

Physique Quantique

Travaux pratiques numérique de Physique Quantique :
L'oscillateur harmonique dans tous ses états (quantiques)

Auteur :

Arthur Quairel
143

Encadrant :

S.Vlaic
N.Bergeal



Table des matières

1	Introduction	2
2	Etude Préliminaire	3
2.1	Puits de potentiel infini	3
2.2	Potentiel harmonique dans un puits infini	4
3	Résolution Numérique	7
3.1	\mathbf{H} pour $N = 6$	8
3.2	Énergies propres	9
3.3	Fonctions propres	11
4	Puits de potentiel fini dans les hétérostructures de semi-conducteurs	12
5	Conclusion	16
6	Annexes Python	16

1 Introduction

Le but des simulations qui vont être effectuées est de trouver les énergies et les fonctions propres, qui sont des solutions approchées de l'équation de Schrödinger 1D dans le cas d'un potentiel harmonique. Pour réaliser cela, on va utiliser la méthode de F. Marsiglio, qui consiste à placer le potentiel harmonique dans un puits de potentiel infini. Ainsi, le potentiel harmonique sera tronqué par les bords du puits. Cela permet d'utiliser les fonctions propres du puits infini comme une base pour exprimer les fonctions propres de l'oscillateur harmonique.

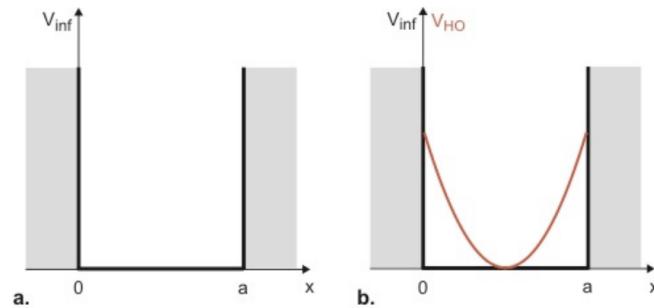


FIGURE 1 – a) puits de potentiel infini unidimensionnel de largeur a . b) Potentiel harmonique unidimensionnel V_{OH} localisé dans un puits infini

2 Etude Préliminaire

2.1 Puits de potentiel infini

On va chercher à déterminer les énergies propres et les fonctions propres à l'intérieur du puits de potentiel infini. On considère donc une particule de masse m dans un puits de potentiel infini de largeur a . L'équation de Schrödinger nous donne :

$$H_0 |\phi\rangle = i\hbar \frac{\partial |\phi\rangle}{\partial t} \quad (1)$$

avec $H_0 = \frac{-\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x)$ l'hamiltonien du système. Cependant les solutions du système vont être supposées stationnaires et le potentiel $V(x)$ du système s'écrit :

$$V(x) = V_{inf}(x) = \begin{cases} 0 & \text{si } x \in [0, a] \\ +\infty & \text{sinon} \end{cases}$$

Ainsi dans le puits de potentiel infini l'équation de Schödinger s'écrit :

$$\frac{-\hbar^2}{2m} \frac{d^2}{dx^2} |\phi\rangle = E^0 |\phi\rangle \quad (2)$$

On applique le bra $\langle x|$ à (2) tel que $\langle x|\phi\rangle = \phi(x)$:

$$\frac{d^2}{dx^2} \phi(x) + \frac{2mE^0}{\hbar^2} \phi(x) = 0 \quad (3)$$

On considère seulement le cas où E^0 est positif sinon les résultats obtenus pour la fonction d'onde seraient une exponentielle décroissante ce qui n'est pas utile pour le problème que l'on souhaite traiter. Ainsi en notant $k = \sqrt{\frac{2mE^0}{\hbar^2}}$, on aura :

$$\phi(x) = A\cos(kx) + B\sin(kx) \quad (4)$$

Il y a continuité de la fonction d'onde à la limite du puits. Or à l'extérieur du puits la fonction d'onde est nulle. Cela implique :

$$\begin{cases} \phi(0) = 0 \implies A = 0 \\ \phi(a) = 0 \implies B\sin(ka) = 0 \implies ka = \frac{n\pi}{a} \text{ avec } n \in \mathbb{N}^* \end{cases}$$

Donc $\phi(x) = B\sin\left(\frac{n\pi x}{a}\right)$ (5)

Or $k_n = \sqrt{\frac{2mE_n^{(0)}}{\hbar^2}}$. Donc on obtient les énergies propres de la particule dans le puits de potentiel infini :

$$E_n^{(0)} = n^2 \frac{\pi^2 \hbar^2}{2ma^2} = n^2 E_1^0 \quad n \in \mathbb{N}^* \quad (6)$$

Le carré du module de la fonction d'onde doit être normalisé sur l'espace. Ainsi on obtient :

$$1 = \int_{-\infty}^{+\infty} |\phi(x)|^2 dx = \int_0^a B^2 \sin^2\left(\frac{n\pi x}{a}\right) dx = B^2 \int_0^a \frac{1 - \cos\left(\frac{2\pi nx}{a}\right)}{2} dx = \frac{B^2 a}{2} - \underbrace{\frac{B^2}{2} \int_0^a \cos\left(\frac{2\pi nx}{a}\right) dx}_{=0}$$

$$\text{Donc } B = \sqrt{\frac{2}{a}}$$

On remplace dans (5) pour obtenir les fonctions propres du système :

$$\boxed{\phi_n(x) = \sqrt{\frac{2}{a}} \sin\left(\frac{n\pi x}{a}\right)} \quad n \in \mathbb{N}^* \quad (7)$$

On a donc ainsi trouvé les fonctions propres et les énergies propres à l'intérieur du puits de potentiel infini.

2.2 Potentiel harmonique dans un puits infini

On se place dans le cas d'un potentiel harmonique unidimensionnel V_{OH} localisé dans un puit infini. Le nouveau potentiel du système est alors :

$$V(x) = V_{inf}(x) + V_{OH}(x) = V_{inf}(x) + \frac{1}{2}m\omega^2(x - \frac{a}{2})^2$$

On peut donc décrire l'hamiltonien du nouveau système :

$$H = H_0 + V_{OH}$$

On cherche donc à trouver les énergies et les fonctions propres de ce système. Les fonctions propres sont les solutions de l'équation de Schrödinger $H|\psi\rangle = E|\psi\rangle$. Soit $|\psi\rangle$ une solution de l'équation précédente, on le développe sur la base Φ .

$$\text{Ainsi } |\psi\rangle = \sum_{m=1}^{\infty} c_m |\phi_m\rangle \quad (8)$$

