Task2_PeriodicMultilayer_Bandgap

May 17, 2016

```
In [3]: import numpy as np
        import scipy as sp
        import scipy.sparse.linalg as linalg
        import matplotlib.pyplot as plt
        %matplotlib inline
```

1 Task 2: 1D Periodic Multilayer Dielectric Slab, Eigenmode formulation

1.1 Maxwell Equations (time domain)

$$\nabla \times E = -\mu \frac{\partial H}{\partial t} \tag{1}$$

$$\nabla \times H = \epsilon \frac{\partial E}{\partial t} \tag{2}$$

1.2 Time Domain $\rightarrow$ Frequency Domain

Using the Fourier transform we get:

$$E(r,t) \rightarrow E(r,\omega) \tag{3}$$

$$\frac{\partial E}{\partial t} \rightarrow i\omega E \tag{4}$$

Thus our wave equation transforms into:

$$\nabla E = -i\omega\mu H \tag{5}$$

$$\nabla E = -i\omega\epsilon E \tag{6}$$

1.3 1D Coordinates, Periodic

$$H_{y_i}, E_{z_{i+\frac{1}{2}}}, i \in \mathcal{N} \tag{7}$$

$$H_{y_0} = H_{y_n} e^{ik_x a} \tag{8}$$

$$E_{z_{0+\frac{1}{2}}} = E_{z_{n+\frac{1}{2}}} e^{ik_x a} \tag{9}$$

$$\frac{\partial E_z}{\partial x} = -i\omega\mu H_y \tag{10}$$

$$\frac{\partial H_y}{\partial x} = i\omega\epsilon E_z \tag{11}$$

1.4 Discretization

$$\frac{\partial E_{zi}}{\partial x} \approx \frac{E_{zi+1} - E_{zi-1}}{\Delta x} \tag{12}$$

$$\frac{\partial E_{z0}}{\partial x} \approx \frac{E_{z1} - E_{zn} e^{ik_x a}}{\Delta x} \tag{13}$$

$$\frac{\partial E_{zn}}{\partial x} \approx \frac{E_{z0} e^{ik_x a} - E_{zn-1}}{\Delta x} \tag{14}$$

Forward difference

$$-i\omega\mu_i E_{z_{i+\frac{1}{2}}} \approx \frac{H_{y_{i+1}} - H_{y_i}}{\Delta x} \tag{15}$$

Backward difference

$$-i\omega\epsilon_i H_{y_i} \approx \frac{E_{z_{i+\frac{1}{2}}} - E_{z_{i-\frac{1}{2}}}}{\Delta x} \tag{16}$$

1.5 Wave equation

$$H_{y_i} = \frac{i}{\omega\mu_i} \frac{\partial E_{zi}}{\partial x} \tag{17}$$

$$E_{zi} = \frac{-i}{\omega\epsilon_i} \frac{\partial H_{y_i}}{\partial x} \tag{18}$$

$$\frac{\partial}{\partial x} \frac{i}{\omega\mu_i} \frac{\partial}{\partial x} E_{zi} = i\omega\epsilon_i E_{zi} \tag{19}$$

$$\frac{1}{\epsilon_i} \frac{\partial}{\partial x} \frac{1}{\mu_i} \frac{\partial}{\partial x} E_{zi} = \omega^2 E_{zi} \tag{20}$$

or

$$\frac{1}{\mu_i} \frac{\partial}{\partial x} \frac{1}{\epsilon_i} \frac{\partial}{\partial x} H_{y_i} = \omega^2 H_{y_i} \tag{21}$$

1.6 Matrix form

$$H_{y_i} = \frac{i}{\omega\mu_i}\frac{\partial E_z}{\partial x} \approx \frac{i}{\omega\mu_i}\frac{E_{z_{i+\frac{1}{2}}} - E_{z_{i-\frac{1}{2}}}}{\Delta x} \tag{22}$$

