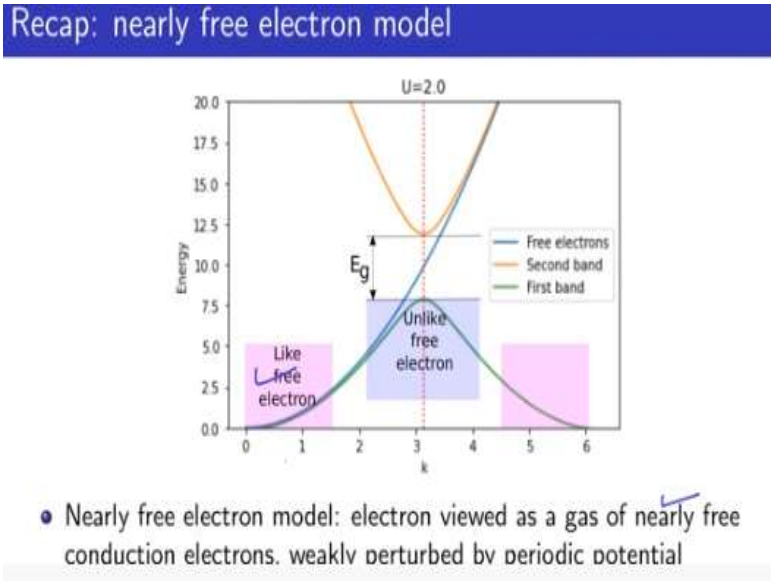
Electronic Properties of the Materials: Computational Approach
Prof. Somanth Bhowmick
Department of Materials Science and Engineering
Indian Institute of Technology, Kanpur

Module No # 07
Lecture No # 34
Tight-Binding Method: Part 1

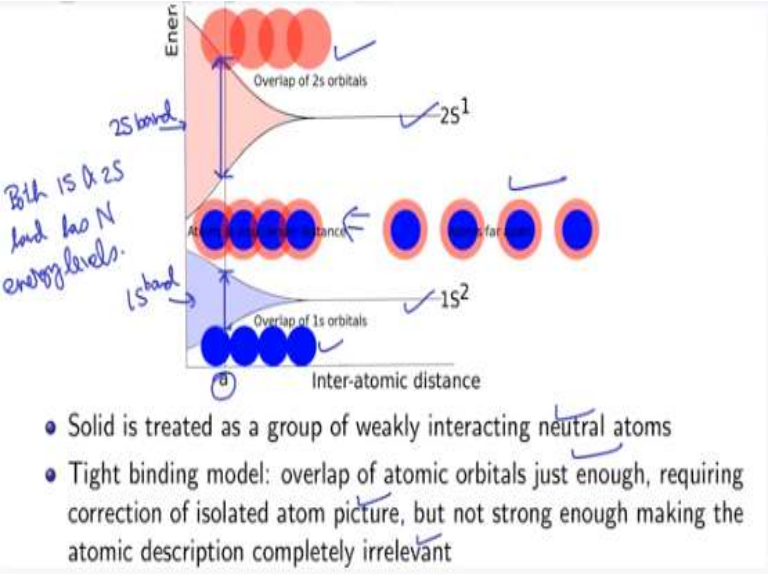
Hello friends I have discussed nearly free electron model in detail why nearly free electron model is very useful for qualitative understanding of effects of periodic potential on electron in a solid. Nearly free electron model is not generally used to calculate electronic damage structure. There are several methods available to calculate electronic band structure such methods vary in terms of accuracy speed and computational cost. In this lecture I am going to introduce a very simple method widely used for electronic band structure calculation the type binding method.

(Refer Slide Time: 01:01)



In nearly free electron model electron is viewed as a gas of nearly free conduction electrons thank you partnered by the periodic potential of nearly free electron is very similar to free electron energy dispersion near the zone center. And effect of weak perturbation by periodic potential is visible only near the black planes marked by the red dashed vertical lines in this figure.

