first = math.sin(alphas[i]) * math.sqrt(10.44)
    if(first<=1):
        arcsin = math.asin(first)
        alphast1044.append(arcsin)
    else:
        alphast1044.append(0)

alphast35 = []
for i in range(len(alphas)):
    first = math.sin(alphas[i]) * math.sqrt(35)
    if(first<=1):
        arcsin = math.asin(first)
        alphast35.append(arcsin)
    else:
        alphast35.append(0)

alphast81 = []
for i in range(len(alphas)):
    first = math.sin(alphas[i]) * math.sqrt(81)
    if(first<=1):
        arcsin = math.asin(first)
        alphast81.append(arcsin)
    else:
        alphast81.append(0)
```

```
reflection_e2e1_1 = []
reflection_e2e1_2 = []
reflection_e2e1_3 = []
reflection_e2e1_4 = []
reflection_e2e1_5 = []

transmission_e2e1_1 = []
transmission_e2e1_2 = []
transmission_e2e1_3 = []
transmission_e2e1_4 = []
transmission_e2e1_5 = []

for i in range(181):
    coeff1 = -e2e1[0]*math.cos(alphas[i])
    coeff2 = math.sqrt(e2e1[0]-(math.sin(alphas[i]))**2 )
    coeff3 = e2e1[0]*math.cos(alphast158[i])
    reflection = (coeff1+coeff2)/(coeff3+coeff2)
    coefftransmission= math.cos(alphas[i])/math.cos(alphast158[i])
    transmission_e2e1_1.append((1+reflection)*coefftransmission)
    reflection_e2e1_1.append(reflection)

for i in range(181):
    coeff1 = -e2e1[1]*math.cos(alphas[i])
    coeff2 = math.sqrt(e2e1[1]-(math.sin(alphas[i]))**2 )
    coeff3 = e2e1[1]*math.cos(alphast222[i])
    reflection = (coeff1+coeff2)/(coeff3+coeff2)
    coefftransmission= math.cos(alphas[i])/math.cos(alphast222[i])
    transmission_e2e1_2.append((1+reflection)*coefftransmission)
    reflection_e2e1_2.append(reflection)

for i in range(181):
    coeff1 = -e2e1[2]*math.cos(alphas[i])
    coeff2 = math.sqrt(e2e1[2]-(math.sin(alphas[i]))**2 )
    coeff3 = e2e1[2]*math.cos(alphast1044[i])
    reflection = (coeff1+coeff2)/(coeff3+coeff2)
    coefftransmission= math.cos(alphas[i])/math.cos(alphast1044[i])
    transmission_e2e1_3.append((1+reflection)*coefftransmission)
    reflection_e2e1_3.append(reflection)

for i in range(181):
    coeff1 = -e2e1[3]*math.cos(alphas[i])
    coeff2 = math.sqrt(e2e1[3]-(math.sin(alphas[i]))**2 )
    coeff3 = e2e1[3]*math.cos(alphast35[i])
    reflection = (coeff1+coeff2)/(coeff3+coeff2)
    coefftransmission= math.cos(alphas[i])/math.cos(alphast35[i])
    transmission_e2e1_4.append((1+reflection)*coefftransmission)
    reflection_e2e1_4.append(reflection)

for i in range(181):
    coeff1 = -e2e1[4]*math.cos(alphas[i])
    coeff2 = math.sqrt(e2e1[4]-(math.sin(alphas[i]))**2 )
    coeff3 = e2e1[4]*math.cos(alphast81[i])
    reflection = (coeff1+coeff2)/(coeff3+coeff2)
    coefftransmission= math.cos(alphas[i])/math.cos(alphast81[i])
    transmission_e2e1_5.append((1+reflection)*coefftransmission)
    reflection_e2e1_5.append(reflection)


plt.plot(reflection_e2e1_1,"g",label='1.58')
```

```
plt.plot(reflection_e2e1_2,"r",label='2.22')
plt.plot(reflection_e2e1_3,"b",label='10.44')
plt.plot(reflection_e2e1_4,"purple",label = '35')
plt.plot(reflection_e2e1_5,"orange", label='81')
plt.legend(framealpha=1,frameon=True)
plt.show()
'''
plt.plot(transmission_e2e1_1,"g",label='1.58')
plt.plot(transmission_e2e1_2,"r",label='2.22')
plt.plot(transmission_e2e1_3,"b",label='10.44')
plt.plot(transmission_e2e1_4,"purple",label = '35')
plt.plot(transmission_e2e1_5,"orange", label='81')
plt.legend(framealpha=1,frameon=True)
plt.show()
'''
```



Bottom scale is 0 to 180 because there are 180 values between 0 and 90 that adds itself 0.5 in every step. Because of the calculations, there is a step we should get arcsin() of a value. When this value is bigger than 1, arcsin function gives error. So i put 0's to these values. We can see that border where the graphs come back near of the start.

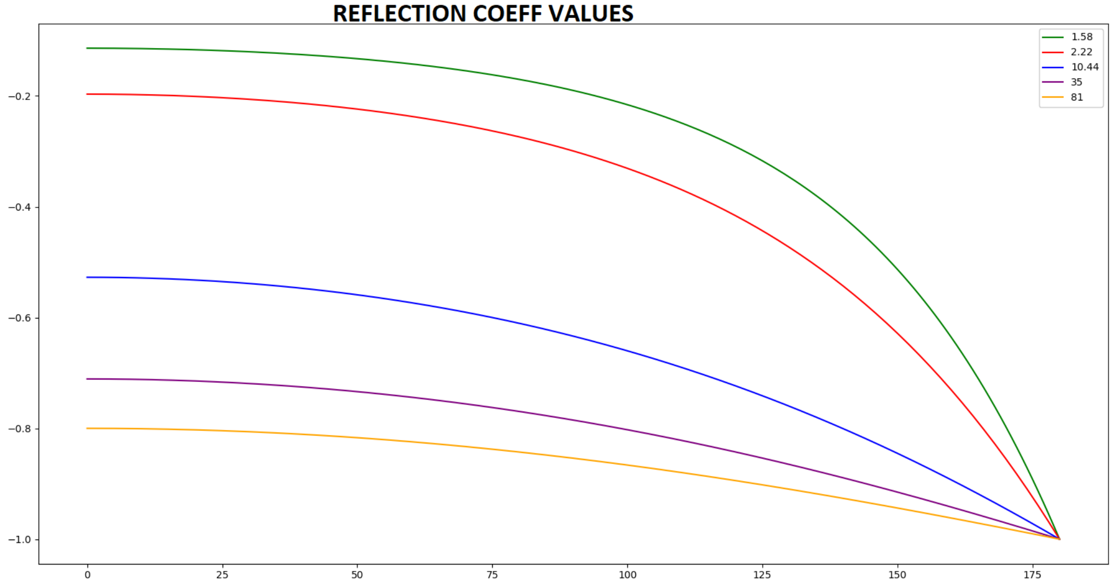
2-E.D)

Source code:

```
import math
import matplotlib.pyplot as plt

e2e1 = [1.58,2.22,10.44,35,81]

alphas = []
for i in range (181):
    alphas.append(math.radians(i*0.5))




reflection_e2e1_1 = []
reflection_e2e1_2 = []
reflection_e2e1_3 = []
reflection_e2e1_4 = []
reflection_e2e1_5 = []

transmission_e2e1_1 = []
transmission_e2e1_2 = []
transmission_e2e1_3 = []
transmission_e2e1_4 = []
transmission_e2e1_5 = []



for i in range(181):
    coeff1 = math.cos(alphas[i])
    coeff2 =  - math.sqrt(e2e1[0]- ((math.sin(alphas[i]))**2))
    coeff3 =  -coeff2
    reflection = (coeff1+coeff2)/(coeff1+coeff3)
    transmission_e2e1_1.append((1+reflection))
    reflection_e2e1_1.append(reflection)


for i in range(181):
    coeff1 = math.cos(alphas[i])
    coeff2 =  - math.sqrt(e2e1[1]- ((math.sin(alphas[i]))**2))
    coeff3 =  -coeff2
    reflection = (coeff1+coeff2)/(coeff1+coeff3)
    transmission_e2e1_2.append((1+reflection))
    reflection_e2e1_2.append(reflection)


for i in range(181):
    coeff1 = math.cos(alphas[i])
    coeff2 =  - math.sqrt(e2e1[2]- ((math.sin(alphas[i]))**2))
    coeff3 =  -coeff2
    reflection = (coeff1+coeff2)/(coeff1+coeff3)
```

```
    transmission_e2e1_3.append((1+reflection))
    reflection_e2e1_3.append(reflection)

for i in range(181):
    coeff1 = math.cos(alphas[i])
    coeff2 =  - math.sqrt(e2e1[3]- ((math.sin(alphas[i]))**2))
    coeff3 =  -coeff2
    reflection = (coeff1+coeff2)/(coeff1+coeff3)
    transmission_e2e1_4.append((1+reflection))
    reflection_e2e1_4.append(reflection)


for i in range(181):
    coeff1 = math.cos(alphas[i])
    coeff2 =  - math.sqrt(e2e1[4]- ((math.sin(alphas[i]))**2))
    coeff3 =  -coeff2
    reflection = (coeff1+coeff2)/(coeff1+coeff3)
    transmission_e2e1_5.append((1+reflection))
    reflection_e2e1_5.append(reflection)



plt.plot(reflection_e2e1_1,"g",label='1.58')
plt.plot(reflection_e2e1_2,"r",label='2.22')
plt.plot(reflection_e2e1_3,"b",label='10.44')
plt.plot(reflection_e2e1_4,"purple",label = '35')
plt.plot(reflection_e2e1_5,"orange", label='81')
plt.legend(framealpha=1,frameon=True)
plt.show()
'''
plt.plot(transmission_e2e1_1,"g",label='1.58')
plt.plot(transmission_e2e1_2,"r",label='2.22')
plt.plot(transmission_e2e1_3,"b",label='10.44')
plt.plot(transmission_e2e1_4,"purple",label = '35')
plt.plot(transmission_e2e1_5,"orange", label='81')
plt.legend(framealpha=1,frameon=True)
plt.show()
'''
```

REFLECTION COEFF VALUES




TRANSMISSION COEFF VALUES