$$\begin{bmatrix} H_{y_0} \\ H_{y_1} \\ \vdots \\ H_{y_{n-1}} \\ H_{y_n} \end{bmatrix} = \frac{i}{\omega\Delta x} \begin{bmatrix} \frac{1}{\mu_0} & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{\mu_1} & 0 & \dots & 0 \\ 0 & 0 & \frac{1}{\mu_2} & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & 0 & \frac{1}{\mu_n} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & \dots & -e^{ik_xa} \\ -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & -1 & 1 \end{bmatrix} \begin{bmatrix} E_{z0+\frac{1}{2}} \\ E_{z1+\frac{1}{2}} \\ \vdots \\ E_{zn-\frac{1}{2}} \\ E_{zn+\frac{1}{2}} \end{bmatrix} \tag{23}$$

$$E_{z_{i+\frac{1}{2}}} = \frac{-i}{\omega\epsilon_i}\frac{\partial H_{y_i}}{\partial x} \approx \frac{-i}{\omega\epsilon_i}\frac{H_{y_{i+1}} - H_{y_i}}{\Delta x} \tag{24}$$

$$\begin{bmatrix} E_{z0+\frac{1}{2}} \\ E_{z1+\frac{1}{2}} \\ \vdots \\ E_{zn-\frac{1}{2}} \\ E_{zn+\frac{1}{2}} \end{bmatrix} = \frac{i}{\omega\Delta x} \begin{bmatrix} \frac{1}{\epsilon_0} & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{\epsilon_1} & 0 & \dots & 0 \\ 0 & 0 & \frac{1}{\epsilon_2} & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & 0 & \frac{1}{\epsilon_n} \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ 0 & 0 & -1 & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 1 \\ e^{ik_xa} & 0 & \dots & 0 & -1 \end{bmatrix} \begin{bmatrix} H_{y_0} \\ H_{y_1} \\ \vdots \\ H_{y_{n-1}} \\ H_{y_n} \end{bmatrix} \tag{25}$$

$$\omega^2 E_z = \frac{1}{\epsilon}\frac{\partial}{\partial x}\frac{1}{\mu}\frac{\partial}{\partial x}E_z \tag{26}$$

$$\omega^2 \begin{bmatrix} E_{z0+\frac{1}{2}} \\ E_{z1+\frac{1}{2}} \\ \vdots \\ E_{zn-\frac{1}{2}} \\ E_{zn+\frac{1}{2}} \end{bmatrix} = \frac{i}{\Delta x} \begin{bmatrix} \frac{1}{\epsilon_0} & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{\epsilon_1} & 0 & \dots & 0 \\ 0 & 0 & \frac{1}{\epsilon_2} & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & 0 & \frac{1}{\epsilon_n} \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ 0 & 0 & -1 & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 1 \\ e^{ik_xa} & 0 & \dots & 0 & -1 \end{bmatrix} \frac{i}{\Delta x} \begin{bmatrix} \frac{1}{\mu_0} & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{\mu_1} & 0 & \dots & 0 \\ 0 & 0 & \frac{1}{\mu_2} & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & 0 & \frac{1}{\mu_n} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \\ 0 & -1 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \tag{27}$$

or, simplifing, and replacing $i + \frac{1}{2}$ with i:

$$\omega^2 \begin{bmatrix} E_{z0} \\ E_{z1} \\ \vdots \\ E_{zn-1} \\ E_{zn} \end{bmatrix} = -\frac{1}{\Delta x^2} \begin{bmatrix} \frac{1}{\epsilon_0} & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{\epsilon_1} & 0 & \dots & 0 \\ 0 & 0 & \frac{1}{\epsilon_2} & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & 0 & \frac{1}{\epsilon_n} \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ 0 & 0 & -1 & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 1 \\ e^{ik_xa} & 0 & \dots & 0 & -1 \end{bmatrix} \begin{bmatrix} \frac{1}{\mu_0} & 0 & 0 & \dots & 0 \\ 0 & \frac{1}{\mu_1} & 0 & \dots & 0 \\ 0 & 0 & \frac{1}{\mu_2} & \dots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & 0 & \frac{1}{\mu_n} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \\ \vdots & \vdots & \vdots \\ 0 & 0 & \dots \end{bmatrix} \tag{28}$$

1.7 Boundary Conditions

In this case the boundary conditions are periodic, satisfied by the e^{ik_xa} elements in the resulting matrix

2 Implementation

2.1 Setup grid

```
In [102]: n=200           # number of grid nodes

          dx=1/n          # discretization step, domain size = 1

          eps1 = 13       # Layer 1
          eps2 = 1        # Layer 2

          num_eigs = 6    # Solver for first # eigenmodes
```