(Refer Slide Time: 01:41)

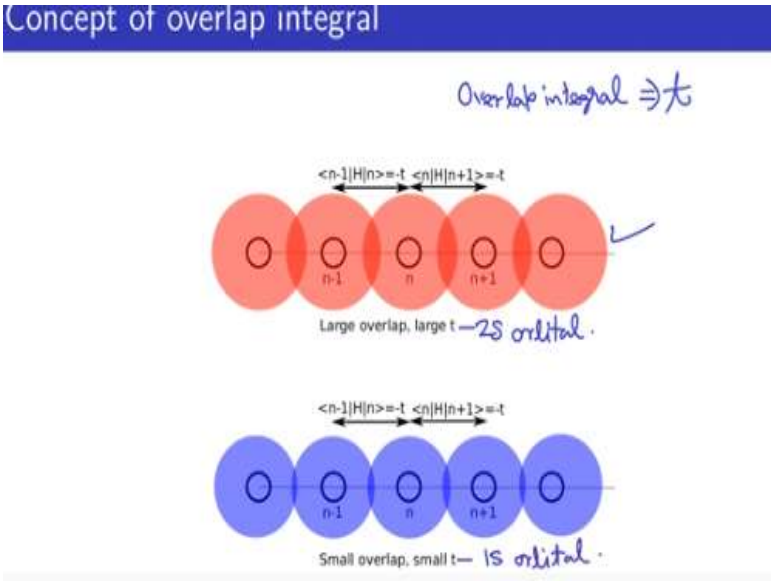


Tight binding model starts from a completely different point of theme it fits the solid as a collection of weekly interacting neutral atoms. For example take lithium atom it has 2 electrons in 1s orbital and one electron in 2s orbital if the atoms are far apart from each other in space energy levels of the orbitals are same as the atomic energy levels corresponding to 1s and 2s orbitals.

However as we move the isolated lithium atoms closer to each other atomic orbitals start to overlap with each other. As a result in an atom start to feel the presence of the neighboring atoms and the energy levels start deviating from the isolated atomic energy levels. In other words from in isolated atomic energy levels we get an energy bank having n states this is the one is band and this is the 2s band each band has N energy levels.

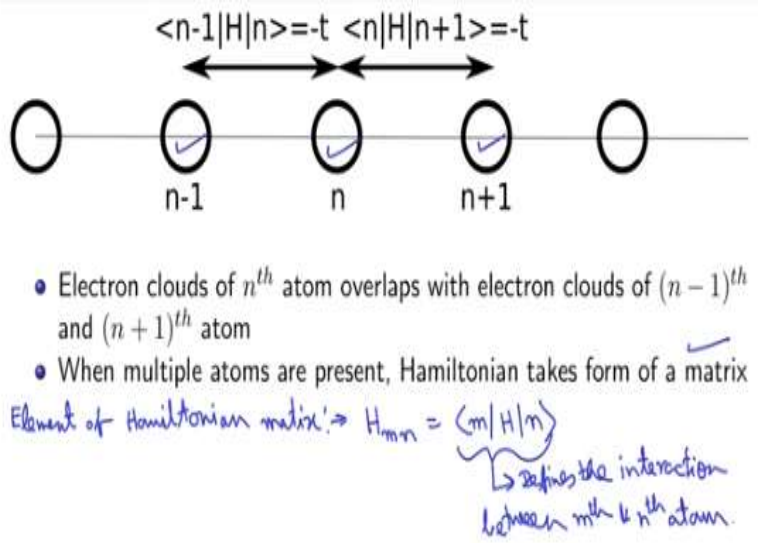
Width of the energy band depends on the overlap of atomic or by chance for example 2s orbitals overlap more than 1s orbitals as a result 2s band in lithium has larger bandwidth than one is note that bandwidth is calculated at the equilibrium separation. Thus this is the width of the 2s band which is much larger than the width of the 1s band note that type binding model deals with a case where overlap of atomic orbitals is just enough requiring correction of isolated atomic picture but not strong enough making the atomic description completely irrelevant.

(Refer Slide Time: 04:41)



Tight binding model is not a having issue method where we can get the energy levels from first principles. The model we are going to use is a parameter based model the parameter is overlap integral denoted by t. You need not to know how to get t at this moment it is sufficient to know that more the overlap between 2 orbitals more will be p if you think in terms of lithium atom 2s orbitals overlap more and p for 2s orbitals is large on the other hand 1s overlap less and t for 1s orbital is small.

