

Oscillateur harmonique à une dimension

Clément Lotteau

May 2020

Résumé

Table des matières

1	Introduction	3
2	L'oscillateur harmonique à 1 dimension	3
2.1	Rappels théoriques	3
2.2	Calcul numérique des polynômes d'Hermite	5
3	Méthode Runge-Kutta d'ordre 4 - Étude d'un puits carré et d'un potentiel harmoniques - États liés, résonance	6
3.1	Projet de modélisation numérique M1 - Résolution RK4	6
3.1.1	Passage de l'équation différentielle à la résolution numérique	6
3.1.2	Le programme de M1 en détails	8
3.2	Adaptation à un puits carré et à un potentiel harmonique	12
3.2.1	Le nouveau programme en détails	12
3.2.2	États liés du puits carré	14
3.2.3	États liés du potentiel harmonique	18
3.2.4	États non-liés, phénomènes de résonance	22
4	Réseaux de neurones	26
4.1	Explications théoriques sur les réseaux de neurones	26
4.1.1	Fonctions d'activation	27
4.1.2	Gradient de la fonction d'activation – réseaux à plusieurs couches	30
4.1.3	Précision du réseau	30
4.2	Approximation de fonctions d'onde par réseaux de neurones	31
4.2.1	Théorème d'approximation universelle	31
4.2.2	fit d'une gaussienne, variations du nombre d'outputs	32
4.2.3	fit d'une gaussienne, variation du batch	35
4.2.4	fit d'une gaussienne, variation des epochs	37
4.2.5	fit d'une gaussienne, variation batch vs epochs	39
4.2.6	Fit des polynômes d'hermite - variation du nombre de points et d'outputs (annexes)	41
	Bibliographie	42
A	Approximation des pol. d'Hermite par réseau de neurones (500 points)	43
B	Approximation des modules carrés des pol. d'Hermite par réseau de neurones (500 points)	47
C	Approximation des pol. d'Hermite par réseau de neurones (10001 pts)	51
D	Approximation des modules carrés des pol. d'Hermite par réseau de neurones (10001 pts)	55

1 Introduction

2 L'oscillateur harmonique à 1 dimension

2.1 Rappels théoriques

On cherche à trouver une solution analytique de l'équation de Schrödinger pour un oscillateur harmonique à une dimension¹. 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 :

$$E_n = (n + \frac{1}{2})\hbar\omega \quad (4)$$

où n est un entier positif ou nul. On introduit aussi les opérateurs d'échelle :

$$a = \frac{1}{\sqrt{2}}(\hat{X} + i\hat{P}) \quad \text{et} \quad a^\dagger = \frac{1}{\sqrt{2}}(\hat{X} - i\hat{P}) \quad \text{avec} \quad \begin{cases} \hat{X} = \sqrt{\frac{m\omega}{\hbar}} X \\ \hat{P} = \sqrt{\frac{1}{m\hbar\omega}} P \end{cases} \quad (5)$$

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 :

$$a|\psi_0\rangle = 0 \quad \Rightarrow \quad \frac{1}{\sqrt{2}} \left(\sqrt{\frac{m\omega}{\hbar}} X + \frac{i}{\sqrt{m\hbar\omega}} P \right) |\psi_0\rangle = 0 \quad (6)$$

Ainsi :

$$\left(\frac{m\omega}{\hbar} x + \frac{d}{dx} \right) \psi_0(x) = 0 \quad (7)$$

1. [3]

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 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 :

$$\psi_n(x) = \left[\frac{1}{2^n n!} \left(\frac{\hbar}{m\omega} \right)^n \right]^{1/2} \left(\frac{m\omega}{\pi\hbar} \right)^{1/4} \left[\frac{m\omega}{\hbar} x - \frac{d}{dx} \right]^n e^{-\frac{m\omega x^2}{2\hbar}} \quad (8)$$

Ce sont des polynômes d'Hermite. Les fonctions d'onde ϕ_0 , ϕ_1 , ϕ_2 et ϕ_3 sont :

$$\begin{aligned} \psi_0(x) &= \left(\frac{m\omega}{\pi\hbar} \right)^{\frac{1}{4}} e^{-\frac{m\omega x^2}{2\hbar}} \\ \psi_1(x) &= \left[\frac{4}{\pi} \left(\frac{m\omega}{\hbar} \right)^3 \right]^{\frac{1}{4}} x e^{-\frac{m\omega x^2}{2\hbar}} \\ \psi_2(x) &= \left(\frac{m\omega}{4\pi\hbar} \right)^{\frac{1}{4}} \left[2 \frac{m\omega}{\hbar} x^2 - 1 \right] e^{-\frac{m\omega x^2}{2\hbar}} \\ \psi_3(x) &= \frac{1}{\sqrt{3}} \left[\frac{1}{\pi} \left(\frac{m\omega}{\hbar} \right)^3 \right]^{\frac{1}{4}} \left[2 \frac{m\omega}{\hbar} x^3 - 3x \right] e^{-\frac{m\omega x^2}{2\hbar}} \end{aligned} \quad (9)$$

2.2 Calcul numérique des polynômes d'Hermite

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 , ω 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 ψ_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 ψ_1 , ψ_2 et ψ_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.

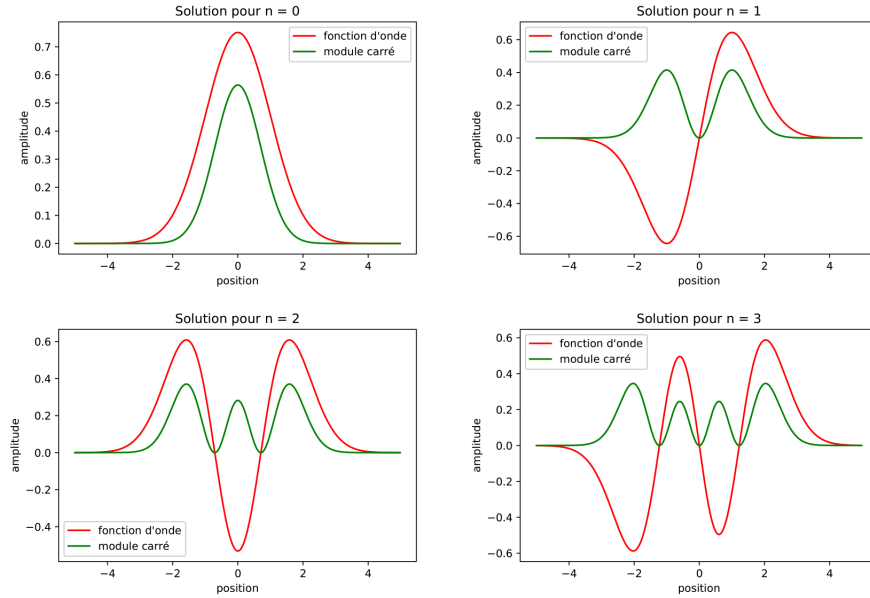


FIGURE 1 – Fonction d'onde (rouge) et module carré (vert). On observe que la parité de la fonction d'onde correspond à la parité de n .

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

3 Méthode Runge-Kutta d'ordre 4 - Étude d'un puits carré et d'un potentiel harmoniques - États liés, résonance

3.1 Projet de modélisation numérique M1 - Résolution RK4

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. 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.

3.1.1 Passage de l'équation différentielle à la résolution numérique

Reprenons l'équation (2) :

$$\frac{d^2\psi(x)}{dx^2} = \frac{2m(V(x) - E)}{\hbar^2}\psi(x) \quad (10)$$

et on pose k le vecteur d'onde :

$$k = \sqrt{\frac{2m(V(x) - E)}{\hbar^2}} \quad (11)$$

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

$$\phi(x) = \frac{d\psi(x)}{dx} \quad (12)$$

Ce qui nous donne :

$$\phi'(x) - k^2\psi(x) = 0 \quad (13)$$

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

$$\begin{aligned} \psi'(x) &= \phi(x) \\ \phi'(x) &= k^2\psi(x) \end{aligned} \quad (14)$$

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

$$\psi_{n+1} = \psi_n + hp \quad (15)$$

$$x_{n+1} = x_n + h \quad (16)$$

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

$$\psi_{n+1} = \psi_n + \frac{h}{6}(p_1 + 2p_2 + 2p_3 + p_4) + o(h^5) \quad (17)$$

où p_1, p_2, p_3, p_4 sont les pentes de ψ_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 ψ_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 ψ_n obtenue grâce à p_2
- On calcule la pente p_4 au point $x+h$ en utilisant une valeur de ψ_n obtenue grâce à p_3

Mathématiquement :

$$\begin{aligned} p_1 &= \psi'_n(x_n, \psi_n) \\ p_2 &= \psi'_n\left(x_n + \frac{h}{2}, \psi_n + \frac{h}{2}p_1\right) \\ p_3 &= \psi'_n\left(x_n + \frac{h}{2}, \psi_n + \frac{h}{2}p_2\right) \\ p_4 &= \psi'_n(x_n + h, \psi_n + hp_3) \end{aligned} \quad (18)$$

Pour calculer une valeur définitive pour ψ_{n+1} , on voit dans l'équation (17) 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 :

$$\begin{aligned} \psi_{n+1} &= \psi_n + \frac{h}{6}(p_1 + 2p_2 + 2p_3 + p_4) + o(h^5) \\ \phi_{n+1} &= \phi_n + \frac{h}{6}(t_1 + 2t_2 + 2t_3 + t_4) + o(h^5) \end{aligned} \quad (19)$$

Et les pentes dépendent les unes des autres. Combiner les équations (14) et (18)

permet d'obtenir :

$$\begin{aligned}
p_1 &= \phi_n \\
t_1 &= k^2 \psi_n \\
p_2 &= \phi_n + \frac{h}{2} t_1 \\
t_2 &= k^2 (\psi_n + \frac{h}{2} p_1) \\
p_3 &= \phi_n + \frac{h}{2} t_2 \\
t_3 &= k^2 (\psi_n + \frac{h}{2} p_2) \\
p_4 &= \phi_n + h t_3 \\
t_4 &= k^2 (\psi_n + h p_3)
\end{aligned} \tag{20}$$

Ces pentes permettent alors d'effectuer le calcul (19) pour obtenir les valeurs de ψ_{n+1} et ϕ_{n+1} au point x_{n+1} . On calcule à nouveaux les pentes au point x_{n+1} en remplaçant ψ_n et ϕ_n par ψ_{n+1} et ϕ_{n+1} obtenues précédemment et on avance au point x_{n+2} ...etc. On note grâce à l'équation (11) 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).