2.2 Build matrix

```
In [103]: # Material parameters distribution

          eps = np.ones(n)
          eps[:int(n/2)-1] = 1/eps1
          eps[int(n/2):] = 1/eps2

          eps = sp.sparse.dia_matrix(([eps], [0]), [n,n])

          # Forward

          diag = np.ones(n) * -1/dx
          up_diag = np.ones(n) * 1/dx

          M_1 = sp.sparse.dia_matrix(([up_diag, diag], [1, 0]), [n,n])
          # Backward

          diag = np.ones(n) * 1/dx
          up_diag = np.ones(n) * -1/dx

          M_2 = sp.sparse.dia_matrix(([up_diag, diag], [-1, 0]), [n,n])

          M_1 = sp.sparse.lil_matrix(M_1)
          M_2 = sp.sparse.lil_matrix(M_2)
```

2.3 Solve for eigenmodes

```
In [117]: kk0 = 2*sp.pi*np.linspace(-0.5,0.5,300) # k-vector in medium with eps1-ep

          k = np.zeros((num_eigs, kk0.size), dtype=complex)
          for ik in range(kk0.size):
              k0=kk0[ik]

              M_1[n-1, 0] = np.cos(k0*1)/dx # to impose periodic boundary conditio
```

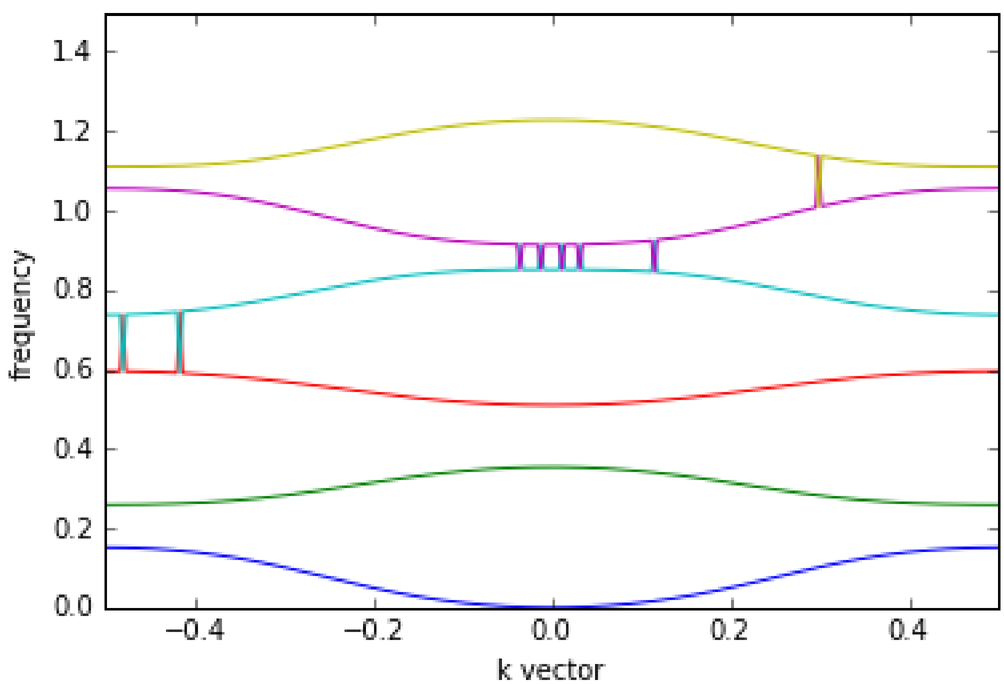
```
        M_2[0, n-1] = -np.cos(k0*1)/dx

        M = -eps*M_2*M_1

        kt = k0/np.sqrt((eps1+eps2)/2); # target k=w/c
        k2, V = linalg.eigs(M, k=num_eigs, M=None, sigma=kt**2)

        k[:,ik] = np.sqrt(k2) # k=w/c
```