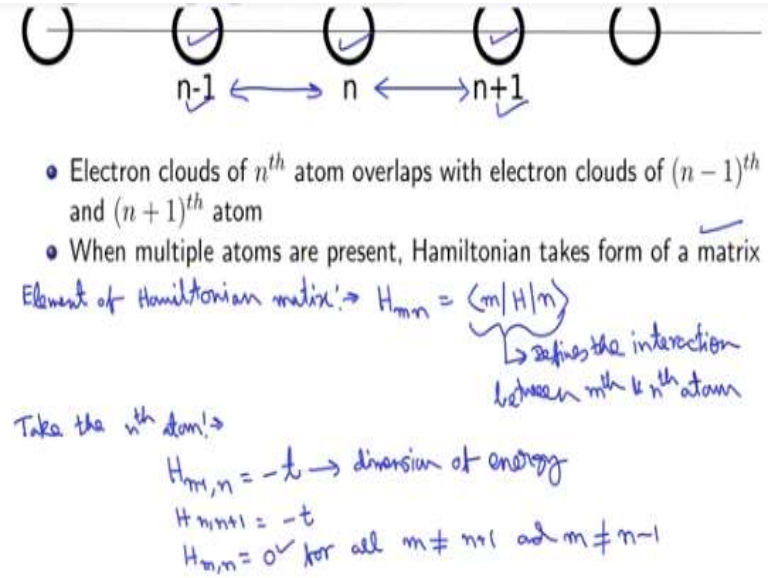
(Refer Slide Time: 05:56)



Let us start with a 1d lattice it is reasonable to assume that the electron cloud of the nth atom is overlapping with the electron cloud of the neighboring atoms that is n- 1 atom and n+ 1 atom. Which are the nearest neighbors of the nth atom when multiple atoms are present the Hamiltonian

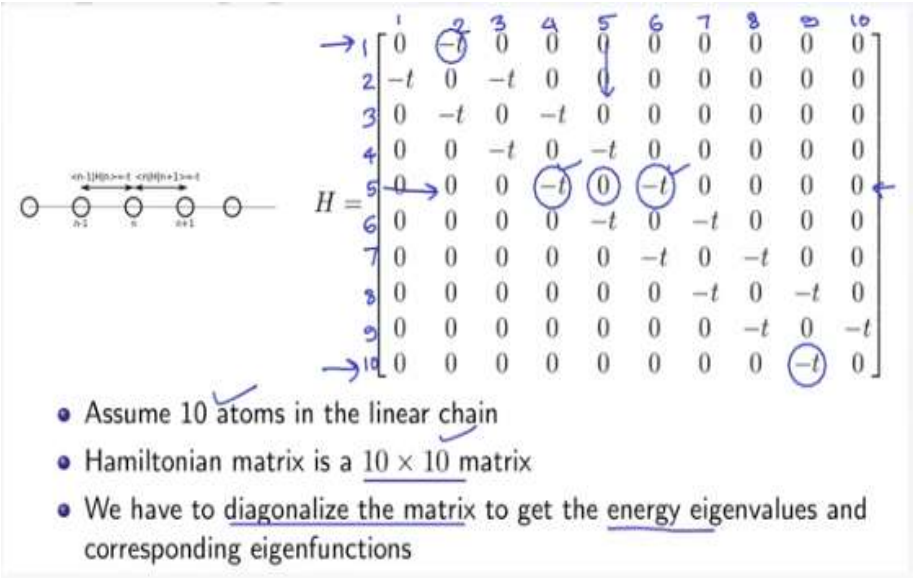
takes the form of a matrix. Let us learn how to build the Hamiltonian matrix, element of Hamiltonian matrix is expressed as h m n which is written in this form this defines the interaction in between mth and nth atom.

(Refer Slide Time: 07:51)



Now take the nth atom it has 2 nearest neighbours n-1 and n+1. Then the element of Hamiltonian matrix corresponding to the nth atom are H n- 1, n = -t where p has the dimension of energy. Similarly for the other neighbor we have H n+ 1 = -t H m n is equals to 0. For all m not equal to n +1 and for all m not equal to n- 1 because there is no interaction other than the interaction between the nearest neighbors. If there is no interaction those terms will be 0 and if there is some interaction between 2 atoms those terms will be equal to –t.

(Refer Slide Time: 09:31)



Let us take some specific example let us assume that we have 10 atoms in the linear chain in that case the Hamiltonian matrix is a n cross 10 matrix as shown here. Let me mark each row and column of the matrix first so this is column 1 2 3 4 5 6 7 8 9 10 and this is row 1 2 3 4 5 6 7 8 9 10. select some atom say the fifth atom this is the row corresponding to the fifth atom. And this is the column corresponding to the fifth atom this is the location of the fifth atom in the matrix.

