

# Travaux préparatoires de Physique Quantique

Thomas Darley

May 22, 2021

## 1 Etude préliminaire

### 1.1 Puits infini

Le Hamiltonien s'écrit, pour ce système

$$H_0 = -\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + V_\infty(x)$$

avec  $V_\infty(x) = +\infty \forall x \in [-\infty, 0] \cup [a, \infty]$  et  $V_\infty(x) = 0 \forall x \in [0, a]$ .

L'équation aux états et valeurs propres s'écrit :

$$H_0 |\varphi_n\rangle = E_n |\varphi_n\rangle$$

Hors du puits,  $\langle x | \varphi_n \rangle = 0$ . Il suffit donc de considérer la zone  $x \in [0, a]$ .

Dans cette dernière, le ket vérifie l'équation différentielle :

$$-\frac{\hbar^2}{2m} \frac{d^2}{dx^2} |\varphi_n\rangle - E_n |\varphi_n\rangle = 0$$

On la résoud aisément et on arrive, en représentation  $x$ , à

$$\varphi_n(x) = A \cos(kx) + B \sin(kx)$$

avec  $k^2 = \frac{2mE_n}{\hbar^2}$ , et a priori  $A, C \in \mathbb{C}^2$ .

Par continuité de la fonction d'onde aux interfaces ( $x=0$  et  $x=a$ ), on obtient des conditions sur la forme de  $\varphi_n$ . En particulier,

$$\varphi_n(0) = 0 \text{ donne } A = 0 \quad \textbf{(a)}$$

$$\varphi_n(a) = 0 \text{ donne } \sin(ka) = 0 \text{ d'où } ka = n\pi, \quad n \in \mathbb{N} \quad \textbf{(b)}$$

De (b), on déduit la forme des énergies propres du système. en remplaçant  $k$  par sa valeur, on arrive bien à

$$E_n = \frac{n^2 \pi^2 \hbar^2}{2ma^2} = n^2 E_1$$

si on note l'énergie fondamentale  $E_1 = \frac{\pi^2 \hbar^2}{2ma^2}$ .

De là, on réécrit la fonction propre

$$\varphi_n(x) = B \sin\left(\frac{n\pi x}{a}\right)$$

On détermine enfin  $B$  par normalisation. La particule se trouve forcément dans le puits (car il est infini), de ce fait, l'intégrale de la probabilité de densité de présence dans le puits vaut 1. Ou encore,

$$\int_0^a |\varphi_n(x)|^2 dx = 1$$

Un rapide calcul mène à  $B = \sqrt{\frac{2}{a}}$ , et on a bien montré les expressions (1) et (2).

## 1.2 Ajout d'un potentiel harmonique

Le Hamiltonien du système est maintenant  $H = H_0 + V_{oh}$  où  $V_{oh}(x) = \frac{1}{2}m\omega^2(x - \frac{a}{2})^2$  potentiel harmonique centré en  $\frac{a}{2}$ .

On cherche maintenant une solution sous la forme  $|\Psi\rangle = \sum_{m=1}^{\infty} c_m |\varphi_m\rangle$ . On remplace d'abord  $|\Psi\rangle$  par cette forme dans l'équation de Schrödinger

$$H|\Psi\rangle = E|\Psi\rangle$$

Cette dernière devient  $\sum_{m=1}^{\infty} c_m H |\varphi_m\rangle = E \sum_{m=1}^{\infty} c_m |\varphi_m\rangle$ .

Le produit scalaire avec  $\langle\varphi_n|$  donne

$$\sum_{m=1}^{\infty} c_m \langle\varphi_n| H |\varphi_m\rangle = E \sum_{m=1}^{\infty} c_m \langle\varphi_n|\varphi_m\rangle \quad (\text{c})$$

Cependant, la base  $\{|\varphi_m\rangle\}$  est orthonormale, et ainsi  $\langle\varphi_n|\varphi_m\rangle = 0 \quad \forall m \neq n$  et  $\langle\varphi_n|\varphi_n\rangle = 1$ .

