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


a1=np.array([0.5,np.sqrt(3)/2])#lattice vectors
a2=np.array([-0.5,np.sqrt(3)/2])
b1=np.array([2*np.pi,2*np.pi/np.sqrt(3)])#reciprocal lattice vecors
b2=np.array([-2*np.pi,2*np.pi/np.sqrt(3)])

N=100
m=np.linspace(-0.5,0.5,N)
n=np.linspace(-1,1,N)
M,N=np.meshgrid(m,n)

kx=M*b1[0]+N*b2[0]
ky=M*b1[1]+N*b2[1]

k1=kx*a1[0]+ky*a1[1]
k2=kx*a2[0]+ky*a2[1]

Jx=1
Jy=1
Jz=1

def refk(k1,k2,Jx,Jy,Jz):
    refk=Jz+Jx*np.cos(k1)+Jy*np.cos(k2)
    return(refk)

def Imfk(k1,k2,Jx,Jy,Jz):
    imfk=Jx*np.sin(k1)+Jy*np.sin(k2)
    return(imfk)

def xy(k1,k2,Jx,Jy,Jz):
    x=refk(k1,k2,Jx,Jy,Jz)
    y=Imfk(k1,k2,Jx,Jy,Jz)
    xy=np.array([x,y])
    return xy


def real(i,j):
    real=xy(k1[i,j],k2[i,j],Jx,Jy,Jz)[0]
    return(real)

def im(i,j):
    im=xy(k1[i,j],k2[i,j],Jx,Jy,Jz)[1]
    return(im)

def Hk(g,h):
    hk=np.array([[0,real(g,h)+(im(g,h))*(1j)],[real(g,h)-(im(g,h))*(1j),0]])
    return(hk)
```

```
def eigenvaluesgreater(g,h):
    eigenvalue=np.linalg.eigvals(Hk(g,h))[0]
    return(eigenvalue)

def eigenvalueslesser(g,h):
    eigenvalue1=np.linalg.eigvals(Hk(g,h))[1]
    return(eigenvalue1)

#print(eigenvalueslesser(9,7))


greatereigval=np.zeros((100,100), dtype=complex)
for i in range(100):
    for j in range(100):
        greatereigval[i,j]=eigenvaluesgreater(i,j)

lessereigval=np.zeros((100,100), dtype=complex)
for i in range(100):
    for j in range(100):
        lessereigval[i,j]=eigenvalueslesser(i,j)


#print(greatereigval)

#print(seteiggreatervalues)

fig=plt.figure(figsize=(10,7))
ax=fig.add_subplot(111,projection='3d')

ax.plot_surface(kx,ky,greatereigval,color='red')
ax.plot_surface(kx,ky,lessereigval,color='blue')

ax.set_xlabel('Kx')
ax.set_ylabel('Ky')
ax.set_zlabel('Ek')

plt.show()
```