AIM:

Plot the following functions with energy at different temperatures.

1. Maxwell-Boltzmann Distribution

2. Fermi-Dirac Distribution

3. Bose-Einstein Distribution

Formula Required :

The general formula for the three distributions is:

> f(E) = 1 / (exp((E-u)/kT) + a)

```
where  f(E) = Probability that the given system has energy E
       E = Energy
       u = Chemical potential
       k = Boltzmann constant
       T = Temperature
```

Values of a for different distributions are:

- Maxwell-Boltzmann distribution : **a = 0**
- Bose-Einstein distribution : **a = -1**
- Fermi-Dirac distribution : **a = +1**

Step-1 : Import necessary libraries (numpy & matplotlib.pyplot)

In [43]:

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

Step-2 : Define an array for Energy (E), with sufficient range

In [44]:

```
E = np.linspace(-0.5,0.5,1001)
```

Step-3 : Define constant values (e,k,u)

In [45]:

```
e = 1.6e-19 #Electric Charge (in Couloumb)
k = 1.38e-23 #Boltzmann Constant (in Joule per Kelvin)
u = 0 #Chemical Potential (Let it be zero)
```

Step-4 : Define a function Fn with parameters T & a, which returns the general fomula result

In [46]:

```
def Fn(T,a):
    return 1/(np.exp(((E-u)*e)/(k*T)) + a) #e is multiplied to make the whole system in
```