De plus, le coefficient  $H_{mn}$  de la matrice de  $H$  dans la base  $\{|\varphi_m\rangle\}$  s'écrit  $\langle\varphi_n| H |\varphi_m\rangle$ .

Avec ces considérations, l'équation (c) s'écrit bien

$$\sum_{m=1}^{\infty} H_{nm} c_m = E c_n \quad (5)$$

Afin de s'affranchir des unités, on normalise le potentiel harmonique par l'énergie fondamentale du puits infini  $v_{oh} = \frac{V_{oh}}{E_1}$ . On détermine alors la valeur de  $v_{oh}$  :

$$v_{oh}(x) = \frac{\frac{1}{2} m \omega^2 (x - \frac{a}{2})^2}{E_1} = \frac{m \omega^2 (x - \frac{a}{2})^2}{2 E_1^2} \frac{\pi^2 \hbar^2}{2 m a^2} = \frac{\pi^2}{4} \left( \frac{\hbar \omega}{E_1} \right)^2 \left( \frac{x}{a} - \frac{1}{2} \right)^2$$

On a fait apparaître le rapport  $R = \frac{\hbar \omega}{E_1}$ . Celui-ci compare l'énergie de l'oscillateur  $\hbar \omega$  à l'énergie propre fondamentale du puits. Si  $R \ll 1$ , c'est le puits qui aura un effet prédominant et on sera dans une situation semblable au puits infini. Si  $R \gg 1$ , l'effet du puits sera négligeable et tout se passera comme si la particule n'était soumise qu'au potentiel harmonique. Le code donnant la figure ci-après est fourni en Annexe (A1).

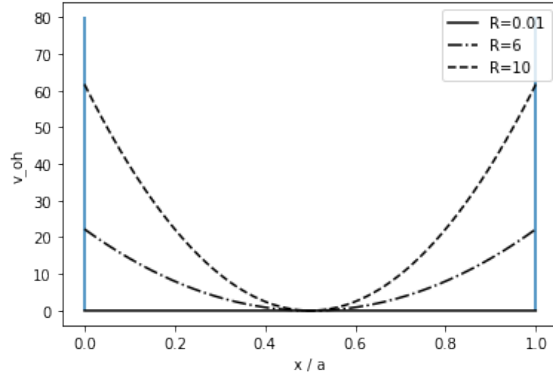


Figure 1: Allure du potentiel normalisé pour différentes valeurs de R

De même, on peut normaliser les coefficients de  $H$  de la même manière, c'est-à-dire  $h_{nm} = \frac{H_{nm}}{E_1}$ . On utilisera la valeur des coefficients fournis en (6).

$$h_{nm} = \frac{E_n}{E_1} \delta_{nm} + \frac{2}{a} \int_0^a \sin\left(\frac{n\pi x}{a}\right) \frac{V_{oh}}{E_1} \sin\left(\frac{m\pi x}{a}\right) dx$$

En effectuant le changement de variable  $x' = \frac{x}{a}$ , licite car  $x \mapsto \frac{x}{a}$  bijectif de  $[0,a]$  dans  $[0,1]$ , avec  $dx = a dx'$ , et en utilisant ce qui a été montré précédemment, on arrive à

$$h_{nm} = n^2 \delta_{nm} + 2 \int_0^1 \sin(n\pi x') v_{oh} \sin(m\pi x') dx' \quad (8)$$

## 2 Résolution numérique

Le code en Annexe (A2) calcule  $h_{nm}$  pour tous m,n et permet de construire la matrice  $H$  normalisée de dimension 6 (ici arrondie au centième):

$$\begin{pmatrix} 47.44 & 0 & 54 & 0 & 10 & 0 \\ 0 & 104.44 & 0 & 64 & 0 & 13.5 \\ 54 & 0 & 119.94 & 0 & 67.5 & 0 \\ 0 & 64 & 0 & 129.94 & 0 & 69.12 \\ 10 & 0 & 67.5 & 0 & 140.56 & 0 \\ 0 & 13.5 & 0 & 69.12 & 0 & 152.44 \end{pmatrix}$$

On observe en particulier que  $\forall i \in \mathbb{N}, H_{i,i\pm 1} = 0$ . En effet, pour tout n,

$$\sin(n\pi x) \sin((n \pm 1)\pi x) = \frac{1}{2} (\cos(n\pi x) - \cos((2n \pm 1)\pi x))$$

De ce fait,

$$\int_0^1 \sin(n\pi x) \sin((n \pm 1)\pi x) dx = \frac{1}{2} \left[ \frac{1}{n\pi} \sin(n\pi x) - \frac{1}{(2n \pm 1)\pi} \sin((2n \pm 1)\pi x) \right]_0^1 = 0$$

Pour les énergies propres, on cherche le spectre de  $H$ . Vu que  $H$  est symétrique réelle, elle est diagonalisable par le théorème spectral, et possède bien N énergies propres distinctes vu qu'aucune n'est dégénérée. Pour N=50, on calcule et trace  $\epsilon(n) = \frac{E(n)}{E_1}$  Cf. Annexe (A3) pour le code.

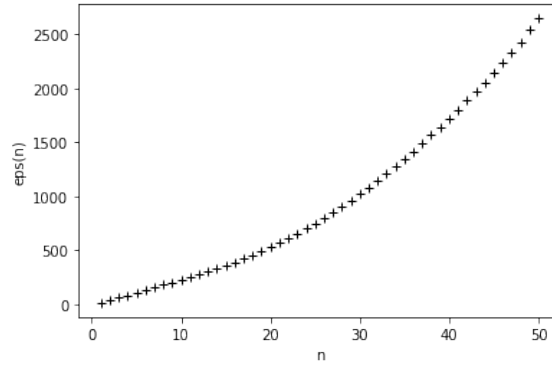


Figure 2: Valeur des énergies propres normalisées en fonction de n

On compare ces valeurs d'énergie aux énergies propres d'un oscillateur harmonique normalisées,  $\epsilon_{OH}(n) = R(n - \frac{1}{2}), \forall n \in \mathbb{N}^*$ .

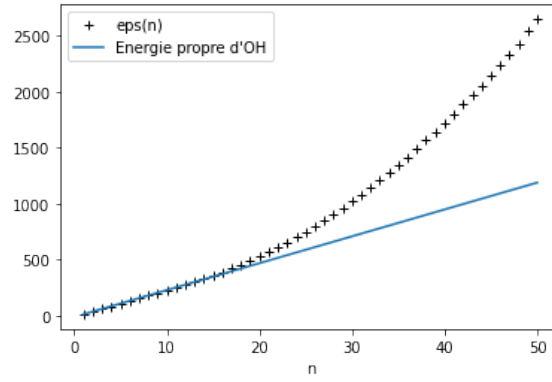


Figure 3: Valeur des énergies propres normalisées en fonction de n

Jusque  $n=17$ , l'approximation est très fidèle (Le code fournissant cette figure est donné en Annexe (A4)).

L'allure de la courbe  $\epsilon(n)$  prend une allure quadratique après  $n=20$ . On fit cette partie de la courbe avec une équation de la forme  $f(n) = n^2 + C$  pour déterminer  $C$ . L'annexe (A5) fournit  $C = 123.03$  et la courbe suivante.

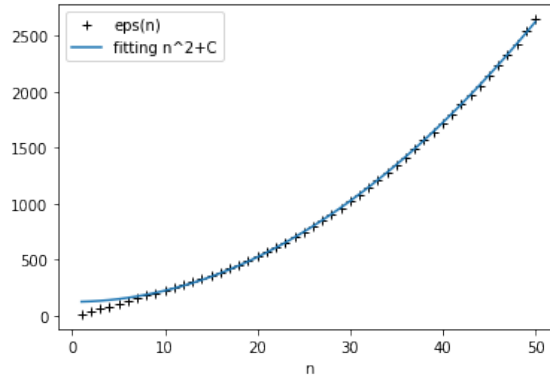


Figure 4: fitting des  $n$  grands

Pour  $n = 15$  on est à peu près à  $R = 4$ , valeur à laquelle le profil du potentiel est entre puits infini et potentiel harmonique (Cf. Fig. 1), il est donc cohérent qu'à cette valeur, on soit à la transition.

