

# Oscillateur harmonique à une dimension

Clément Lotteau

May 2020

**Résumé**

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Rappels théoriques sur l'oscillateur harmonique à 1 dimension</b>	<b>3</b>
<b>3</b>	<b>Calcul numérique des fonctions d'onde</b>	<b>4</b>
3.1	Tracer des fonctions d'onde et de leurs modules carrés . . . . .	5
3.2	Méthode Runge-Kutta d'ordre 4 . . . . .	5
3.2.1	Projet de modélisation numérique M1 . . . . .	5
3.2.2	Adaptation à l'oscillateur harmonique . . . . .	11
3.3	Machine Learning . . . . .	16
3.3.1	Les tutos . . . . .	16
3.3.2	Premier exemple : fit d'une droite linéaire, illustration du théorème de représentation universelle . . . . .	16
	<b>Bibliographie</b>	<b>17</b>

## 1 Introduction

## 2 Rappels théoriques sur l'oscillateur harmonique à 1 dimension

On cherche à trouver une solution analytique de l'équation de Schrödinger pour un oscillateur harmonique à une dimension<sup>1</sup>. On rappelle quelques notions. Equation de Schrödinger indépendante du temps :

$$H|\psi\rangle = E|\psi\rangle \quad (1)$$

où  $H$  est l'opérateur hamiltonien, et  $E$  l'énergie de l'état quantique  $|\psi\rangle$ . À une dimension pour un potentiel quelconque et en représentation  $|x\rangle$  :

$$\left[ -\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + V(x) \right] \psi(x) = E\psi(x) \quad \Leftrightarrow \quad \frac{d^2\psi(x)}{dx^2} = \frac{2m(V(x) - E)}{\hbar^2} \psi(x) \quad (2)$$

Pour un oscillateur harmonique :

$$\left[ -\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + \frac{1}{2} m\omega^2 x^2 \right] \psi(x) = E\psi(x) \quad (3)$$

On peut montrer que les valeurs de l'énergie sont discrètes et s'écrivent :  $E_n = (n + \frac{1}{2})\hbar\omega$  où  $n$  est un entier positif ou nul. On introduit aussi les opérateurs d'échelle :

$$a = \frac{1}{\sqrt{2}}(\hat{X} + i\hat{P}) \quad \text{et} \quad a^\dagger = \frac{1}{\sqrt{2}}(\hat{X} - i\hat{P}) \quad \text{avec} \quad \begin{cases} \hat{X} = \sqrt{\frac{m\omega}{\hbar}} X \\ \hat{P} = \sqrt{\frac{1}{m\hbar\omega}} P \end{cases} \quad (4)$$

$a$  et  $a^\dagger$  sont respectivement les opérateurs d'annihilation et de création. Ils permettent de retirer et d'ajouter la quantité  $\hbar\omega$  à l'énergie  $E_n$ . L'état fondamental du système a une énergie  $E_0 = \frac{1}{2}\hbar\omega$ . L'action de  $a^\dagger$  sur l'état quantique fondamental  $|\psi_0\rangle$  a pour effet de passer de l'énergie  $E_0$  à l'énergie  $E_1 = \frac{3}{2}\hbar\omega$ . L'énergie ne pouvant pas être inférieure à celle de l'état fondamental, l'action de  $a$  sur  $|\psi_0\rangle$  est nulle :  $a|\psi_0\rangle = 0$ . Cette conséquence nous permet de déduire la fonction d'onde de l'état fondamental :

$$a|\psi_0\rangle = 0 \quad \Rightarrow \quad \frac{1}{\sqrt{2}} \left( \sqrt{\frac{m\omega}{\hbar}} X + \frac{i}{\sqrt{m\hbar\omega}} P \right) |\psi_0\rangle = 0 \quad (5)$$

Ainsi :

$$\left( \frac{m\omega}{\hbar} x + \frac{d}{dx} \right) \psi_0(x) = 0 \quad (6)$$

Cette équation a pour solution :  $\psi_0(x) = c^{te} e^{-\frac{m\omega x^2}{2\hbar}}$  où  $c^{te}$  est une constante de normalisation. L'action de  $a^\dagger$  sur cette fonction d'onde nous permet d'obtenir

---

1. [1]

celles des états d'énergie supérieure. En répétant l'opération  $n$  fois, on obtient la fonction d'onde de l'état d'énergie  $E_n$  :

$$\psi_n(x) = \left[ \frac{1}{2^n n!} \left( \frac{\hbar}{m\omega} \right)^n \right]^{1/2} \left( \frac{m\omega}{\pi \hbar} \right)^{1/4} \left[ \frac{m\omega}{\hbar} x - \frac{d}{dx} \right]^n e^{-\frac{m\omega x^2}{2\hbar}} \quad (7)$$