3.1.2 Le programme de M1 en détails

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 ψ et ϕ indépendamment.

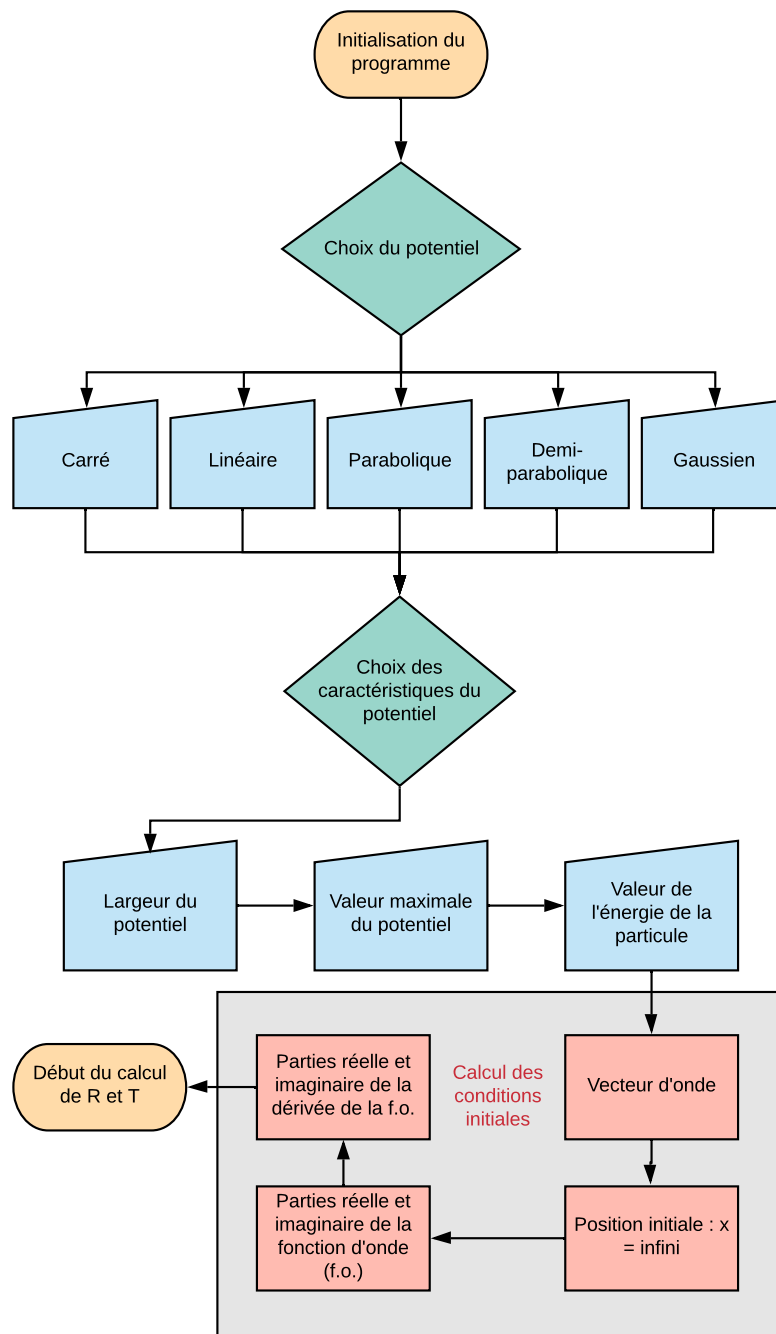


FIGURE 2 – Logigramme de l'initialisation du programme

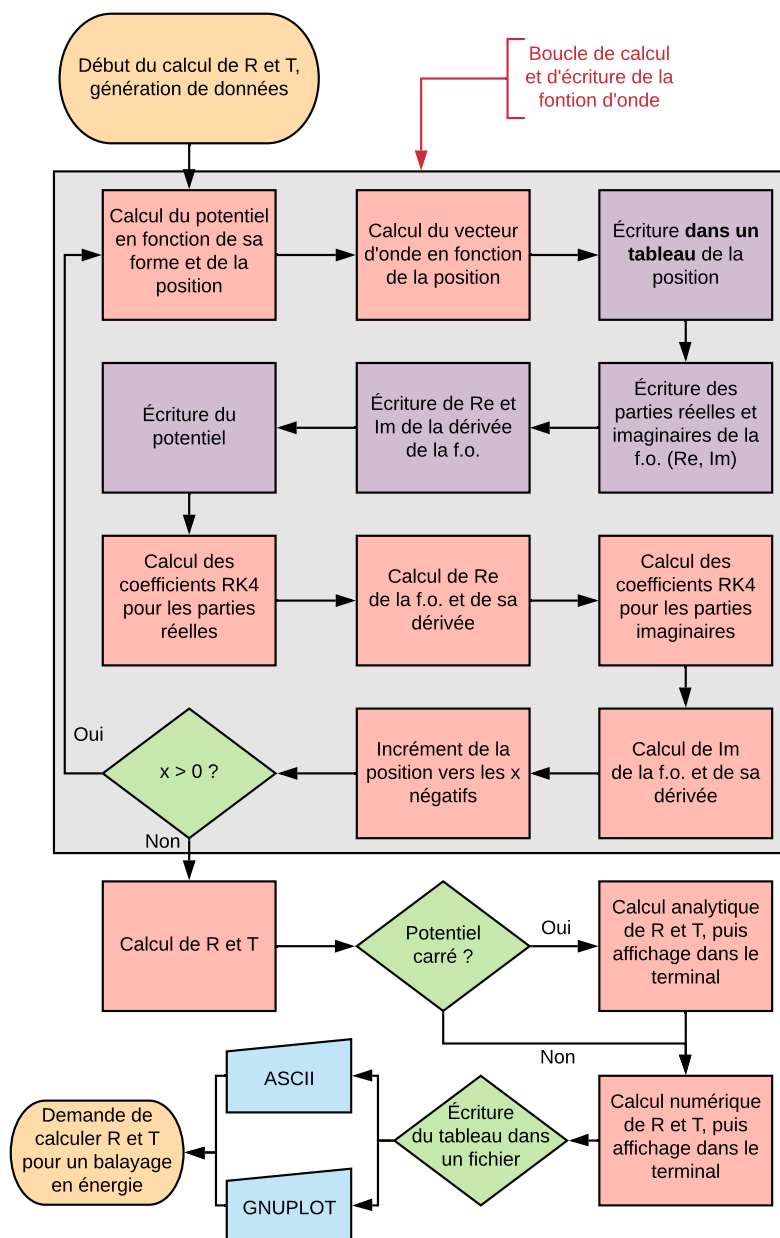


FIGURE 3 – Logigramme du calcul de R et T par le programme

Résultats : ?

3.2 Adaptation à un puits carré et à un potentiel harmonique

3.2.1 Le nouveau programme en détails

AJOUTER UNE EXPLICATION SUR LES ÉTATS LIÉS + CONDITION AUX LIMITES DE LA F.O.

Pour des énergies négatives, une explosion nulle (une convergence de la fonction d'onde) peut indiquer que l'énergie est celle d'un état propre de l'hamiltonien. J'ai donc créé une fonction effectuant un balayage en énergie du minimum au maximum du puit et j'ai cherché les valeurs absolues minimales de l'explosion de la fonction d'onde en $x = 0$ (on rappelle que la propagation se fait de $+\infty$ vers $-\infty$). C'est à dire pour quelles énergies $|\psi(0)|$ est minimale.

Pour cela, un pas d'incrémentatation est déterminé en fonction des énergies de départ et d'arrivée du balayage :

$$Pas = \frac{E_{max} - E_{min}}{1000} \quad (21)$$

On a donc 1000 points pour le balayage.

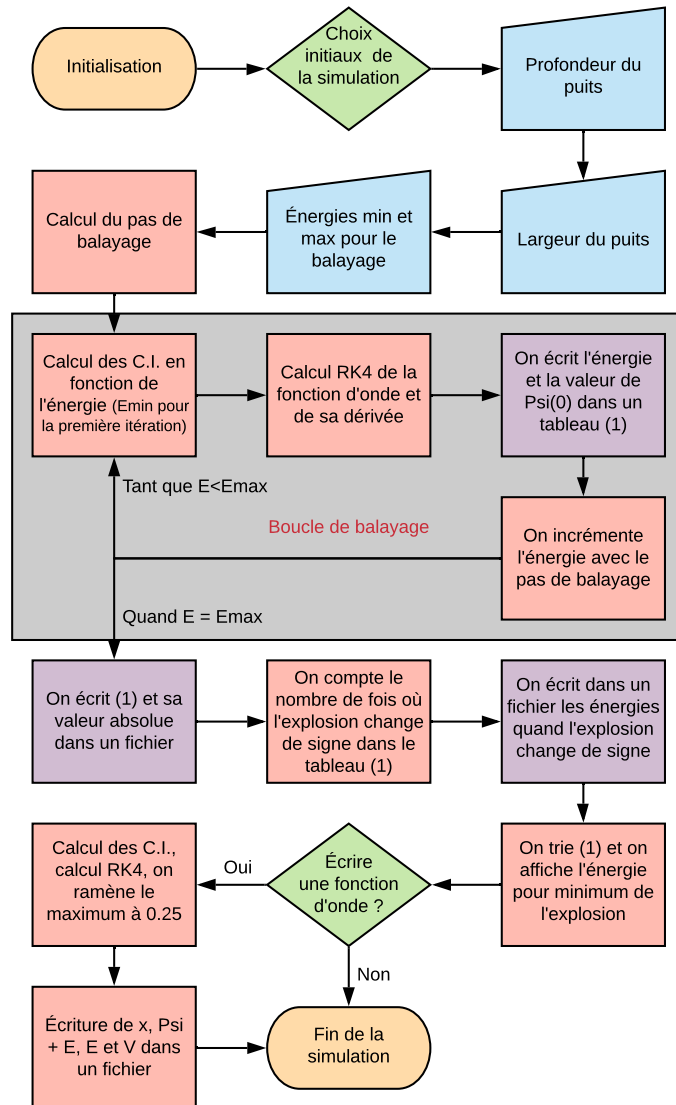


FIGURE 5 – Logigramme du nouveau programme. Le trie et l’affichage de l’énergie pour le minimum de l’explosion ne sont utiles que pour les affinages des changements de signe (expliqué plus loin).

