AIM:

Plot the distribution of particles w.r.t. Energy (dN/dE vs E) in 3D for :-

1. Relativistic & Non-Relativistic Bosons both at low & high temperatures.

2. Relativistic & Non-Relativistic Fermions both at low & high temperatures.

THEORY:

(Source : Practical Hope Youtube Channel)

The density of particles is the product of the density of states and the average number of particles in each state,

$$\frac{dN}{dE} = g(E).\bar{n}(E) = \frac{C_n\sqrt{E}}{e^{\beta(E-\mu)} - 1} \quad \text{(Non-relativistic Bosons)}$$

$$\frac{dN}{dE} = g(E).\bar{n}(E) = \frac{C_r E^2}{e^{\beta(E-\mu)} - 1} \quad \text{(Relativistic Bosons)}$$

$$\frac{dN}{dE} = g(E).\bar{n}(E) = \frac{C_n\sqrt{E}}{e^{\beta(E-\mu)} + 1} \quad \text{(Non-relativistic Fermions)}$$

$$\frac{dN}{dE} = g(E).\bar{n}(E) = \frac{C_r E^2}{e^{\beta(E-\mu)} + 1} \quad \text{(Relativistic Fermions)}$$

$$C_n = (2s+1)\frac{2\pi V(2m)^{3/2}}{h^3} \qquad C_r = 2s\frac{4\pi V}{h^3 c^3}$$

1. Non-Relativistic Fermions

Step-1 : Import necessary libraries

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

Step-2 : Define required Constants

```
e = 1.6e-19 #Electronic Charge
kb = 1.38e-23 #Boltzmann Constant
h = 6.626e-34 #Planck's Constant
s = 0.5 #Spin
u = 1 #Chemical Potential in eV
V = 1 #Volume
m = 9.1e-31 #Mass of electron
```

Step-3 : Define Energy & Temperature Range

```
E = np.arange(0,2,0.001) #Energy in eV
T = np.array([100,1000]) #Temperature in Kelvin (100K & 1000K)
```

Step-4 : Evaluate Cn using above mentioned formula

```
Cn = (2*s + 1)*(2*3.14*V*(2*m)**1.5)/(h**3)
```

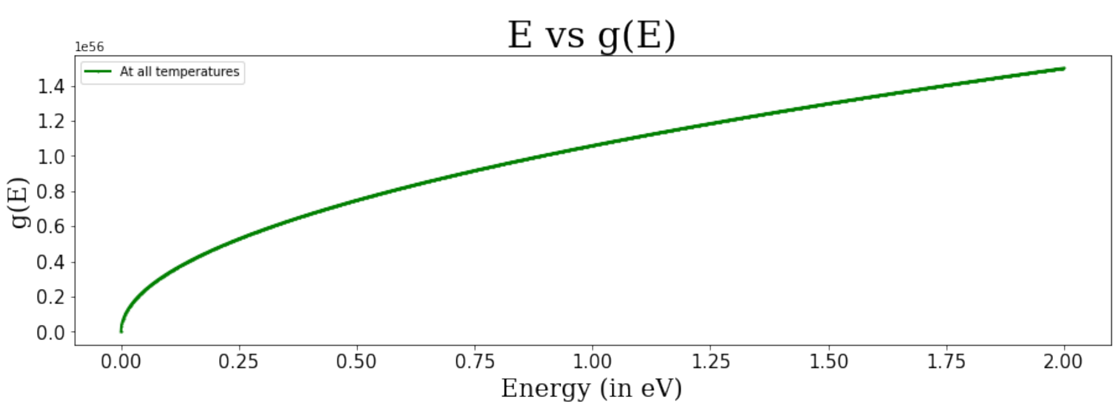
Step-5 : Evaluate g(E), n(E) & dN/dE using above mentioned formulaes

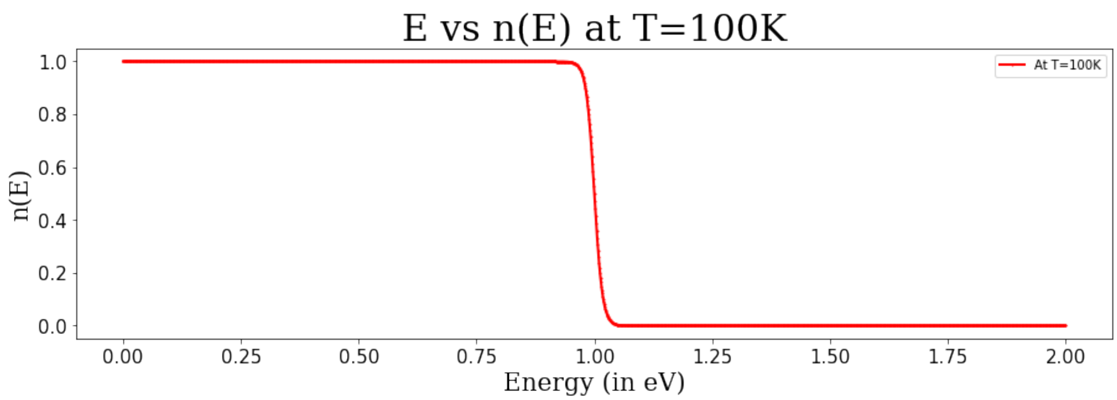
```
b = 1/(kb*T)
g = Cn * np.sqrt(E) #Density of States
n100 = 1/(np.exp((E-u)*e*b[0])+1) #At 100K
n1000 = 1/(np.exp((E-u)*e*b[1])+1) #At 1000K
f100 = n100*g #dN/dE at 100K
f1000 = n1000*g #dN/dE at 1000K
```

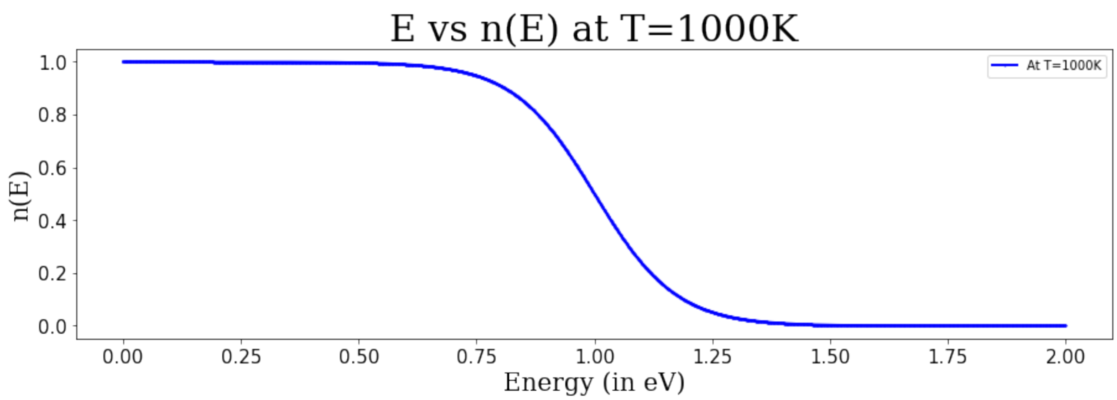
Step-6 : Plot required graphs

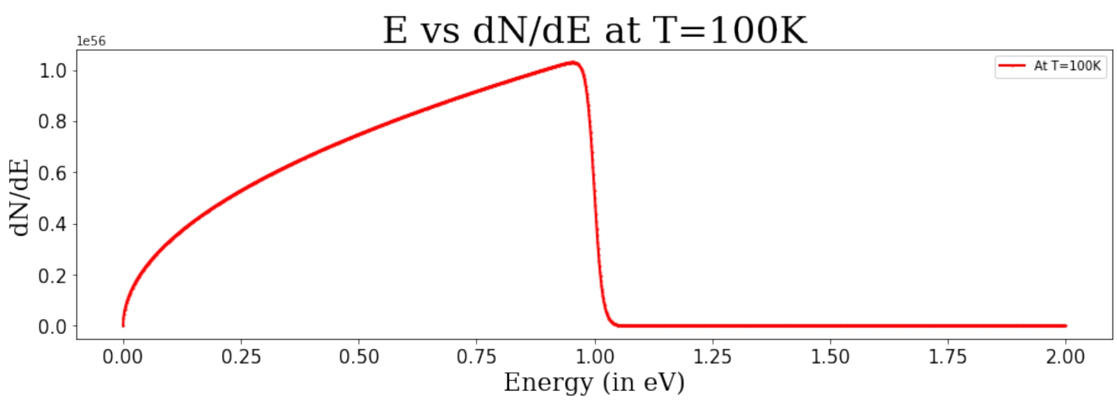
```
### Step-6 : Plot required graphs

fontji = {'family':'serif','size':20}
fontji2 = {'family':'serif','size':30}

#plt.suptitle("Non-Relativistic Fermions")
plt.figure(figsize=(15,25)) #Setting size of the figure
plt.subplot(5,1,1)
plt.plot(E,g,"o-g",lw="2",ms="1",label="At all temperatures")
plt.legend(loc="best")
plt.xlabel("Energy (in eV)",fontdict=fontji)
plt.ylabel("g(E)",fontdict=fontji)
plt.title("E vs g(E)",fontdict=fontji2)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()

plt.figure(figsize=(15,25)) #Setting size of the figure
plt.subplot(5,1,2)
plt.plot(E,n100,"o-r",lw="2",ms="1",label="At T=100K")
plt.legend(loc="best")
plt.xlabel("Energy (in eV)",fontdict=fontji)
plt.ylabel("n(E)",fontdict=fontji)
plt.title("E vs n(E) at T=100K",fontdict=fontji2)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()

plt.figure(figsize=(15,25)) #Setting size of the figure
plt.subplot(5,1,3)
plt.plot(E,n1000,"o-b",lw="2",ms="1",label="At T=1000K")
plt.legend(loc="best")
plt.xlabel("Energy (in eV)",fontdict=fontji)
plt.ylabel("n(E)",fontdict=fontji)
```

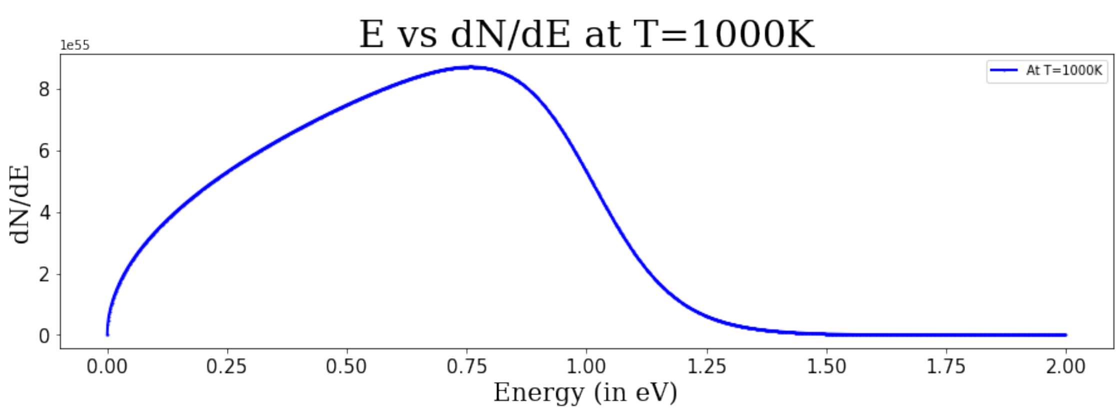
```
plt.title("E vs n(E) at T=1000K",fontdict=fontji2)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()

plt.figure(figsize=(15,25)) #Setting size of the figure
plt.subplot(5,1,4)
plt.plot(E,f100,"o-r",lw="2",ms="1",label="At T=100K")
plt.legend(loc="best")
plt.xlabel("Energy (in eV)",fontdict=fontji)
plt.ylabel("dN/dE",fontdict=fontji)
plt.title("E vs dN/dE at T=100K",fontdict=fontji2)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()

plt.figure(figsize=(15,25)) #Setting size of the figure
plt.subplot(5,1,5)
plt.plot(E,f1000,"o-b",lw="2",ms="1",label="At T=1000K")
plt.legend(loc="best")
plt.xlabel("Energy (in eV)",fontdict=fontji)
plt.ylabel("dN/dE",fontdict=fontji)
plt.title("E vs dN/dE at T=1000K",fontdict=fontji2)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```




E vs n(E) at T=100K
At T=100K
n(E)
Energy (in eV)


E vs n(E) at T=1000K
At T=1000K
n(E)
Energy (in eV)

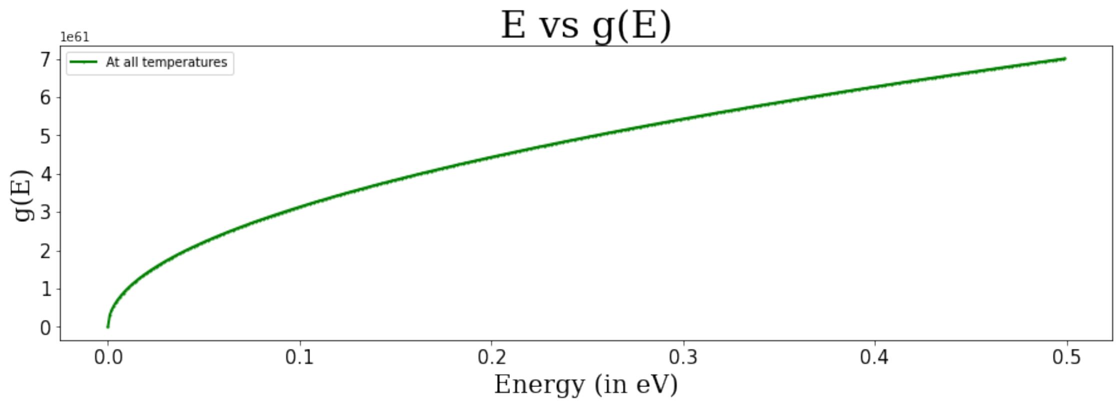

E vs dN/dE at T=100K
1e56
At T=100K
dN/dE
Energy (in eV)

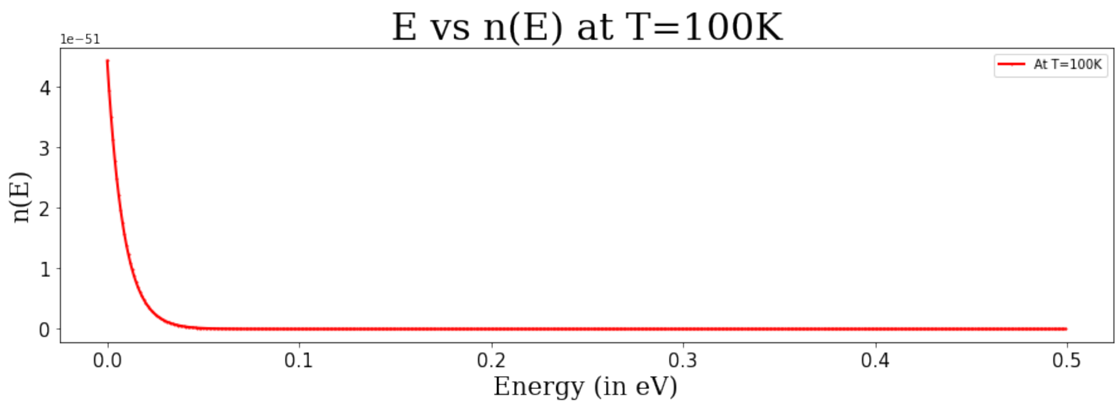

E vs dN/dE at T=1000K
1e55
At T=1000K
dN/dE
Energy (in eV)

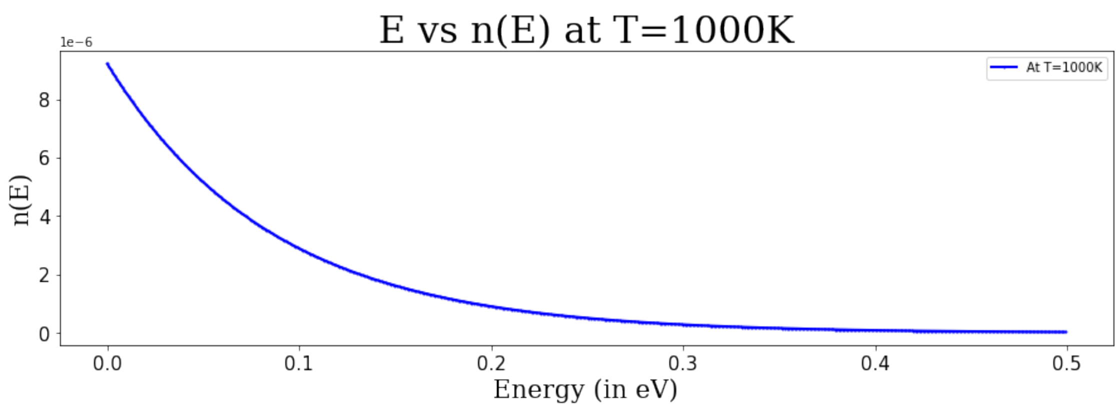
2. Non-Relativistic Bosons (same as above with few changes)

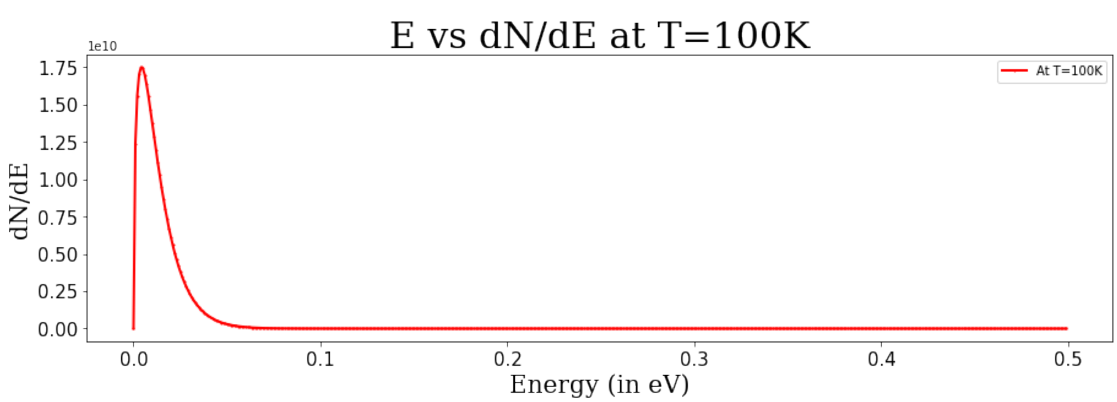
```
### Step-1 : Import necessary libraries

import numpy as np
import matplotlib.pyplot as plt

### Step-2 : Define required Constants

e = 1.6e-19 #Electronic Charge
kb = 1.38e-23 #Boltzmann Constant
h = 6.626e-34 #Planck's Constant
s = 1 #Spin
u = -1 #Chemical Potential in eV
V = 1 #Volume
m = 4*1.66e-27 #Mass = 4amu (1amu = 1.66e-27)

### Step-3 : Define Energy & Temperature Range

E = np.arange(0,0.5,0.001) #Energy in eV
T = np.array([100,1000]) #Temperature in Kelvin (100K & 1000K)

### Step-4 : Evaluate Cn using above mentioned formula

Cn = (2*s + 1)*(2*3.14*V*(2*m)**1.5)/(h**3)

### Step-5 : Evaluate g(E), n(E) & dN/dE using above mentioned
formulaes

b = 1/(kb*T)
g = Cn * np.sqrt(E) #Density of States
n100 = 1/(np.exp((E-u)*e*b[0])-1) #At 100K
n1000 = 1/(np.exp((E-u)*e*b[1])-1) #At 1000K
f100 = n100*g #dN/dE at 100K
f1000 = n1000*g #dN/dE at 1000K

### Step-6 : Plot required graphs

fontji = {'family':'serif','size':20}
fontji2 = {'family':'serif','size':30}

#plt.suptitle("Non-Relativistic Bosons")
plt.figure(figsize=(15,25)) #Setting size of the figure
plt.subplot(5,1,1)
plt.plot(E,g,"o-g",lw="2",ms="1",label="At all temperatures")
plt.legend(loc="best")
plt.xlabel("Energy (in eV)",fontdict=fontji)
plt.ylabel("g(E)",fontdict=fontji)
plt.title("E vs g(E)",fontdict=fontji2)
plt.xticks(fontsize=15)
```

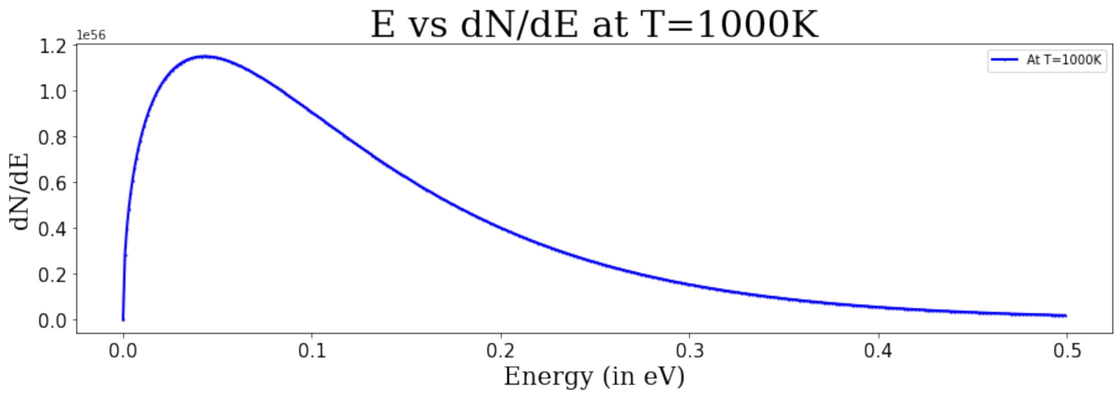
```python
plt.yticks(fontsize=15)
plt.show()

plt.figure(figsize=(15,25)) #Setting size of the figure
plt.subplot(5,1,2)
plt.plot(E,n100,"o-r",lw="2",ms="1",label="At T=100K")
plt.legend(loc="best")
plt.xlabel("Energy (in eV)",fontdict=fontji)
plt.ylabel("n(E)",fontdict=fontji)
plt.title("E vs n(E) at T=100K",fontdict=fontji2)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()

plt.figure(figsize=(15,25)) #Setting size of the figure
plt.subplot(5,1,3)
plt.plot(E,n1000,"o-b",lw="2",ms="1",label="At T=1000K")
plt.legend(loc="best")
plt.xlabel("Energy (in eV)",fontdict=fontji)
plt.ylabel("n(E)",fontdict=fontji)
plt.title("E vs n(E) at T=1000K",fontdict=fontji2)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()

