

Maxwell-Boltzmann distribution as a function of particle speed at different temperatures

All calculations for a Helium Atom with atomic mass = 4 amu

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import simpson
```

The Different Temperatures

```
T = [10,100,300,500,1000,5000,10000] #in units of K
mass = 4 #mass in a.m.u
#print(E)
col = ["red", "green", "blue", "cyan", "magenta", "purple", "black"]
#print (col)
leg = ["10 K", "100 K", "300 K", "500 K", "1000 K", "5000 K", "10000 K"]
#print(leg)
```

```
k_B = 8.617*10**(-2) # units of meV K-1)
```

Initializing the speeds and the distribution

```
v = np.arange(0,1000,0.1)
print('length of v =', len(v))
n_temp = 7
npoints = 10000
vD = np.zeros([n_temp,npoints])
#vD.shape
```

```
length of v = 10000
```

This function calculates the MB Distribution at temperature T_{temp}

```
def MBDis(v,Temp):
    return (np.sqrt(2/np.pi))*((mass/(k_B*Temp))**(1.5))*(v*v)*np.exp(-(mass*v*v)/(2*k_B*Temp))
```

```
for i in range(n_temp):
    vD[i] = MBDis(v,T[i])
```

```
plt.close()
plt.rcParams.update({'font.size': 14})
plt.rcParams["figure.figsize"] = [12,9]
```

```
for i in range(n_temp):
    plt.plot(v,vD[i,:], color = col[i], label = leg[i], ls = '-', linewidth = 2)
    #plt.plot(v,vD[i,:], dashlist[i], color = col[i], label = leg[i], ls = '.', linewidth = 2)
```

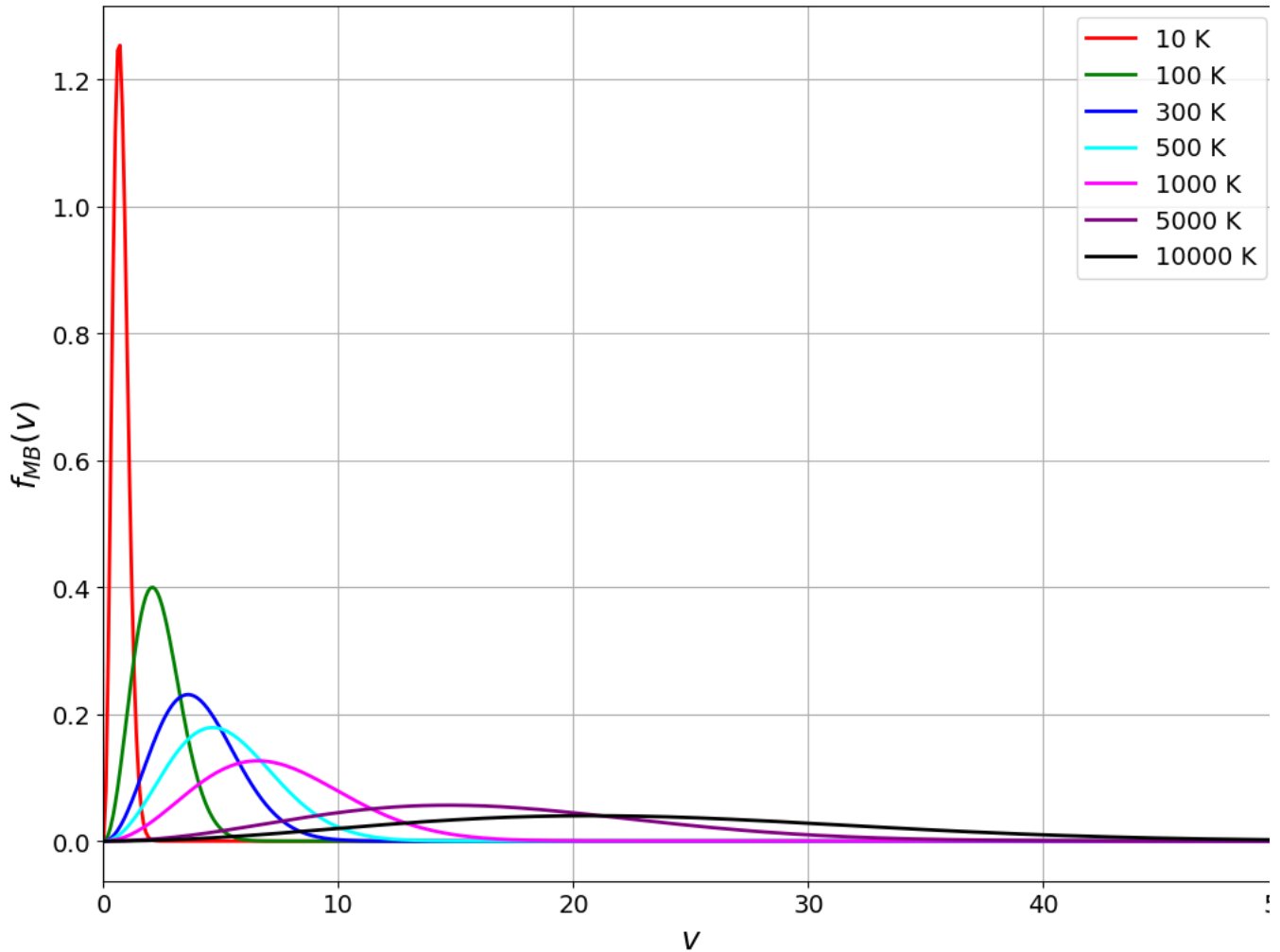
```
#plt.plot((E/(k_B*T)),BD)
#plt.xscale("log")
#plt.yscale("log")
plt.grid(True, which="both")
plt.title('Speed Distribution', fontweight = 'bold', pad = 30)
plt.xlabel(r'$v$', fontsize=18, color='black')
plt.ylabel(r'$f_{MB}(v)$', fontsize=18, color='black')
plt.legend()
plt.xlim(0,50)
#... vlim(0, 1)
```

