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

Lecture - 15
Sommerfeld model: Selected Numerical Exercise Using Python

Hello friends, we have discussed Sommerfeld theory of free electrons in detail. In this lecture, we are going to verify some of the analytical results using simple Python codes.

(Refer Slide Time: 00:26)

Let us start our discussion with density of states. So, how did we discover density of states? We know that thermodynamic property can be calculated by k space integration of this form where f of E is the Fermi function given by $\frac{1}{e^{(E(k)-\mu)/k_B T}+1}$ .The other function F of k depends on the property that we want to calculate. For example, $F(\vec{k}) = 1$, for electron density and $F(\vec{k}) = \frac{\hbar^2 k^2}{2m}$ for energy density.

For example, we can calculate the electron density by calculating the integral of this for note that the integral is over k. Now, using E = h cross square k square divided by 2m, we can convert the k space integration to energy integration.

(Refer Slide Time: 01:43)

And we can write N by V as integral of density of states times the Fermi function and this integral is over the energy thus, we discovered density of states in the process of converting k space integration to energy integration.

(Refer Slide Time: 02:07)

How did we discover density of states (DOS)?

- $n = \dfrac{N}{V} = \int_0^\infty d\varepsilon \underbrace{\dfrac{m}{\hbar^2\pi^2}\sqrt{\dfrac{2m\varepsilon}{\hbar^2}}}_{g(\varepsilon)=DOS} f(\varepsilon) = \int_{-\infty}^\infty d\varepsilon\, g(\varepsilon) f(\varepsilon)$

  $T=0K,\ f(\varepsilon)=1,\ \varepsilon<\varepsilon_F$
  $f(\varepsilon)=0,\ \varepsilon>\varepsilon_F$

- For 3D free electron gas:
  - $g(\varepsilon) = \dfrac{m}{\hbar^2\pi^2}\sqrt{\dfrac{2m\varepsilon}{\hbar^2}}$
  - $g(\varepsilon) = 0, \varepsilon < 0$
- What is density of states (DOS)?
- At $T = 0$ K, $n = \dfrac{N}{V} = \int_0^{\varepsilon_F} d\varepsilon\, g(\varepsilon)$
- Total area under curve $g(\varepsilon) \equiv \dfrac{1}{V} \times$ (total number of energy levels between 0 and ε_F)

Electron density is given by the integral of density of states times the Fermi function, where density of states for 3D electron gas goes as the square root of E note that density of states is equal to 0 when energy is less than 0, meaning of density of states can be easily understood by calculating the electron density at T = 0 Kelvin. At T = 0 Kelvin, electron density is given by integral 0 to E F d times gE note that since g of E = 0 for energy less than 0.

The lower limit of this integral is set to 0 on the other hand, Fermi function is a step function at T = 0 Kelvin, that is at T = 0 Kelvin f of E = 1 for E less than E F and f of E = 0 or E greater

than E F. As a result of this the upper limit of the integral is set to E of F integral is total area under the curve as shown in the diagram, total area under the curve shaded in red is equal to the number of energy levels between 0 and E F.

(Refer Slide Time: 03:49)

If we take a small strip of width dE, the area of the strip is gE dE as shown by the blue shaded region in that figure, since that total area under the density of states curve represents the total number of energy levels this small strip also should represent some small number of energy levels thus g of E dE is equal to number of energy levels between E and E + dE. Let us see, instead of getting density of states as by-product of change of variable from K to E is there a direct way of deriving density of states.

(Refer Slide Time: 04:34)

Let me show a numerical scheme to get density of states. I show it for 2D electron gas and the same scheme will work for 1D or 3D electron gets in 2D energy is given by h cross square by 2m k x square + k y square. We know that each point in k space represents an energy level filling k space with electrons is same as filling energy levels with electrons. For example, filling the 0 0 point in k space this is like filling the lowest energy level.

Filling the 1 0 point and there are 4 such points of equivalent energy in k space is like filling the next energy level and so on. In the rightmost diagram, I show total number of states at each energy for example, when energy = 0, there are 2 states when energy = 1 there are 8 states when energy = 2 there are 8 states and so on.

(Refer Slide Time: 05:46)

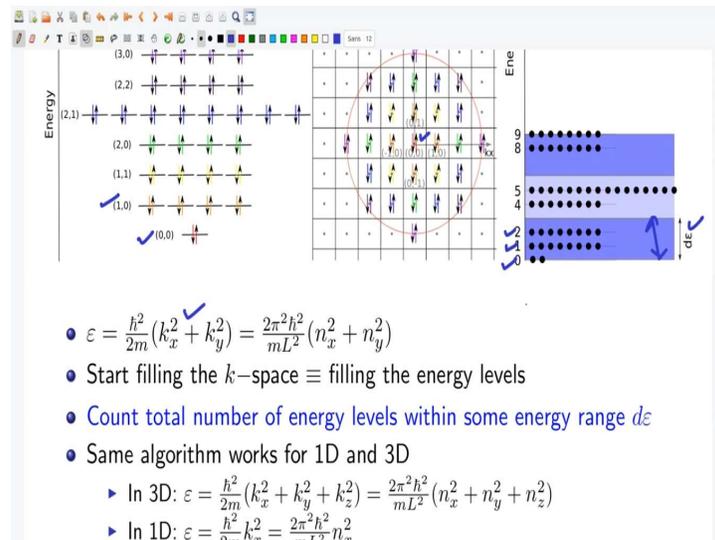- $\varepsilon = \frac{\hbar^2}{2m}(k_x^2 + k_y^2) = \frac{2\pi^2\hbar^2}{mL^2}(n_x^2 + n_y^2)$
- Start filling the $k-$space $\equiv$ filling the energy levels
- Count total number of energy levels within some energy range $d\varepsilon$
- Same algorithm works for 1D and 3D
  - In 3D: $\varepsilon = \frac{\hbar^2}{2m}(k_x^2 + k_y^2 + k_z^2) = \frac{2\pi^2\hbar^2}{mL^2}(n_x^2 + n_y^2 + n_z^2)$
  - In 1D: $\varepsilon = \frac{\hbar^2}{2m}k_x^2 = \frac{2\pi^2\hbar^2}{mL^2}n_x^2$

Now, we split the entire energy range in small energy strips of width dE. Finally, we count the total number of energy levels within the energy range dE for example, in this case, within this energy range; we have 8 + 8 16 + 2 18 energy levels. This is the algorithm to numerically calculate terms, the same algorithm will work for 1D and 3D.