On trouve c_m les coefficients du développement de $|\psi\rangle$ sur Φ . De plus on peut décrire l'hamiltonien comme une matrice avec les $\{H_{nm}\}_{(n,m) \in \mathbb{N}^{*2}} = \{\langle \phi_n | H | \phi_m \rangle\}_{(n,m) \in \mathbb{N}^{*2}}$ les éléments de matrice. On veut maintenant prouver que dans la base Φ :

$$\sum_{m=1}^{\infty} H_{nm} c_m = E c_n \quad (9)$$

Soit $n \in \mathbb{N}^*$ on applique $\langle \phi_n |$ à l'équation de Schrödinger. De plus comme les $\{|\phi_n\rangle\}_{n \in \mathbb{N}^*}$ forment une base orthonormée, on a :

$$\langle \phi_n | \phi_m \rangle = \delta_{nm}$$

Ainsi lorsqu'on applique $\langle \phi_n |$ à $E \sum_{m=1}^{\infty} c_m |\phi_m\rangle$ on obtient $\boxed{E c_n}$. Pour le membre de gauche on fait le calcul :

$$\langle \phi_n | H | \psi \rangle = \langle \phi_n | \sum_{m=1}^{\infty} c_m H |\phi_m\rangle \rangle = \sum_{m=1}^{\infty} c_m \underbrace{\langle \phi_n | H |\phi_m\rangle}_{H_{nm}} = \sum_{m=1}^{\infty} c_m H_{nm}$$

$$\text{Ainsi } \boxed{\sum_{m=1}^{\infty} c_m H_{nm} = E c_n} \quad (10)$$

Cette formule peut s'écrire différemment en utilisant des vecteurs et des matrices :

$$\underbrace{\begin{pmatrix} H_{11} & \dots & \dots & H_{1m} & \dots \\ \vdots & & & \vdots & \\ H_{n1} & \dots & \dots & H_{nm} & \dots \\ \vdots & & & \vdots & \\ \vdots & & & \vdots & \end{pmatrix}}_H \underbrace{\begin{pmatrix} c_1 \\ \vdots \\ c_m \\ \vdots \end{pmatrix}}_\Psi = E \underbrace{\begin{pmatrix} c_1 \\ \vdots \\ c_m \\ \vdots \end{pmatrix}}_\Psi \quad (11)$$

De cette formule on peut comprendre qu'il faut trouver les valeurs propres et les vecteur propres de H , afin d'obtenir les fonctions propres et énergies propres du système.

On commence par calculer les coefficients de H dans la base Φ :

$$H_{nm} = \langle \phi_n | H | \phi_m \rangle = \langle \phi_n | \underbrace{H_0 | \phi_m \rangle}_{=m^2 E_1^0} + \langle \phi_n | V_{OH} | \phi_m \rangle = m^2 E_1^0 \underbrace{\langle \phi_n | \psi_m \rangle}_{=\delta_{nm}} + \langle \phi_n | V_{OH} | \phi_m \rangle \quad (12)$$

$$= n^2 E_1^0 \delta_{nm} + \langle \phi_n | V_{OH} | \phi_m \rangle \quad (13)$$

Pour la suite de l'exercice on va poser un potentiel harmonique normalisé :

$$\begin{aligned} \nu_{OH}(x) &= \frac{V_{OH}(x)}{E_1^0} = \frac{m\omega^2}{2E_1^0} \left(x - \frac{a}{2}\right)^2 \quad \text{or } E_1^0 = \frac{\pi^2 \hbar^2}{2ma^2} \\ &= \frac{\pi^2 m \hbar^2 \omega^2}{4ma^2 E_1^{02}} \left(x - \frac{a}{2}\right)^2 \end{aligned}$$

$$\text{Donc } \nu_{OH}(x) = \frac{\pi^2}{4} \left(\frac{\hbar\omega}{E_1^0}\right)^2 \left(\frac{x}{a} - \frac{1}{2}\right)^2$$

(14)

Calculons maintenant ce produit scalaire :

$$\begin{aligned}
\langle \phi_n | V_{OH} | \phi_m \rangle &= \int_0^a \phi_n(x) V_{OH}(x) \phi_m(x) dx \\
&= \int_0^a \frac{2}{a} \sin\left(\frac{n\pi x}{a}\right) V_{OH}(x) \sin\left(\frac{m\pi x}{a}\right) dx \\
&= 2E_1^0 \int_0^1 \sin(n\pi x') \nu_{OH}(x') \sin(m\pi x') dx' \quad \text{où on a posé } x' = \frac{x}{a}
\end{aligned}$$

Ainsi l'expression du coefficient H_{nm} dans la base Φ est :

$$H_{nm} = n^2 E_1^0 \delta_{nm} + 2E_1^0 \int_0^1 \sin(n\pi x) \nu_{OH}(x) \sin(m\pi x) dx \quad (15)$$

On pose maintenant $\mathbf{H} = \frac{1}{E_1^0} H$ l'hamiltonien normalisé du système. Cette normalisation nous permet de généraliser le comportement de la particule de masse m à toute sorte de particule qui n'aurait pas la même énergie fondamentale. Les coefficients de \mathbf{H} sont :

$$h_{nm} = n^2 \delta_{nm} + 2 \int_0^1 \sin(n\pi x) \nu_{OH}(x) \sin(m\pi x) dx \quad (16)$$

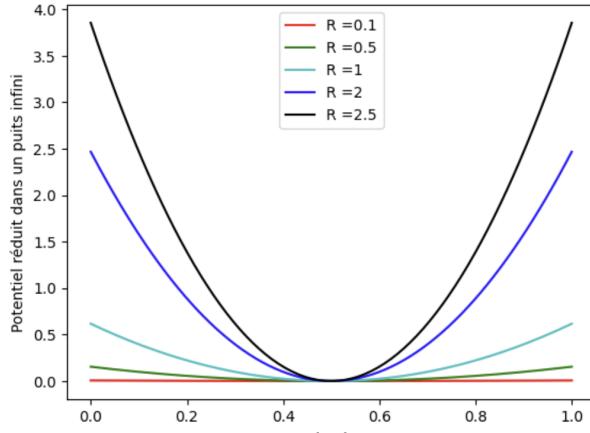


FIGURE 2 – Graphe du potentiel harmonique normalisé dans le puits infini pour différents R en fonction $x' = \frac{x}{a}$

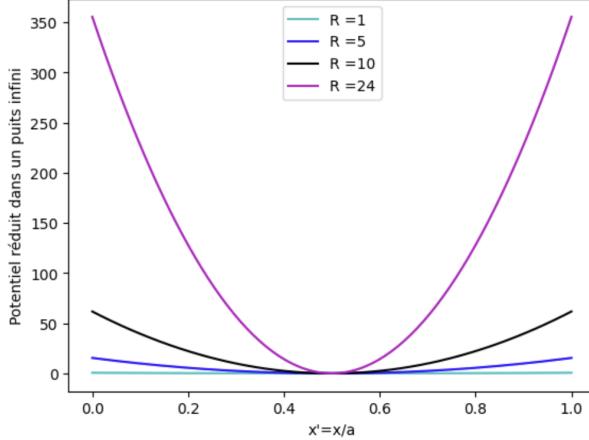


FIGURE 3 – Graphe pour des R plus importants

Le rapport $R = \frac{\hbar\omega}{E_1^0}$ permet de définir le comportement de la particule dans le potentiel harmonique tronqué. On peut remarquer que si $R \ll 1$, l'énergie du mode fondamental du puits domine l'énergie de l'oscillateur harmonique. On peut considérer que, dans ce cas, le potentiel auquel est soumise la particule est modifié, et la particule ne voit plus le potentiel global $V = V_{OH} + V_{inf}$ mais seulement le potentiel V_{inf} . On en déduit donc que l'effet de l'oscillateur harmonique est négligeable.