Ce sont des polynômes d'Hermite. Les fonctions d'onde  $\phi_0$ ,  $\phi_1$ ,  $\phi_2$  et  $\phi_3$  sont :

$$\begin{aligned} \psi_0(x) &= \left( \frac{m\omega}{\pi \hbar} \right)^{\frac{1}{4}} e^{-\frac{m\omega x^2}{2\hbar}} \\ \psi_1(x) &= \left[ \frac{4}{\pi} \left( \frac{m\omega}{\hbar} \right)^3 \right]^{\frac{1}{4}} x e^{-\frac{m\omega x^2}{2\hbar}} \\ \psi_2(x) &= \left( \frac{m\omega}{4\pi \hbar} \right)^{\frac{1}{4}} \left[ 2 \frac{m\omega}{\hbar} x^2 - 1 \right] e^{-\frac{m\omega x^2}{2\hbar}} \\ \psi_3(x) &= \frac{1}{\sqrt{3}} \left[ \frac{1}{\pi} \left( \frac{m\omega}{\hbar} \right)^3 \right]^{\frac{1}{4}} \left[ 2 \frac{m\omega}{\hbar} x^3 - 3x \right] e^{-\frac{m\omega x^2}{2\hbar}} \end{aligned} \quad (8)$$

### 3 Calcul numérique des fonctions d'onde

Dans cette section, on s'intéresse au tracer de ces quatre fonctions d'onde ainsi que de leur module carré. Le programme utilisé pour gérer les données est codé en C++.  $m$ ,  $\omega$  et  $\hbar$  ont tous été choisis égaux à 1 et le calcul est effectué entre  $x = -5$  et  $x = +5$  avec un pas de 0,02. Le code calcule dans un premier temps  $\psi_0$  puis écrit la position, l'amplitude de la fonction d'onde et de son module carré dans un fichier. Il recommence ensuite l'opération avec  $\psi_1$ ,  $\psi_2$  et  $\psi_3$  pour un total de quatre fichiers. Les données générées sont ensuite traitées dans un programme écrit en Python consistant à tracer les fonctions d'onde et leur modules carré grâce à l'outil Matplotlib.

### 3.1 Tracer des fonctions d'onde et de leurs modules carrés

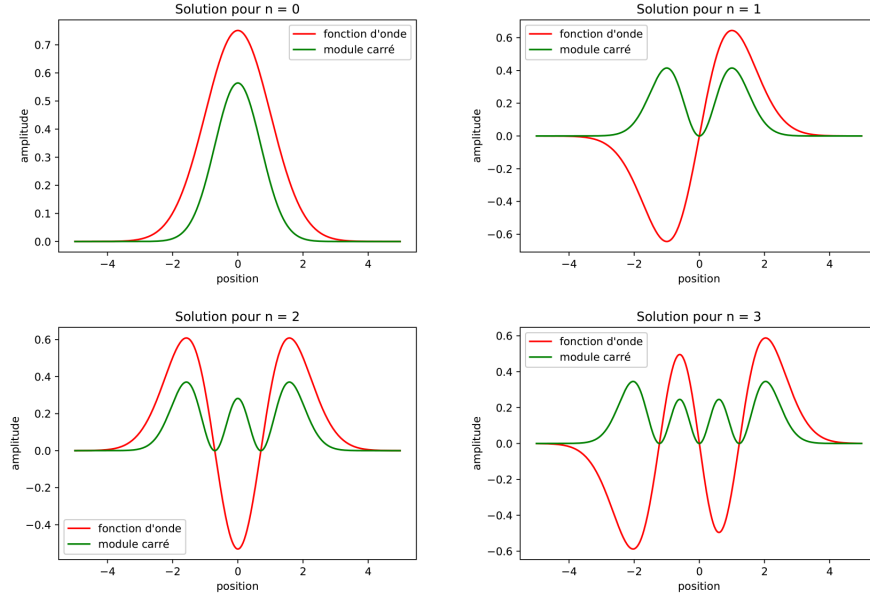


FIGURE 1 – Fonction d'onde (rouge) et module carré (vert). On observe que la parité de la fonction d'onde correspond à la parité de  $n$ .

Ici ajouter que  $\psi_n$  a  $n$  0 (p.551). Faire le lien entre valeurs notables de  $\psi_n$  et énergie (p.520)  $\rightarrow E_p = E_c$  croissent avec  $n$  (p.523)

### 3.2 Méthode Runge-Kutta d'ordre 4

La méthode RK4 utilisée dans ce projet est une adaptation de la méthode que j'ai créé pour un projet de modélisation numérique en M1.

#### 3.2.1 Projet de modélisation numérique M1

Dans cette section, j'explique comment passer de l'équation différentielle à la résolution numérique, puis je décris le programme en détails avec des logigrammes.