2.4 Plot eigenmodes

2.4.1 Bands

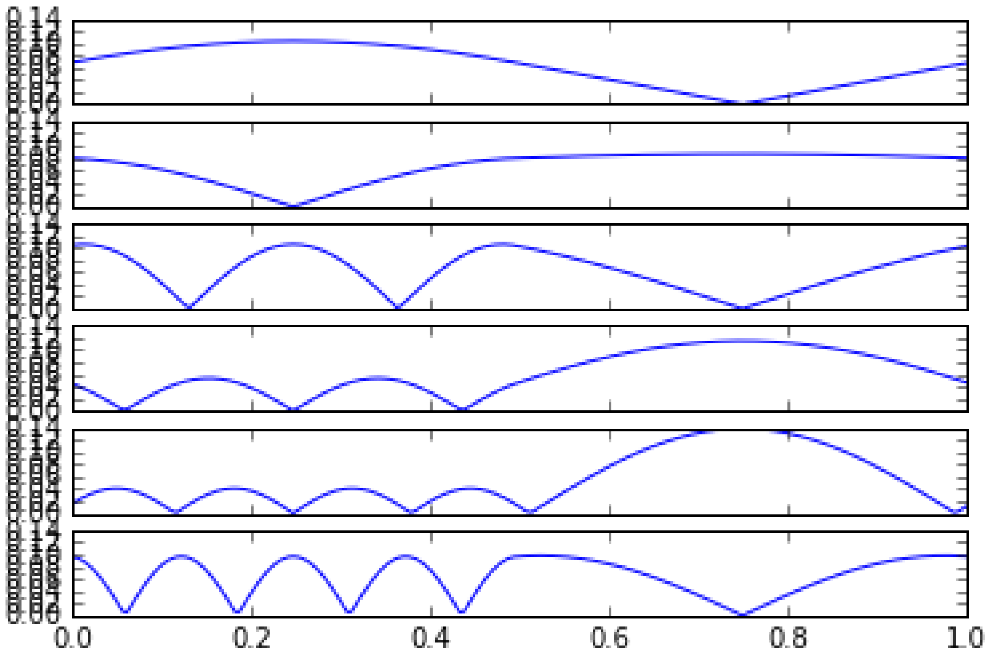
```
In [126]: for i in range(num_eigs):
              plt.hold(True)
              plt.plot(kk0/(2*sp.pi), np.real(k[i,:]/(2*sp.pi)) ,'-')

          plt.xlabel("k vector")
          plt.ylabel("frequency")
          plt.xlim([-0.5, 0.5])
          plt.ylim([0.0, 1.5])
```

```
Out[126]: (0.0, 1.5)
```



2.4.2 Field distribution

```
In [119]: f, ax = plt.subplots(6,1, sharex=True, sharey=True)

          for i in range(num_eigs):
              ax[i].plot(np.linspace(0, 1, n), np.abs(V[:,i]),'-')
```