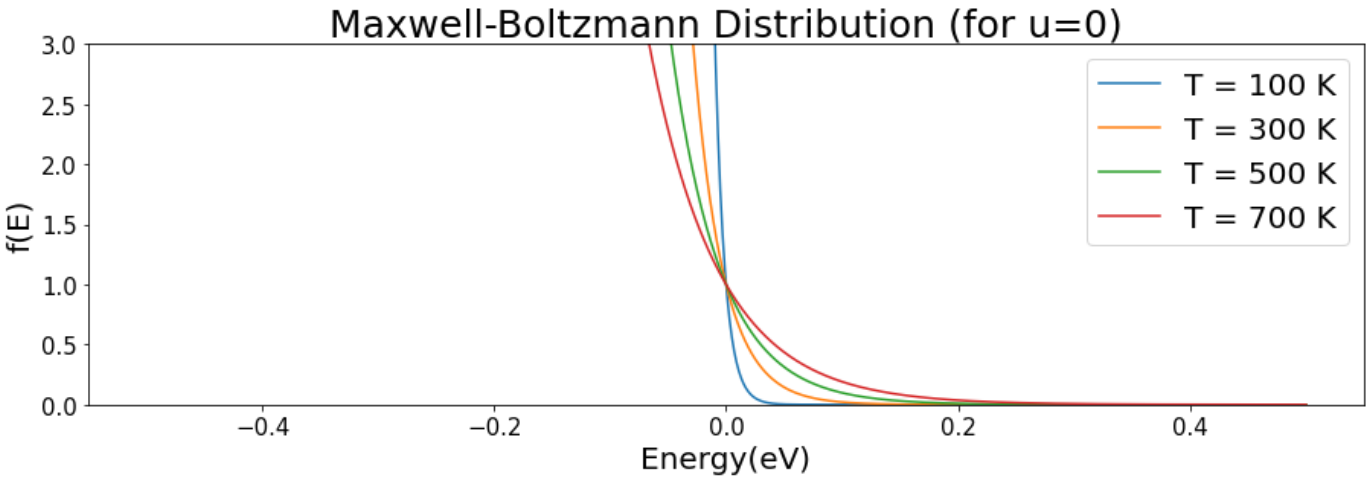
Step-5 : Plotting required graphs

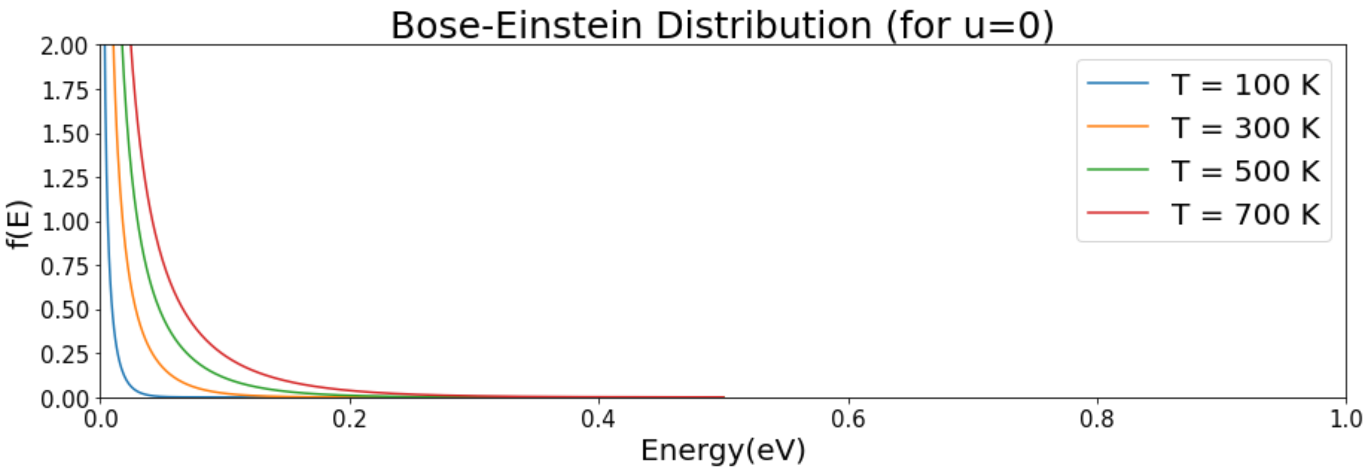
In [47]:

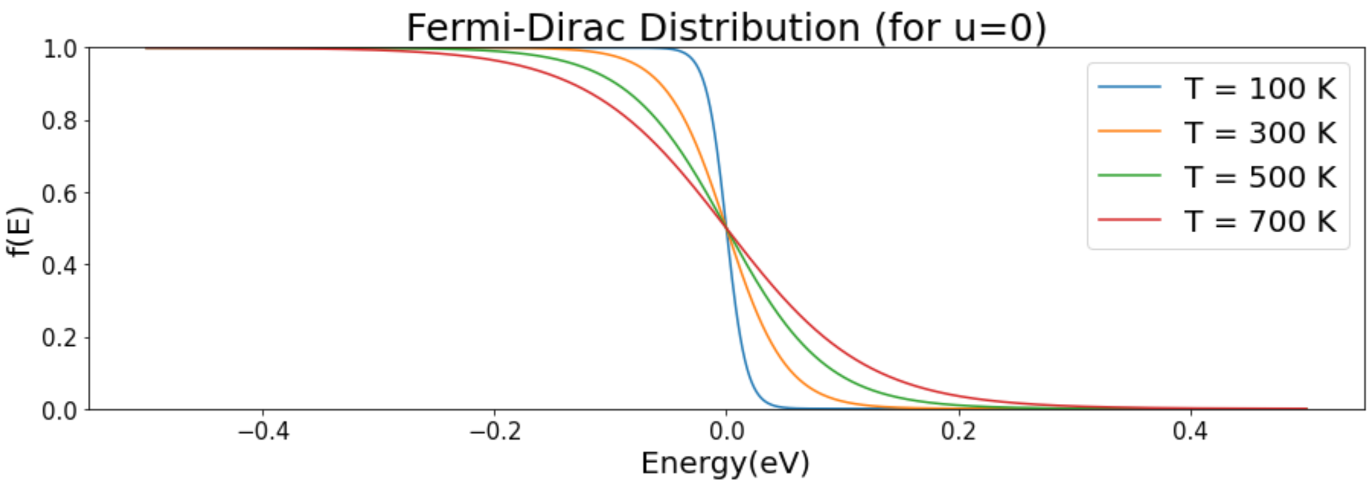
```
#Defining the Super Title of all the 4 graphs
#plt.suptitle("Plot of the following functions at different temperatures",size = 20,co

#1. Maxwell-Boltzmann Distribution
plt.figure(figsize=(15,20))
plt.subplot(4,1,1)
plt.plot(E,Fn(100,0),label='T = 100 K')
plt.plot(E,Fn(300,0),label='T = 300 K')
plt.plot(E,Fn(500,0),label='T = 500 K')
plt.plot(E,Fn(700,0),label='T = 700 K')
plt.ylim(0,3)
plt.xlabel('Energy(eV)',fontsize=20)
plt.ylabel('f(E)',fontsize=20)
plt.legend(loc='best',prop={'size':20})
plt.title("Maxwell-Boltzmann Distribution (for u=0)",fontsize=25)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)

#2. Bose-Einstein Distribution
plt.figure(figsize=(15,20))
plt.subplot(4,1,2)
plt.plot(E,Fn(100,-1),label='T = 100 K')
plt.plot(E,Fn(300,-1),label='T = 300 K')
plt.plot(E,Fn(500,-1),label='T = 500 K')
plt.plot(E,Fn(700,-1),label='T = 700 K')
plt.xlim(0,1)
plt.ylim(0,2)
plt.xlabel('Energy(eV)',fontsize=20)
plt.ylabel('f(E)',fontsize=20)
plt.legend(loc='best',prop={'size':20})
plt.title("Bose-Einstein Distribution (for u=0)",fontsize=25)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)

#3. Fermi-Dirac Distribution
plt.figure(figsize=(15,20))
plt.subplot(4,1,3)
plt.plot(E,Fn(100,+1),label='T = 100 K')
plt.plot(E,Fn(300,+1),label='T = 300 K')
plt.plot(E,Fn(500,+1),label='T = 500 K')
plt.plot(E,Fn(700,+1),label='T = 700 K')
plt.legend(loc='best',prop={'size':20})
plt.ylim(0,1)
plt.xlabel('Energy(eV)',fontsize=20)
plt.ylabel('f(E)',fontsize=20)
plt.title("Fermi-Dirac Distribution (for u=0)",fontsize=25)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)

#4. Comparing all 3 distributions at specific temperature (T=500K)
plt.figure(figsize=(15,20))
plt.subplot(4,1,4)
plt.plot(E,Fn(500,0),label='M-B distribution')
plt.plot(E,Fn(500,-1),label='B-E distribution')
plt.plot(E,Fn(500,+1),label='F-D distribution')
plt.legend(loc='best',prop={'size':20})
plt.ylim(0,4)
plt.xlabel('Energy(eV)',fontsize=20)
plt.ylabel('f(E)',fontsize=20)
plt.title("Comparing all 3 Distributions (At Temperature = 500 K)",fontsize=25)
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)

#Showing the plot
plt.show()
```