#### Passage de l'équation différentielle à la résolution numérique :

Reprenons l'équation (2) :

$$\frac{d^2\psi(x)}{dx^2} = \frac{2m(V(x) - E)}{\hbar^2}\psi(x) \quad (9)$$

et on pose  $k$  le vecteur d'onde :

$$k = \sqrt{\frac{2m(V(x) - E)}{\hbar^2}} \quad (10)$$

La méthode Runge-Kutta ne permet pas de résoudre d'équation différentielle d'ordre 2. Néanmoins, après quelques manipulation, il est possible de la transformer en deux équations différentielles couplées d'ordre 1. On commence par poser :

$$\phi(x) = \frac{d\psi(x)}{dx} \quad (11)$$

Ce qui nous donne :

$$\phi'(x) - k^2\psi(x) = 0 \quad (12)$$

On a ainsi deux équations différentielles couplées d'ordre 1 :

$$\begin{aligned} \psi'(x) &= \phi(x) \\ \phi'(x) &= k^2\psi(x) \end{aligned} \quad (13)$$

La méthode Runge-Kutta permet de transformer une équation différentielle en une suite de terme de la manière suivante :

$$\psi_{n+1} = \psi_n + hp \quad (14)$$

$$x_{n+1} = x_n + h \quad (15)$$

où  $h$  est l'incrément de la variable  $x$ , et où  $p$  est la pente  $\psi'_n$  de  $\psi_n$  au point  $x_n$ . Pour la méthode d'ordre 4 utilisée dans le programme :

$$\psi_{n+1} = \psi_n + \frac{h}{6}(p_1 + 2p_2 + 2p_3 + p_4) + o(h^5) \quad (16)$$

où  $p_1, p_2, p_3, p_4$  sont les pentes de  $\psi_n$  calculées en différents points. Pour calculer ces pentes, on procède de la manière suivante :

- On calcule la pente  $p_1$  au point  $x_n$
- On calcule la pente  $p_2$  au point  $x + \frac{h}{2}$  en utilisant une nouvelle valeur de  $\psi_n$  obtenue grâce à  $p_1$
- On calcule la pente  $p_3$  au point  $x + \frac{h}{2}$  une fois encore mais en utilisant cette fois une valeur de  $\psi_n$  obtenue grâce à  $p_2$
- On calcule la pente  $p_4$  au point  $x + h$  en utilisant une valeur de  $\psi_n$  obtenue grâce à  $p_3$

Mathématiquement :

$$\begin{aligned} p_1 &= \psi'_n(x_n, \psi_n) \\ p_2 &= \psi'_n(x_n + \frac{h}{2}, \psi_n + \frac{h}{2}p_1) \\ p_3 &= \psi'_n(x_n + \frac{h}{2}, \psi_n + \frac{h}{2}p_2) \\ p_4 &= \psi'_n(x_n + h, \psi_n + hp_3) \end{aligned} \quad (17)$$

Pour calculer une valeur définitive pour  $\psi_{n+1}$ , on voit dans l'équation (16) qu'on fait une moyenne pondérée des pentes en accordant plus de poids à celles calculées en  $\frac{h}{2}$ .

Dans notre cas, nous avons affaire à deux équations différentielles couplées et il faut ainsi les résoudre en même temps. On a donc :

$$\begin{aligned}\psi_{n+1} &= \psi_n + \frac{h}{6}(p_1 + 2p_2 + 2p_3 + p_4) + o(h^5) \\ \phi_{n+1} &= \phi_n + \frac{h}{6}(t_1 + 2t_2 + 2t_3 + t_4) + o(h^5)\end{aligned}\tag{18}$$

Et les pentes dépendent les unes des autres. Combiner les équations (13) et (17) permet d'obtenir :

$$\begin{aligned}p_1 &= \phi_n \\ t_1 &= k^2 \psi_n \\ p_2 &= \phi_n + \frac{h}{2} t_1 \\ t_2 &= k^2 \left( \psi_n + \frac{h}{2} p_1 \right) \\ p_3 &= \phi_n + \frac{h}{2} t_2 \\ t_3 &= k^2 \left( \psi_n + \frac{h}{2} p_2 \right) \\ p_4 &= \phi_n + h t_3 \\ t_4 &= k^2 (\psi_n + h p_3)\end{aligned}\tag{19}$$

Ces pentes permettent alors d'effectuer le calcul (18) pour obtenir les valeurs de  $\psi_{n+1}$  et  $\phi_{n+1}$  au point  $x_{n+1}$ . On calcule à nouveaux les pentes au point  $x_{n+1}$  en remplaçant  $\psi_n$  et  $\phi_n$  par  $\psi_{n+1}$  et  $\phi_{n+1}$  obtenues précédemment et on avance au point  $x_{n+2}$ ...etc. On note grâce à l'équation (10) que le vecteur d'onde, et donc le potentiel, doivent être recalculés à chaque itération en changeant éventuellement le signe de  $k$  selon si  $V(x) - E$  est positif ( $k$  réel) ou négatif ( $k$  imaginaire).

#### **Le programme en détail :**

Le but de ce projet était de créer un programme permettant de calculer les coefficients de transmission et de réflexion d'une particule à travers une barrière de potentiel. On propage de  $+\infty$  vers  $-\infty$  et on calcule les parties réelles et imaginaires de  $\psi$  et  $\phi$  indépendamment.

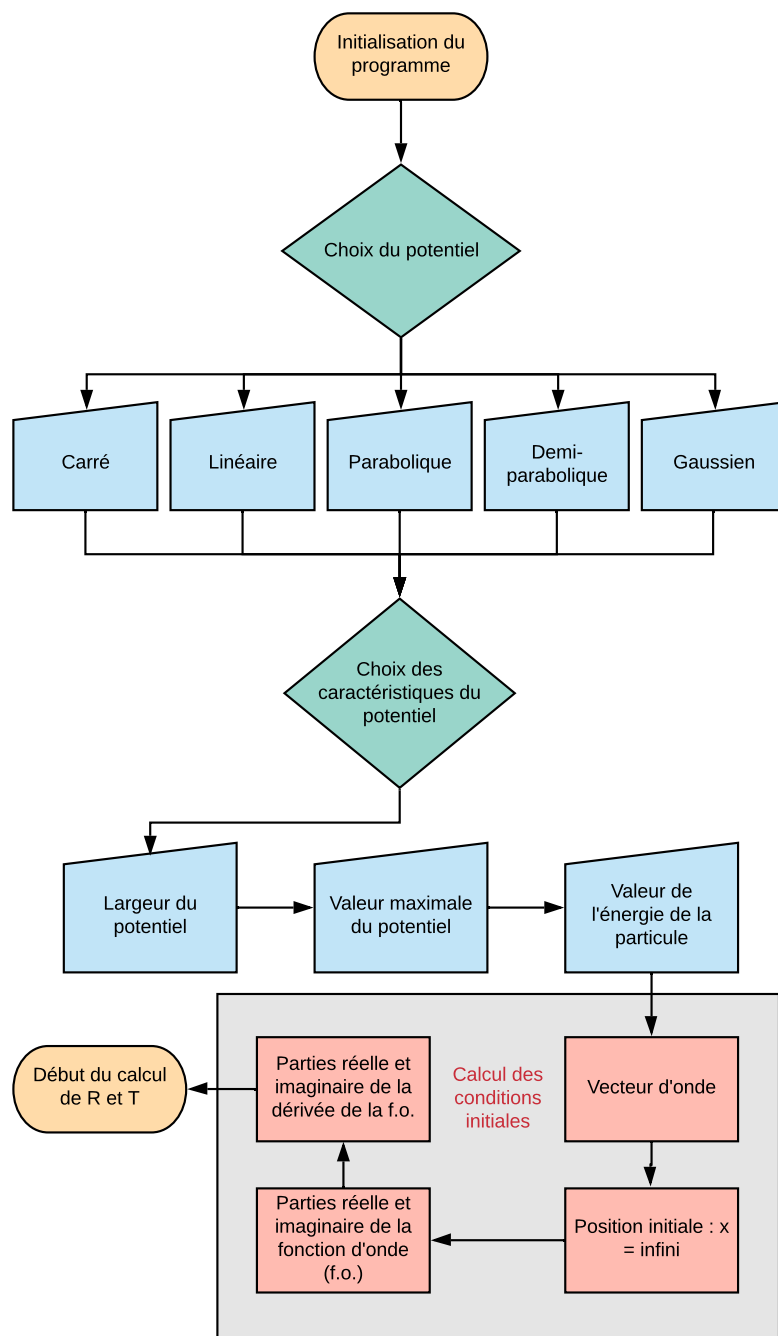


FIGURE 2 – Logigramme de l'initialisation du programme



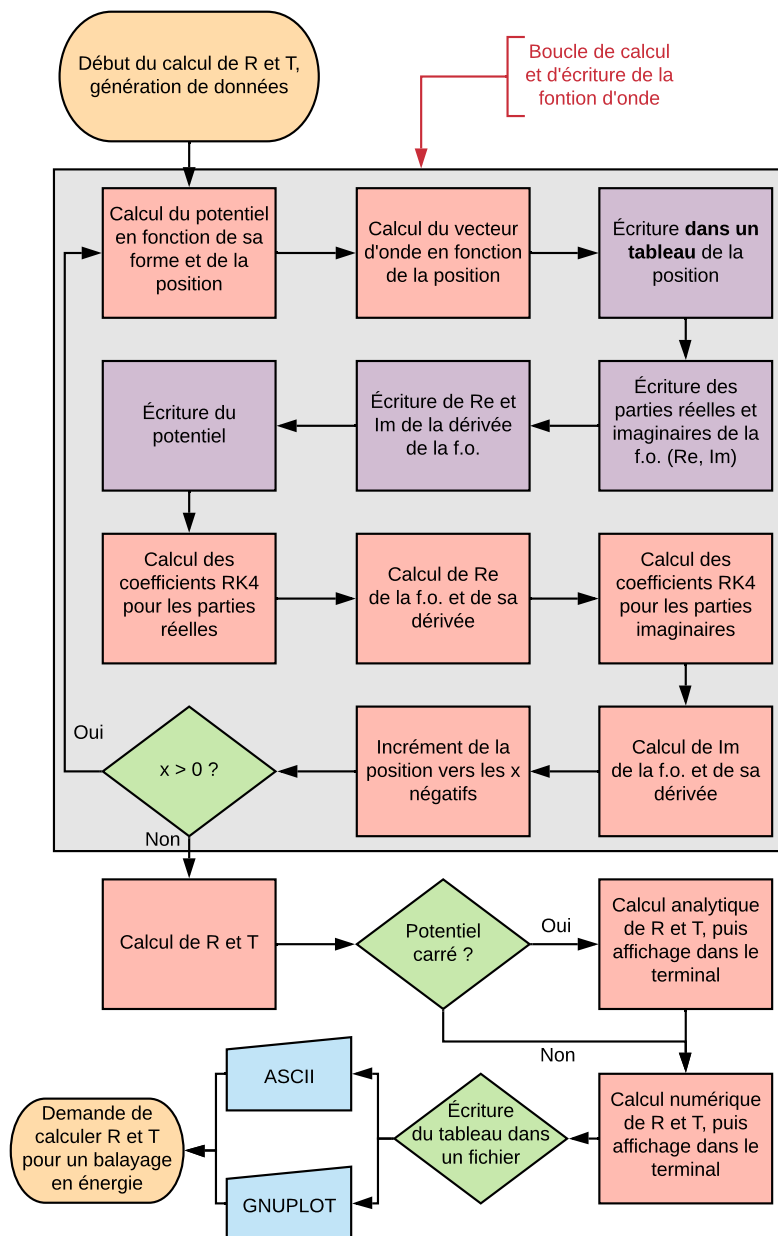


FIGURE 3 – Logigramme du calcul de R et T par le programme

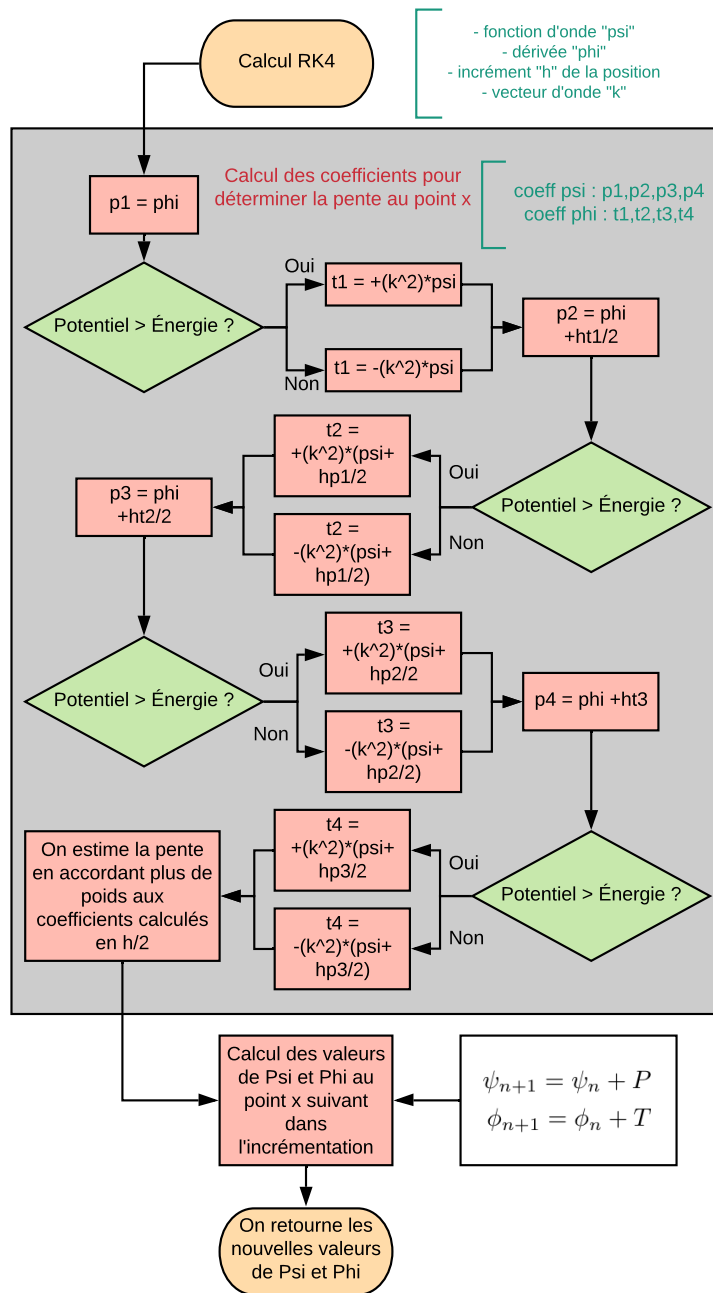


FIGURE 4 – Logigramme de la méthode RK4 utilisée pour calculer la fonction d'onde.

Résultats : ?

### 3.2.2 Adaptation à l'oscillateur harmonique

Je me propose dans un premier temps d'adapter mon programme à un puit de potentiel carré, puis à un potentiel harmonique.