On veut enfin tracer  $\forall x \in [0, 1], \Psi_n(x) = \langle x | \Psi_n \rangle$ . La fonction "Psi" en Annexe (A6) renvoie la fonction  $\Psi_n$  pour N quelconque et un ordre quelconque.

On l'utilise donc à l'ordre 0 pour différents N (3, 5, 8, 15) et on la compare à la fonction d'onde de l'oscillateur harmonique à l'ordre 0

$$\Psi_0^{HO}(x) = \left(\frac{m\omega}{\pi\hbar}\right)^{1/4} \exp\left(-\frac{m\omega}{2\hbar} \left(x - \frac{a}{2}\right)^2\right) = \left(\frac{\pi}{2a^2}R\right)^{1/4} \exp\left(-\frac{\pi^2}{4}R \left(\frac{x}{a} - \frac{1}{2}\right)^2\right)$$

Le code en Annexe (A6.1) donne l'allure de l'ordre 0

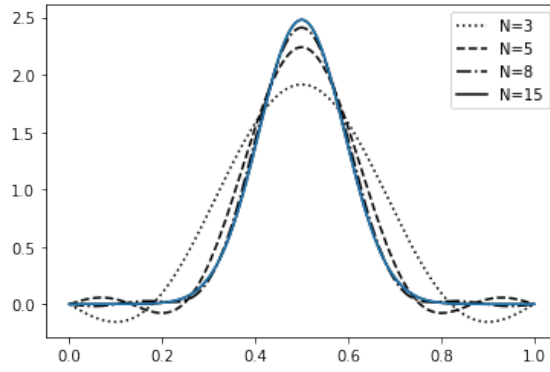


Figure 5: Comparaison de la simulation à différents ordres avec la fonction d'onde harmonique, ordre 0

On fait de même à l'ordre 1, où la fonction d'onde harmonique vaut

$$\Psi_1^{HO}(x) = \left( \frac{\pi^5}{2a^2} R^3 \right)^{1/4} \left( \frac{x}{a} - \frac{1}{2} \right) \exp \left( -\frac{\pi^2}{4} R \left( \frac{x}{a} - \frac{1}{2} \right)^2 \right)$$

Le code en Annexe (A6.2) donne l'allure de l'ordre 1

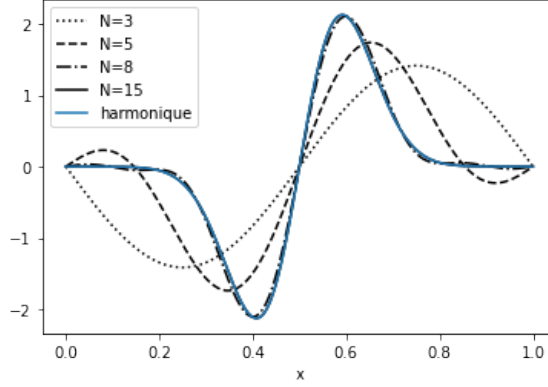


Figure 6: Comparaison de la simulation à différents ordres avec la fonction d'onde harmonique, ordre 1



Enfin à l'ordre 2, la fonction d'onde harmonique vaut

$$\Psi_1^{HO}(x) = \left(\frac{\pi}{8a^2}R\right)^{1/4} \left(\pi^2 R \left(\frac{x}{a} - \frac{1}{2}\right)^2 - 1\right) \exp\left(-\frac{\pi^2}{4}R \left(\frac{x}{a} - \frac{1}{2}\right)^2\right)$$

