Exercise 4)

(a)

most probable velocity is given by $\frac{d}{dv} f(v) \overset{!}{=} 0$:

$$\Leftrightarrow \quad 0 = \frac{d}{dv}\left[N e^{-\frac{mv^2}{2k_BT}} \cdot 4\pi v^2 \right]$$

$$\Leftrightarrow \quad 0 = 8\pi v N e^{-\frac{mv^2}{2k_BT}} + 4\pi v^2 N \cdot \frac{mv}{k_BT} e^{-\frac{mv^2}{2k_BT}}$$

$$\Leftrightarrow \quad 0 = \underbrace{N e^{-\frac{mv^2}{2k_BT}}}_{\neq 0} \left[8\pi v + \frac{4\pi v^3 m}{k_BT} \right]$$

$$\Rightarrow \quad 0 = v \left(8\pi + \frac{4\pi m}{k_BT} v^2 \right)$$

$$\Leftrightarrow \quad v_1 = 0 \quad \vee \quad 0 = v^2 + \frac{\cancel{8\pi}}{\cancel{4\pi}} m k_BT \cdot 2$$

$$\Leftrightarrow \quad v^2 = \frac{2k_BT}{m} \quad \Rightarrow v_2 = \pm\sqrt{\frac{2k_BT}{m}}$$

$\Rightarrow$ the most probable velocity is $v_m = +\sqrt{\frac{2k_BT}{m}}$ as $v_1 = 0$ makes no sense.

Determine the normalization constant N:

$$\int_0^\infty N e^{-\frac{mv^2}{2k_BT}} 4\pi v^2 \, d^3r \overset{!}{=} 1$$

$$\Leftrightarrow \quad 4\pi N \underbrace{\int_0^\infty v^2 e^{-\frac{mv^2}{2k_BT}}}_{\text{gaussian integral where } a = \sqrt{\frac{2k_BT}{m}}} = 1$$

$$\int_0^\infty x^{2n} e^{-\frac{x^2}{a^2}} dx = \sqrt{\pi} \frac{a^{2n+1}(2n-1)!}{2^{n+1}} \quad , n = 1$$

$$\Leftrightarrow \quad 1 = \cancel{4}\pi N \cdot \sqrt{\pi} \left(\frac{2k_BT}{m} \right)^{3/2} \frac{\cancel{1!}}{\cancel{4}}$$

$$\Leftrightarrow \quad N = \left(\frac{m}{2\pi k_BT} \right)^{3/2}$$

(b)

(b)

In [ ]:
```
# import the packages needed
import numpy as np
import scipy.constants as const
import scipy as sc
import matplotlib.pyplot as plt
```