**Puit carré :**

AJOUTER UNE EXPLICATION SUR LES ÉTATS LIÉS + CONDITIONS INITIALES + CONDITION AUX LIMITES DE LA F.O. AJOUTER DES LOGIGRAMMES DU PROGRAMME

Dans un premier temps, j'ai inversé la forme du potentiel carré en passant d'une marche à un puit. Pour vérifier que la propagation fonctionne, j'ai effectué un balayage en énergie de 0 jusqu'au haut du puit et j'ai cherché les explosions minimales en  $x = 0$  de la fonction d'onde (c'est à dire pour quelles énergies  $|\psi(0)|$  est minimale). Les figures et tableaux suivants illustrent ma démarche, le puit choisi est profond de 3 eV et de largeur 2, on commence par effectuer un premier balayage de 0 à 3 eV.

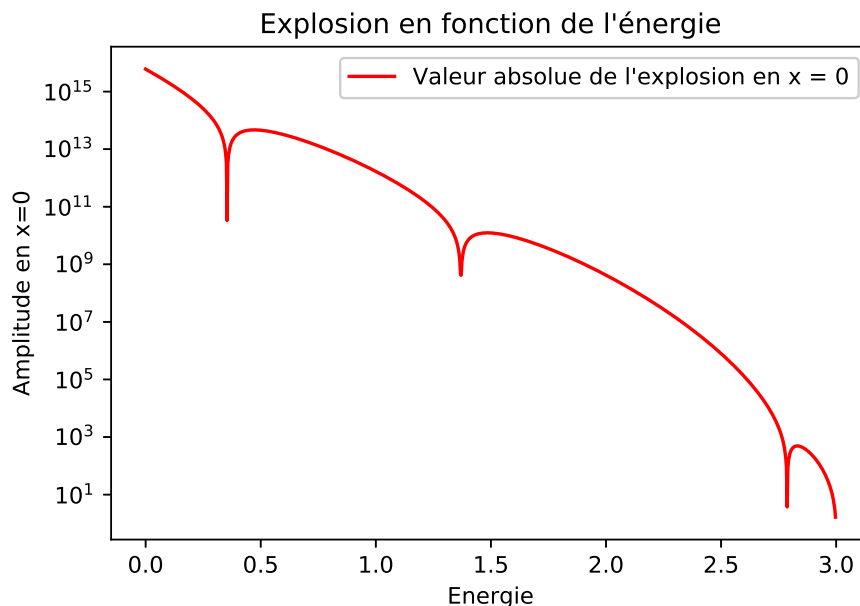


FIGURE 5 – On observe de brusques diminutions de l'explosion autour de certaines valeurs de l'énergie. ICI J'AI SUPPOSÉ QU'IL S'AGISSAIT D'ÉNERGIES PROPRES.

Energie (eV)	$\psi(0)$
1.362	$2.28 \times 10^9$
1.365	$1.34 \times 10^9$
1.368	$4.40 \times 10^8$
1.371	$-4.16 \times 10^8$
1.374	$-1.23 \times 10^9$
1,377	$-2.00 \times 10^9$
1,38	$-2.74 \times 10^9$
1,383	$-3.44 \times 10^9$

On voit sur ce tableau que, lorsque l'énergie passe de 1.368 eV à 1.371 eV, la valeur de  $\psi(0)$  change brusquement de signe. L'explosion de la fonction d'onde est donc passée par 0 entre ces deux énergies.

Energie (eV)	$\psi(0)$
2.781	-188.383
2.784	-86.7335
2.787	3.72786
2.793	154.463

Ici aussi, entre 2.784 eV et 2.787 eV. Essayons de vérifier qu'il s'agit bien d'une énergie propre. Pour cela, nous allons affiner la valeur de l'énergie pour laquelle l'explosion change de signe.