(Refer Slide Time: 06:25)

```python
import numpy as np
import matplotlib.pyplot as plt
#Number of k-points
n=100
#Energy intervals for DOS
ndos=50
dos = np.zeros(ndos,float)
en = np.zeros(ndos,float)
de = 120
#Loop to get DOS for 3D
for i in range(ndos):
    emin = i * de
    emax = (i+1) * de
    en[i] = emin + de/2
    for j in range(n):
        for k in range(n):
            for l in range(n):
                eval = (j * j + k * k + l * l)
                if emin < eval < emax:
                    dos[i] += 1
maxdos = max(dos)
maxen = max(en)
#Plotting
plt.ylim(0,1)
plt.xlabel("Energy")
```

- Specify max value of n_x, n_y, n_z
- Run for loop to get : $\varepsilon = \frac{2\pi^2\hbar^2}{mL^2}(n_x^2 + n_y^2 + n_z^2)$
- Split energy in small strips of width $d\varepsilon$ for DOS calculation
- DOS: count how many energy levels lying in a particular energy strip

Let us write a code to calculate density of states for a free electron gas in 3D, we need to specify a range of n x, n y and n z and we take all of them to be equal to 100 as shown here, we run a for loop to get the total energy E as a function of n x, n y and n z and the energy value is calculated here.

(Refer Slide Time: 06:54)


```python
#Energy intervals for DOS
ndos=50
dos = np.zeros(ndos,float)
en = np.zeros(ndos,float)
de = 120
#Loop to get DOS for 3D
for i in range(ndos):
    emin = i * de
    emax = (i+1) * de
    en[i] = emin + de/2
    for j in range(n):
        for k in range(n):
            for l in range(n):
                eval = (j * j + k * k + l * l)
                if emin < eval < emax:
                    dos[i] += 1
maxdos = max(dos)
maxen = max(en)
#Plotting
plt.ylim(0,1)
plt.xlabel("Energy")
plt.ylabel("DOS")
plt.plot(en,dos/maxdos, label='3D DOS', marker='o')
plt.plot(en,np.sqrt(en/maxen), color='r', label='$\sqrt{E}$')
plt.legend()
plt.show()
```

- Run for loop to get : $\varepsilon = \frac{2\pi^2\hbar^2}{mL^2}(n_x^2 + n_y^2 + n_z^2)$
- Split energy in small strips of width $d\varepsilon$ for DOS calculation
- DOS: count how many energy levels lying in a particular energy strip

We split energy in small strips of width de for dos calculation. For every small strip ranging from emin to emin + de we count the number of energy levels in that energy range, this gives us the value of density of states finally, we plot density of states as a function of energy.
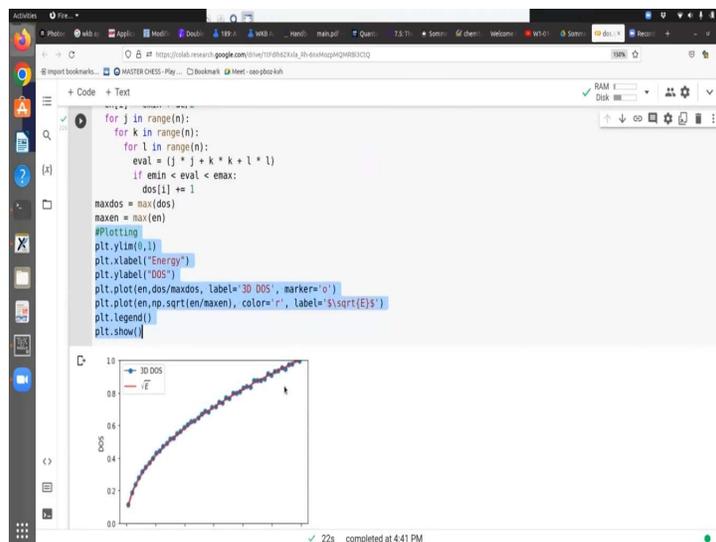
(Refer Slide Time: 07:33)

```
#Code to calculate DOS for 3D
import numpy as np
import matplotlib.pyplot as plt
#Number of k-points
n=100
#Energy intervals for DOS
ndos=50
dos = np.zeros(ndos,float)
en = np.zeros(ndos,float)
de = 120
#Loop to get DOS for 3D
for i in range(ndos):
    emin = i * de
    emax = (i+1) * de
    en[i] = emin + de/2
    for j in range(n):
        for k in range(n):
            for l in range(n):
                eval = (j * j + k * k + l * l)
                if emin < eval < emax:
                    dos[i] += 1
maxdos = max(dos)
maxen = max(en)
#Plotting
plt.ylim(0,1)
plt.xlabel("Energy")
plt.ylabel("DOS")
plt.plot(en,dos/maxdos, label='3D DOS', marker='o')
plt.plot(en,np.sqrt(en/maxen), color='r', label='$\sqrt{E}$')
```

This is the code for calculating DOS in 3D. Here we define the number of k- points this is where we define the energy intervals for dos, we have created 1 array to store the value of dos, then we run a for loop to get the density of states. So, in the for loop, first we define the energy interval and then we run for loop to get the total energy the total energy is calculated in this step. Then if the total energy for a given value of j, k and l lies within the energy interval we defined here, then we just update our array for storing the value of dos.

**(Refer Slide Time: 08:34)**

And then we plot the density of states as a function of energy in this step. If you run the code then this is what you will get.

(Refer Slide Time: 08:45)
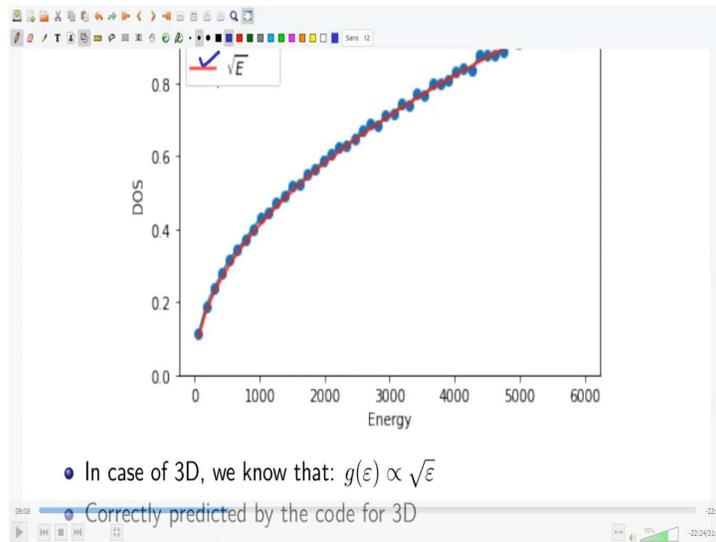
Thus the density of states for 3D electron gas fits very well with the light square root of E this is shown by the red colour we already know that for 3D electron gas g of E goes as square root of E thus the code managed to find the density of states for 3D free electron gas correctly, write the code to find the density of states for 1D and 2D free electron gas.

(Refer Slide Time: 09:14)
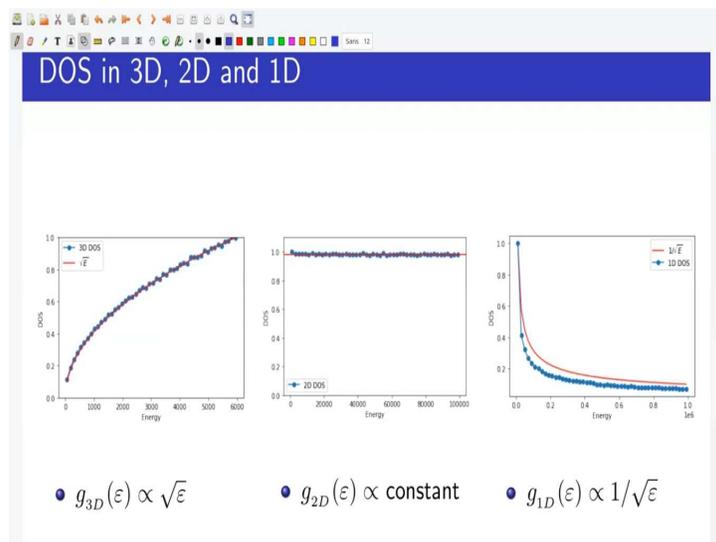
In this slide, I show the density of states offer 3D 2D and 1D free electron gas side by side. Note that in case of 3D the density of states goes as square root of E. In case of 2D, the density of states is a constant and in case of 1D, the density of states goes as 1 by square root of E.

(Refer Slide Time: 09:40)

Derivation of DOS for 3D and 2D

$\Omega_{3D} = \frac{4}{3}\pi k^3$ $\Omega_{2D} = \pi k^2$

$\Delta k_{3D} = \frac{8\pi^3}{V}$ $\Delta k_{2D} = \frac{4\pi^2}{A}$

$\frac{\Omega_{3D}}{\Delta k_{3D}}$ $\frac{\Omega_{2D}}{\Delta k_{2D}}$

- $k^3 = 3\pi^2 n$, where $n = N/V$
- Replace k by ε using: $\varepsilon = \frac{\hbar^2 k^2}{2m}$

- $k^2 = 2\pi n$, where $n = N/A$
- Replace k by ε using: $\varepsilon = \frac{\hbar^2 k^2}{2m}$
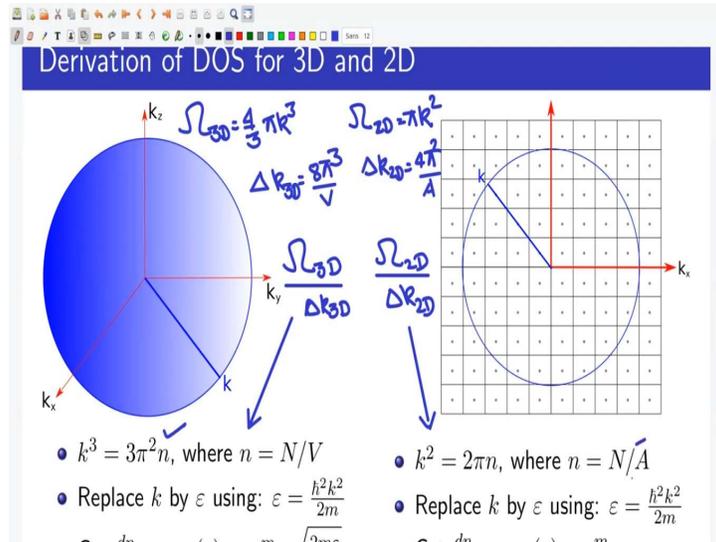
Let us derive density of states for 3D and 2D free electron gas. In case of 3D we take a sphere and volume of the sphere in case this is given by $\Omega_{3D} = \frac{4}{3}\pi k^3$ in case of 2D we take a circle and area of the circle k space is given by $\Omega_{2D} = \pi k^2$. Now, we know that the volume occupied by a single point in k space in 3D is $\Delta k_{3D} = \frac{8\pi^3}{V}$.

Similarly, in case of 2D the area occupied by a single k point is delta k 2D = 4 pi square divided by A. Now, we divide omega 3D by delta k 3D $\frac{\Omega_{3D}}{\Delta k_{3D}}$ and multiply by 2 to account for pauli exclusion principle which allows 2 electrons per step if we do that, then we get k cube = 3 pi square n in case of 3D. Similarly, in case of 2D we have to divide omega of 2D divided by delta k of 2D and multiplied by 2 to account for pauli exclusion principle.

And if we do that, then we can derive k square = 2 pi n in case of 3D n = number of electrons per unit volume and in case of 2D n = number of electrons per unit area.

(Refer Slide Time: 11:52)

Now, we replace k by E using this equation such that k = square root of 2mE by h cross square. If we do that, then in case of 3D we get n = 1 by 3 pi square and in case of k we just right square root of 2mE h cross square whole power 3 and then we can calculate dn by dE which is equal to 1 by 3 pi square into 3 by 2 2mE by h cross square whole power half 2m by h cross square.

And = dn dE is the density of states thus, we can express density of states in case of 3D free electron gas in this form. Similarly, in case of 2D we just write n = 1 by 2 pi and in place of k square we are just going to replace 2mE divided by h cross square and in this case, we get dn by dE = m by pi h cross square using similar method, you can derive the density of states for 1D free electron gas this is left as an exercise for you.

(Refer Slide Time: 13:55)

Now, that we have figured out how does the density of states look like for 1D 2D and 3D free electron gas, we can plot Fermi function times density of states at T greater than 0. Let us do it for room temperature, at room temperature k B T = 0.025 electron volts, where k B is the Boltzmann constant and T is the temperature let us take Fermi energy to be equal to 1 electron volt, we define the Fermi function here and for a 3D free electron gas density of states is equal to square root of E.

Then we plot energy versus Fermi function, energy versus density of states and energy versus Fermi function times the density of states.

(Refer Slide Time: 14:55)
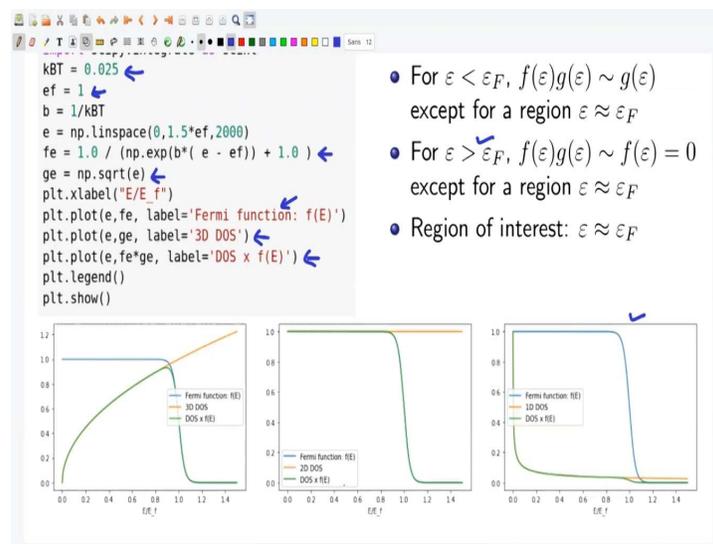
The plots for 3D, 2D and 1D looks like this. Now, Fermi function differs from its limiting values 0 and 1 only in the region E close to the Fermi energy. As a result for E less than E F, Fermi function times density of states is almost equal to density of states, except for a region E very close to the Fermi energy, you will see that orange curve is the density of states and the green curve is the density of states times the Fermi function and for E less than E F for most of the regions that orange curve where the green curve are identical.

On the other hand for E greater than E F, Fermi function times density of states is very close to the Fermi function except for region E close to E F in the plot, the green curve is the density of states times the Fermi function and the blue curve is the Fermi function and you see that for E greater than E 0, they are identical and they differ only in a region where E is very close to the E F. Thus, to understand the behaviour of free electron gas at T greater than 0, we need to mainly focus in the region close to the Fermi energy.

```
#3D: f(e) x g(e) at T~0 and T>0
import numpy as np
import matplotlib.pyplot as plt
import scipy.integrate as scint
kBT0 = 0.001
kBT=0.025
ef = 1
b0 = 1/kBT0
b = 1/kBT
e = np.linspace(0,1.5*ef,2000)
fe0 = 1.0 / (np.exp(b0*( e - ef)) + 1.0 )
fe = 1.0 / (np.exp(b*( e - ef)) + 1.0 )
ge = np.sqrt(e)
plt.xlabel("E/E_f")
plt.title("f(E) x g(E) for 3D")
plt.plot(e,fe0*ge, label='T~0')
plt.plot(e,fe*ge, label='T>0')
plt.legend()
plt.show()
```

- $f(\varepsilon) \times g(\varepsilon)$ identical at $T = 0$ & $T > 0$, except a region $\varepsilon \approx \varepsilon_F$
- DIY for 2D and 1D
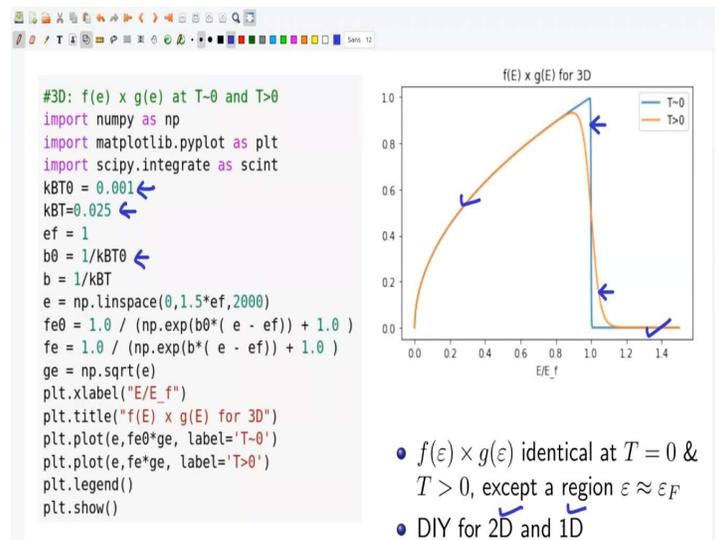
To understand what changes in finite temperature, we plot Fermi function times density of states at T = 0 and P greater than 0, we cannot take T exactly equal to 0 because in that case inverse temperature 1 by kBT becomes infinity. Thus, we take a very small temperature kBT 0.001 and compare it with room temperature where kBT = 0.025 electron volt. In this plot, the blue curve is fE times gE at a very low temperature.

And the orange curve is fE times gE at room temperature note that fE times gE is identical at T = 0 and T greater than 0 except a region close to the Fermi energy that is they are identical in this region as well as there are identical in this region, but only in the region close to the Fermi energy the blue and orange curve differ from each other. Here I have shown it for 3D free electron gas, you can write the code for 2D and 1D free electron gas.

```
#3D: e x f(e) x g(e) at T~0 and T>0
import numpy as np
import matplotlib.pyplot as plt
import scipy.integrate as scint
kBT0 = 0.001
kBT=0.025
ef = 1.0
b0 = 1/kBT0
b = 1/kBT
e = np.linspace(0,1.5*ef,2000)
fe0 = 1.0 / (np.exp(b0*( e - ef)) + 1.0 )
fe = 1.0 / (np.exp(b*( e - ef)) + 1.0 )
ge = np.sqrt(e)
plt.xlabel("E/E_f")
plt.title("E x f(E) x g(E) for 3D")
plt.plot(e,e*fe0*ge, label='T~0')
plt.plot(e,e*fe*ge, label='T>0')
plt.legend()
plt.show()
```