Dans le régime $R \gg 1$, comme on l'observe sur la courbe avec $R = 10$, on remarque que l'ordonnée d'intersection entre le puits infini et le potentiel harmonique est grande, et que la particule se comporte comme dans un potentiel harmonique simple.

Ainsi, l'interprétation physique de R est celle d'un facteur qui délimite les comportements de la particule entre le puits de potentiel infini ou le potentiel harmonique.

On peut aussi le constater mathématiquement en exprimant le rayon de courbure en $x = \frac{a}{2}$ de notre potentiel harmonique :

$$R_c = \frac{(1 + \nu'_{OH}^2(\frac{a}{2}))^{\frac{3}{2}}}{\nu''_{OH}(\frac{a}{2})} \text{ or } \nu_{OH}(x) = \frac{\pi^2}{4} R^2 (\frac{x}{a} - \frac{1}{2})^2$$

Donc $R_c = \frac{2a^2}{\pi^2} \frac{1}{R^2}$

Avec cette expression, on remarque que plus R est grand, plus le rayon de courbure est important, et plus le potentiel harmonique adopte un profil proche d'une droite horizontale sur le schéma. Le potentiel sera donc négligeable. Il est donc nécessaire de considérer une valeur de R suffisamment grande pour étudier la particule.

3 Résolution Numérique

Théoriquement, le vecteur $|\psi\rangle$ introduit dans la section précédente s'exprime comme une combinaison linéaire d'une infinité de vecteurs $|\phi_n\rangle$. Cependant, dans une approche numérique, il est impossible de manipuler une somme infinie, sauf à disposer d'un temps de calcul illimité.

D'après la distribution de Boltzmann, la probabilité qu'une particule possède une énergie E est proportionnelle à $\exp\left(-\frac{E}{k_b T}\right)$, où k_b est la constante de Boltzmann et T la

température du système. Cette relation montre que les états d'énergie élevée sont moins peuplés.

Ainsi, pour rendre notre approche numérique réalisable, nous nous limiterons aux $N \in \mathbb{N}^*$ premiers vecteurs $|\phi_n\rangle$ de la base Φ , correspondant aux N niveaux d'énergie les plus bas. Plus N est grand, plus nos résultats se rapprocheront de la solution théorique.

Enfin, nous fixerons $\mathbf{R} = \mathbf{24}$ pour l'ensemble de nos simulations numériques.

3.1 H pour $N = 6$

Pour $N=6$, avec le code python présent en annexe, on trouve, en utilisant seulement 4 chiffres significatifs, que la matrice de l'hamiltonien est

$$\mathbf{H} = \begin{pmatrix} 47.4353 & 0 & 54 & 0 & 10 & 0 \\ 0 & 104.4353 & 0 & 64 & 0 & 13.5 \\ 54 & 0 & 119.4353 & 0 & 67.5 & 0 \\ 0 & 64 & 0 & 129.9353 & 0 & 69.12 \\ 10 & 0 & 67.5 & 0 & 140.5553 & 0 \\ 0 & 13.5 & 0 & 69.12 & 0 & 152.4353 \end{pmatrix} \quad (17)$$

On remarque qu'un terme non diagonal sur deux est nul. Pour cela, on peut donner plusieurs explications. Premièrement, physiquement, la base Φ n'est pas une base de vecteurs propres de \mathbf{H} , sinon la matrice \mathbf{H} serait diagonale. Ainsi, le terme h_{nm} représente le couplage énergétique entre le vecteur $|\phi_n\rangle$ et le vecteur $|\phi_m\rangle$. Si ce terme est nul, cela implique que ces deux états ont une énergie d'interaction nulle. Or, $\phi_n(x)$ est proportionnelle à $\sin(n\pi x)$, et cette fonction est paire (respectivement impaire) par rapport à \mathbf{D} , la droite verticale passant par $x = 0.5$ si n est impair (respectivement pair).

On trouve que les fonctions paires et impaires par rapport à \mathbf{D} sont découplées. Il y a l'apparition de deux classes de vecteurs. L'interprétation physique de ces coefficients nuls étant donnée, prouvons qu'ils sont bien nuls théoriquement.

On va donc le montrer théoriquement :

$$h_{nm} = n^2 \delta nm + \frac{\pi^2 R^2}{2} \int_0^1 \sin(n\pi x) \sin(m\pi x) (x - \frac{1}{2})^2 dx \quad (18)$$

On va donc procéder au calcul de l'intégrale, on suppose $n \geq m$:

$$I_{n,m} = \int_0^1 \sin(n\pi x) \sin(m\pi x) (x - \frac{1}{2})^2 dx \quad (19)$$

La formule du produit de 2 sinus donne :

$$I_{n,m} = \frac{1}{2} \int_0^1 (x - \frac{1}{2})^2 (\cos(\pi x(n-m)) - \cos(\pi x(n+m))) dx \quad (20)$$

De plus $\forall M \in \mathbb{N}^*$ avec deux intégrations par parties on peut avoir la valeur de :

$$J_M = \int_0^1 (x - \frac{1}{2})^2 \cos(\pi x M) dx = \frac{1}{(\pi M)^2} ((-1)^M + 1) = \begin{cases} 0 & \text{si } M \text{ est impair} \\ \frac{2}{(\pi M)^2} & \text{si } M \text{ est pair} \end{cases} \quad (21)$$

$$\text{Donc si } n \neq m \quad I_{n,m} = \frac{1}{2} (J_{n-m} - J_{n+m}) = \begin{cases} 0 & \text{si } n \text{ et } m \text{ sont de parité différente} \\ \frac{4nm}{\pi^2(n^2 - m^2)^2} & \text{si } n \text{ et } m \text{ sont de même parité} \end{cases} \quad (22)$$

En effet la somme ou la différence de 2 entiers naturels de parité égale (respectivement opposée) est un entier naturel pair (respectivement impair).

$$\text{Dans le cas } n=m \quad I_{n,n} = \frac{1}{2} \left(\int_0^1 (x - \frac{1}{2})^2 dx - J_{2n} \right) = \frac{1}{4} \left(\frac{1}{6} - \frac{1}{\pi^2 n^2} \right) \quad (23)$$

Voici l'expression finale des coefficients de \mathbf{H} :

$$\boxed{\text{Si } n=m \quad h_{nm} = n^2 + \frac{R^2}{8} \left(\frac{\pi^2}{6} - \frac{1}{n^2} \right)} \quad (24)$$

$$\boxed{\text{si } n \neq m \quad h_{nm} = \begin{cases} 0 & \text{si } n \text{ et } m \text{ ne sont pas de même parité} \\ \frac{R^2}{2} \frac{4nm}{(n^2 - m^2)^2} & \text{sinon} \end{cases}} \quad (25)$$

On a donc montré théoriquement qu'un terme non diagonal sur deux de \mathbf{H} était nul.