Atomic orbital of the fifth atom overlaps with the atomic orbital of the fourth atom that is why this term is non-zero. Similarly atomic orbital of the fifth atom overlaps with the atomic orbital of the sixth atom that is why this term is non-zero. The atomic orbital of the fifth atom is not overlapping with any other atom that is why rest of the terms are 0 in this row. Thus in each row we have 2 non-zero terms the only exception is the first and last row.

The first atom has only one neighbor the second atom because the first atom is located at one edge of the chain similarly the last atom has only 1 neighbor because the last atom is located at 1 edge of the chain. Thus we have a cluster Hamiltonian matrix we have to diagonalizable the matrix to get the energy Eigen values and corresponding Eigen functions.

(Refer Slide Time: 12:11)

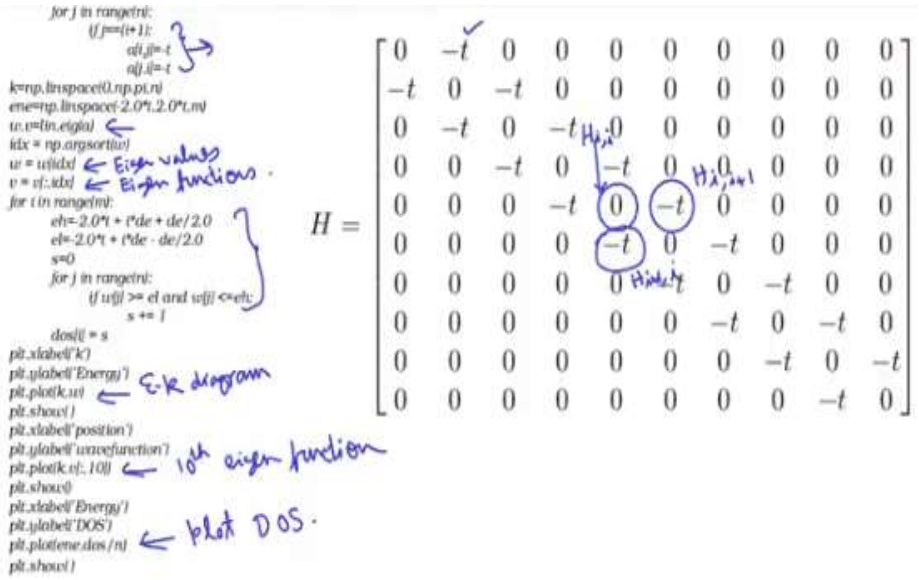
```
n=int(input("Enter the size of the chain: "))
m=30
t=1.0
a=np.zeros((n,n),float)
dos=np.zeros(m,float)
de=4.0*t/m
for i in range(n):
    for j in range(n):
        if j==(i+1):
            a[i,j]=-t
            a[j,i]=-t
k=np.linspace(0,np.pi,n)
ene=np.linspace(-2.0*t,2.0*t,m)
w,v=lin.eig(a)
idx = np.argsort(w)
w = w[idx]
v = v[:,idx]
for i in range(m):
    eh=-2.0*t + i*de + de/2.0
    el=-2.0*t + i*de - de/2.0
    s=0
    for j in range(n):
        if w[j] >= el and w[j] <=eh:
            s += 1
    dos[i] = s
plt.xlabel('k')
plt.ylabel('Energy')
plt.plot(k,w)
plt.show()
plt.xlabel('position')
```

$$H = \begin{bmatrix} 0 & -t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -t & 0 & -t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -t & 0 & -t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -t & 0 & -t & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -t & 0 & -t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -t & 0 & -t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -t & 0 & -t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -t & 0 & -t & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -t & 0 & -t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -t & 0 \end{bmatrix}$$

Let me show a python code to construct the Hamiltonian matrix get the energy Eigen values and also get the Eigen functions. The code is going to ask for the number of atoms in the linear chain m is the number of energy intervals for density of states calculation t has the dimension of energy I create the n cross n Hamiltonian matrix in this step. This is an array to store the values of density of states; d e is the energy interval for density of states calculation.

Now I construct the Hamiltonian matrix take ith atom say this one this is the h i i term. This term is h i i+ 1 and this term is h i i+ 1 i and both of them are equal to minus t. This is what I am defining here this line calculates the Eigen values and Eigen vectors of the Hamiltonian matrix; w holds the Eigen values and v holds the Eigen functions. Then we find the density of states by calculating number of energy Eigen values in a given interval.