- Total energy:

$$\varepsilon = \int_0^\infty \varepsilon f(\varepsilon) g(\varepsilon) d\varepsilon$$

- $\Delta \varepsilon = \varepsilon - \varepsilon_{(T=0)}$
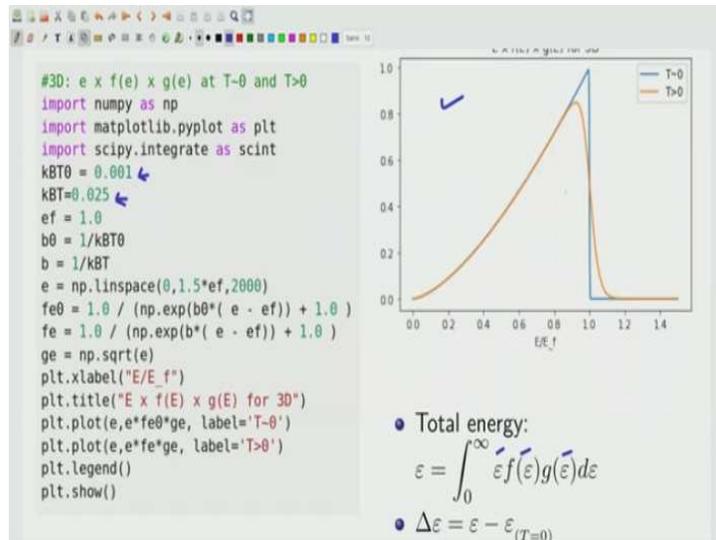
Let us find out the fraction of electrons getting excited about the Fermi level at T greater than 0 total number of states is given by the total area under the curve fE times gE number of states above E F can be calculated from the area as shown here. In the code, we calculate the total area under the curve here. And we calculate the area of the hash region here let us run the code.

(Refer Slide Time: 18:44)
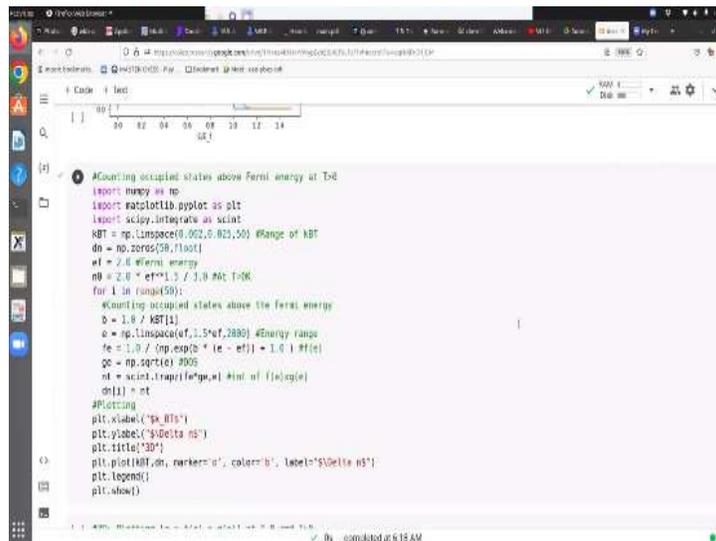
This is the code for counting the occupied states above Fermi energy at T greater than 0. This is where we define the range of kBT and we start at a very low temperature 0.002 and we go all the way to room temperature. This is the array where we are going to store the number of occupied states above the Fermi energy the number of states at 0 Kelvin can either we calculated by finding the total area under the curve or it can be obtained from the analytical result as shown here.
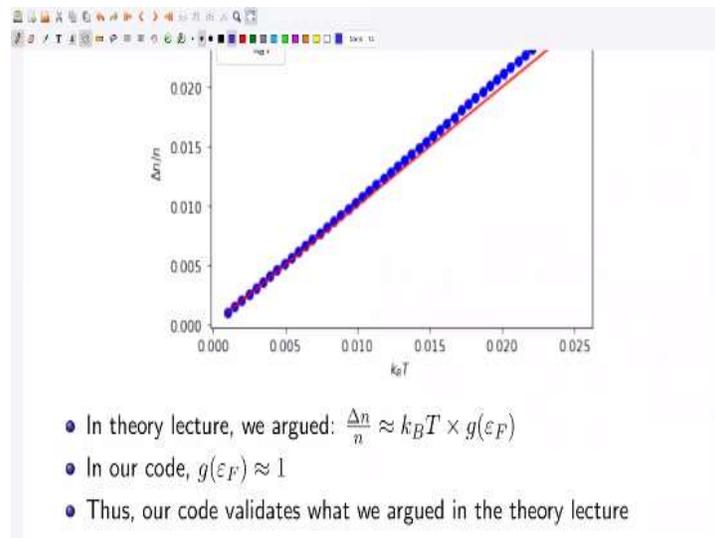
Then, we find the area under the curve of the hashed region that is above the Fermi energy in this step. And this area equal to number of occupied states about the Fermi energy. Let us run the code and find the number of occupied states above that Fermi energy at T greater than 0.

(Refer Slide Time: 20:07)

- In theory lecture, we argued: $\frac{\Delta n}{n} \approx k_B T \times g(\varepsilon_F)$
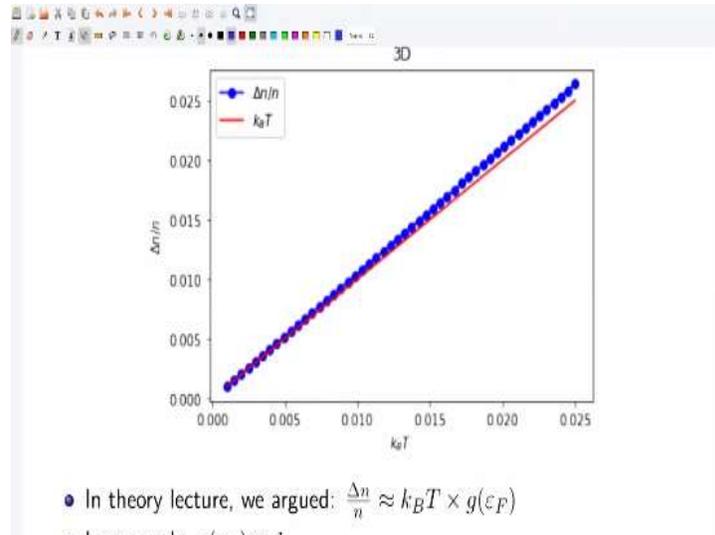
Clearly, fraction of electrons excited about the Fermi level increases linearly with k B T.