3.2 Énergies propres

Nous allons maintenant nous intéresser aux énergies propres normalisées de la particule $\varepsilon(n) = \frac{E(n)}{E_1^0}$. Nous allons comparer les énergies normalisées théoriques avec les énergies propres d'un oscillateur harmonique sans puits infini. On a que les expressions des énergies d'un oscillateur harmonique sont : $\varepsilon_{OH}(n) = R(n - \frac{1}{2})$ avec $n \in \mathbb{N}^*$. On représente $\varepsilon(n)$ et $\varepsilon_{OH}(n)$ sur un même graphique.

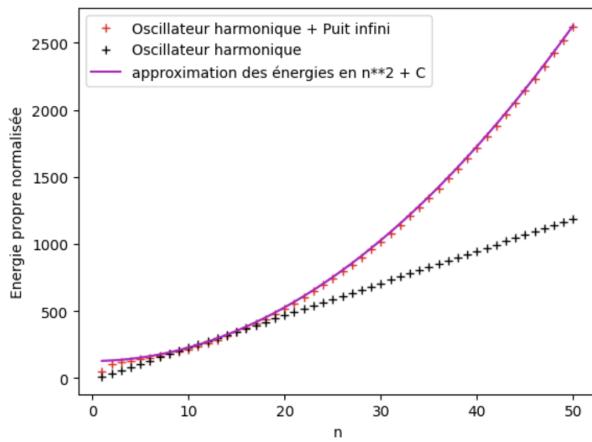


FIGURE 4 – Comparaison des $\varepsilon(n)$ et $\varepsilon_{OH}(n)$

On remarque que les points des deux énergies se superposent très bien pour $n \leq 15$. Pour $n \geq 15$, l'écart entre les $\varepsilon(n)$ et les $\varepsilon_{OH}(n)$ augmente. Le comportement des énergies propres lorsque n augmente ne correspond plus à un potentiel harmonique. On remarque que $\varepsilon(n)$ décrit une parabole quadratique.

On aurait donc $\varepsilon(n) = n^2 + C$, avec C une constante que l'on détermine avec le code Python. On trouve $C = 118.343$, et la courbe est tracée sur la figure précédente. On observe un bon accord avec $\varepsilon(n)$ pour $n \geq 15$.

Nous allons maintenant essayer de trouver d'où provient cette constante C . On remarque que dans l'équation pour trouver les coefficients diagonaux de \mathbf{H} dans la base Φ est de la forme d'un terme quadratique n^2 et d'un terme en inverse quadratique $\frac{R^2}{8n^2}$, et d'un terme constant. $\frac{\pi^2 R^2}{48}$. On néglige le terme d'inverse quadratique pour des n grands.

$$\text{Les coefficients diagonaux s'écrivent donc } h_{nn} \underset{n \text{ grand}}{\approx} n^2 + \frac{\pi^2 R^2}{48}. \quad (26)$$

Pour $R=24$ on obtient $\frac{\pi^2 R^2}{48} = 118.43$ ce qui est très proche de $C = 118.343$ qui est la valeur que nous avons trouvée avec le script python. On va s'assurer que C est bien $\frac{\pi^2 R^2}{48}$, on trace C en fonction de différentes valeurs de R . On peut dire que $\frac{\pi^2 R^2}{48}$ est **la valeur moyenne du potentiel harmonique réduit** ($\nu_{OH}(x)$). On trace les valeurs de C calculées numériquement et la valeur moyenne du potentiel harmonique réduit :

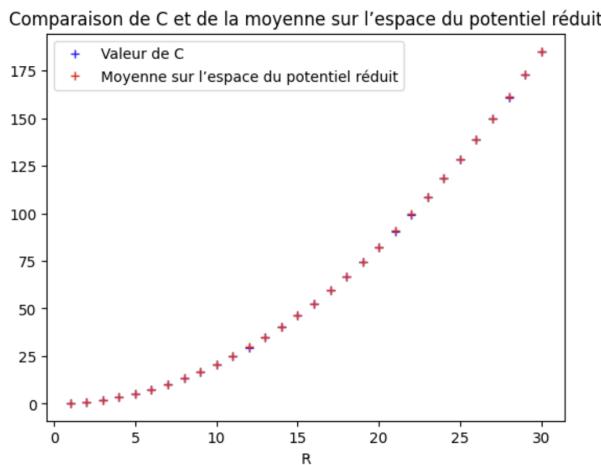


FIGURE 5 – Comparaison entre C calculé numériquement et la valeur moyenne du potentiel harmonique réduit en fonction de R

Les deux courbes sont très proches. On peut donc en conclure que $C(R)$ vaut la valeur moyenne du potentiel harmonique réduit pour un R donné.

On peut donner une interprétation physique à cela. En effet, pour n grand, $\varepsilon(n)$ correspond aux énergies d'un puits de potentiel infini dont la valeur du plateau serait fixée à C . Or, nous avons prouvé que C correspond à la moyenne du potentiel harmonique réduit. La particule ne "voit" donc plus le potentiel harmonique, mais uniquement sa moyenne pour les énergies élevées.

Nous allons maintenant essayer d'expliquer pourquoi la transition entre les 2 régimes identifiés précédemment se produit pour $n \approx 15$. On peut raisonner en termes d'énergie

propres et se dire que le changement d'état arrive lorsque l'énergie normalisé de la particule dépasse la valeur du potentiel harmonique normalisé sur les bords du puits infini. Si on considère une particule dont l'énergie est supérieure à cette limite $\frac{\pi^2 R^2}{16}$ alors l'impact du potentiel harmonique devient négligeable devant celui du puits infini.

Pour $R=24$ on peut calculer $\varepsilon(14) = 328.28$ et $\varepsilon(15) = 356.14$. Or pour $R=24$ $\nu_{OH}(1) = 355.30$, cette limite est donc bien dépassée par $\varepsilon(n)$ quand n passe à 15.

3.3 Fonctions propres

Les vecteurs propres de la matrice \mathbf{H} forment les fonctions propres $|\psi_n\rangle$ du système avec n le numéro du n^{me} état excité. Ces vecteurs s'expriment dans la base Φ :

$$|\psi_n\rangle = \sum_{\Phi} c_i^n |\phi_i\rangle \quad (27)$$

où les coefficients $\{C_i^n\}_{i \in \mathbb{N}^*}$ sont les coordonnées du vecteur $|\psi\rangle_n$ dans la base Φ . On peut donc tracer ces fonctions, $\psi_n(x) = \langle x|\psi_n\rangle$, avec notre code Python . On ajoute les courbes des fonctions théoriques pour étudier si l'accord entre le numérique et le théorique est bon ; pour cela, on va développer à l'ordre 2 l'expression théorique donnée par l'énoncé (qui est la fonction d'onde vue en cours) :

$$\psi_0^{th}(x) = \left(\frac{\pi}{2a^2}R\right)^{1/4} \exp\left(-\frac{\pi^2}{4}R\left(\frac{x}{a} - \frac{1}{2}\right)^2\right) \quad (28)$$

$$\psi_1^{th}(x) = \left(\frac{\pi^5}{2a^2}R^3\right)^{1/4} \left(\frac{x}{a} - \frac{1}{2}\right) \exp\left(-\frac{\pi^2}{4}R\left(\frac{x}{a} - \frac{1}{2}\right)^2\right) \quad (29)$$

$$\psi_2^{th}(x) = \left(\frac{\pi}{8a^2}R\right)^{1/4} \left(\pi^2R\left(\frac{x}{a} - \frac{1}{2}\right)^2 - 1\right) \exp\left(-\frac{\pi^2}{4}R\left(\frac{x}{a} - \frac{1}{2}\right)^2\right) \quad (30)$$