3.2.2 États liés du puits carré

DÉTAILS CONDITIONS INITIALES

Le potentiel carré étudié ici prend la forme :

$$V(x) = \begin{cases} 0 & \text{si } x < 0.5 \\ -3 & \text{si } 0.5 \leq x \leq 5.5 \\ 0 & \text{si } x > 5.5 \end{cases} \quad (22)$$

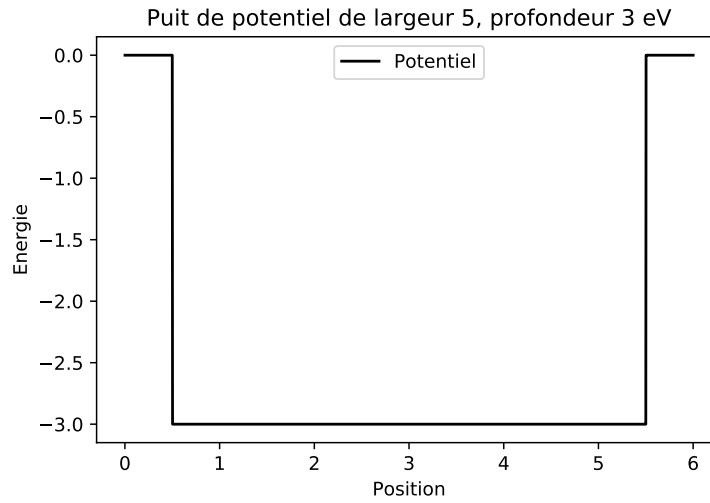


FIGURE 6 – Forme du puits étudié ici.

On commence par faire un premier balayage en énergie de -3 eV à 0 eV (on balaye tout le puits).

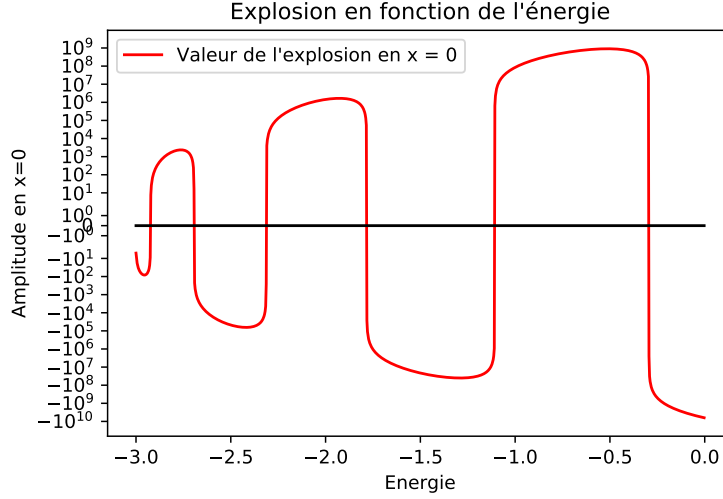


FIGURE 7 – Valeur de $\psi(0)$ en fonction de l'énergie pour le potentiel (22).

Comme on s'y attendait, on voit sur ce graphe que la valeur absolue de l'explosion en $x = 0$ diminue brusquement pour certaines valeurs de l'énergie. Ces diminutions correspondent toutes aux changements de signe de l'explosion. La fonction déterminant les intervalles d'énergie dans lesquelles $\psi(0)$ change de signe nous donne :

Energie (eV)	$\psi(0)$
-3	9.23436×10^{-313}
-2.997	-10.4924
-2.925	-9.00711
-2.922	7.60979
-2.694	16.3943
-2.691	-226.777
-2.313	-256.525
-2.31	4086.43
-1.785	54529.6
-1.782	-26857.6
-1.11	-925473
-1.107	646346
-0.297	2.67906×10^7
-0.294	-2.50448×10^6

On cherche maintenant à affiner ces intervalles pour déterminer avec précision les valeurs propres des états liés. Après affinage, on obtient pour le second changement de signe :

Energie (eV)	$\psi(0)$
-2.92334	-0.0328983
-2.92333	-0.0161811
-2.92333	0.017259
-2.92333	0.017259

On détermine donc l'énergie propre supposée :

$$E = -2.92333 \pm 0.00001 \text{ eV} \quad (23)$$

Et on trace la fonction d'onde correspondante :

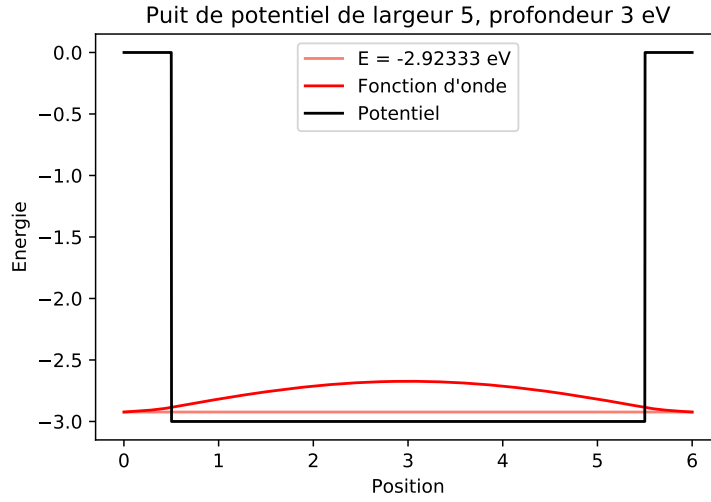


FIGURE 8 – On voit ici la fonction d'onde obtenue par la méthode RK4 pour l'énergie -2.92333 eV précisément. Elle est alignée avec son énergie pour des raisons de lisibilité. On rappelle que le programme ramène le maximum (ou le minimum) des fonctions d'onde à 0.25 puis qu'il ajoute son énergie à toute la fonction d'onde.

Il s'agit de l'état fondamental de notre système. On répète ces étapes pour les autres énergies et on obtient alors :

$$\begin{aligned}
E_1 &= -2.92333 \pm 0.00001 \text{ eV} \\
E_2 &= -2.69379 \pm 0.00001 \text{ eV} \\
E_3 &= -2.31282 \pm 0.00001 \text{ eV} \\
E_4 &= -1.78298 \pm 0.00001 \text{ eV} \\
E_5 &= -1.10822 \pm 0.00001 \text{ eV} \\
E_6 &= -0.294254 \pm 0.000001 \text{ eV}
\end{aligned} \quad (24)$$

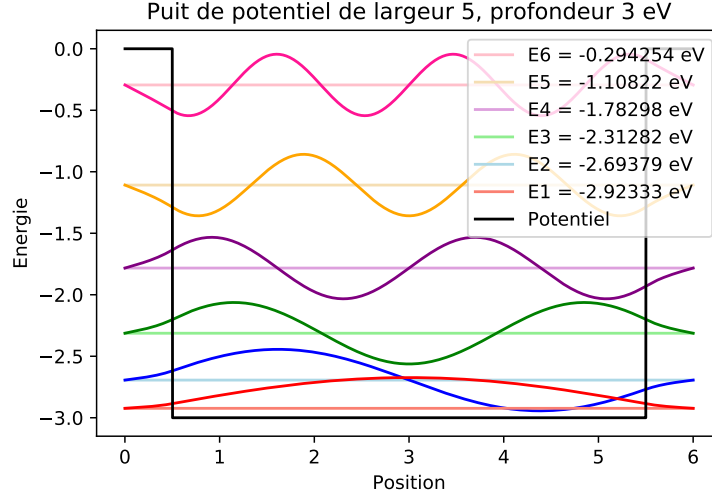


FIGURE 9 – Tracer des 6 états liés avec les énergies propres déterminées précédemment pour le potentiel (22). Les fonctions d'onde ne sont pas légendées pour des raisons de lisibilité, elles sont toutes alignées sur leur énergie.