(Refer Slide Time: 20:16)

- In theory lecture, we argued: $\frac{\Delta n}{n} \approx k_B T \times g(\varepsilon_F)$
- In our code, $g(\varepsilon_F) \approx 1$
- Thus, our code validates what we argued in the theory lecture

In theory lecture, we argued that delta n divided by n that is the number of electrons excited about the Fermi level divided by the total number of electrons goes as k B T time's density of states at the Fermi level. In our code density of states at the Fermi level is a number which is close to 1. Thus, our code validates what we argued in the theory lecture. I showed for 3D free electron gas, it is left as an exercise for you to write the code for 1D and 2D free electron gas.

Total energy can be calculated by finding the area under the curve E times fE times gE, 2 of them what at a temperature very close to 0 and another other room temperature are shown in this diagram. The energy difference can be calculated by finding the difference of the area under the curve shown in orange and blue colour. The code given here is going to generate the plot as shown in this slide.

(Refer Slide Time: 21:37)
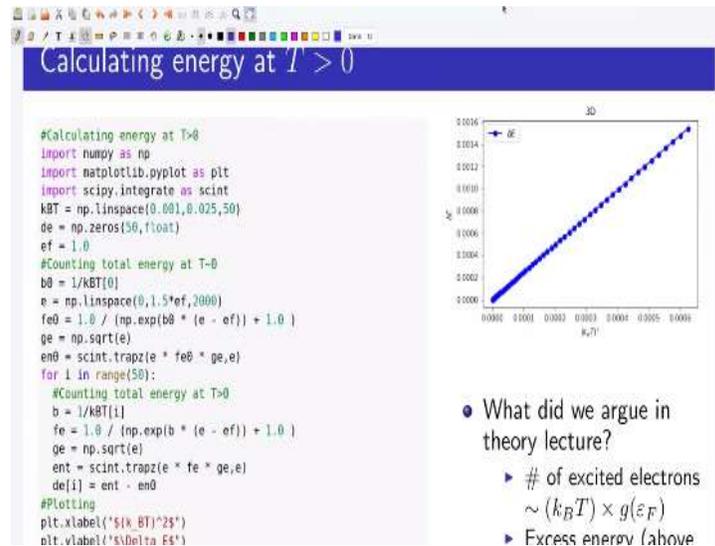
Let us write a code to calculate the energy difference between T = 0 and T greater than 0. This is where we calculate the area under the curve where T = 0 and this is where we calculate the area under the curve at T greater than 0, we store the energy difference at different temperature in an array, this is the code for calculating the energy difference for temperature T = 0 and T greater than 0, this is where we define the temperature range and this is where we calculate the total energy at a temperature which is very close to 0.

Then, we run for loop and we get the total energy at T greater than 0 this is where it is done. Finally, we stored the energy difference in some array and then we plot the energy difference as a function of k BT square. So, let us know run the code. clearly, delta E is a quadratic function of temperature.

(Refer Slide Time: 23:08)

```
ef = 1.0
#Counting total energy at T=0
b0 = 1/kBT[0]
e = np.linspace(0,1.5*ef,2000)
fe0 = 1.0 / (np.exp(b0 * (e - ef)) + 1.0 )
ge = np.sqrt(e)
en0 = scint.trapz(e * fe0 * ge,e)
for i in range(50):
    #Counting total energy at T>0
    b = 1/kBT[i]
    fe = 1.0 / (np.exp(b * (e - ef)) + 1.0 )
    ge = np.sqrt(e)
    ent = scint.trapz(e * fe * ge,e)
    de[i] = ent - en0
#Plotting
plt.xlabel('$(k_BT)^2$')
plt.ylabel('$\Delta E$')
plt.title('3D')
plt.plot(kBT*kBT,de, marker='o', color='b', label='$\Delta E$')
plt.legend()
plt.show()
```

- What did we argue in theory lecture?
  - ▸ # of excited electrons $\sim (k_B T) \times g(\varepsilon_F)$
  - ▸ Excess energy (above ground state) of these electrons: $k_B T$
  - ▸ $\Delta \varepsilon \approx (k_B T)^2 g(\varepsilon_F)$

- The code provides validation of what we argued in theory lecture

Let us go back to what we argued in the theory lecture number of excited electrons is given by k BT times the density of states at Fermi energy. Now, excess energy above the ground state of these electrons is k BT. Thus, delta E is k BT time g of E F time k BT that is k BT squared times g of E F in our code, we indeed see that delta E is a quadratic function of temperature. Thus, the code provides validation of what we argued in the theory lecture. I showed for 3D free electron gas it is left as an exercise for you to write the code for 1D and 2D.

(Refer Slide Time: 23:58)

How much does μ change with T?

- At $T = 0$ K, we know that $\mu = \varepsilon_F$
- In numerical exercise, so far we assumed $\mu = \varepsilon_F$ at $T > 0$
- How good/bad is the assumption?
- From Sommerfeld expansion: $\mu(T) = \varepsilon_F \left[1 - \frac{\pi^2}{12} \left(\frac{T}{T_F} \right)^2 \right]$
- Sommerfeld expansion is an approximation, is there a direct way of getting μ?
- At $T = 0$ K: $n_0 = \int_0^{\varepsilon_F} g(\varepsilon) d\varepsilon$
- At $T > 0$ K: $n(T, \mu) = \int_0^\infty \frac{g(\varepsilon)}{e^{\beta(\varepsilon - \mu)} + 1} d\varepsilon$, where $\beta = \frac{1}{k_B T}$

We know that at T = 0 Kelvin, chemical potential is equal to the Fermi energy in numerical exercises. So, far we have assumed mu = E F even at a temperature which is greater than 0. Let us assess how good or bad is the assumption from Sommerfeld expansion we know that the chemical potential for 3D free electron gas decreases quadratically with increasing

temperature. However, sommerfeld expansion is an approximation is there a direct way of calculating mu.

(Refer Slide Time: 24:38)

We know that at T = 0 Kelvin, the number of electrons can be obtained by integrating density of states between 0 and E F at T greater than 0. The number of electrons can be obtained by evaluating this integral in case of a constant volume process n naught should be is equal to n. Now, from this equation, it is clear that for a given value of temperature n the number of electron depends on the value of mu. This is going to be our algorithm for calculating mu as a function of temperature.

At a given temperature, we are going to start with a guess value of mu and we are going to unit until we reach a point where n = n naught.

(Refer Slide Time: 25:34)

```
#Calculating \mu at different T
import numpy as np
import matplotlib.pyplot as plt
import scipy.integrate as scint
kBT = np.linspace(0.01,0.2,25) #Range of kBT
mu = np.zeros(25,float)
ef = 5.0 #Fermi energy
n0 = 2.0 * ef**1.5 / 3.0 #At T=0K
def fint(b,m):
    e = np.linspace(0,1.5*ef,5000) #Energy range
    fe = 1.0 / (np.exp(b * (e - m)) + 1.0 ) #f(e)
    ge = np.sqrt(e) #DOS
    value = scint.trapz(fe*ge,e) #int of f(e)xg(e)
    return value
#Loop to get \mu
m = ef
for i in range(25):
    b = 1.0 / kBT[i]
    nt = n0 + 0.5
    while abs(nt-n0) > 0.00001:
        m = m - 0.000001
        nt = fint(b,m)
    mu[i] = m
#Plotting
```

- $g(\varepsilon) = \sqrt{\varepsilon}$

- $n_0 = \displaystyle\int_0^{\varepsilon_F} g(\varepsilon)d\varepsilon = \frac{2}{3}\varepsilon_F^{3/2}$

- $n(\mu) = \displaystyle\int_0^{\infty} \frac{g(\varepsilon)}{e^{\beta(\varepsilon-\mu)}+1}d\varepsilon$

- For some given T, vary μ until
  $n(\mu) = n_0$

Let us write a code to get mu as a function of temperature, we shall consider 3D free electron gas such that density of states is equal to square root of E total number of electrons at T = 0k can be calculated by integrating density of states from 0 to the Fermi energy and it is equal to 2 3rd times E F to the power 3 by 2 n naught is calculated here in the code at some temperature greater than 0 number of electrons can be calculated by integrating density of states times the Fermi function.

This integral is calculated here in this user defined function, we start with some initial guess value of mu. For example, we can start with a guess value mu, which is equal to ef we calculate the number of electrons for the guess value of mu and verify whether it is equal to n naught or not we vary mu until we reach a point where n = n naught this is done by using a while loop all this condition is satisfied, we have the value of mu at a given temperature, we do it for a wide range of temperature, where the temperature is varied using a formula.

(Refer Slide Time: 27:07)

This is the code to calculate mu at different temperature for the 3D free electron gas. This is where we define the range of k BT and this is where we define the Fermi energy this is where we calculate the number of electrons that T = 0 Kelvin. In this user defined function, we calculate the number of electrons at finite temperature for some guess value of mu we start with a guess value of mu = ef that is the Fermi energy then, we have a while loop where we calculate the number of electrons at some finite temperature for a given value of mu.
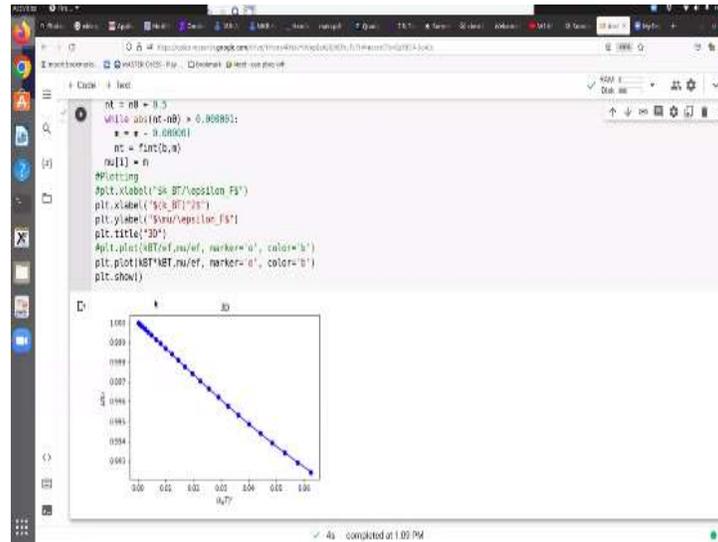
Then we check whether nt - n naught is greater than this number or not. Once this condition is satisfied, that is when the value of nt the number of electrons at finite temperature is very close to the number of electrons at 0 temperature at that point, we decide that the convergence is achieved and we store the value of mu in some arrays.
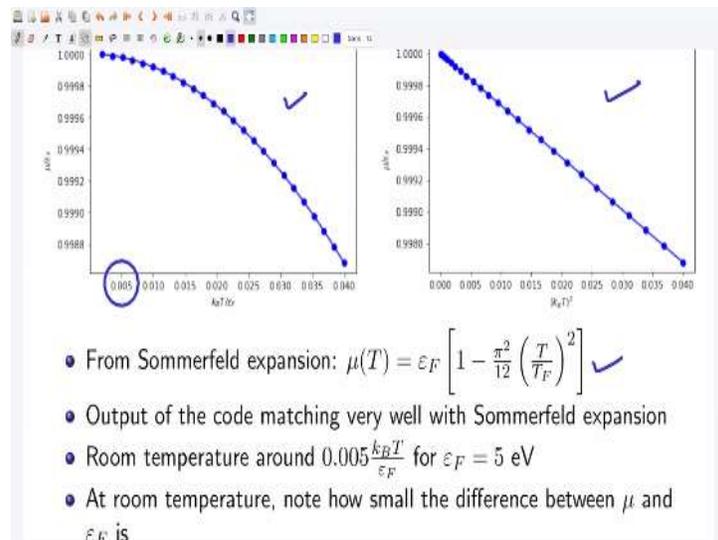
(Refer Slide Time: 28:47)
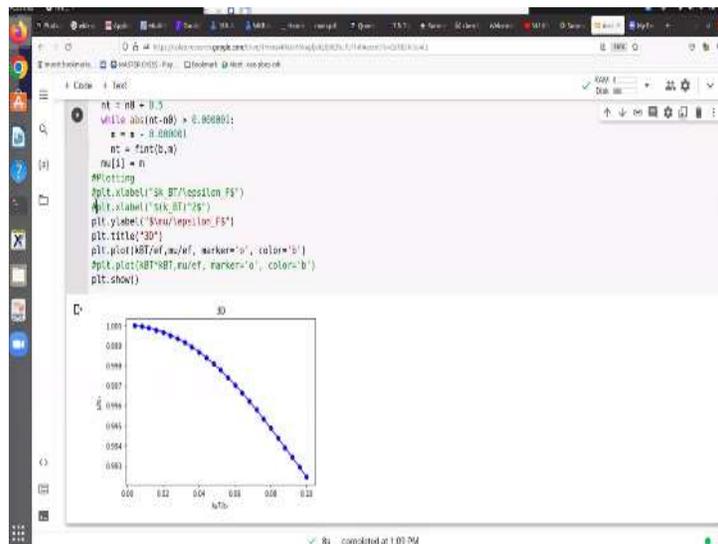
Finally, we got mu as a function of kBT. So, let us run the code this is how mu varies as a function of kBT, we can even plot it as a function of kBT square. So, let us do that. So, previously we were plotting mu as a function of kBT. Now, we plot it as a function of kBT square.

