

# Représentation du potentiel de Lennard-Jones

L'utilisation de fonctions en python permet de nombreuses applications par la création de graphiques. En utilisant la "bibliothèque matplotlib/pylab", vous pourrez donc aisément créer des graphes de fonction.

Exemple du [potentiel de Lennard-Jones](#) de l'argon :

$$V_{LJ} = 4\varepsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right]$$

$$V_{LJ} = \varepsilon \left[ \left( \frac{r_m}{r} \right)^{12} - 2 \left( \frac{r_m}{r} \right)^6 \right]$$

où  $\sigma$  est la distance à laquelle le potentiel entre les particules s'annule et  $\varepsilon$  est l'énergie du puits de potentiel d'interaction. La distance  $r_m$  à laquelle le potentiel a cette valeur minimale (pour une interaction entre seulement deux atomes) est reliée à  $\sigma$  par la relation suivante :  $r_m = 2^{1/6} \sigma$

```
<sxh python; title : Lennard-Jones-01.py> #!/usr/bin/env python # -*- coding: utf-8 -*- """
Représentation du potentiel de Lennard-Jones Argon :  $\sigma = 3.405 \text{ \AA}$ ,  $\varepsilon/k_B = 118.2 \text{ K}$   $k_B = 1.3806488(13) \times 10^{-23} \text{ J}$  ->  $\varepsilon = 1.632 \cdot 10^{-21} \text{ J}$  """
from pylab import * def f@: sigma=3.405
#angstrom ! epsilon=1.632 # 10-21 J
```

```
s = (sigma/r)**6
s2= 4.*epsilon* (s**2. - s)
return s2
```

```
r=[] u=[] x=3. while x < 10:
```

```
    r.append(x)
    u.append(f(x))
    x=x+0.1
```

```
plot(r, u) show() </sxh>
```

Suggestion : récrire ce programme en utilisant des directives d'importation standard des bibliothèques Matplotlib/NumPy

## Application : forces de cohésion dans les cristaux de gaz rares

À basse température, les gaz rares peuvent donner des cristaux de structure cubique à face centrée ou hexagonale compacte. La phase cubique centrée peut également être investiguée. Dans chaque de ces structures, chaque atome possède alors d'autres atomes dans son voisinage immédiat (la

coordination 12 ou 8), mais aussi au delà. Pour caractériser l'énergie de cohésion et relier le paramètre du réseau cristallin à  $\sigma$ , il faut considérer la somme de toutes ces interactions, ou du moins de celles correspondant à un voisinage suffisant pour converger. En sommant pour N atomes, on obtient :

$$U_{\text{tot}} = \frac{1}{2} N (4\epsilon) \left[ \sum_j \left( \frac{\sigma}{p_j R} \right)^{12} - \sum_j \left( \frac{\sigma}{p_j R} \right)^6 \right]$$

où R est la distance de séparation entre plus proches voisins et  $p_j$  représente les distances non dimensionnelles (réduites par rapport à R) entre les paires possibles d'atomes par rapport à l'atome de référence.

La distance d'équilibre  $R_0$  pourra être obtenue facilement en minimisant  $U_{\text{tot}}$ . Cela nécessite de calculer pour le réseau envisagé les valeurs de  $\sum_j p_j^{-12}$  et  $\sum_j p_j^{-6}$ .

Voici un programme non optimisé calculant les deux sommes pour la structure cubique à face centrée. Il serait bien sûr intéressant de modifier le programme pour réduire le nombre de calculs nécessaires, et aussi envisager les autres structures (hexagonal compact et cubique centré).

```
<sxh python; title : Lennard-Jones-sum_fcc-01.py> #!/usr/bin/env python # -*- coding: utf-8 -*- """
Calcul des sommes utilisées dans l'énergie de cohésion d'un cristal de gaz rare en utilisant le
potentiel de Lennard-Jones
```

```
Version non optimisée comptabilisant toutes les interactions dans un cristal fictif de structure cubique
à face centrée. Pour une telle maille, les positions des atomes à considérer sont en coordonnées
cristallographiques (0,0,0), (0.5,0.5,0), (0.5,0,0.5) et (0,0.5,0.5). """ from math import *
```

```
motif = [(0,0,0),(0.5,0.5,0),(0.5,0,0.5),(0,0.5,0.5)] R2= 0.52+0.52 # carré de la distance entre
atomes voisins print R2 nmax = 5 # nombre de mailles entre l'atome central et le bord sum6=0.
sum12=0. for ic in range (-nmax,nmax,1):
```

```
    for jc in range (-nmax,nmax,1):
        for kc in range (-nmax,nmax,1):
            # ic, jc, kc est la coordonnée du noeud de référence
            # de la maille cubique.
            noeud=[ic,jc,kc]
            # print noeud # impression pour debug
            for atome in motif:
                d2=0
                for direct in range(3):
                    d2 += (noeud[direct]+atome[direct])**2
                    #print direct, d2 # impression pour debug
                if d2 != 0:
                    inv_pj6 = (R2/d2)**3
                    inv_pj12 = inv_pj6**2
                    sum6 += inv_pj6
                    sum12 += inv_pj12
```

```
print 'La somme "6" vaut ', sum6 print 'La somme "12" vaut ', sum12 </sxh>
```

## Références

- Kittel, physique de l'état solide, chapitre 3
- <http://www.cmp.ucl.ac.uk/~ikr/3225/Section%203.pdf>
- [http://th.fhi-berlin.mpg.de/th/lectures/materialscience-2004/vorlesung\\_2004/fest-k6-2004.pdf](http://th.fhi-berlin.mpg.de/th/lectures/materialscience-2004/vorlesung_2004/fest-k6-2004.pdf)
- Données sur les gaz,  $\sigma$  du potentiel de Lennard-Jones (<http://www.hindawi.com/journals/jther/2013/828620/>) :
  - He : 262.8 pm
  - Ne : 277.5 pm
  - Ar : 340.1 pm
  - Kr : 360.1 pm
  - Xe : 405.5 pm
- Données cristaux (cubic close packed ou face centered, <http://www.webelements.com/>)
  - He : 424.2 pm
  - Ne : 442.9 pm
  - Ar : 525.6 pm
  - Kr : 570.6 pm
  - Xe : 620.23 pm

From:

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

Permanent link:

<https://dvillers.umons.ac.be/wiki/teaching:progappchim:lennard-jones>

Last update: **2015/03/05 12:14**