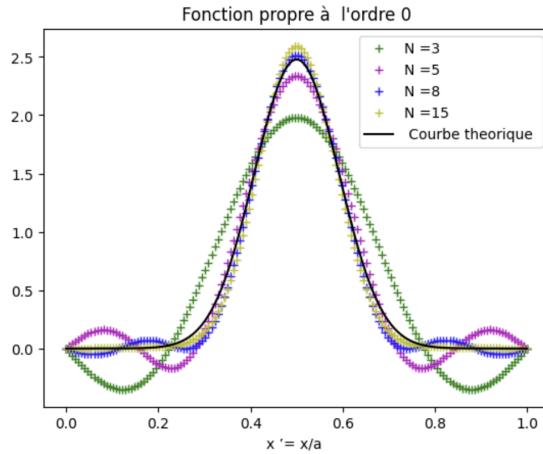


FIGURE 6 – Tracé des fonctions d'ondes pour différents états exités à l'ordre 0

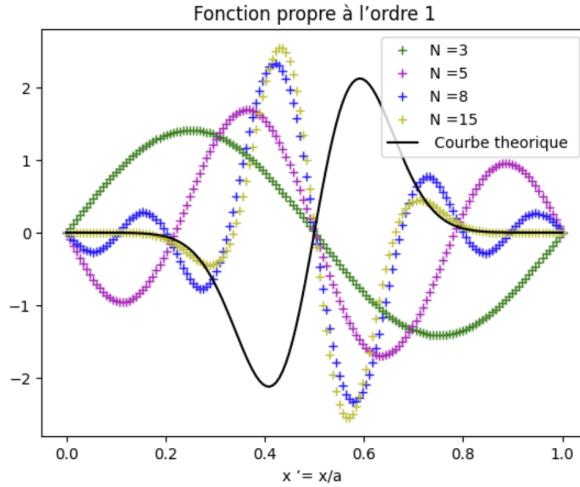


FIGURE 7 – Tracé des fonctions d’ondes pour différents états excités à l’ordre 1

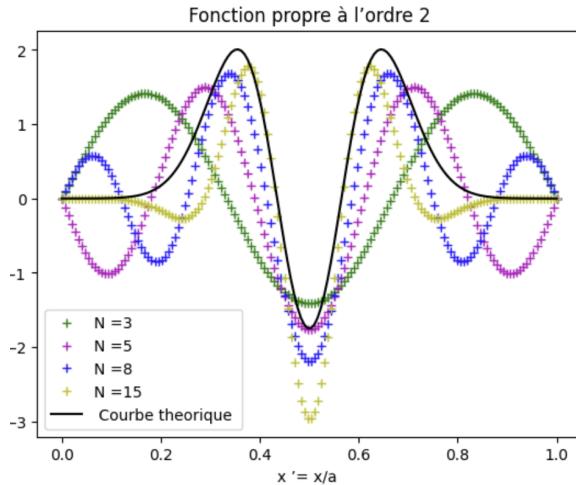


FIGURE 8 – Tracé des fonctions d’ondes pour différents états excités à l’ordre 2

La méthode numérique pour trouver les fonctions propres de l’oscillateur harmonique quantique fonctionne donc très bien, plus N est grand plus les résultats de simulation s’approchent des résultats théoriques ce qui était bien attendu par l’énoncé.

4 Puits de potentiel fini dans les hétérostructures de semi-conducteurs

Dans la partie IA) on a prouvé que dans un puits de potentiel infini de largeur a entre 0 et a , la fonction propre du puit était :

$$\boxed{\phi_n(x) = \sqrt{\frac{2}{a}} \sin\left(\frac{n\pi x}{a}\right)} \quad n \in \mathbb{N}^* \quad (31)$$

En remplaçant a , la longueur du puits précédent par $2b$ pour décrire le puits donnée par l’énoncé et en centrant le puits en 0 on va donc devoir réaliser un changement de

variable dans le sinus, on pose $x = z + b$ pour centrer le puits en 0. On obtient alors :

$$\boxed{\phi_n(z) = \sqrt{\frac{1}{b}} \sin\left(\frac{n\pi}{2b}(z+b)\right)} \quad n \in \mathbb{N}^* \quad (32)$$

Dans la partie précédente, on avait trouvé que les énergies propres du puits étaient :

$$\boxed{E_n^{(0)} = n^2 \frac{\pi^2 \hbar^2}{2ma^2}} \quad n \in \mathbb{N}^* \quad (33)$$

Or avec l'énoncé on sait que les énergies propres sont décalées de $-V_0$ par rapport au puits étudiés précédemment, de plus on applique le changement $a = 2b$ ce qui nous donne que les énergies propres de ce nouveau puits sont :

$$\boxed{E_n^0 = n^2 \frac{\pi^2 \hbar^2}{8mb^2} - V_0} \quad n \in \mathbb{N}^* \quad (34)$$

On va maintenant appliquer la méthode de la partie 1 pour trouver les fonctions et les énergies propres du puits fini. Les termes de la matrice de l'hamiltonien sont donc :

$$H_{nm} = n^2 E_1^0 \delta_{nm} + \int_{-b}^b \phi_n(z) * V(z) * \phi_m(z) dx \quad (35)$$

Notre code python permet de trouver les 3 états propres confinés dans le puits fini. En augmentant la précision pour $N > 100$, on trouve les énergies suivantes ; on a affiché les 15 valeurs propres les plus basses. On observe bien le fait que on a 3 états propres confinés dans le puits fini car leurs énergies sont négatives et donc inférieures à celle du puits.

```
Les 15 plus basses valeurs propres (en eV) :
[-0.44372193 -0.28171122 -0.05073253  0.07317706  0.11175119  0.24446636
 0.36088625  0.49118571  0.69187783  0.86182766  1.07434256  1.32611921
 1.55991267  1.84150886  2.1477969 ]
```

FIGURE 9 – Les 15 valeurs propres les plus basses du système avec un puits de potentiel fini

On trouve aussi les 3 énergies des états confinés :

Énergies propres < 0 :

État 1: $E = -0.443722$ eV

État 2: $E = -0.281711$ eV

État 3: $E = -0.050733$ eV

Nombre d'états liés ($E < 0$) = 3

FIGURE 10 – Les 3 énergies des états confinés du système

Après avoir déterminé les énergies des états propres confinés,, nous allons maintenant tracer leurs fonctions d'ondes grâce à notre script python :