(Refer Slide Time: 15:03)



Finally we plot the e k diagram; we plot the nth Eigen function; and plot the density of states.

(Refer Slide Time: 15:32)

```
import numpy as np
import matplotlib.pyplot as plt
import numpy.linalg as lin
n=int(input("Enter the size of the chain: "))
m=30
t=1.0
#Hamiltonian matrix
a=np.zeros([n,n],float)
#Array to store DOS
dos=np.zeros(m,float)
#Energy interval to get DOS
de=4.0*t/m
#For loop to create Hamiltonian matrix
for i in range(n):
    for j in range(n):
        if j==(i+1):
            a[i,j]=-t
            a[j,i]=-t
k=np.linspace(0,np.pi,n)
x=np.linspace(0,n,n)
ene=np.linspace(-2.0*t,2.0*t,m)
#Finding eigenvalues and eigenvectors of Hamiltonian matrix
w,v=lin.eig(a)
idx = np.argsort(w)
w = w[idx]
v = v[:,idx]
#Finding DOS
for i in range(m):
```

This is the python code to get the energy Eigen values and energy Eigen vectors of a 1d lattice. You are already familiar with numpy and mat plot library. I have added 1 new library for the linear algebra calculation this is going to be used for calculating the energy Eigen values and energy Eigen functions this is where the code asks for the size of the chain. That is you have to enter the number of atoms in the linear chain m is the number of intervals for dos calculation the Hamiltonian matrix is defined here this is the array for storing density of states. This is the energy interval for density of states calculation; this is the loop to create the Hamiltonian matrix.

(Refer Slide Time: 16:36)

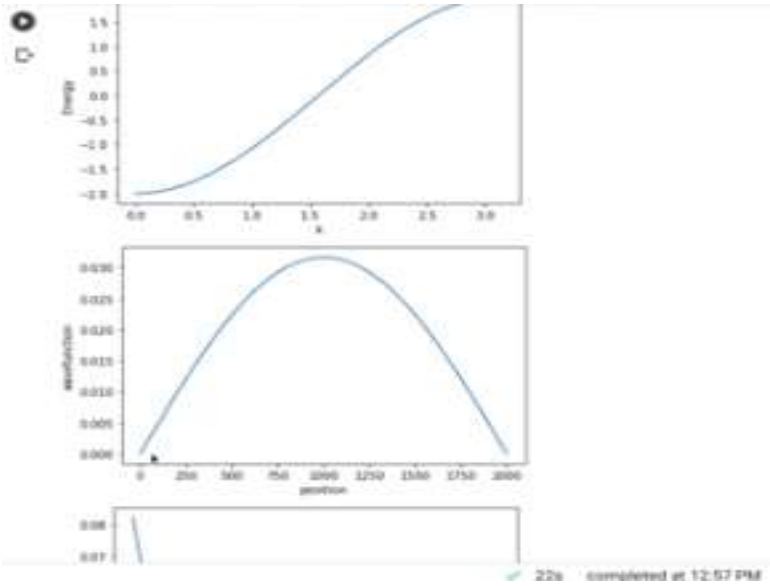
```
#Finding DOS
for i in range(m):
    eh=-2.0*t + i*de + de/2.0
    el=-2.0*t + i*de - de/2.0
    s=0
    for j in range(n):
        if w[j] >= el and w[j] <=eh:
            s += 1
    dos[i] = s
plt.xlabel('k')
plt.ylabel('Energy')
#Plotting energy dispersion
plt.plot(k,w)
plt.show()
#Plotting wavefunction
plt.xlabel('position')
plt.ylabel('wavefunction')
plt.plot(x,v[:,0])
plt.show()
#Plotting DOS
plt.xlabel('Energy')
plt.ylabel('DOS')
plt.plot(ene,dos/n)
plt.show()
```

```
Enter the size of the chain: 2000
```

✓ 23s completed at 12:51 PM

This is where we are finding the Eigen values and Eigen functions of the Hamiltonian matrix this is where we are calculating the density of states. Finally we plot the energy dispersion we plot the wave function and we plot the density of states. Let us plot the ground state wave function.