(Refer Slide Time: 29:16)

 (Refer Slide Time: 29:20)

- From Sommerfeld expansion: $\mu(T) = \varepsilon_F \left[1 - \frac{\pi^2}{12} \left(\frac{T}{T_F} \right)^2 \right]$
- Output of the code matching very well with Sommerfeld expansion
- Room temperature around $0.005 \frac{k_B T}{\varepsilon_F}$ for $\varepsilon_F = 5$ eV
- At room temperature, note how small the difference between μ and ε_F is

And then this is how the plot looks like. Thus, we get a curve if we plot mu as a function of k BT. Further by plotting mu as a function of k BT square, we get a straight line, which confirms that mu decreases quadratically with increasing temperature, the outcome is seen as obtained from that Sommerfeld expansion note that the room temperature is around 0.005. Thus, at room temperature mu hardly changes from F. So, our assumption of mu = E F in previous numerical examples is not do that.

- Sommerfeld expansion is an approximation, but there is no approximation in our code
- $n = \underbrace{\int_{-\infty}^{\varepsilon_F} g(\varepsilon)d\varepsilon} + (\mu - \varepsilon_F)g(\varepsilon_F) + \frac{\pi^2}{6}\left(\frac{1}{\beta}\right)^2 g'(\varepsilon_F)$
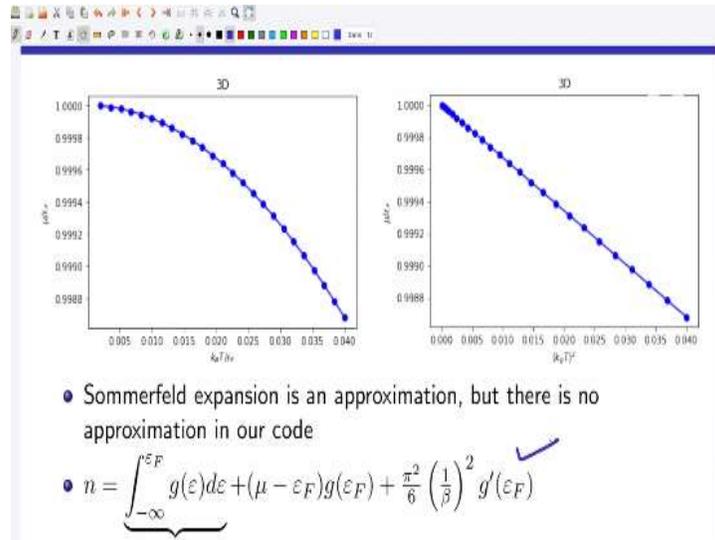
Let us discuss an example where Sommerfeld expansion is going to fail we know that using sommerfeld expansion, we can write the electron density at finite temperature in this form.

- Sommerfeld expansion is an approximation, but there is no approximation in our code
- $n = \underbrace{\int_{-\infty}^{\varepsilon_F} g(\varepsilon)d\varepsilon}_{n_0} + (\mu - \varepsilon_F)g(\varepsilon_F) + \frac{\pi^2}{6}\left(\frac{1}{\beta}\right)^2 g'(\varepsilon_F)$ $g' = 0$
  $\mu = \varepsilon_F$ at any T
- If $g(\varepsilon)$ is a constant (like for 2D free electron gas), can not use Sommerfeld expansion, which predicts $\mu = \varepsilon_F$ at any T
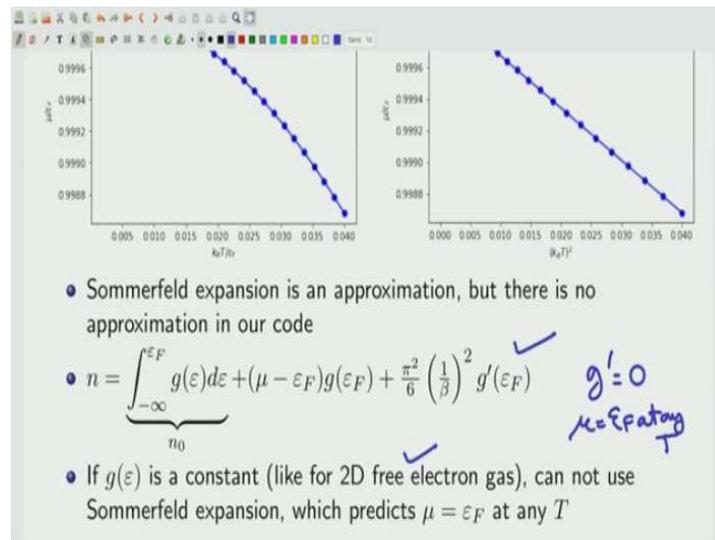
Thus, if density of states is a constant, for example, in a 2D free electron gas we cannot use Sommerfeld expansion because G dash = 0. In that case Sommerfeld expansion always predicts that mu will be equal to E f for any temperature, which is not correct. In such cases, we have to find mu not from Sommerfeld expansion, but using some other method for example, the numerical method discussed in the lecture.