Les écarts entre niveaux d'énergie sont :

$$\begin{aligned}
 E_2 - E_1 &= 0.22954 \pm 0.00002 \text{ eV} \\
 E_3 - E_2 &= 0.38097 \pm 0.00002 \text{ eV} \\
 E_4 - E_3 &= 0.52984 \pm 0.00002 \text{ eV} \\
 E_5 - E_4 &= 0.67476 \pm 0.00002 \text{ eV} \\
 E_6 - E_5 &= 0.813966 \pm 0.000011 \text{ eV}
 \end{aligned}
 \tag{25}$$

3.2.3 États liés du potentiel harmonique

Pour nous rapprocher du puits étudié précédemment, le potentiel traité ici prend la forme suivante :

$$V(x) = -\frac{x^2}{3} + 2x \quad (26)$$

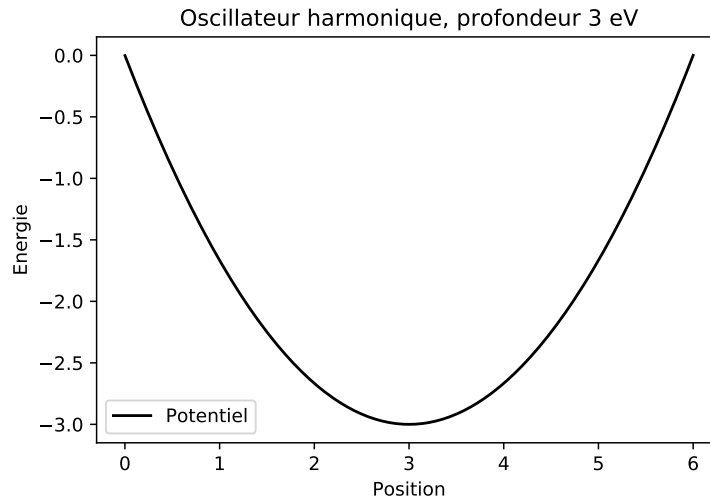


FIGURE 10 – Forme de notre oscillateur harmonique. Minimum de -3 eV en $x = 3$, deux racines en $x = 0$ et en $x = 6$.

On effectue un premier balayage de -3 à 0 eV et on regarde l'explosion.

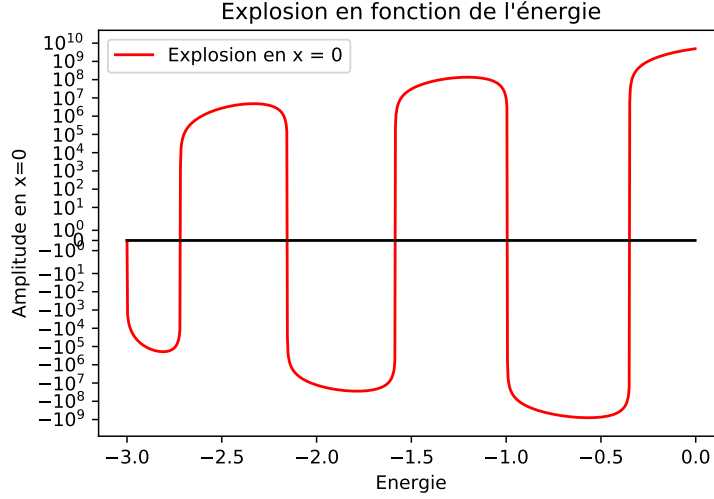


FIGURE 11 – Valeur de $\psi(0)$ en fonction de l'énergie pour le potentiel (26).

Les intervalles d'énergie dans lesquels $\psi(0)$ change de signe sont :

Energie (eV)	$\psi(0)$
-3	0
-2.997	-1879.25
-2.721	-12145.2
-2.718	1931.49
-2.157	147972
-2.154	-28019.9
-1.587	-605865
-1.584	279478
-0.996	3.55846×10^6
-0.993	-655163
-0.351	-1.81079×10^7
-0.348	6.21782×10^6

Un premier affinage autour du second changement de signe nous donne les valeurs suivantes² :

Energie (eV)	$\psi(0)$
-2.71841	-19.2058
-2.7184	-4.90427
-2.7184	9.3979
-2.7184	23.7007

2. La dernière décimale est manquante pour l'énergie -2.7184 eV. Il s'agit de l'affichage du terminal auquel je n'ai pas touché. La vraie valeur est -2.71840 eV.

On a l'énergie du premier changement de signe et on trace la fonction d'onde correspondante :

$$E = -2.71840 \pm 0.00001 \text{ eV} \quad (27)$$

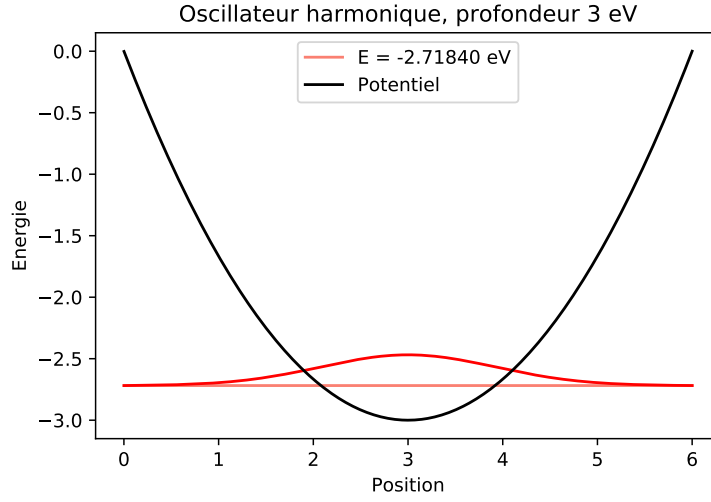


FIGURE 12 – Fonction d'onde obtenue pour l'énergie -2.71840 eV précisément. Elle est alignée avec son énergie pour des raisons de lisibilité.

Des affinages successifs du balayage nous permettent de déterminer les énergies suivantes et de tracer les fonctions d'onde correspondantes :

$$\begin{aligned} E_1 &= -2.71840 \pm 0.00001 \text{ eV} \\ E_2 &= -2.15447 \pm 0.00001 \text{ eV} \\ E_3 &= -1.58494 \pm 0.00001 \text{ eV} \\ E_4 &= -0.993463 \pm 0.000001 \text{ eV} \\ E_5 &= -0.348762 \pm 0.000001 \text{ eV} \end{aligned} \quad (28)$$

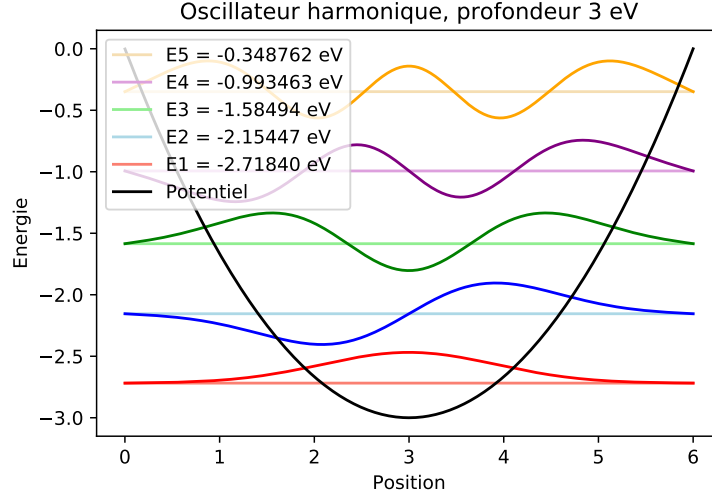


FIGURE 13 – Tracer des 5 états liés avec les énergies propres déterminées précédemment pour le potentiel (26).

Les écarts entre les niveaux successifs sont :

$$\begin{aligned}
 E_2 - E_1 &= 0.56393 \pm 0.00002 \text{ eV} \\
 E_3 - E_2 &= 0.56953 \pm 0.00002 \text{ eV} \\
 E_4 - E_3 &= 0.591477 \pm 0.000011 \text{ eV} \\
 E_5 - E_4 &= 0.644701 \pm 0.000002 \text{ eV}
 \end{aligned}
 \tag{29}$$

L'augmentation de l'écart est significative en comparaison de la prédiction (4) selon laquelle l'écart entre deux valeurs de l'énergie dans le spectre de l'oscillateur harmonique est constant ($\hbar\omega$).

3.2.4 États non-liés, phénomènes de résonance

On effectue maintenant le balayage de 0 eV à +3 eV au dessus du potentiel (22).

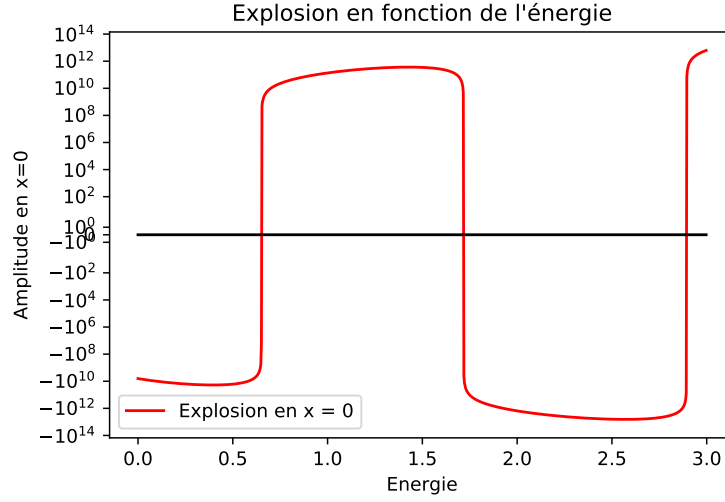


FIGURE 14 – Explosion en fonction de l'énergie pour le potentiel (22). On balaye de 0 à +3eV. On observe là encore des chutes brusques de l'explosion pour certaines valeurs de l'énergie. ANTI-RÉSONANCE ?

On observe là encore de brusques chutes dans l'explosion en $x = 0$. Le premier balayage nous donne les intervalles suivants :

Energie (eV)	$\psi(0)$
0.651	-1.5618×10^7
0.654	5.18651×10^8
1.716	4.90288×10^9
1.719	-4.08953×10^9
2.892	-7.59726×10^{10}
2.895	6.85421×10^{10}

L'affinage de ces énergies nous donne les valeurs suivantes :

$$\begin{aligned}
 E_1 &= 0.651088 \pm 0.000001 \text{ eV} \\
 E_2 &= 1.71764 \pm 0.00001 \text{ eV} \\
 E_3 &= 2.89358 \pm 0.00001 \text{ eV}
 \end{aligned} \tag{30}$$

On trace les fonctions d'ondes pour ces énergies :

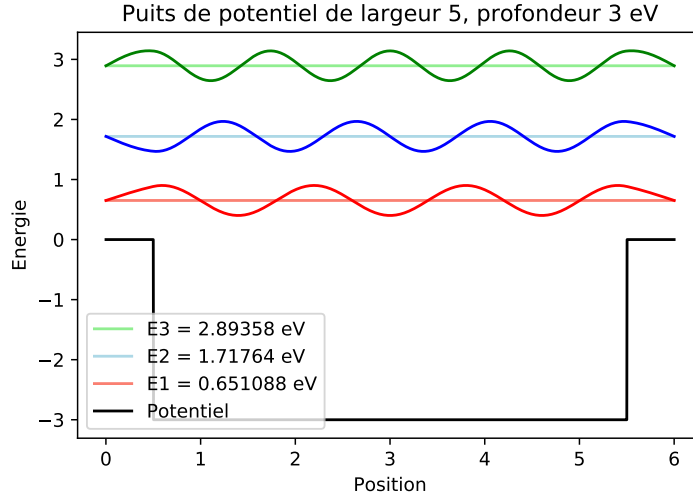


FIGURE 15 – Là encore, les fonctions d'onde ne sont pas légendées, elles sont alignées sur leurs énergies. ANTI-RÉSONANCE ?

Les écarts entre niveaux successifs sont :

$$\begin{aligned}
 E_2 - E_1 &= 1.066552 \pm 0.000011 \text{ eV} \\
 E_3 - E_2 &= 1.17594 \pm 0.00002 \text{ eV}
 \end{aligned}
 \tag{31}$$

On refait le même balayage de 0 à +3 eV mais cette fois au dessus du potentiel harmonique (26).

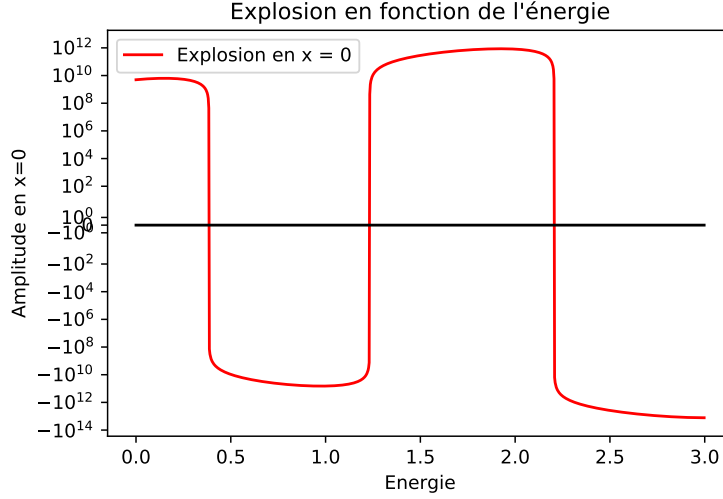


FIGURE 16 – Valeur absolue de l’explosion en fonction de l’énergie pour le potentiel (22). On balaye de 0 à +3eV. On observe là encore des chutes brusques de l’explosion pour certaines valeurs de l’énergie. ANTI-RÉSONANCE ?

Le premier balayage nous donne les intervalles suivants :

Energie (eV)	$\psi(0)$
0.384	4.48584×10^7
0.387	1.40469×10^8
1.23	-1.35319×10^9
1.233	4.38442×10^8
2.205	5.77642×10^9
2.208	-1.5672×10^{10}

L’affinage de ces énergies nous donne les valeurs suivantes :

$$\begin{aligned}
 E_1 &= 0.384731 \pm 0.000001 \text{ eV} \\
 E_2 &= 1.23227 \pm 0.00001 \text{ eV} \\
 E_3 &= 2.20581 \pm 0.00001 \text{ eV}
 \end{aligned} \tag{32}$$

On trace les fonctions d’ondes pour ces énergies :

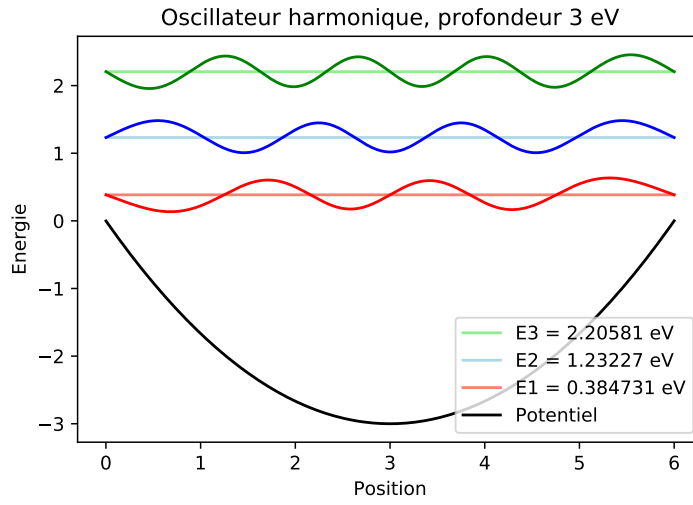


FIGURE 17 – ANTI-RÉSONANCE ?

Les écarts entre niveaux successifs sont :

$$\begin{aligned}
 E_2 - E_1 &= 0.847539 \pm 0.000011 \text{ eV} \\
 E_3 - E_2 &= 0.97354 \pm 0.00002 \text{ eV}
 \end{aligned}
 \tag{33}$$

4 Réseaux de neurones

4.1 Explications théoriques sur les réseaux de neurones

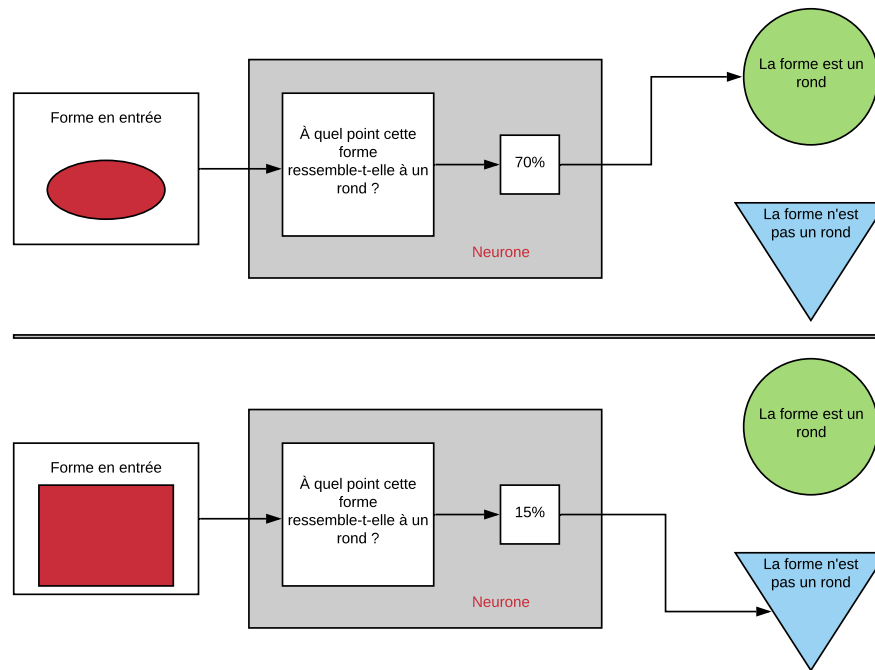


FIGURE 18 – Représentation d'un réseau à un neurone classant des formes (input) dans deux catégories (output) : "rond" ou "autre". Le neurone détermine un "pourcentage de ressemblance" et, lorsqu'il dépasse 50, le neurone s'allume. En dessous de 50% le neurone reste éteint et un choix par défaut est effectué. **Attention, cette représentation est simpliste au sens où un tel réseau nécessiterait plusieurs neurones, elle ne sert qu'à saisir l'idée générale.**

Le manque de choix en sortie rend ce "réseau" assez limité en apparence, mais il permet de répondre aisément à un besoin de classification binaire. Nous verrons plus loin qu'un tel réseau peut nous aider à approximer des fonctions mais parlons des fonctions d'activation³ des neurones dans un premier temps.

3. [2]

4.1.1 Fonctions d'activation

Une fonction d'activation, comme son nom l'indique, active ou non le neurone pour une entrée donnée. La fonction la plus simple qu'on peut imaginer est la **fonction de Heaviside** :

$$H(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x > 0 \end{cases} \quad (34)$$

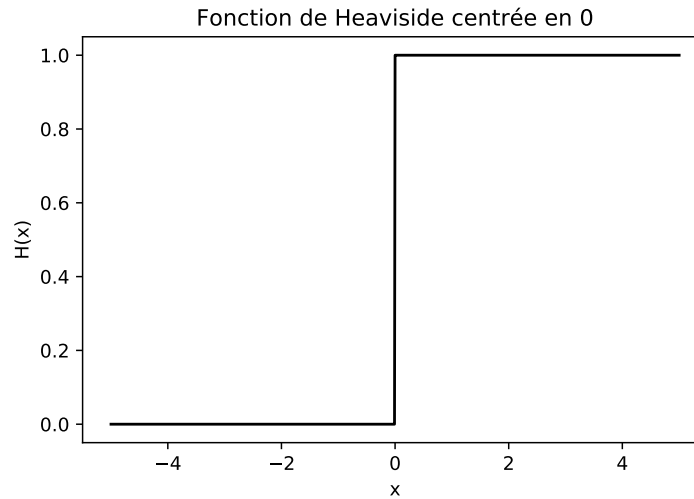


FIGURE 19 – Fonction d'activation de Heaviside. Le neurone est activé lorsque $x > 0$.

Une telle fonction d'activation est très limitée. Reprenons notre exemple de reconnaissance de formes. La sortie d'une telle fonction est binaire : "le neurone reconnaît un rond" ou "le neurone ne reconnaît pas un rond". On voudrait connaître à quel point un rond a été reconnu (ce qui est représenté sur le schéma). Pour ce faire, on peut utiliser la **fonction sigmoïde**⁴.

$$f(z) = \frac{1}{1 + e^{-z}} \quad (35)$$

où z peut être réécrit :

$$z = wx + b \quad (36)$$

où w est le poids de la variable x et b le biais.