```

plt.savefig("f_vs_v.pdf")
plt.show()

```

Speed Distribution



Calculating the V_{rms} , V_{mp} and V_{mean} of a Maxwell-Boltzmann Distribution at temperature $Temp$

V_{rms} \equiv Root Mean Square speed; V_{mp} \equiv Most Probable Speed; V_{mean} \equiv average speed

Initializing V_{rms} , V_{mp} , V_{mean}

```

v_rms = np.zeros(n_temp, dtype = float)
v_mp = np.zeros(n_temp, dtype = float)
v_mean = np.zeros(n_temp, dtype = float)
maxval_p_index = np.zeros(n_temp, dtype = int)

```

The most probable speed V_{mp}

```

maxval_p_index = np.argmax(vD, axis = 1)
#print (maxval_p_index)
v_mp = v[maxval_p_index]
#for i in range(len(maxval_p_index)):
#    #print(maxval_p_index[i], v[maxval_p_index[i]], vD[i,maxval_p_index[i]])

print ("The most probable speeds at different temperatures")
for i in range (n_temp):
    print ("Temperature", T[i], "K: " "%4.2f" %v_mp[i], "m/s")

```

```
The most probable speeds at different temperatures
Temperature 10 K: 0.70 m/s
Temperature 100 K: 2.10 m/s
Temperature 300 K: 3.60 m/s
Temperature 500 K: 4.60 m/s
Temperature 1000 K: 6.60 m/s
Temperature 5000 K: 14.70 m/s
Temperature 10000 K: 20.80 m/s
```

▼ The mean speed V_{mean}

```
def integ_v(v, Temp):
    return np.sqrt(2/np.pi)*((mass/(k_B*Temp))**(1.5))*(v**3)*np.exp(-(mass*v*v)/(2*k_B*Temp))

def MeanSpeed(v,Temp):
    integrand = integ_v(v, Temp)
    #print('From MeanSpeed integrand =', integrand)
    return simps(integrand, v)

for i in range(n_temp):
    v_mean[i] = MeanSpeed(v,T[i])

print ("The mean speeds at different temperatures")
for i in range (n_temp):
    print ("Temperature", T[i], "K: " "%4.2f" %v_mean[i], "m/s")

The mean speeds at different temperatures
Temperature 10 K: 0.74 m/s
Temperature 100 K: 2.34 m/s
Temperature 300 K: 4.06 m/s
Temperature 500 K: 5.24 m/s
Temperature 1000 K: 7.41 m/s
Temperature 5000 K: 16.56 m/s
Temperature 10000 K: 23.42 m/s
```

▼ The Root Mean Square speed V_{rms}

```
def integ_vsq(v, Temp):
    return np.sqrt(2/np.pi)*((mass/(k_B*Temp))**(1.5))*(v**4)*np.exp(-(mass*v*v)/(2*k_B*Temp))

def MeanSpeedSq(v,Temp):
    a = min(v)
    b = max(v)
    integrand = integ_vsq(v, Temp)
    #print('From MeanSpeed integrand =', integrand)
    return simps(integrand, v)

for i in range(n_temp):
    v_rms[i] = np.sqrt(MeanSpeedSq(v,T[i]))

print ("The root mean square speeds at different temperatures")
for i in range (n_temp):
    print ("Temperature", T[i], "K: " "%4.2f" %v_rms[i], "m/s")

The root mean square speeds at different temperatures
Temperature 10 K: 0.80 m/s
Temperature 100 K: 2.54 m/s
Temperature 300 K: 4.40 m/s
Temperature 500 K: 5.68 m/s
Temperature 1000 K: 8.04 m/s
Temperature 5000 K: 17.98 m/s
Temperature 10000 K: 25.42 m/s
```

▼ Tabling V_{mp} , V_{mean} , V_{rms}

```
print ()
```

▾ Comparing the ratios V_{mean}/V_{mp} and V_{rms}/V_{mp}

```
vel_ratios = np.zeros((n_temp,3))
print ("The theoretical ratios at any temperature T are = ",1.128, 1.224)
print ("The Calculated ratios are ")
for i in range(n_temp):
    vel_ratios[i,0] = v_mp[i]/v_mp[i]
    vel_ratios[i,1] = v_mean[i]/v_mp[i]
    vel_ratios[i,2] = v_rms[i]/v_mp[i]
    print(" %8.3f    %8.3f    " %(vel_ratios[i,1], vel_ratios[i,2] ), "for temp =", T[i],"K")
```

The theoretical ratios at any temperature T are = 1.128 1.224

The Calculated ratios are

1.058	1.148	for temp = 10 K
1.115	1.211	for temp = 100 K
1.127	1.223	for temp = 300 K
1.139	1.236	for temp = 500 K
1.122	1.218	for temp = 1000 K
1.127	1.223	for temp = 5000 K
1.126	1.222	for temp = 10000 K

✓ 1s completed at 2:57 PM

