

```

# -*- coding: utf-8 -*-
"""
Created on Fri Aug 5 10:29:22 2022

@author: fabrice
"""
##dessin des fonctions d'onde radiales de l'atome d'hydrogène avec sympy

from __future__ import division
import numpy as np
from math import *
import matplotlib.pyplot as plt
from sympy import *
x, y, z, t = symbols('x y z t')
k, m, n = symbols('k m n', integer=True)
f, g, h = symbols('f g h', cls=Function)

from sympy.physics.hydrogen import R_nl
from sympy.abc import r, Z
from sympy.plotting import plot

##déclaration graphique (2 graphiques)
fig, ax = plt.subplots(nrows = 3, ncols = 2, figsize=(10, 16))

##définition des styles de lignes, maximum 5 graphes
linestyles = [{ 'ls': '-' }, { 'ls': '--' }, { 'ls': ':' }, { 'ls': '-.' },
               { 'dashes': [2,4,2,4,8,4] }]

##fichier des valeurs de r
x = np.linspace(0, 20, 100)

## n=numéros quantique principal maximal à représenter et angulaire l lié
n = 3
l = n-1

##rayon de bohr égal à 1 pour les graphiques, l'unité de l'abscisse est r/a0
a0 = 1
Z=1/a0

##calcul des fichiers fonction d'onde et densité de probabilité
for i in range(1, n+1):
    for j in range(0, i):

        g=[] ##fichier vide fonction
        f=[] ##fichier vide probabilité

        for r in x:
            lag=R_nl(i, j, r, Z) ##subtilité: Les fonctions sont importées de sympy
            g.append(lag)
            f.append(r*r*lag*lag)

##dessin graphiques
    line, = ax[i-1, 0].plot(x, g, label = 'n='+str(i) + ' l='+str(j), c='k', **linestyles[j])
    line, = ax[i-1, 1].plot(x, f, label = 'n='+str(i) + ' l='+str(j), c='k', **linestyles[j])

##fignoLage
    ax[i-1, 0].set_xlabel('r/a0', fontsize=12)
    ax[i-1, 0].set_ylabel('$\mathrm{Rnl(r)}$', fontsize=12)
    ax[i-1, 0].set_title("Fonction d'onde radiale Rnl(r)", fontsize=12)
    ax[i-1, 0].axhline(0, 0, 20, color='k')
    ax[i-1, 0].legend()
    ax[i-1, 0].minorticks_on()

    ax[i-1, 1].set_xlabel('r/a0', fontsize=12)
    ax[i-1, 1].set_ylabel('$\mathrm{r^2 Rnl(r)^2}$', fontsize=12)
    ax[i-1, 1].set_title('Densité de probabilité Pnl', fontsize=12)
    ax[i-1, 1].legend()
    ax[i-1, 1].minorticks_on()

fig.tight_layout()
plt.show()

```