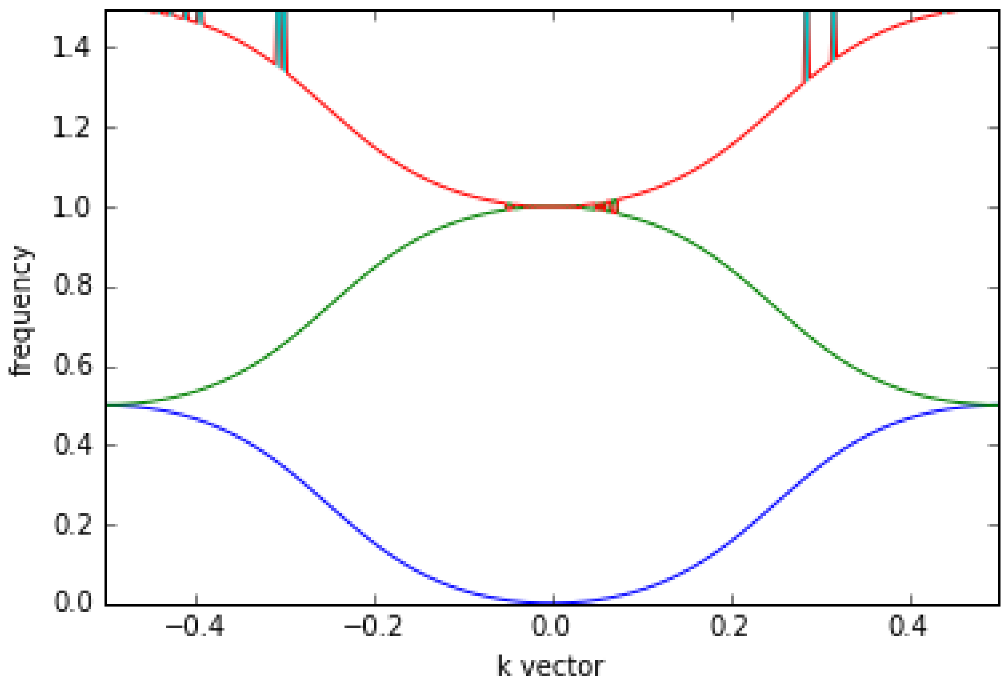
2.5 Permittivity Contrast Bandgap dependence

```
In [128]: def build_bandgap_diagram(eps1, eps2, num_eigs=6):
              n=200           # number of grid nodes

              dx=1/n          # discretization step, domain size = 1

              # Material parameters distribution

              eps = np.ones(n)
              eps[:int(n/2)-1] = 1/eps1
              eps[int(n/2):] = 1/eps2

              eps = sp.sparse.dia_matrix(([eps], [0]), [n,n])

              # Forward

              diag = np.ones(n) * -1/dx
              up_diag = np.ones(n) * 1/dx
```

```
        M_1 = sp.sparse.dia_matrix(([up_diag, diag], [1, 0]), [n,n])
        # Backward

        diag = np.ones(n) * 1/dx
        up_diag = np.ones(n) * -1/dx

        M_2 = sp.sparse.dia_matrix(([up_diag, diag], [-1, 0]), [n,n])

        M_1 = sp.sparse.lil_matrix(M_1)
        M_2 = sp.sparse.lil_matrix(M_2)

        kk0 = 2*sp.pi*np.linspace(-0.5,0.5,300) # k-vector in medium with eps

        k = np.zeros((num_eigs, kk0.size), dtype=complex)
        for ik in range(kk0.size):
            k0=kk0[ik]

            M_1[n-1, 0] = np.cos(k0*1)/dx # to impose periodic boundary condi
            M_2[0, n-1] = -np.cos(k0*1)/dx

            M = -eps*M_2*M_1

            kt = k0/np.sqrt((eps1+eps2)/2); # target k=w/c
            k2, V = linalg.eigs(M, k=num_eigs, M=None, sigma=kt**2)

            k[:,ik] = np.sqrt(k2) # k=w/c

        return kk0/(2*sp.pi), np.real(k/(2*sp.pi))
```

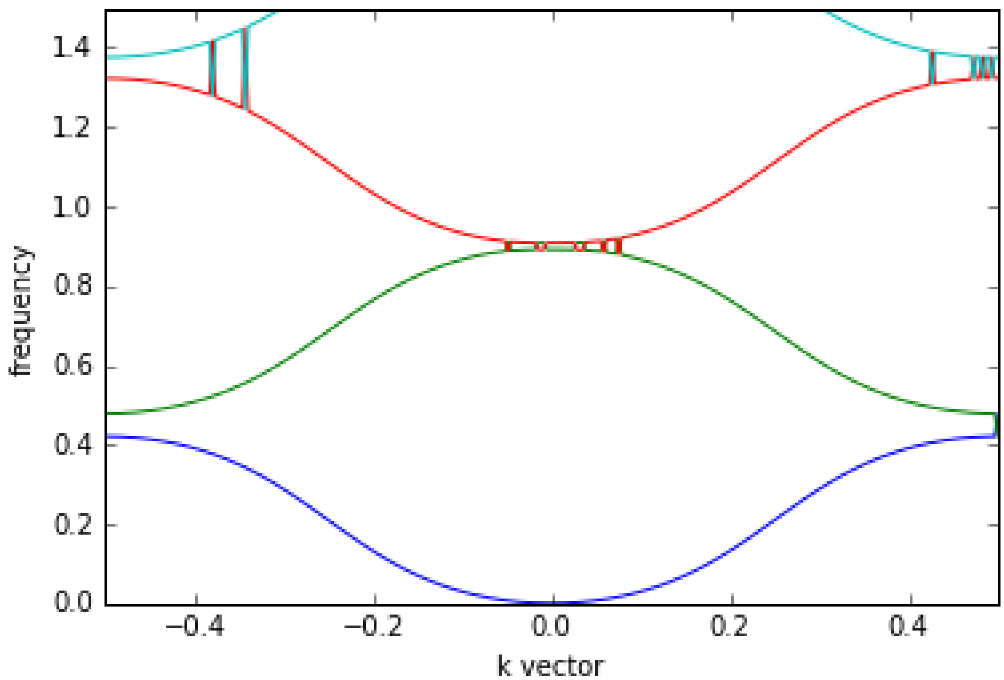
2.5.1 $\epsilon_1 = 1, \epsilon_2 = 1$

```
In [130]: k0, k = build_bandgap_diagram(1, 1, num_eigs)
          for i in range(num_eigs):
              plt.hold(True)
              plt.plot(k0, k[i,:],'-')

          plt.xlabel("k vector")
          plt.ylabel("frequency")
          plt.xlim([-0.5, 0.5])
          plt.ylim([0.0, 1.5])

Out[130]: (0.0, 1.5)
```