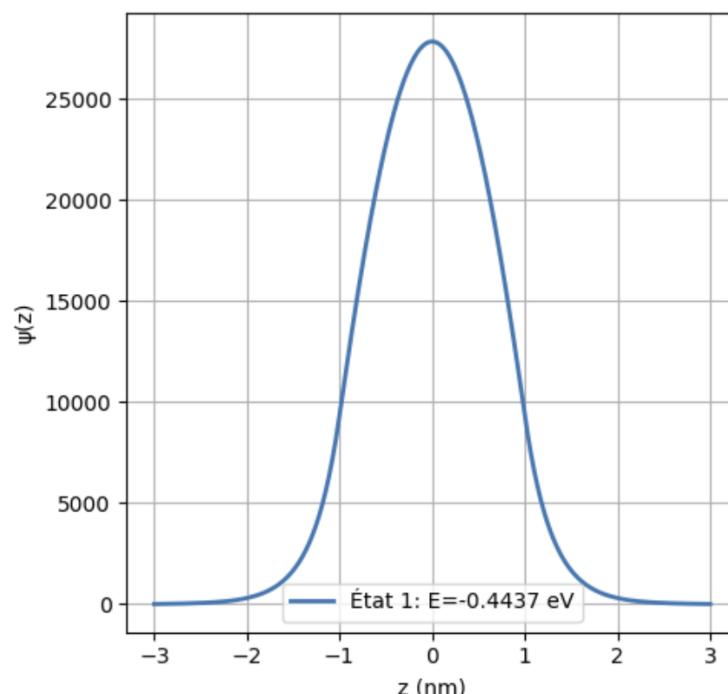


FIGURE 11 – Représentation de l'état confiné n°1

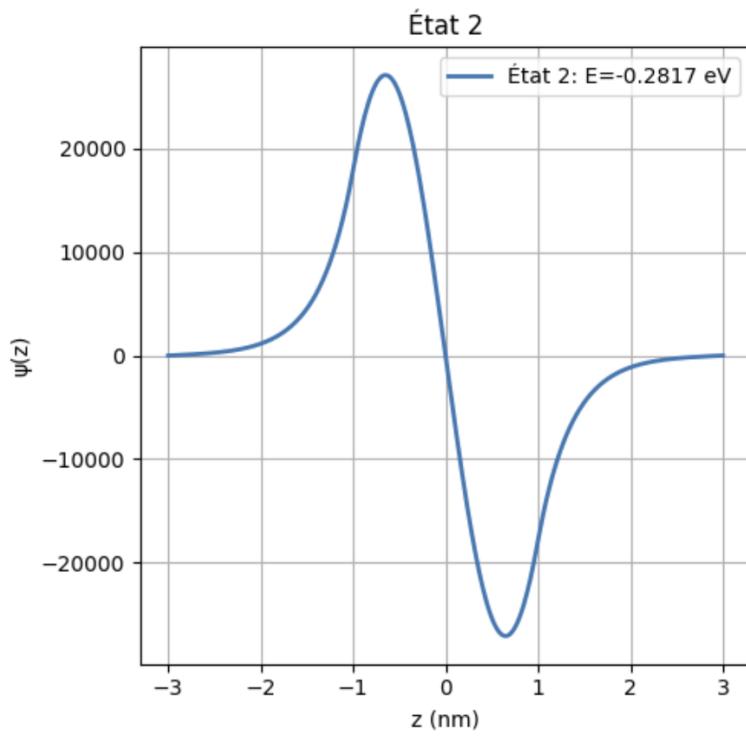


FIGURE 12 – Représentation de l'état confiné n°2

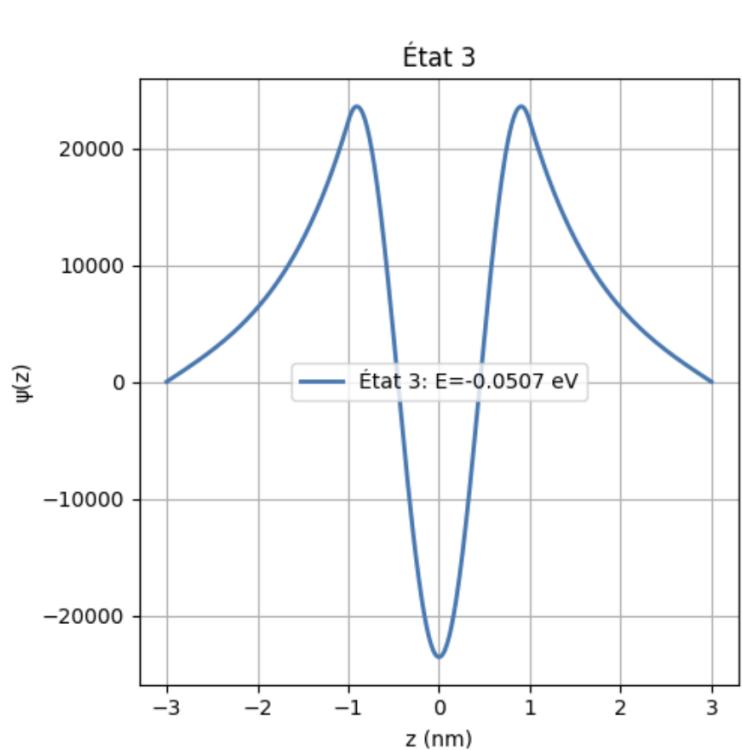


FIGURE 13 – Représentation de l'état confiné n°3

On observe donc les 3 états recherchés par l'énoncé. Nous allons maintenant tracer les niveaux d'énergies des 3 états précédents dans le puits de potentiel en fonction de z et en superposant les différents états sur la figure.

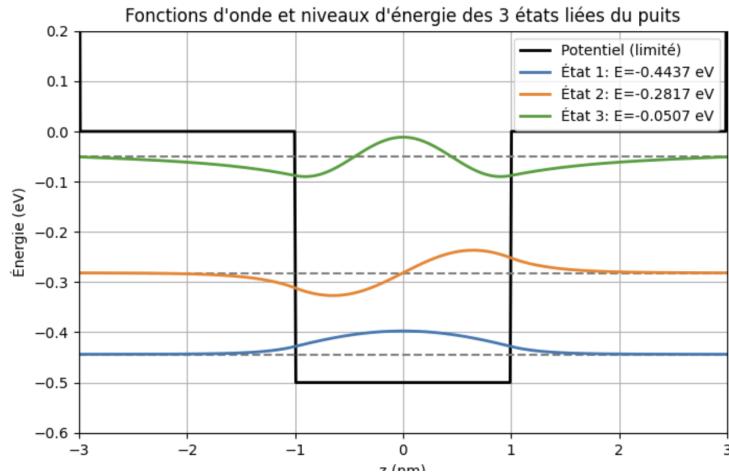


FIGURE 14 – Représentation des niveaux d'énergies des 3 états dans le puits de potentiel avec les fonctions d'ondes superposées

Je n'ai pas eu le temps d'aboutir à un code pour répondre à la question 4 de la partie 2, en effet je comptais finir ce travail durant les vacances mais après un accident de ski qui a engendré une rupture complète de mon ligament croisé, à cause de tous les rendez-vous médicaux que j'ai dû passer je n'ai pas pu réaliser ce travail. Je vous prie de m'excuser pour cela.

