

Représentation de la distribution de vitesse de Maxwell-Boltzmann

Pour la théorie, cf. le cours de physico-chimie ou la [page Wikipédia sur la distribution de vitesse de Maxwell-Boltzmann](http://en.wikipedia.org/wiki/Maxwell-Boltzmann_distribution#Distribution_of_speeds)

Sans NumPy

```
<sxh python; title : Maxwell-Boltzmann_01.py> #!/usr/bin/env python # -*- coding: utf-8 -*- """
NumPy/Matplotlib : representation de la distribution de vitesses de Maxwell-Boltzmann version SANS
utilisation de NumPy cf cours et
http://en.wikipedia.org/wiki/Maxwell-Boltzmann\_distribution#Distribution\_of\_speeds """ import
matplotlib.pyplot as plt from math import pi,exp # importation des fonctions mathématiques exp et pi
plt.figure() #from pylab import *
```

```
def fonc_dist(v):
```

```
    kB = 1.3806504e-23 # constante de Boltzmann
    NA = 6.02214179e23 # nombre d'Avogadro
    T = 298.15 # température (25 Celcius)
    mmg= 28 # masse molaire en g
    m=mmg/(1000*NA)
    factor=4.*pi * (m/(2.*pi*kB*T))**1.5
    expofactor= -m/(2.*kB*T)
    s = factor * v**2 * exp(expofactor * v**2)
    return s
```

```
x=[] # liste standard en python y=[] xx=1. while xx < 2000:
```

```
    x.append(xx)
    y.append(fonc_dist(xx))
    xx=xx+10.
```

```
plt.plot(x,y) plt.show() </sxh>
```

Avec NumPy

```
<sxh python; title : Maxwell-Boltzmann_02.py> #!/usr/bin/env python # -*- coding: utf-8 -*- """
NumPy/Matplotlib : representation de la distribution de vitesses de Maxwell-Boltzmann version AVEC
utilisation de NumPy cf cours et
http://en.wikipedia.org/wiki/Maxwell-Boltzmann\_distribution#Distribution\_of\_speeds """ import
matplotlib.pyplot as plt import numpy as np # utilisation des tableaux numpy, cf.
http://docs.scipy.org/doc/ #les fonctions mathématiques exp et pi seront issues de numppy
```

```
plt.figure() #from pylab import *
```

```
def fonc_dist(v):
```

```
    kB = 1.3806504e-23 # constante de Boltzmann
    NA = 6.02214179e23 # nombre d'Avogadro
    T = 298.15 # température (25 Celcius)
    mmg= 28 # masse molaire en g
    m=mmg/(1000*NA)
    factor=4.*np.pi * (m/(2.*np.pi*kB*T))**1.5
    expofactor= -m/(2.*kB*T)
    s = factor * v**2 * np.exp(expofactor * v**2)
    return s
```

```
x = np.arange(0., 2000., 10.) # array numpy
```

```
plt.plot(x,func_dist(x)) plt.show() </sxh>
```

Suggestion : ajouter des fonctionnalités pour superposer des distributions avec des températures différentes, ou des masses molaires différentes.

From:

<https://dvillers.umons.ac.be/wiki/> - **Didier Villers, UMONS - wiki**

Permanent link:

<https://dvillers.umons.ac.be/wiki/teaching:progappchim:maxwell-boltzmann>

Last update: **2014/01/30 04:19**