plt.figure(figsize=(15,25)) #Setting size of the figure
plt.subplot(5,1,4)
plt.plot(E,f100,"o-r",lw="2",ms="1",label="At T=100K")
plt.legend(loc="best")
plt.xlabel("Energy (in eV)",fontdict=fontji)
plt.ylabel("dN/dE",fontdict=fontji)
plt.title("E vs dN/dE at T=100K",fontdict=fontji2)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()

plt.figure(figsize=(15,25)) #Setting size of the figure
plt.subplot(5,1,5)
plt.plot(E,f1000,"o-b",lw="2",ms="1",label="At T=1000K")
plt.legend(loc="best")
plt.xlabel("Energy (in eV)",fontdict=fontji)
plt.ylabel("dN/dE",fontdict=fontji)
plt.title("E vs dN/dE at T=1000K",fontdict=fontji2)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```


E vs g(E)
1e61
At all temperatures
g(E)
Energy (in eV)


E vs n(E) at T=100K
1e−51
At T=100K
n(E)
Energy (in eV)


E vs n(E) at T=1000K
1e−6
At T=1000K
n(E)
Energy (in eV)

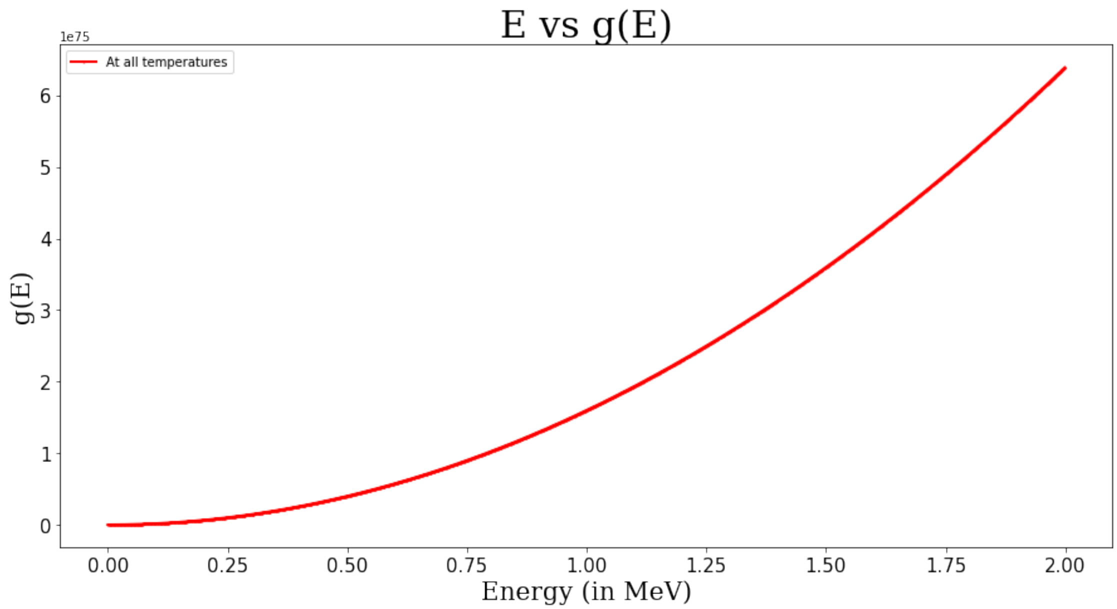

E vs dN/dE at T=100K
1e10
At T=100K
dN/dE
Energy (in eV)