5 Conclusion

Ce TP numérique a permis d'implémenter un algorithme pour trouver les états propres de l'oscillateur harmonique et de trouver et décrire les 3 états liés dans un puits de potentiel fini pour les hétérostructures de semi-conducteurs. Il permet aussi de se rendre compte de l'importance de la simulation numérique en physique quantique et des atouts de cette méthode. En effet, la simulation numérique permet, avec une précision très importante, de décrire différentes fonctions d'ondes et énergies sans avoir à poser les calculs et à trouver à la main les sous-espaces propres.

6 Annexes Python

Voici le code python qui a servi à réaliser ce TP numérique :

```

import numpy as np
from math import *
import scipy.optimize as opt
import matplotlib.pyplot as plt
import scipy.integrate as int

#Oscillateur harmonique dans tous ses états (quantiques)

#A) Etude préliminaires
#Question 4 : Signification de R

a =1 # largeur du puits
X = np . linspace ( 0 ,a ,150 ) # creation de l ' espace du puits
R =24

def Pot_Q4(x,R):
    return(((np.pi)*R/2)*(x/a-0.5))**2

Rbis=[[1,'c'],[5,'b'],[10,'k'],[24, 'm']] #Créations des différents R pour le graphique
for i in range ( len ( Rbis ) ) :
    plt . plot ( X , Pot_Q4 ( X , Rbis [ i ][0] ) , color = Rbis [ i ][1] , label = f'R ={ Rbis [ i ][0]}' ) #Affichage des potentiels avec les différents R

plt.legend()
plt.xlabel("x'=x/a")
plt.ylabel("Potentiel réduit dans un puits infini")
plt.show()

N1=6
N2=50
#B) Résolution numérique
#Matrice de l'Hamiltonien diagonalisé pour N= 6
plt.close()

def H_diag(x,n,m,R) : #Fct à intégrer pour trouver les coefficients de la matrice H
    return sin(n*np.pi*x)*sin(m*np.pi*x)*Pot_Q4(x,R)

def H(N,R):
    H = np.zeros (( N , N ))
    H_affichage = np.zeros (( N , N )) # cette matrice servira à l ' affichage
    for n in range ( N ) :
        for m in range ( n +1 ) :
            Int , erreur = int . quad ( lambda x : H_diag ( x ,n +1 ,m +1,R ) , 0,1 ) # intégration de la fonction définie auparavant avec l'état fondamental qui commence à n=1, d'où le +1 en python
            H [n,m ]+=2* Int
            H [m,n ]= H [n,m ] # H est symétrique
            if n == m : # ajout du terme supplémentaire sur la diagonale du au symbole de kronecker
                H[n,m] += (n+1)**2
                H_affichage [n,m ]= H [n,m ]
    return H_affichage #Affichage de la matrice H réduite

print(H(N1,R))
#Question 2

def VP(N,R): #fonction qui donne les énergies propres dans l'ordre croissant et les vecteurs propres associés de l'hamiltonien H
    V= np.linalg.eig(H(N,R))
    vp=V[0] #On extrait les valeurs propres
    Vp=V[1] #On extrait les vecteurs propres
    indices=np.argsort(vp) #On trie les valeurs propres dans le bon ordre
    vp=vp[indices]
    Vp=Vp[:,indices]
    return vp,Vp

#On va tracer les graphes demandés maintenant :
n=np.linspace(1,N2,N2)
plt.plot(n,VP(N2,R)[0],'+r',label=f'Oscillateur harmonique + Puit infini')
plt.plot(n,R*(n-1/2),'+k',label=f'Oscillateur harmonique')
plt.xlabel('n')
plt.ylabel('Energie normalisée')
plt.legend()

print(VP(N2,R)[0])

#Détermination de C
C=np.sum((VP(N2,R)[0]-n**2)[15:])/len(VP(N2,R)[0][15:]) #On fait la moyenne des écarts des valeurs propres n**2 pour des énergies d'ordres supérieures
print(f'\n \n C=(C)')
plt.plot(n,n**2+C+10,'-m',label=f'approximation des énergies en n**2 + C')
plt.xlabel('n')
plt.ylabel('Energie propre normalisée')
plt.legend()
plt.show()

```

```

def comparaison(Rmax):
    Comp = np.zeros(Rmax)
    Pot_moyen = np.zeros(Rmax) # Initiation des matrices
    N = np.linspace(1, N2, N2) # Defi len(obj, /)
    for R in range(1, Rmax + 1):
        V = VP(N2, R)[0] # Return the number of items in a container.
        C = np.sum((V - N**2)[15:]) / len(V[15:])#La valeur 20 a été choisit arbitrairement mais on remarque que en changeant cette valeur les 2 courbes ne change pas beaucoup.

        Comp[R - 1] = C
        Pot_moyen[R - 1] = int.quad(lambda x: Pot_Q4(x, R), 0, 1)[0]
        # Calcul de la moyenne sur l'espace du potentiel réduit

    plt.plot(np.linspace(1, Rmax, Rmax), Comp, '+b', label="Valeur de C")
    plt.plot(np.linspace(1, Rmax, Rmax), Pot_moyen, '+r', label="Moyenne sur l'espace du potentiel réduit")
    plt.legend()
    plt.title("Comparaison de C et de la moyenne sur l'espace du potentiel réduit")
    plt.xlabel("")
    plt.show()

return None

comparaison(30) # Comparaison des valeurs de C et de la moyenne sur l'espace du potentiel réduit pour R allant de 1 à 30

#Fonctions propres

def psi(n,N,R): #cette fonction renvoie la fonction d'onde à l'ordre n, en considérant N modes et R.
    V=VP(N,R)[1]
    X=np.linspace(0,a,150)
    Psi=np.zeros(len(X))
    for i in range (len(X)):
        for j in range (1,N+1):
            Psi[i]+=V[j-1,n]*np.sin(j*np.pi*X[i])*np.sqrt(2)
    return Psi

#Ordre 0:
def psi0 (x) :
    return ((np.pi*R ) /(2*a**2))**(.25)*np.exp(-np.pi**2*R/4*(x/a-1/2) **2)

plt.plot (X ,psi(0,3,R) , '+ g' , label = f'N =3')
plt.plot (X ,psi(0,5,R) , '+ m' , label = f'N =5')
plt.plot (X ,psi(0,8,R) , '+ b' , label = f'N =8')
plt.plot (X ,psi(0,15,R) , '+y' , label = f'N =15')

plt.plot (X,psi0(X) , 'k' , label = " Courbe theorique " )
plt.legend ()
plt.xlabel ("x '= x/a " )
plt.title ( " Fonction propre à l'ordre 0 " )
plt.show ()

#ordre 1
def psi1(x):
    return ((np.pi**5 / (2 * a**2) * R**3)**(1/4)) * ((x/a - 1/2)) * np.exp(- (np.pi**2 / 4) * R * (x/a - 1/2)**2)

plt.plot (X ,psi(1,3,R) , '+ g' , label = f'N =3')
plt.plot (X ,psi(1,5,R) , '+ m' , label = f'N =5')
plt.plot (X ,psi(1,8,R) , '+ b' , label = f'N =8')
plt.plot (X ,psi(1,15,R) , '+y' , label = f'N =15')

plt.plot (X,psi1(X) , 'k' , label = " Courbe theorique " )
plt.legend ()
plt.xlabel ("x '= x/a " )
plt.title ( " Fonction propre à l'ordre 1 " )
plt.show ()

#ordre 2
def psi2(x):
    return ((np.pi / (8 * a**2) * R)**(1/4)) * ((np.pi**2 * R * (x/a - 1/2)**2) - 1) * np.exp(- (np.pi**2 / 4) * R * (x/a - 1/2)**2)

plt.plot (X ,psi(2,3,R) , '+ g' , label = f'N =3')
plt.plot (X ,psi(2,5,R) , '+ m' , label = f'N =5')
plt.plot (X ,psi(2,8,R) , '+ b' , label = f'N =8')
plt.plot (X ,psi(2,15,R) , '+y' , label = f'N =15')

plt.plot (X,psi2(X) , 'k' , label = " Courbe theorique " )
plt.legend ()
plt.xlabel ("x '= x/a " )
plt.title ( " Fonction propre à l'ordre 2 " )
plt.show ()