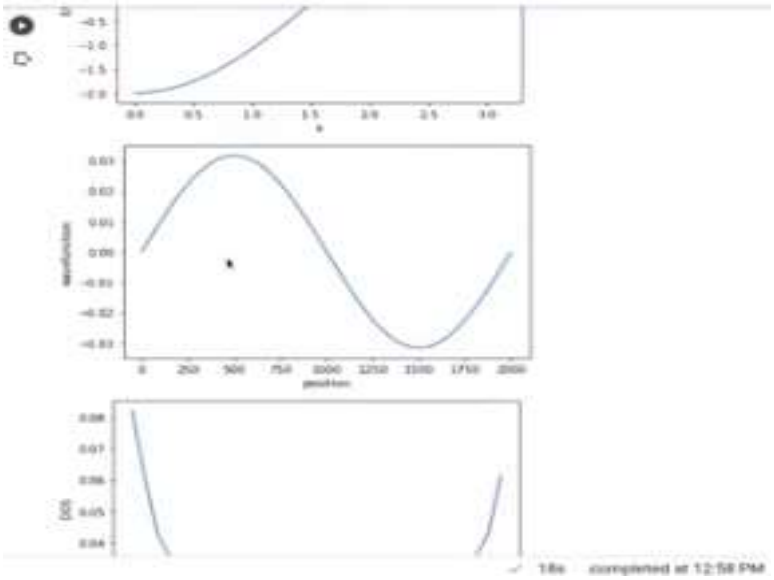
(Refer Slide Time: 17:14)



✓ 22s completed at 12:57 PM

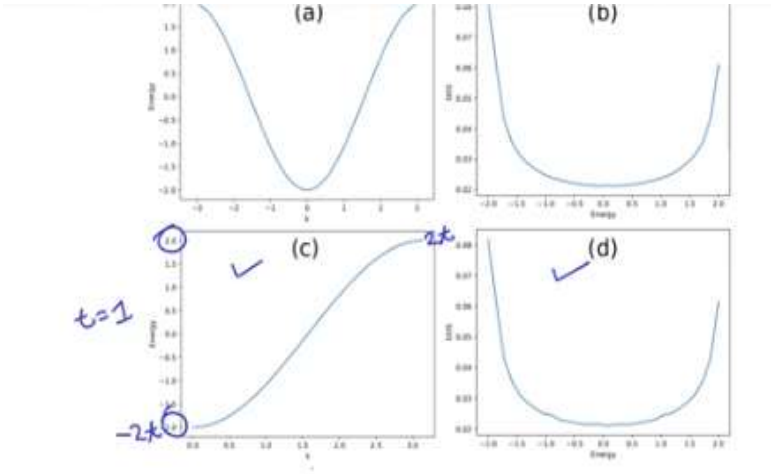
The code is asking for number of atoms in the linear chain let us enter 2000. This is how the energy dispersion looks like this is the plot of the ground state wave function note that it has no nodes as expected. And this is the plot of density of states for a 1d chain. Let us plot some other wave function say that first excited state again let us enter 2000 and run the code.

(Refer Slide Time: 17:54)



So, this is the first excited state wave function and you see that there is only one node. Now let us do for nth excited state let us enter 2000 and this is the plot for the 10th excited state Eigen function.

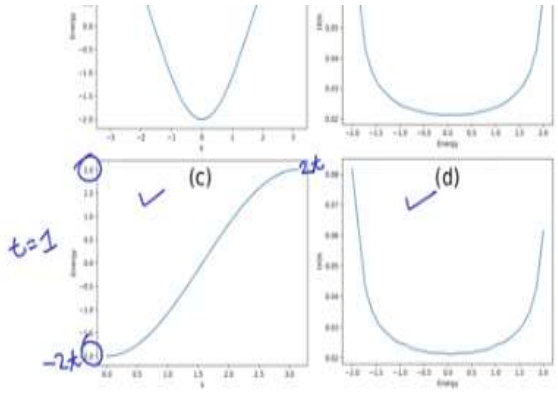
(Refer Slide Time: 18:15)



- $\varepsilon_m = -2t$ for $\varepsilon < 0$ and $\varepsilon_m = +2t$ for $\varepsilon > 0$
- Width of the band (range of max and min ε in band): $4t$

From our code we obtain the energy dispersion and density of states for a 1d lattice having 2000 atoms. We have used t = 1; and the band minimum band maximum has a value of -2t and +2t the width of the band is equal to 4t. The weight is defined as the range of maximum and minimum energy in the band.