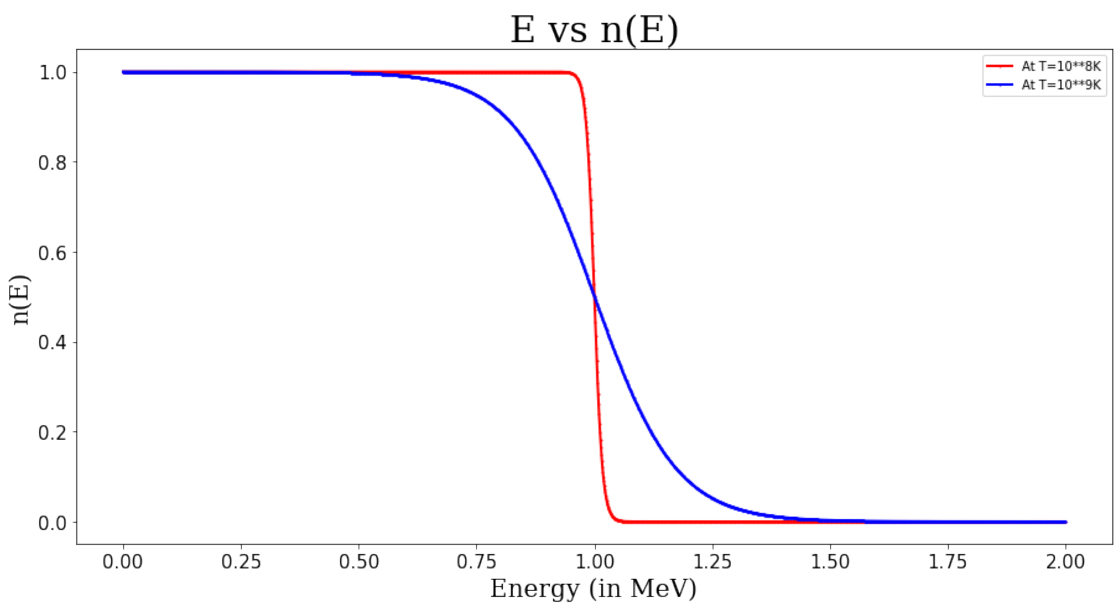


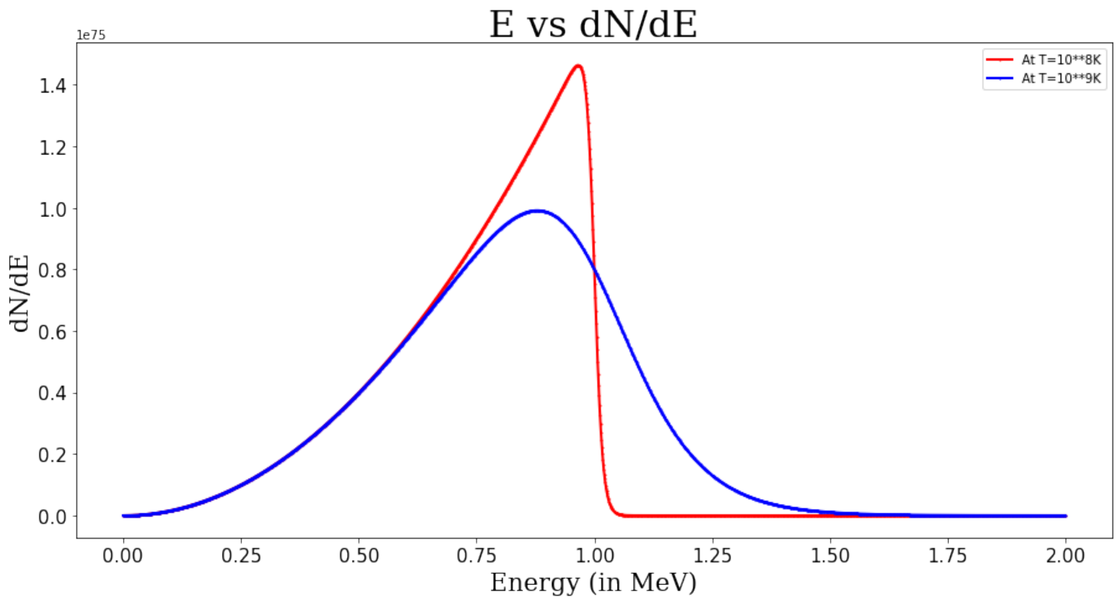
3. Relativistic Fermions (same as above with few changes)

```
### Step-1 : Import necessary libraries

import numpy as np
import matplotlib.pyplot as plt

### Step-2 : Define required Constants

e = 1.6e-19 #Electronic Charge
kb = 1.38e-23 #Boltzmann Constant
h = 6.626e-34 #Planck's Constant
s = 0.5 #Spin
u = 1 #Chemical Potential in MeV
V = 1 #Volume
#m = 9.1e-31 #Mass of electron
c = 3e8 #Speed of Light

### Step-3 : Define Energy & Temperature Range

E = np.arange(0,2,0.001) #Energy in MeV
T = np.array([10**8,10**9]) #Temperature in Kelvin

### Step-4 : Evaluate Cr using above mentioned formula

Cr = (2*s*4*3.14*V)/((h**3)*(c**3))

### Step-5 : Evaluate g(E), n(E) & dN/dE using above mentioned
formulaes

b = 1/(kb*T)
g = Cr * (E)**2 #Density of States
n100 = 1/(np.exp((E-u)*e*(10**6)*b[0])+1) #At 10**8K
n1000 = 1/(np.exp((E-u)*e*(10**6)*b[1])+1) #At 10**9K
f100 = n100*g #dN/dE at 10**8K
f1000 = n1000*g #dN/dE at 10**9K
```

```python
### Step-6 : Plot required graphs

fontji = {'family':'serif','size':20}
fontji2 = {'family':'serif','size':30}

#plt.suptitle("Relativistic Fermions")
plt.figure(figsize=(15,25)) #Setting size of the figure
plt.subplot(3,1,1)
plt.plot(E,g,"o-r",lw="2",ms="1",label="At all temperatures")
plt.legend(loc="best")
plt.xlabel("Energy (in MeV)",fontdict=fontji)
plt.ylabel("g(E)",fontdict=fontji)
plt.title("E vs g(E)",fontdict=fontji2)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()

plt.figure(figsize=(15,25)) #Setting size of the figure
plt.subplot(3,1,2)
plt.plot(E,n100,"o-r",lw="2",ms="1",label="At T=10**8K")
plt.plot(E,n1000,"o-b",lw="2",ms="1",label="At T=10**9K")
plt.legend(loc="best")
plt.xlabel("Energy (in MeV)",fontdict=fontji)
plt.ylabel("n(E)",fontdict=fontji)
plt.title("E vs n(E)",fontdict=fontji2)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()

plt.figure(figsize=(15,25)) #Setting size of the figure
plt.subplot(3,1,3)
plt.plot(E,f100,"o-r",lw="2",ms="1",label="At T=10**8K")
plt.plot(E,f1000,"o-b",lw="2",ms="1",label="At T=10**9K")
plt.legend(loc="best")
plt.xlabel("Energy (in MeV)",fontdict=fontji)
plt.ylabel("dN/dE",fontdict=fontji)
plt.title("E vs dN/dE",fontdict=fontji2)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```


E vs g(E)
1e75
At all temperatures
0
1
2
3
4
5
6
g(E)
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Energy (in MeV)


E vs n(E)
At T=10**8K
At T=10**9K
0.0
0.2
0.4
0.6
0.8
1.0
n(E)
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Energy (in MeV)

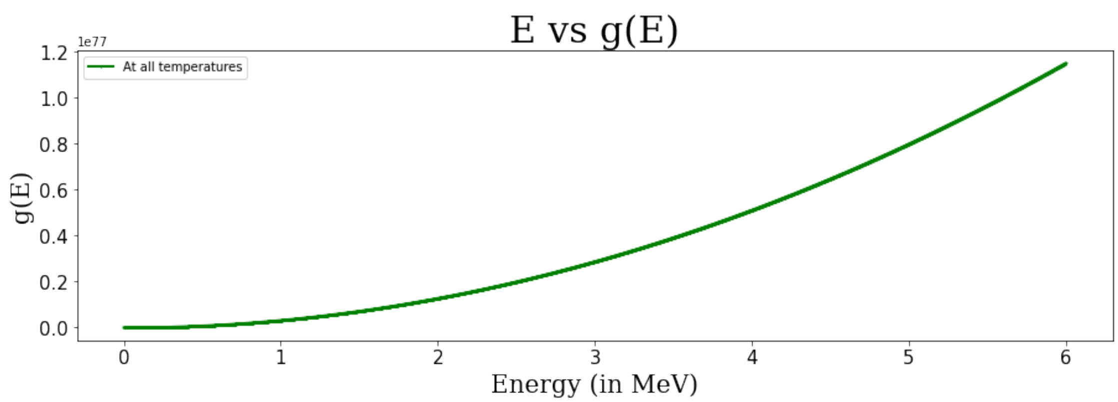


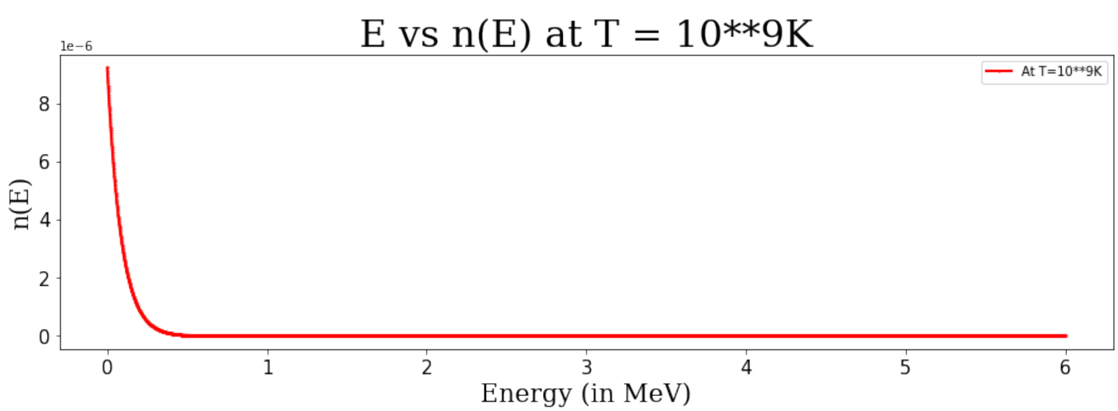
4. Relativistic Bosons (same as above with few changes)

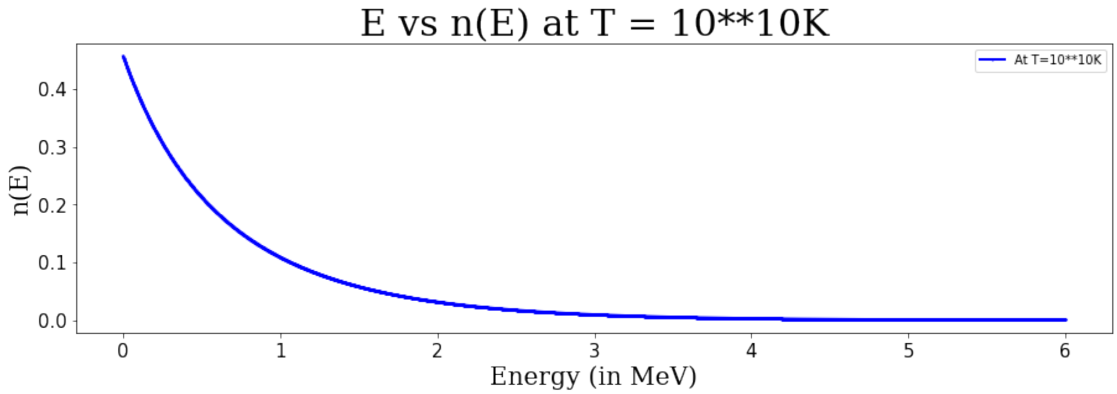
```
### Step-1 : Import necessary libraries