4. [5]

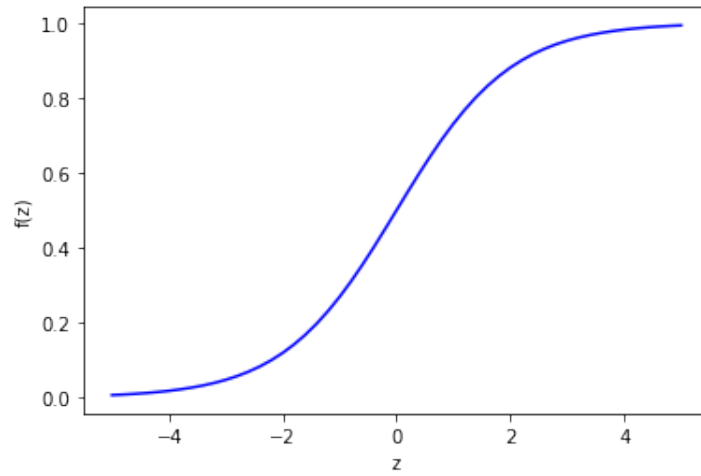


FIGURE 20 – Fonction sigmoïde de biais 0 et de poids 1.

Ici, le neurone est activé si $f(z) > 0.5$. C'est à dire si $x > 0$. Autrement dit, notre neurone considérera que la forme est un rond au delà de 50% de reconnaissance.

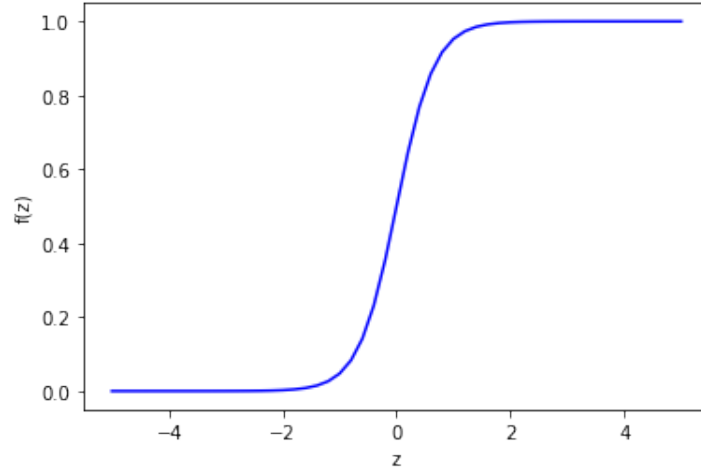


FIGURE 21 – Fonction sigmoïde de biais 0 et de poids 3.

On voit ici que la sigmoïde se rapproche d'une fonction de Heaviside en augmentant le poids w . Essayons de modifier le biais b .

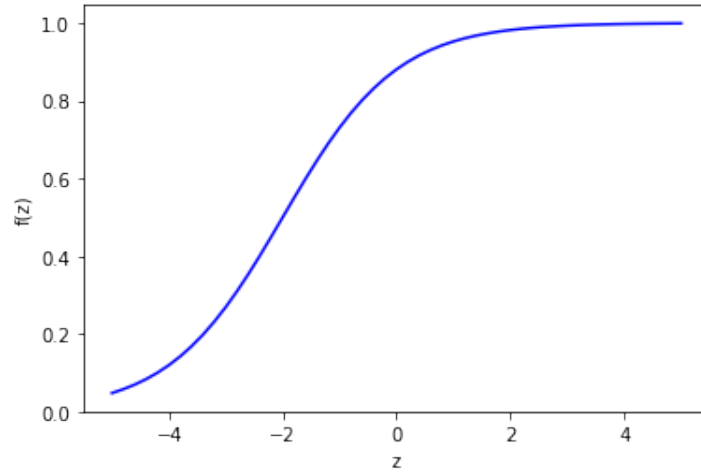


FIGURE 22 – Fonction sigmoïde de biais 2 et de poids 1.

Ici, on remarque que $f(z)$ dépassera les 50% pour une valeur plus basse de x . Autrement dit, avec un biais très grand ($b \gg 0$), le neurone reconnaîtra toujours un rond, et inversement. Il reste donc à déterminer quels sont les paramètres w et b permettant la meilleure classification des formes. Pour ça, il faut entraîner le réseau sur un échantillon connu par le réseau qui lui permettra d'adapter ses paramètres (w et b) à chaque erreur de classification.

La fonction d'activation que nous utilisons ici est la **fonction ReLU** (rectified linear unit) :

$$f(x) = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases} \quad (37)$$

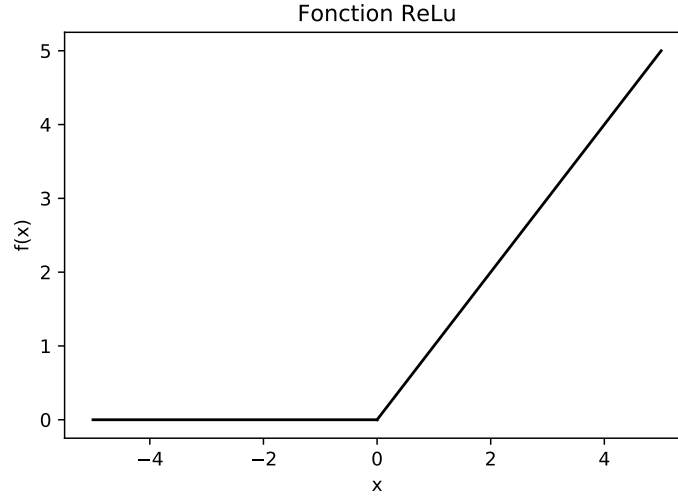


FIGURE 23 – Fonction d’activation ReLU. Le neurone est toujours éteint pour $x < 0$ et s’allume linéairement pour $x \geq 0$. La position en abscisse de l’angle peut être adaptée par entraînement du réseau.

Nous verrons dans le point suivant en quoi la fonction ReLU est mieux adaptée que la sigmoïde pour notre utilisation. Mais d’abord, nous devons nous pencher sur le gradient de la fonction d’activation et sur les réseaux à plusieurs couches.

4.1.2 Gradient de la fonction d’activation – réseaux à plusieurs couches

Une fonction d’activation ReLU est donc plus efficace qu’une sigmoïde ⁵.

4.1.3 Précision du réseau

Une fois l’entraînement effectué, on peut tester le réseau sur un nouvel échantillon (connu de l’utilisateur mais pas du réseau) pour déterminer la précision ⁶ de la classification ⁷ :

$$a = 1 - \frac{M}{N} \quad (38)$$

où a signifie "accuracy", M est le nombre d’éléments mal classés et N le nombre total d’éléments dans l’échantillon. Ce paramètre doit tendre vers 1 dans l’idéal.

5. [ReLU_vs_Sigm]

6. [8]

7. [8]

4.2 Approximation de fonctions d'onde par réseaux de neurones

4.2.1 Théorème d'approximation universelle

Selon ce théorème⁸, un réseau de neurones à propagation avant (sans boucle) ayant une seule couche cachée et contenant un nombre fini de neurones peut approximer toute fonction continue à support compact de \mathbb{R}^n . Notre but, dans un premier temps, est d'illustrer ce théorème [1] en approximant une fonction gaussienne.

8. [2]

4.2.2 fit d'une gaussienne, variations du nombre d'outputs

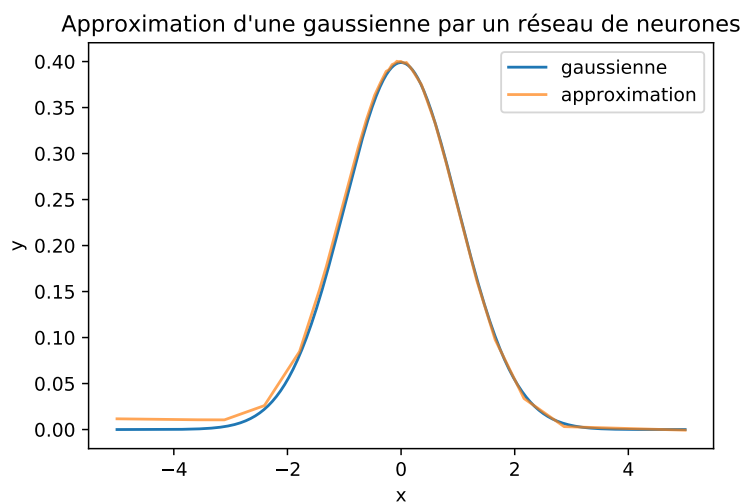


FIGURE 24 – Fit d'une gaussienne. 10001 points. outputs : 20, 20, 1. Total params : 481 Trainable params : 481. Epochs = 30, batch = 20

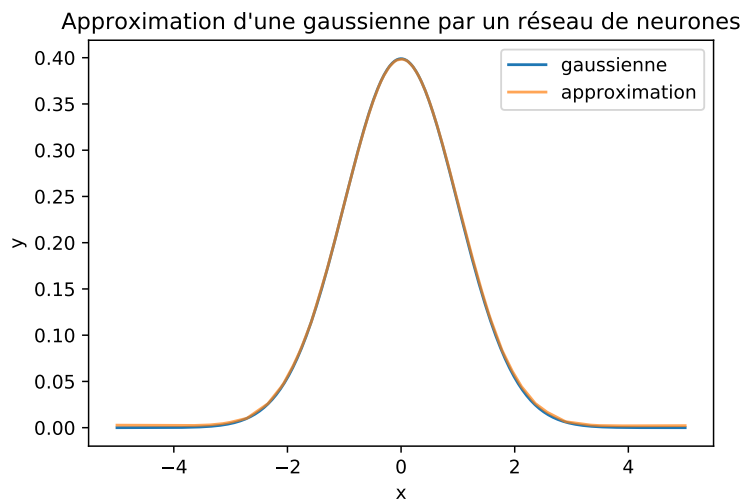


FIGURE 25 – Fit d'une gaussienne. 10001 points. outputs : 200, 200, 1. Total params : 40,801 Trainable params : 40,801. Epochs = 30, batch = 20

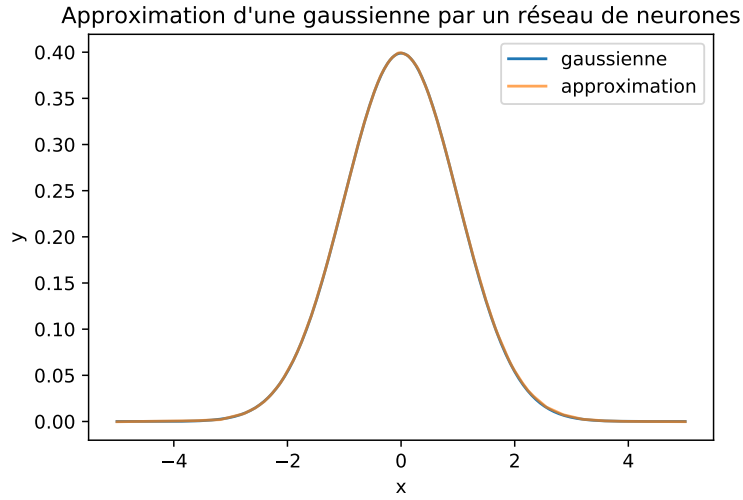


FIGURE 26 – Fit d'une gaussienne. 10001 points. outputs : 400, 400, 1. Total params : 161,601 Trainable params : 161,601. Epochs = 30, batch = 20

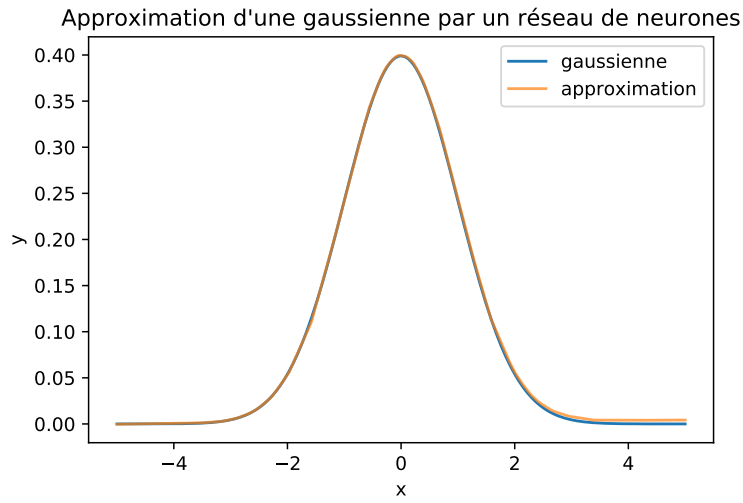


FIGURE 27 – Fit d'une gaussienne. 10001 points. outputs : 200, 20, 1. Total params : 4,441 Trainable params : 4,441. Epochs = 30, batch = 20

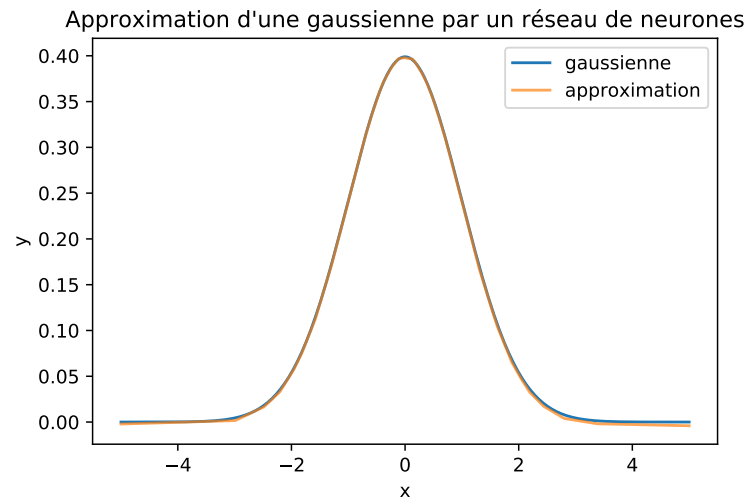


FIGURE 28 – Fit d'une gaussienne. 10001 points. outputs : 20, 200, 1. Total params : 4,441 Trainable params : 4,441. Epochs = 30, batch = 20

4.2.3 fit d'une gaussienne, variation du batch

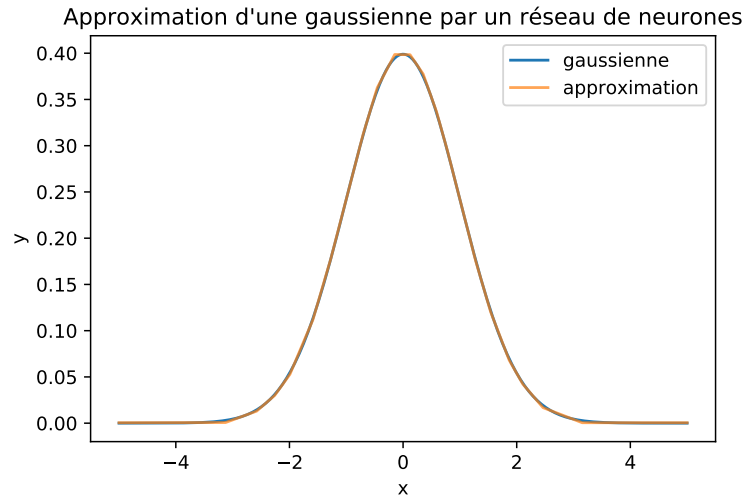


FIGURE 29 – Fit d'une gaussienne. 10001 points. outputs : 200, 200, 1. Total params : 40,801 Trainable params : 40,801. Epochs = 30, batch = 2

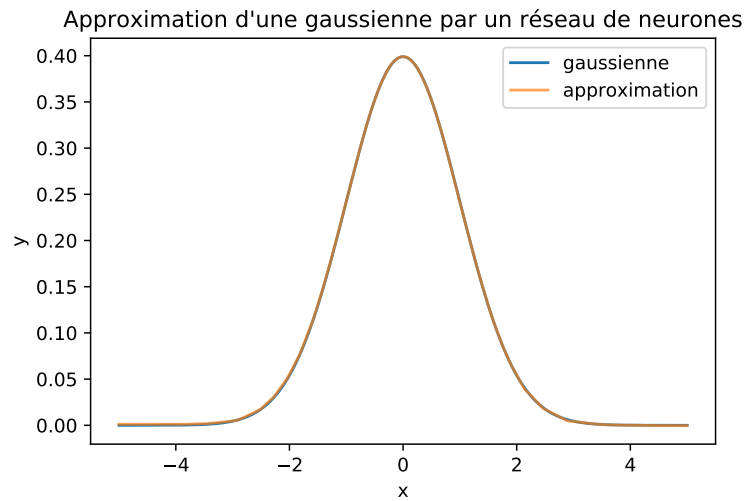


FIGURE 30 – Fit d'une gaussienne. 10001 points. outputs : 200, 200, 1. Total params : 40,801 Trainable params : 40,801. Epochs = 30, batch = 20

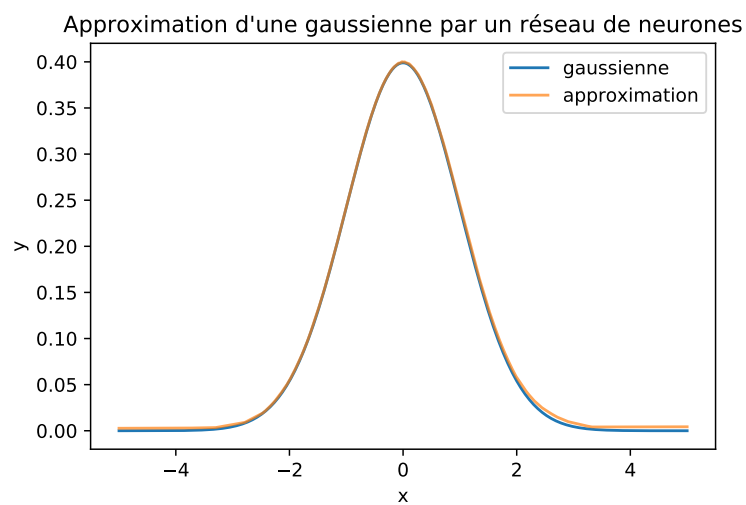


FIGURE 31 – Fit d'une gaussienne. 10001 points. outputs : 200, 200, 1. Total params : 40,801 Trainable params : 40,801. Epochs = 30, batch = 200

4.2.4 fit d'une gaussienne, variation des epochs

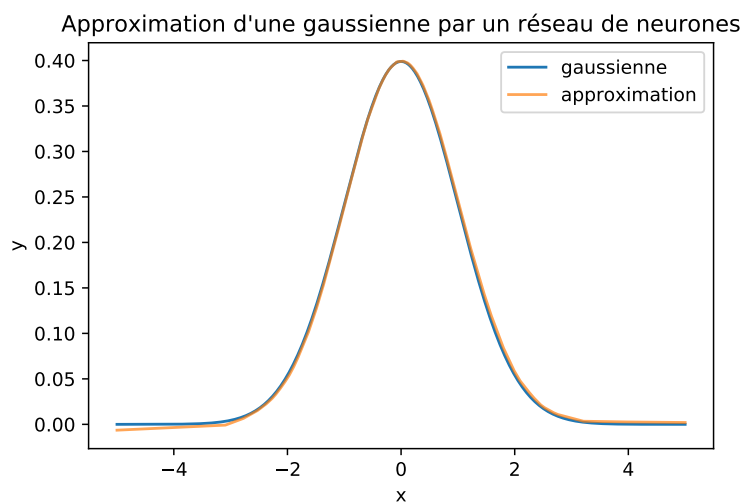


FIGURE 32 – Fit d'une gaussienne. 10001 points. outputs : 200, 200, 1. Total params : 40,801 Trainable params : 40,801. Epochs = 3, batch = 20