Le code en Annexe (A6.3) donne l'allure de l'ordre 2

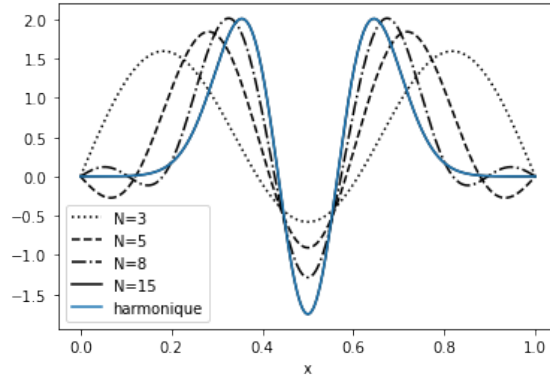


Figure 7: Comparaison de la simulation à différents ordres avec la fonction d'onde harmonique, ordre 2

A tout ordre, plus N augmente, plus la fonction d'onde du système se rapproche de la fonction d'onde harmonique, tout ceci est bien cohérent.

### 3 Annexes

#### Annexe 1

```
1 #A1
2 import numpy as np
3 import matplotlib.pyplot as plt
4 global a
5 a=1
6 def v_oh(R,x):
7     return np.pi**2/4*R**2*(x/a-.5)**2
8 x=np.linspace(0,a,int(1e5))
9
10 plt.plot(x,v_oh(1e-2,x),color='k',label='R=0.01')
11 plt.plot(x,v_oh(6,x),'k-.',label='R=6')
12 plt.plot(x,v_oh(1e1,x),'k--',label='R=10')
13 plt.vlines(0,0,80)
14 plt.vlines(a,0,80)
15 plt.xlabel('x / a')
16 plt.ylabel('v_oh')
17 plt.legend()
```

#### Annexe 2

```
1 #A2
2 import scipy
3 import numpy as np
4 from scipy.constants import hbar
5 from scipy.integrate import quad
6 global R, om, a, m, N
7 R=24
8 a=1
9 m=1
10 om=1
11 eps=1e-8
12
13 N=50
14
15 E_n= lambda n : (n*np.pi*hbar/a)**2/(2*m)
16 v_oh= lambda x : (np.pi/2*R*(x/a-.5))**2
17 delta= lambda n,m : int(n==m)
18 intg= lambda n,m : quad(lambda x : v_oh(x)*np.sin(n*np.pi*x)*np.sin
19 (m*np.pi*x),0,1)
19 H_nm= lambda n,m : n**2*delta(n,m)+2*intg(n,m)[0]
20
21 H=np.zeros((N,N))
22 for i in range(N):
23     for j in range(N):
24         H[i,j]=H_nm(i+1,j+1)
25
26
27 H[np.abs(H) < eps] = 0 #pour viter les art facts de calcul qui
28 devraient valoir 0
29 H=np.around(H,2)
30 print(H)
```

### Annexe 3

```
1 #A3
2 import numpy.linalg as alg
3 import matplotlib.pyplot as plt
4
5 spectre=np.array(sorted(alg.eig(H)[0]))
6 vec_eps=spectre/E_n(1)
7 vec_n=np.linspace(1,N,N)
8
9 plt.plot(vec_n,spectre,'k+')
10 plt.xlabel('n')
11 plt.ylabel('eps(n)')
```

### Annexe 4

```
1 #A4
2 import numpy.linalg as alg
3 import matplotlib.pyplot as plt
4
5 spectre=np.array(sorted(alg.eig(H)[0]))
6 vec_eps=spectre/E_n(1)
7 vec_n=np.linspace(1,N,N)
8 vec_OH=R*np.linspace(.5,N-.5,N)
9 plt.plot(vec_n,spectre,'k+',label='eps(n)')
10 plt.plot(vec_n,vec_OH,label="Energie propre d'OH")
11
12 plt.xlabel('n')
13 plt.legend()
```

### Annexe 5

```
1 #A5
2 from scipy.optimize import curve_fit
3 C=curve_fit(lambda n,C:n**2+C,vec_n[20:],spectre[20:])[0][0]
4 fit=lambda n:n**2+C
5 vec_fit=[fit(n+1) for n in range(N)]
6 plt.plot(vec_n,spectre,'k+',label='eps(n)')
7 plt.plot(vec_n,vec_fit,label="fitting n^2+C")
8 plt.xlabel('n')
9 plt.legend()
```