Energie (eV)	$\psi(0)$
2.78686	-0.114703
2.78687	-0.0290847
2.78687	0.0565229
2.78687	0.14212

Essayons maintenant de tracer la fonction d'onde  $\psi(x)$  pour une énergie de 2.78687 eV.

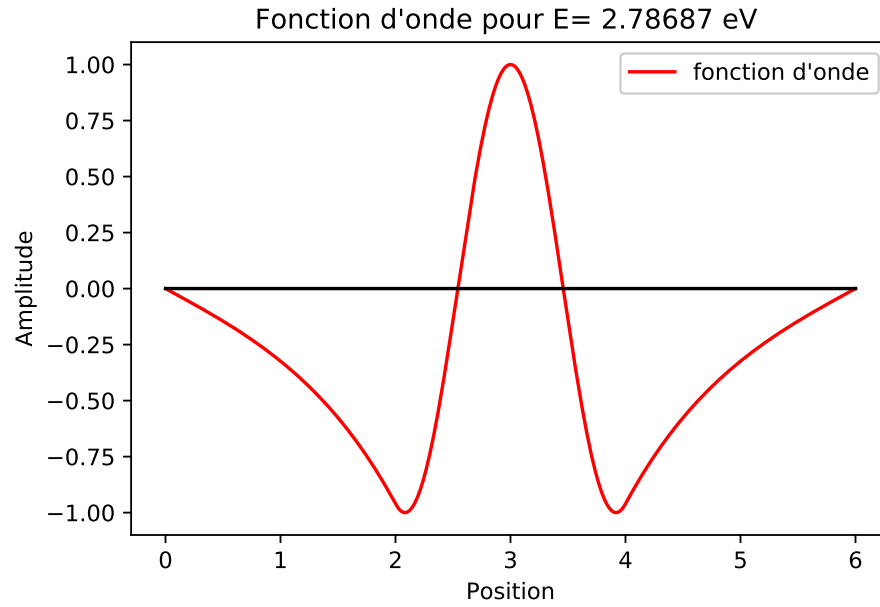


FIGURE 6 – Tracer de la fonction d'onde pour  $E = 2.78687 \text{ eV}$ . Il s'agit du mode propre associé à la valeur propre  $E_3$ .

On fait pareil pour d'autres valeurs propres SUPPOSÉES de l'énergie :  $E_1$  et  $E_2$ .

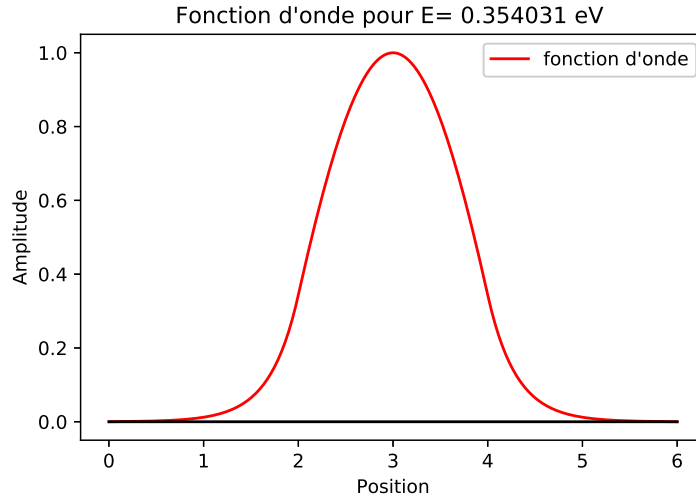


FIGURE 7 – Tracer de la fonction d'onde pour  $E = 0.354031 \text{ eV}$ . Il s'agit du mode propre associé à la valeur propre  $E_1$ .

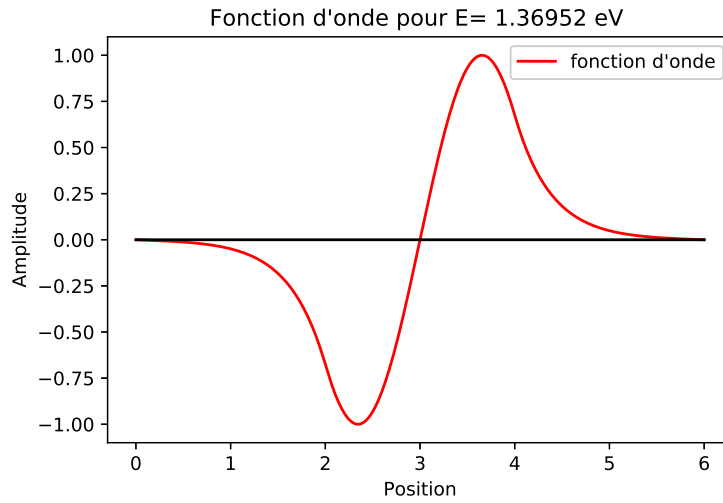


FIGURE 8 – Tracer de la fonction d'onde pour  $E = 1.36952 \text{ eV}$ . Il s'agit du mode propre associé à la valeur propre  $E_2$ .

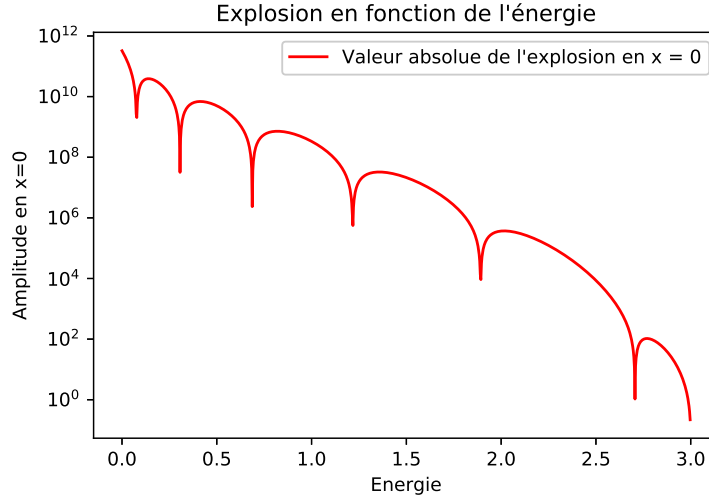


FIGURE 9 – Explosion en fonction de l'énergie pour un potentiel profond de 3 eV et de largeur 5. On observe plus de chutes vers 0 et elles semblent de plus en plus espacées.

MAINTENANT AVEC DES ÉNERGIES ET UN POTENTIEL NÉGATIFS.

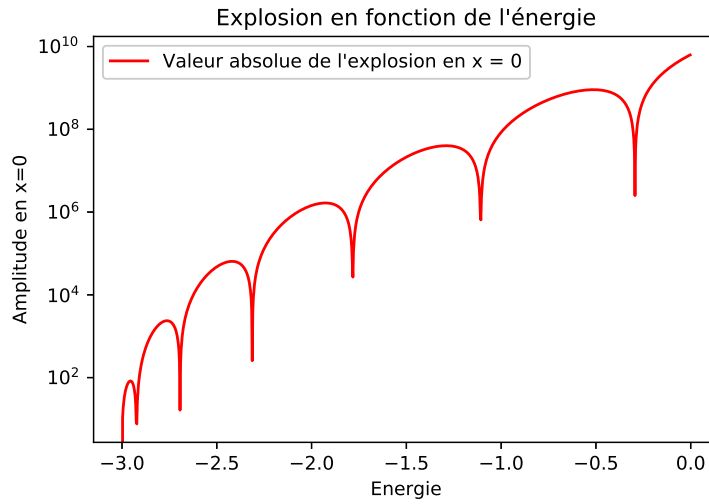


FIGURE 10 – Explosion en fonction de l'énergie pour un potentiel profond de 3 eV ( $V = -3$  eV) et de largeur 5.

Potentiel harmonique :

### **3.3 Machine Learning**

#### **3.3.1 Les tutos**

#### **3.3.2 Premier exemple : fit d'une droite linéaire, illustration du théorème de représentation universelle**

Mémo : [3] [9] [7] [4] [6] [5] [2] [8]



## Bibliographie

- [1] Claude COHEN-TANNOUDJI, Bernard DIU et Franck LALOË. *Mécanique quantique Tome 1*. EDP Sciences, 2018, p. 371-378. ISBN : 9782759822874.
- [2] Colin BERNET. *Handwritten Digit Recognition with scikit-learn*. URL : <https://thedatafrog.com/en/articles/handwritten-digit-recognition-scikit-learn/>.
- [3] Colin BERNET. *Le réseau à un neurone : régression logistique*. URL : <https://thedatafrog.com/fr/articles/logistic-regression/>.
- [4] Colin BERNET. *Le surentraînement*. URL : <https://thedatafrog.com/fr/articles/overfitting-illustrated/>.
- [5] Colin BERNET. *Matplotlib for Machine Learning*. URL : <https://thedatafrog.com/en/articles/matplotlib-machine-learning/>.
- [6] Colin BERNET. *Numpy Crash Course for Machine Learning*. URL : <https://thedatafrog.com/en/articles/numpy-crash-course-machine-learning/>.
- [7] Colin BERNET. *Premier réseau de neurones avec keras*. URL : <https://thedatafrog.com/fr/articles/first-neural-network-keras/>.
- [8] Colin BERNET. *Python Crash Course for Machine Learning*. URL : <https://thedatafrog.com/en/articles/python-crash-course-machine-learning/>.
- [9] Colin BERNET. *Régression Logistique vs Réseau de Neurones : Non Linéarités*. URL : <https://thedatafrog.com/fr/articles/logistic-regression-neural-network/>.