import numpy as np
import matplotlib.pyplot as plt

### Step-2 : Define required Constants

e = 1.6e-19 #Electronic Charge
kb = 1.38e-23 #Boltzmann Constant
h = 6.626e-34 #Planck's Constant
s = 1 #Spin
u = -1 #Chemical Potential in MeV
V = 1 #Volume
#m = 9.1e-31 #Mass of electron
c = 3e8 #Speed of Light

### Step-3 : Define Energy & Temperature Range

E = np.arange(0,6,0.001) #Energy in MeV
T = np.array([10**9,10**10]) #Temperature in Kelvin

### Step-4 : Evaluate Cr using above mentioned formula

Cr = (2*s*4*3.14*V)/((h**3)*(c**3))

### Step-5 : Evaluate g(E), n(E) & dN/dE using above mentioned
formulaes

b = 1/(kb*T)
```

```python
g = Cr * (E)**2 #Density of States
n100 = 1/(np.exp((E-u)*e*(10**6)*b[0])-1) #At 10**9K
n1000 = 1/(np.exp((E-u)*e*(10**6)*b[1])-1) #At 10**10K
f100 = n100*g #dN/dE at 10**9K
f1000 = n1000*g #dN/dE at 10**10K

### Step-6 : Plot required graphs

fontji = {'family':'serif','size':20}
fontji2 = {'family':'serif','size':30}

#plt.suptitle("Relativistic Bosons")
plt.figure(figsize=(15,25)) #Setting size of the figure
plt.subplot(5,1,1)
plt.plot(E,g,"o-g",lw="2",ms="1",label="At all temperatures")
plt.legend(loc="best")
plt.xlabel("Energy (in MeV)",fontdict=fontji)
plt.ylabel("g(E)",fontdict=fontji)
plt.title("E vs g(E)",fontdict=fontji2)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()

plt.figure(figsize=(15,25)) #Setting size of the figure
plt.subplot(5,1,2)
plt.plot(E,n100,"o-r",lw="2",ms="1",label="At T=10**9K")
plt.legend(loc="best")
plt.xlabel("Energy (in MeV)",fontdict=fontji)
plt.ylabel("n(E)",fontdict=fontji)
plt.title("E vs n(E) at T = 10**9K",fontdict=fontji2)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()

plt.figure(figsize=(15,25)) #Setting size of the figure
plt.subplot(5,1,3)
plt.plot(E,n1000,"o-b",lw="2",ms="1",label="At T=10**10K")
plt.legend(loc="best")
plt.xlabel("Energy (in MeV)",fontdict=fontji)
plt.ylabel("n(E)",fontdict=fontji)
plt.title("E vs n(E) at T = 10**10K",fontdict=fontji2)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()

plt.figure(figsize=(15,25)) #Setting size of the figure
plt.subplot(5,1,4)
plt.plot(E,f100,"o-r",lw="2",ms="1",label="At T=10**9K")
plt.legend(loc="best")
plt.xlabel("Energy (in MeV)",fontdict=fontji)
```

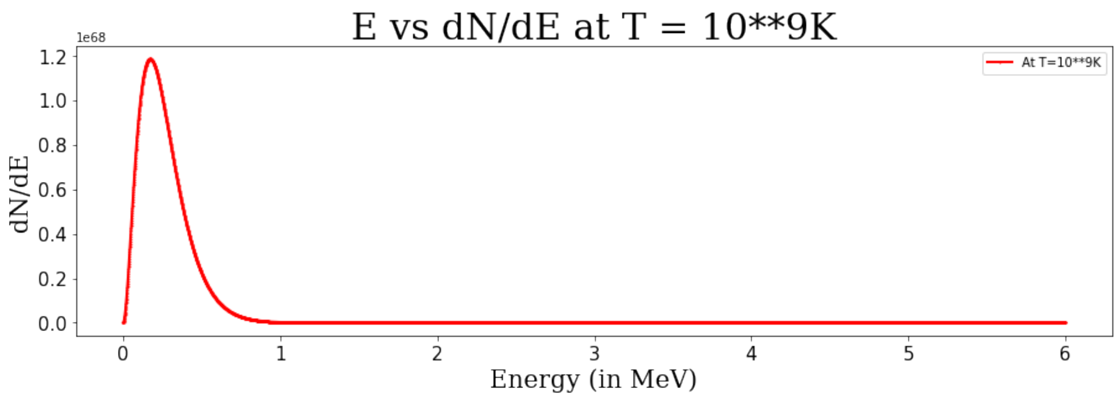
```
plt.ylabel("dN/dE",fontdict=fontji)
plt.title("E vs dN/dE at T = 10**9K",fontdict=fontji2)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()

plt.figure(figsize=(15,25)) #Setting size of the figure
plt.subplot(5,1,5)
plt.plot(E,f1000,"o-b",lw="2",ms="1",label="At T=10**10K")
plt.legend(loc="best")
plt.xlabel("Energy (in MeV)",fontdict=fontji)
plt.ylabel("dN/dE",fontdict=fontji)
plt.title("E vs dN/dE at T = 10**10K",fontdict=fontji2)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.show()
```



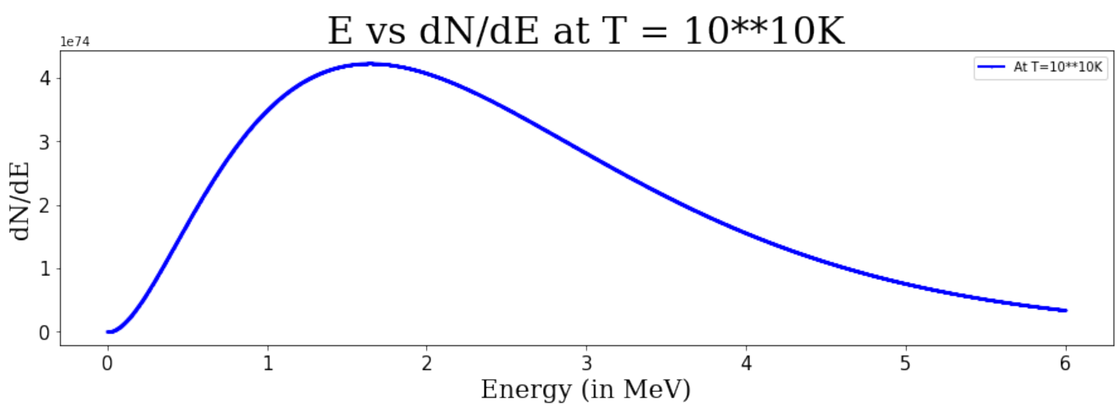




E vs n(E) at T = 10**10K
At T=10**10K
n(E)
0.0
0.1
0.2
0.3
0.4
0
1
2
3
4
5
6
Energy (in MeV)



E vs dN/dE at T = 10**9K
1e68
At T=10**9K
dN/dE
0.0
0.2
0.4
0.6
0.8
1.0
1.2
0
1
2
3
4
5
6
Energy (in MeV)



E vs dN/dE at T = 10**10K
1e74
At T=10**10K
dN/dE
0
1
2
3
4
0
1
2
3
4
5
6
Energy (in MeV)