## Annexe 6

```

1 #A6
2 def Psi(N,n_od): #n_od:ordre de l' tat excit
3     E_n= lambda n : (n*np.pi*hbar/a)**2/(2*m)
4     v_oh= lambda x : (np.pi/2*R*(x/a-.5))**2
5     delta= lambda n,m : int(n==m)
6     intg= lambda n,m : quad(lambda x : v_oh(x)*np.sin(n*np.pi*x)*np
7         .sin(m*np.pi*x),0,1)
8     H_nm= lambda n,m : n**2*delta(n,m)+2*intg(n,m)[0]
9
10    H=np.zeros((N,N))
11    for i in range(N):
12        for j in range(N):
13            H[i,j]=H_nm(i+1,j+1)
14
15    phi= lambda n,x : np.sqrt(2/a)*np.sin(n*np.pi*x/a)
16
17    [spectre,vecs_p]=alg.eig(H)
18    #tri des vecteurs propres
19    ordre=spectre.argsort()
20    vecs_p=vecs_p[:,ordre]
21
22    vec=vecs_p[:,n_od]
23
24    def f(x):
25        f_=0
26        for i in range(N):
27            f_+=vec[i]*phi(i+1,x)
28        return f_ #signe
29
30    return f

```

### Annexe 6.1

```
1 #A6.1
2 psi_oh_0= lambda x : (np.pi/(2*a**2)*R)**0.25*np.e**(-np.pi**2/4*R
3 *(x/a-.5)**2)
4 vec_x=np.linspace(0,1,10000)
5 plt.plot(vec_x,-Psi(3,0)(vec_x),'k:',label='N=3')
6 plt.plot(vec_x,Psi(5,0)(vec_x),'k--',label='N=5')
7 plt.plot(vec_x,Psi(8,0)(vec_x),'k-.',label='N=8')
8 plt.plot(vec_x,-Psi(15,0)(vec_x),'k',label='N=15')
9 plt.plot(vec_x,psi_oh_0(vec_x),label='harmonique')
10 plt.legend()
plt.xlabel('x')
```

### Annexe 6.2

```
1 #A6.2
2 psi_oh_1= lambda x : (np.pi**5/(2*a**2)*R**3)**0.25*(x/a-.5)*np.e
3 **(-np.pi**2/4*R*(x/a-.5)**2)
4 vec_x=np.linspace(0,1,10000)
5 plt.plot(vec_x,-Psi(3,1)(vec_x),'k:',label='N=3')
6 plt.plot(vec_x,-Psi(5,1)(vec_x),'k--',label='N=5')
7 plt.plot(vec_x,Psi(8,1)(vec_x),'k-.',label='N=8')
8 plt.plot(vec_x,-Psi(15,1)(vec_x),'k',label='N=15')
9 plt.plot(vec_x,psi_oh_1(vec_x),label='harmonique')
10 plt.legend()
plt.xlabel('x')
```

### Annexe 6.3

```
1 #A6.3
2 psi_oh_2= lambda x : (np.pi/(8*a**2)*R)**0.25*(np.pi**2*R*(x/a-.5)
3 **2-1)*np.e**(-np.pi**2/4*R*(x/a-.5)**2)
4 vec_x=np.linspace(0,1,10000)
5 plt.plot(vec_x,-Psi(3,2)(vec_x),'k:',label='N=3')
6 plt.plot(vec_x,-Psi(5,2)(vec_x),'k--',label='N=5')
7 plt.plot(vec_x,Psi(8,2)(vec_x),'k-.',label='N=8')
8 plt.plot(vec_x,Psi(15,2)(vec_x),'k',label='N=15')
9 plt.plot(vec_x,psi_oh_2(vec_x),label='harmonique')
10 plt.legend()
plt.xlabel('x')
```