

Rotateur biatomique

Cf. [cette page](#).

Code source : <sxh python; title : rotateur_biatomique-01.py> #!/usr/bin/env python # -*- coding: utf-8 -*- """ Somme d'état (ensemble canonique) de rotation (rotateur biatomique) """

from math import exp # on a juste besoin de l'exponentielle import matplotlib.pyplot as plt # directive d'importation standard de Matplotlib

T = 100. # (température réduite = T / Theta) Zrot = 0. # somme d'état Jmax = 30 # valeur maximale de J print '-----' print '| J | dégénérescence | J*(J+1) | terme de Z |' print '-----' Js=[] terms=[] for J in range(Jmax+1):

```
g = 2*J+1 # dégénérescence
term= (2*J+1)*exp(-J*(J+1)/T)
print '| %3d | %4d | %4d | %.8e |' % (J, g, J*(J+1), term)
Zrot += term
Js.append(J)
terms.append(term)
```

print '-----'

print "Somme d'état à T=%3d jusqu'au niveau %2d : %.8e " % (T,Jmax,Zrot)

fig = plt.figure() plt.title(u"Rotateur biatomique et somme d'état à T = "+str(T)) plt.plot(Js, terms, 'bo', label=r'\$\sum_{J=0}^{Jmax} (2J+1) \exp\{-J(J+1)\theta_{rot}/T\}\$') plt.legend(loc='upper right') plt.vlines(Js, [0], terms) plt.xlabel(u"Niveaux J") plt.ylabel(u"Termes de la somme d'état") eq = r'\$Z_{rot} = \sum_{J=0}^{Jmax} (2J+1) \exp\{-J(J+1)\theta_{rot}/T\} = '+str(round(Zrot,3))+"\$" size = 14 x, y = 12,7 plt.text(x, y, eq, fontsize=size, clip_on=True)

plt.show() # vue interactive de la figure </sxh>

From:

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

Permanent link:

https://dvillers.umons.ac.be/wiki/teaching:progappchim:matplotlib_gallery:rotateur_biatomique?rev=1430394992

Last update: 2015/04/30 13:56