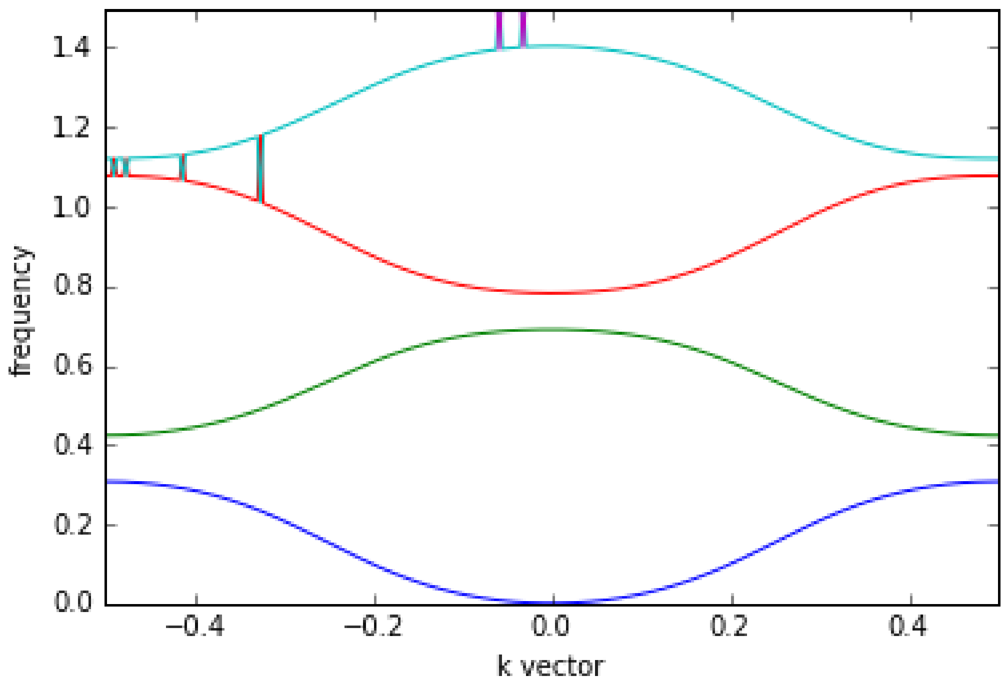
2.5.2 $\epsilon_1 = 1.5, \epsilon_2 = 1$

```
In [136]: k0, k = build_bandgap_diagram(1.5, 1, num_eigs)
          for i in range(num_eigs):
              plt.hold(True)
              plt.plot(k0, k[i,:],'-')

          plt.xlabel("k vector")
          plt.ylabel("frequency")
          plt.xlim([-0.5, 0.5])
          plt.ylim([0.0, 1.5])

Out[136]: (0.0, 1.5)
```



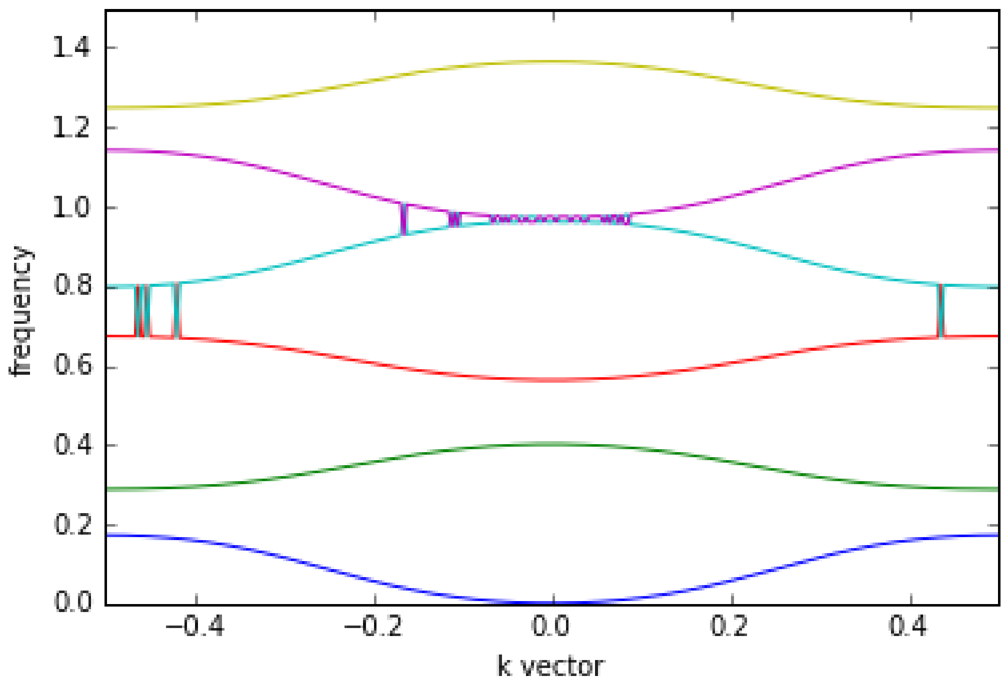
2.5.3 $\epsilon_1 = 3, \epsilon_2 = 1$

```
In [137]: k0, k = build_bandgap_diagram(3, 1, num_eigs)
          for i in range(num_eigs):
              plt.hold(True)
              plt.plot(k0, k[i,:],'-')

          plt.xlabel("k vector")
          plt.ylabel("frequency")
          plt.xlim([-0.5, 0.5])
          plt.ylim([0.0, 1.5])

Out[137]: (0.0, 1.5)
```



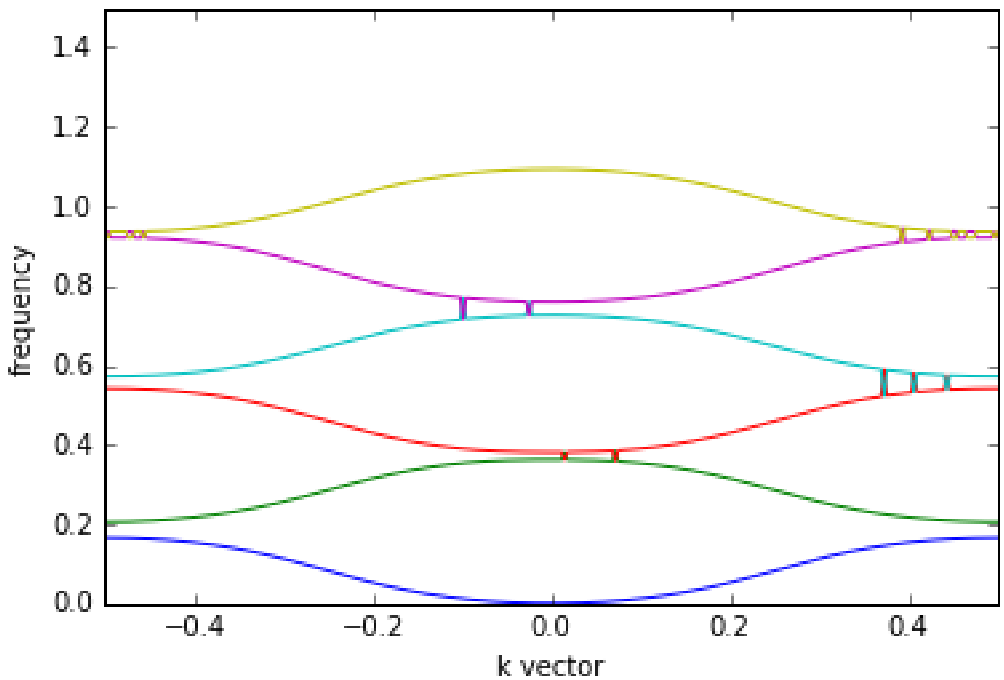
2.5.4 $\epsilon_1 = 10, \epsilon_2 = 1$

```
In [138]: k0, k = build_bandgap_diagram(10, 1, num_eigs)
          for i in range(num_eigs):
              plt.hold(True)
              plt.plot(k0, k[i,:],'-')

          plt.xlabel("k vector")
          plt.ylabel("frequency")
          plt.xlim([-0.5, 0.5])
          plt.ylim([0.0, 1.5])

Out[138]: (0.0, 1.5)
```