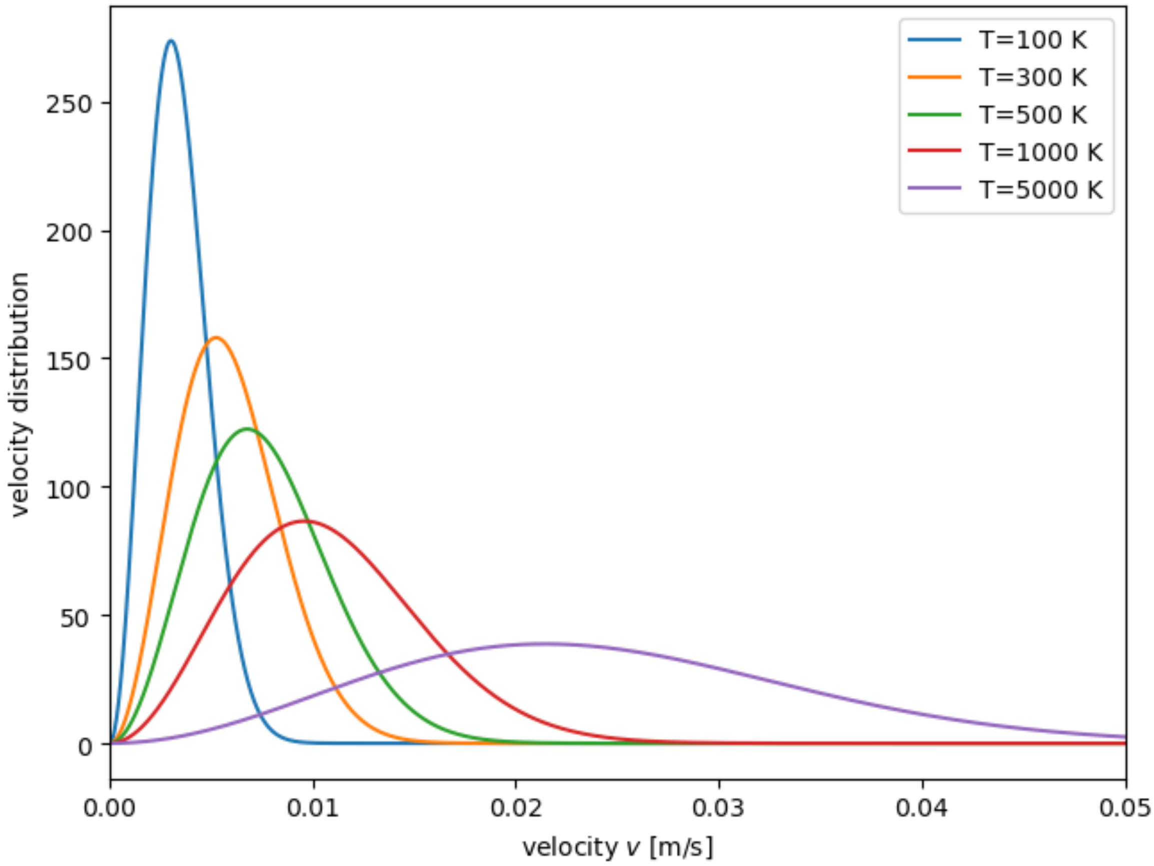
In [ ]:
```
# define all the constants

k_B = const.Boltzmann
m = const.physical_constants["molar mass constant"]
m = m[0]*m[2] # implementing the molecular mass as a float64, instead of
temperature = np.array([100, 300, 500, 1000, 5000])#.astype('float') #sav
N = (m/(2*np.pi*k_B*temperature))**(3/2)
v_m = np.sqrt((2*k_B*temperature)/m)
#print(v_m) #print the the most probable velocities to

v = np.linspace(0,0.05,1000)

for temp in temperature: # define the velocity dist. and plot it
    N = (m/(2*np.pi*k_B*temp))**(3/2)
    f_temp = N * np.exp(-(m*v**2)/(2*k_B*temp))*4*np.pi*v**2
    plt.plot(v, f_temp, label='T={} K'.format(temp))
    plt.legend()
    plt.tight_layout(pad=0, h_pad=1.08, w_pad=1.08)
    plt.xlim(0,0.05)
    plt.xlabel(r'velocity $v$ [m/s]')
    plt.ylabel(r'velocity distribution')

plt.show()
```



In [ ]:
```
# calculate the mean values of each velocity distribution

for temp in temperature:
#    print(f_temp)
    print(r'<v_{}> = '.format(temp), np.mean(f_temp))
```

```
<v_100> =  19.731860288607106
<v_300> =  19.731860288607106
<v_500> =  19.731860288607106
<v_1000> =  19.731860288607106
<v_5000> =  19.731860288607106
```

(c)

In [ ]:
```
from scipy.signal import find_peaks, peak_widths

# calculate the median of each velocity distribution

for temp in temperature:
    peaks, properties = find_peaks(f_temp)
    print(r'v_{} = '.format(temp), f_temp[peaks])
```

```
v_100 =  [38.69981517]
v_300 =  [38.69981517]
v_500 =  [38.69981517]
v_1000 =  [38.69981517]
v_5000 =  [38.69981517]
```

(d)

In [ ]:
```
for temp in temperature:
    results_half = peak_widths(f_temp, peaks, rel_height=0.5)
    print(results_half[0])  # widths
```

```
[472.98198105]
[472.98198105]
[472.98198105]
[472.98198105]
[472.98198105]
```

(e)

In [ ]:
```
for temp in temperature:
    print(r'sigma_v({}) = '.format(temp), np.std(f_temp))
```

```
sigma_v(100) =  12.948458601596856
sigma_v(300) =  12.948458601596856
sigma_v(500) =  12.948458601596856
sigma_v(1000) =  12.948458601596856
sigma_v(5000) =  12.948458601596856
```

In [ ]: