

# Spectre IR du CO

Différentes techniques de spectroscopie utilisent des représentations standardisées des spectres. En [spectroscopie Infrarouge](#), l'absorbance est traditionnellement représentée en fonction des nombres d'ondes décroissants exprimés en  $\text{cm}^{-1}$ . Pour rappel, en spectroscopie, le [nombre d'ondes](#)  $\tilde{\nu}$  est l'inverse de la longueur d'onde, elle-même reliée à la vitesse de la lumière et la fréquence de vibration :

$$\bullet \tilde{\nu} = 1/\lambda = \nu/c$$

Le monoxyde de carbone est une molécule biatomique hétéronucléaire présentant un moment dipolaire dont la valeur peut varier au cours de la vibration. Cette vibration est donc active en IR, mais la transition du niveau fondamental (l'état majoritairement occupé à température ambiante) au premier état excité s'accompagne toujours d'un changement de l'état de rotation ( $\Delta J = \pm 1$ ), ce qui donne une multitude de transitions possible, de l'ordre d'une cinquantaine pour le CO à température ambiante. Nous allons voir comment représenter ce spectre et ces transitions, en utilisant un spectre du monoxyde de carbone mesuré sur un FTIR BRUKER IFS-113v à l'UMONS en 1990, dans une cellule à gaz (cf. [ce fichier](#) au format [csv](#)).

## Graphique des données

Le programme (lecture des données csv, mise en "array" de numpy et représentation par Matplotlib) :

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
"""
Représentation du spectre infrarouge du CO
"""
import csv # module nécessaire "comma separated values"
import numpy as np
import matplotlib.pyplot as plt

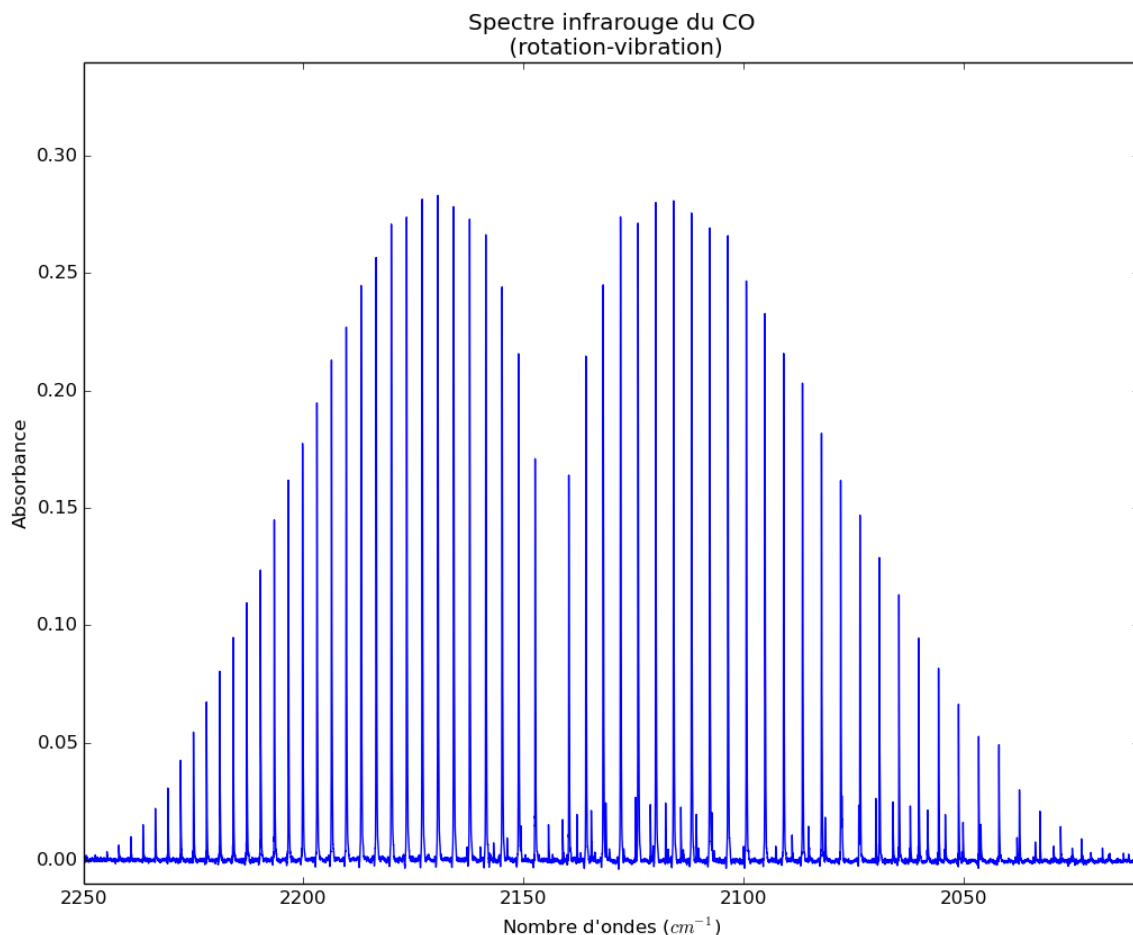
with open("rotvibco.csv", "rb") as csvfile:
    reader = csv.reader(csvfile, delimiter=';')
    no,ab=[],[] # initialisation des listes nombres d'ondes et absorbances
    for row in reader: # on parcourt les lignes successives du fichier
d'entrée
        no.append(float(row[0])) # ajout à la liste no
        ab.append(float(row[1])) # ajout à la liste ab

plt.figure(figsize=(12, 9), dpi=80)
plt.title(u"Spectre infrarouge du CO \n (rotation-vibration)")
x,y=np.array(no),np.array(ab)
plt.xlim(2250.,2010.)
plt.ylim(-0.01,0.34)
plt.xticks(np.linspace(2250., 2050., 5, endpoint=True))
plt.yticks(np.linspace(0., 0.3, 7, endpoint=True))
```

```
plt.xlabel(u"Nombre d'ondes ( $\text{cm}^{-1}$ )")
plt.ylabel(u"Absorbance")

plt.plot(x,y)
plt.show()
```

Le graphique obtenu (version statique) :



## Recherche et affichage des maxima

 : à achever !

- maxima local ? règle logique
- programmation avec liste et parcours
- numpy et slices + opérateurs logiques
- représentation -> maxima parasites
- travailler avec un "subset" -> idée : s'affranchir du bruit, seuil minimum
- moyennes continues
- [numpy.convolve](#) (+ [ref1](#)),  
[<http://glowingpython.blogspot.be/2012/02/convolution-with-numpy.html>|[ref2](#)]
- [scipy.signal](#) maxima searching methods

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

Permanent link:  
[https://dvillers.umons.ac.be/wiki/teaching:progappchim:matplotlib\\_gallery:ir\\_spectrum\\_co](https://dvillers.umons.ac.be/wiki/teaching:progappchim:matplotlib_gallery:ir_spectrum_co)

Last update: **2015/05/12 10:04**