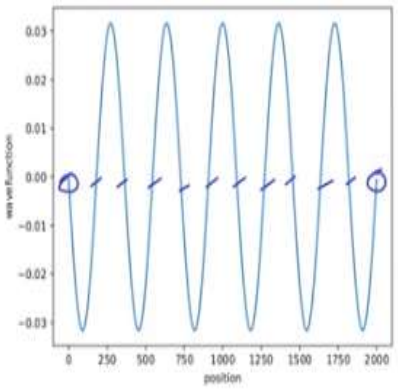
(Refer Slide Time: 18:57)



- $\varepsilon_m = -2t$ for $\varepsilon < 0$ and $\varepsilon_m = +2t$ for $\varepsilon > 0$
- Width of the band (range of max and min ε in band): $4t$
- Analytical result: $\varepsilon(k) = -2t\cos(ka)$ and $g(\varepsilon) \sim \frac{1}{\sqrt{(\varepsilon_m^2 - \varepsilon^2)}}$

It can be shown analytically that the energy dispersion relation is E k - 2t Cos ka and that density of states is equal to 1 divided by e m square minus e square; where E m square = 4 t square. Analytical results are plotted in a and b it matches very well with what we obtain from our python code. Note that in our code we have plotted E k only for k greater than 0; y for the analytical plot E k has been plotted for positive as well as negative k values in the first Brillouin zone.

(Refer Slide Time: 20:03)

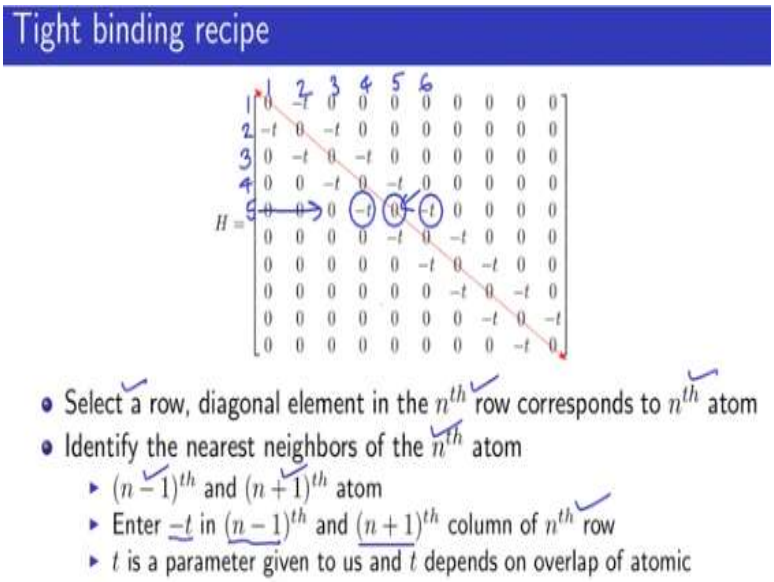


- Plot of the 10^{th} eigenfunction from the code
- Verify: the eigenfunction has 10 nodes
- The first and last zero of the eigenfunction are not counted as nodes,

We also have plotted the energy Eigen functions using our code this is the plot for the tenth Eigen function from the code we can verify whether it has 10 nodes or not node 1 2 3 4 5 6 7 8 9 10 that is the wave function indeed has 10 nodes. Note that the first 0 of the wire function and the last 0

of the y function are not counted as nodes as they happen as a consequence of a fixed boundary condition.

(Refer Slide Time: 20:42)



Let me give you a simple tight binding recipe which can be easily implemented to take into account nearest neighbor interactions as well as beyond the nearest neighbor interactions. Later this recipe is also going to help us to switch to higher dimension like 2 dimensions and 3 dimension. Step 1 select a row diagonal element in the nth row corresponds to the nth atom. For example let us take the fifth row 1 2 3 4 5 diagonal element in the fifth row corresponds to the fifth atom.

Step 2 identify the nearest neighbors of the nth atom in case of this is the n- 1 and n+ 1 atom; enter -t in n- 1 and n+ 1 column of the nth row. For example take the fifth atom fourth and sixth atom are the nearest neighbors of the fifth atom. So, we take the fifth row and we go to column 4 of the fifth row 1 2 3 4. So, that means this is the column 4 of fifth row and this is the column 6 of fifth row and we enter minus t in both the places.