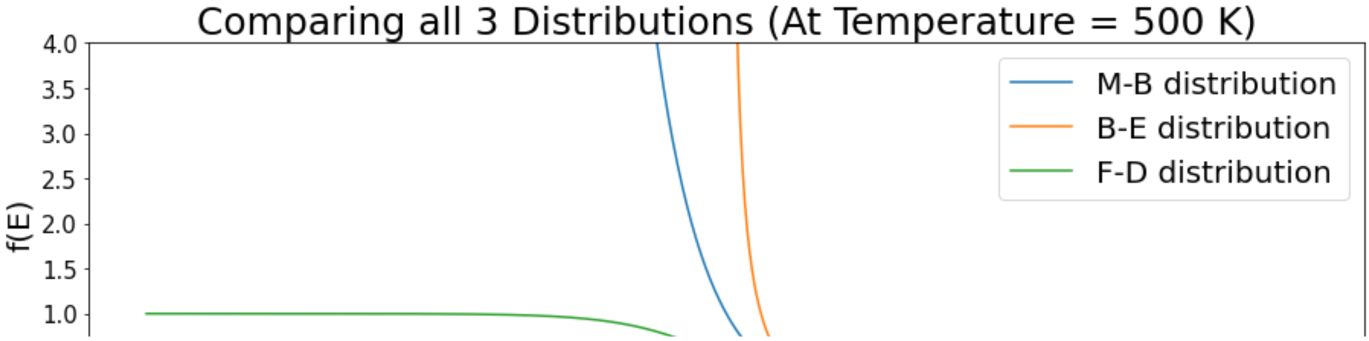
```
/tmp/ipykernel_4786/1138838089.py:2: RuntimeWarning: divide by zero encountered in true
```

```
_divide
  return 1/(np.exp(((E-u)*e)/(k*T)) + a) #e is multiplied to make the whole system in e
V
/tmp/ipykernel_4786/1138838089.py:2: RuntimeWarning: divide by zero encountered in true
_divide
  return 1/(np.exp(((E-u)*e)/(k*T)) + a) #e is multiplied to make the whole system in e
V
/tmp/ipykernel_4786/1138838089.py:2: RuntimeWarning: divide by zero encountered in true
_divide
  return 1/(np.exp(((E-u)*e)/(k*T)) + a) #e is multiplied to make the whole system in e
V
/tmp/ipykernel_4786/1138838089.py:2: RuntimeWarning: divide by zero encountered in true
_divide
  return 1/(np.exp(((E-u)*e)/(k*T)) + a) #e is multiplied to make the whole system in e
V
/tmp/ipykernel_4786/1138838089.py:2: RuntimeWarning: divide by zero encountered in true
_divide
  return 1/(np.exp(((E-u)*e)/(k*T)) + a) #e is multiplied to make the whole system in e
V
```








Comparing all 3 Distributions (At Temperature = 500 K)
f(E)
4.0
3.5
3.0
2.5
2.0
1.5
1.0
M-B distribution
B-E distribution
F-D distribution