```

```

#Données du problème

N = 100 #Nombre d'expériences
a = 1e-9
b = 3e-9
V0 = 0.5
me = 9.31e-31 #Approximation faites dans le problème
hbar_SI = 1.054571628e-34 #valeur de Hbarre
eV_to_J = 1.602176634e-19 #Conversion eV en J

# 1)Fonction d'état
def phi(n, z, b):
    return np.sqrt(1.0/b) * np.sin((n * np.pi / (2.0*b)) * (z + b))

# 2)Potentiel du puits
def Vz(z, a, b, V0):
    V = np.full_like(z, 1e9)
    mask_inside_well = np.abs(z) < a
    mask_inside_barrier = (np.abs(z) >= a) & (np.abs(z) < b)
    V[mask_inside_well] = -V0
    V[mask_inside_barrier] = 0.0
    return V

def pot_2(z, n, m):
    return phi(n, z, b) * Vz(z, a, b, V0) * phi(m, z, b)

def pot_matrice_element(n, m):
    val, _ = int.quad(pot_2, -b, b, args=(n, m), epsabs=1e-9, epsrel=1e-9)
    return val

```

```

# 4)Définition des énergies
def energies_p(n):
    T_joule = ((n * hbar_SI * np.pi)**2) / (8.0 * me * (b**2))
    T_eV = T_joule / eV_to_J
    return T_eV

# 5)Calcul de l'hamiltonien
def H2(N):
    H = np.zeros((N, N))
    for n in range(1, N+1):
        for m in range(1, N+1):
            T = energies_p(n) if n == m else 0.0
            V_nm = pot_matrice_element(n, m)
            H[n-1, m-1] = T + V_nm
    return H

#Extrait des valeurs propres et vecteurs propres de l'hamiltonien
H = H2(N)
vals,vecs = np.linalg.eig(H)
#On met les vecteurs propres et valeurs propres dans l'ordre croissant
idx_sort = np.argsort(vals)
vals_c = vals[idx_sort]
vecs_c = vecs[:, idx_sort]

print("Les 15 plus basses valeurs propres (en eV) :")
print(vals_c[:15])

#Expression de l'énergie des 3 états propres du système
bound_idx = np.where(vals_c < 0)[0]
bound_energies = vals_c[bound_idx]
print("\nÉnergies propres < 0 :")
for i, E in enumerate(bound_energies):
    print(f" État {i+1}: E = {E:.6f} eV")
print(f"\nNombre d'états liés (E < 0) = {len(bound_energies)}") #Affichage du nombre d'états liés

```

```

def valeurs_propres(N):
    H = H2(N)
    valeurs, vecteurs = np.linalg.eig(H)
    valeurs = np.round(valeurs, 3)
    return valeurs, vecteurs

#Affichage des valeurs propres et vecteurs propres
valeurs, vecteurs = valeurs_propres(N)
valeurs_norm = np.sort(valeurs)
print(f"\nMatrice H pour N={N} :\n", H2(N))
print(f"\nValeurs propres de la matrice H pour N={N} :\n", valeurs)
print(f"\nVecteurs propres associés :\n", vecteurs)
# 7)On va tracer les fonctions d'ondes des 3 états
x_vals = np.linspace(-b, b, 500)
#Définition de la fonction d'onde
def psi4(c, x):
    psi_val = np.zeros_like(x)
    for n in range(1, N+1):
        psi_val += c[n-1] * phi(n, x, b)
    return psi_val
max_to_plot = min(3, len(bound_energies))
#Tracer des fonctions d'ondes :
plt.figure(figsize=(15, 5))
for i in range(max_to_plot):
    plt.subplot(1, 3, i+1)
    c = vecs[:, bound_idx[i]]
    psi_vals = psi4(c, x_vals)
    norm = np.sqrt(np.trapz(np.abs(psi_vals)**2, x_vals))
    psi_vals /= norm

```

```

if i == 2:
    psi_vals *= -1
plt.plot(x_vals * 1e9, psi_vals, lw=2, label=f"État {i+1}: E={bound_energies[i]:.4f} eV")
plt.xlabel("z (nm)")
plt.ylabel("psi(z)")
plt.title(f"État {i+1}")
plt.grid(True)
plt.legend()
#Cette boucle permet de tracer les différentes fonctions d'ondes.
plt.tight_layout()
plt.show()
#On récupère les valeurs du potentiels
pot_vals = Vz(x_vals, a, b, V0)
#On va limiter l'affichage et tracer les niveaux dans le puits
plt.figure(figsize=(8,5))
# 1)
pot_vals_display = np.minimum(pot_vals, 1.0)
# Tracé du potentiel limité
plt.plot(x_vals*1e9, pot_vals_display, 'k-', lw=2, label="Potentiel (limité)")
# 2) On trace maintenant les niveaux d'énergies dans le puits
max_a_plot = min(3, len(bound_energies))
echelle_facteur = 0.05
for i in range(max_to_plot):
    # Énergie propre de l'état i
    E = bound_energies[i]
    #Les coefficients du vecteurs propres
    c = vecs_c[:, bound_idx[i]]

    plt.hlines(E, -b*1e9, b*1e9, linestyles='--', color='gray')

    if i==1:
        psi_vals = (-1)*psi4(c, x_vals)

    if i==1:
        psi_vals = (-1)*psi4(c, x_vals)
    else:
        psi_vals = psi4(c, x_vals)
    norm = np.sqrt(np.trapz(np.abs(psi_vals)**2, x_vals))
    psi_vals /= norm

    ondes_vals = E + (echelle_facteur * psi_vals)/30e3
    #On trace alors la fonction d'ondes

    plt.plot(x_vals*1e9, ondes_vals, lw=2, label=f"État {i+1}: E={E:.4f} eV")

# 3)
plt.xlabel("z ")
plt.ylabel("Énergie ")
plt.title("Fonctions d'onde et niveaux d'énergie")
plt.grid(True)
plt.legend()
plt.ylim(-0.6, 0.2)
plt.xlim(-b*1e9, b*1e9)
plt.show()

```