2.5.5 $\epsilon_1 = 10, \epsilon_2 = 5$

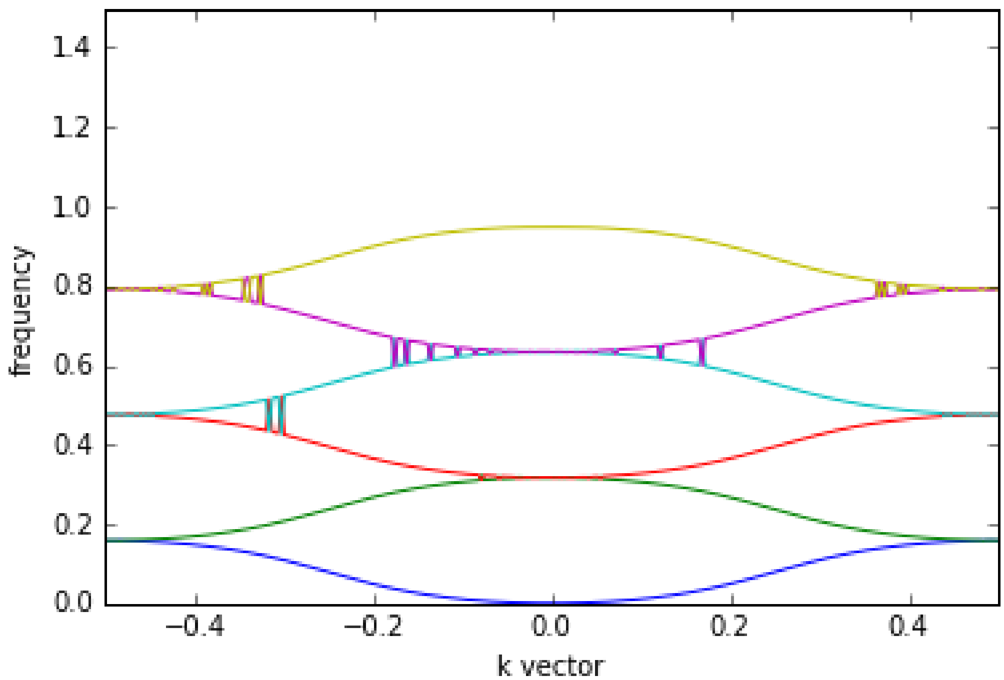
```
In [139]: k0, k = build_bandgap_diagram(10, 5, num_eigs)
          for i in range(num_eigs):
              plt.hold(True)
              plt.plot(k0, k[i,:],'-')

          plt.xlabel("k vector")
          plt.ylabel("frequency")
          plt.xlim([-0.5, 0.5])
          plt.ylim([0.0, 1.5])

Out[139]: (0.0, 1.5)
```



2.5.6 $\epsilon_1 = 10, \epsilon_2 = 10$

In [140]:
```
k0, k = build_bandgap_diagram(10, 10, num_eigs)
for i in range(num_eigs):
    plt.hold(True)
    plt.plot(k0, k[i,:],'-')

plt.xlabel("k vector")
plt.ylabel("frequency")
plt.xlim([-0.5, 0.5])
plt.ylim([0.0, 1.5])
```

Out[140]:
```
(0.0, 1.5)
```




```
In [ ]:
```