(Refer Slide Time: 22:24)

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -t & 0 & -t & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -t & 0 & -t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -t & 0 \end{bmatrix}$$

- Select a row, diagonal element in the n^{th} row corresponds to n^{th} atom
- Identify the nearest neighbors of the n^{th} atom
  - $(n-1)^{th}$ and $(n+1)^{th}$ atom
  - Enter $-t$ in $(n-1)^{th}$ and $(n+1)^{th}$ column of n^{th} row
  - t is a parameter given to us and t depends on overlap of atomic orbitals of nearest neighbors
- Extend the model: identify the next nearest neighbors of the n^{th} atom
  - $(n-2)^{th}$ and $(n+2)^{th}$ atom
  - Enter $-t'$ in $(n-2)^{th}$ and $(n+2)^{th}$ column of n^{th} row
  - $t' << t$, as they depend on overlap of atomic orbitals

Remember that t is a parameter given to us and t depends on overlap of atomic orbitals of the nearest neighbours. Now we can extend the model to include the next near neighbour interactions do that we have to identify the next near neighbours of the nth atom. In case of 1d lattice this is the n- 2th and n+ 2th atom. And we have to enter -t dashed in n- 2th and n+ 2th column of the nth row.

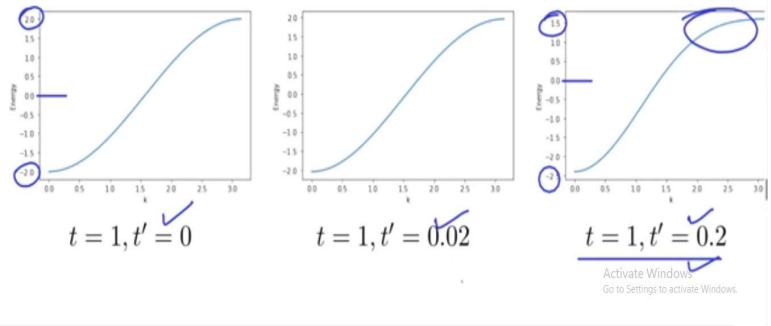
For example again let us take the fifth atom this one now the third atom and the seventh atom is the second near neighbor of the fifth atom. So, we select the fifth row and we go to the third column of the fifth row and enter - t dashed similarly if we go to the seventh column of the fifth row and enter - t dashed. Since p depends on overlap of atomic orbitals t dashed has to be significantly smaller than t.

(Refer Slide Time: 23:47)

```
m=30
t=1.0
t1=0.02
#Hamiltonian matrix
a=np.zeros([n,n],float)
#Array to store DOS
#dos=np.zeros(m,float)
#Energy interval to get DOS
#de=4.0 * t/m
#For loop to create Hamiltonian matrix
for i in range(n):
  for j in range(n):
    #Nearest neighbor
    if j==(i+1):
      a[i,j]=-t
      a[j,i]=-t
    #Next nearest neighbor
    if j==(i+2):
      a[i,j]=-t1
      a[j,i]=-t1
k=np.linspace(0,np.pi,n)
x=np.linspace(0,n,n)
#ene=np.linspace(-2.0*t,2.0*t,m)
#Finding eigenvalues and eigenvectors of Hamiltonian matrix
w,v=lin.eig(a)
idx = np.argsort(w)
w = w[idx]
v = v[:,idx]
#Finding DOS
```

We can modify the existing code to impute next nearest neighbours let me do that. If j = i+ 2 a i j = -t1 a g i = -.t1 the value of p1 i have chosen to be equal to 0.02 now let us run the code and plot the energy dispersion relation so the code is asking for the number of atoms in the chain so let me enter 2000 this is how the energy dispersion looks like. Let me enter a larger value of t1 so that you can clearly see the effect of second near neighbour interaction so let me enter t1 = 0.2 and run the code. So again we enter 2000 so this is how the energy dispersion looks like.

(Refer Slide Time: 25:05)



Comparing the plots of energy dispersion for different values of next near neighbour interaction we find significant difference when t dashed become comparable to t as shown in the third figure; E k curve becomes flatter near the top of the band moreover in case of t dashed = 0 the spread of the band is symmetric about 0. So this is the zero energy maximum value of energy in the band is + 2 and minimum value of the energy in band is -2.

However in case of deduct equal to 0.2 the spread of the band above 0 is no longer symmetric the maximum value of energy in the band is 1.5 and the minimum value is -2.5. However note that t dashed = 0.2 is a very high value for a 1d lattice for t = 1 t dash should be much smaller than 0.2.