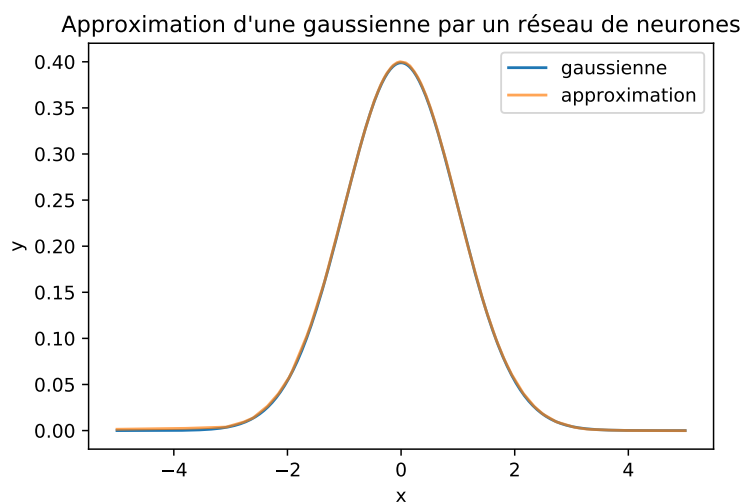


FIGURE 33 – Fit d'une gaussienne. 10001 points. outputs : 200, 200, 1. Total params : 40,801 Trainable params : 40,801. Epochs = 30, batch = 20

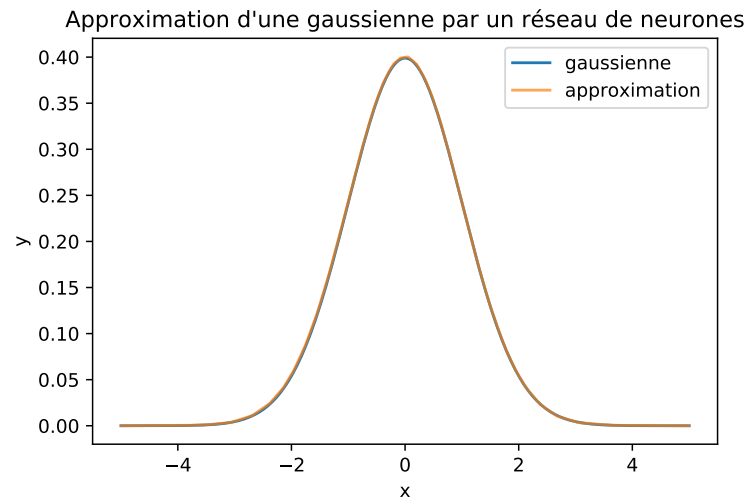


FIGURE 34 – Fit d'une gaussienne. 10001 points. outputs : 200, 200, 1. Total params : 40,801 Trainable params : 40,801. Epochs = 300, batch = 20

4.2.5 fit d'une gaussienne, variation batch vs epochs

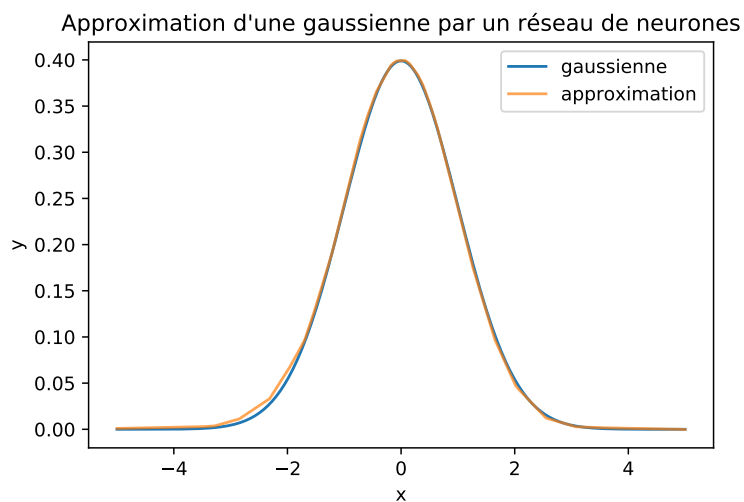


FIGURE 35 – Fit d'une gaussienne. 10001 points. outputs : 200, 200, 1. Total params : 40,801 Trainable params : 40,801. Epochs = 3, batch = 2. 77 secondes de calcul.

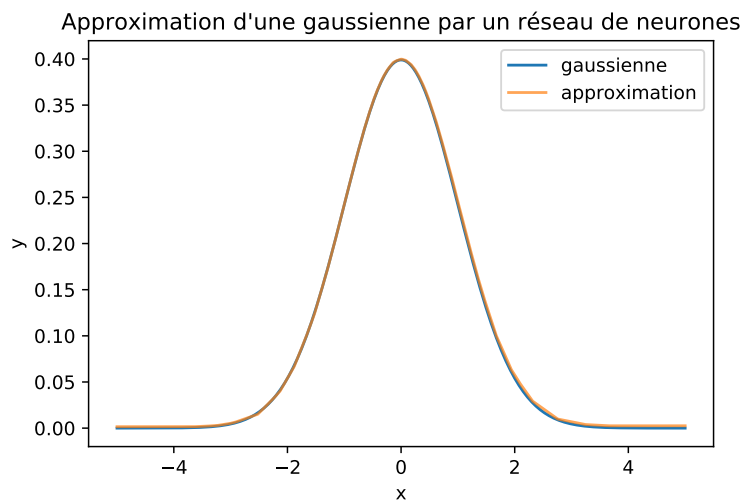


FIGURE 36 – Fit d'une gaussienne. 10001 points. outputs : 200, 200, 1. Total params : 40,801 Trainable params : 40,801. Epochs = 30, batch = 20. 83 secondes de calcul.

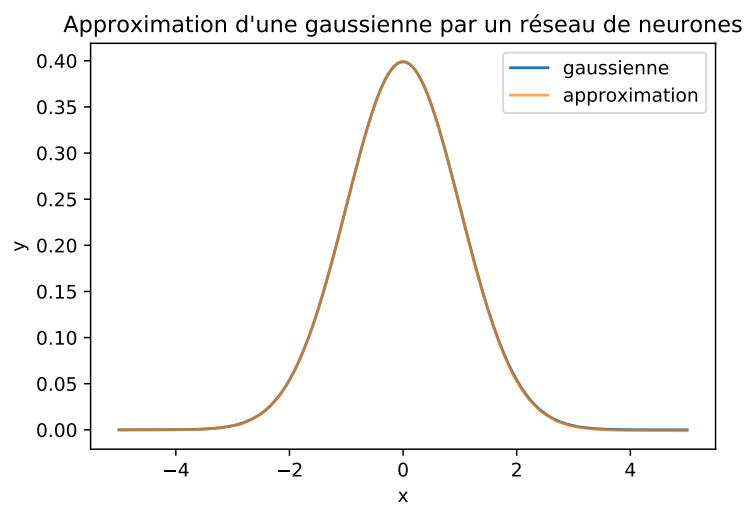


FIGURE 37 – Fit d'une gaussienne. 10001 points. outputs : 200, 200, 1. Total params : 40,801 Trainable params : 40,801. Epochs = 300, batch = 200. 96 secondes de calcul.

4.2.6 Fit des polynômes d’hermite - variation du nombre de points et d’outputs (annexes)

On voit en annexes A et B que, pour un même réseau et un échantillon de 500 points, l’approximation d’une fonction d’onde semble plus précise que l’approximation du module carré. Cette différence, si elle existe, est plus difficile à détecter visuellement pour un échantillon à 10001 points (annexes C et D).

Mémo : [5] [11] [9] [6] [8] [7] [4] [10]

Bibliographie

- [1] Moshe LESHNOI et al. “Multilayer feedforward networks with a nonpolynomial activation function can approximate any function”. In : *Neural Networks* 6 (1993), p. 861-867. DOI : 10.1016/S0893-6080(05)80131-5.
- [2] Balázs Csanád CSÁJI. “Approximation with Artificial Neural Networks”. In : *MSc Thesis, Eötvös Loránd University (ELTE), Budapest, Hungary* 24 :48 (2001). URL : <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.2647&rep=rep1&type=pdf>.
- [3] Claude COHEN-TANNOUDJI, Bernard DIU et Franck LALOË. *Mécanique quantique Tome 1*. EDP Sciences, 2018, p. 371-378. ISBN : 9782759822874.
- [4] Colin BERNET. *Handwritten Digit Recognition with scikit-learn*. URL : <https://thedatafrog.com/en/articles/handwritten-digit-recognition-scikit-learn/>.
- [5] Colin BERNET. *Le réseau à un neurone : régression logistique*. URL : <https://thedatafrog.com/fr/articles/logistic-regression/>.
- [6] Colin BERNET. *Le surentraînement*. URL : <https://thedatafrog.com/fr/articles/overfitting-illustrated/>.
- [7] Colin BERNET. *Matplotlib for Machine Learning*. URL : <https://thedatafrog.com/en/articles/matplotlib-machine-learning/>.
- [8] Colin BERNET. *Numpy Crash Course for Machine Learning*. URL : <https://thedatafrog.com/en/articles/numpy-crash-course-machine-learning/>.
- [9] Colin BERNET. *Premier réseau de neurones avec keras*. URL : <https://thedatafrog.com/fr/articles/first-neural-network-keras/>.
- [10] Colin BERNET. *Python Crash Course for Machine Learning*. URL : <https://thedatafrog.com/en/articles/python-crash-course-machine-learning/>.
- [11] Colin BERNET. *Régression Logistique vs Réseau de Neurones : Non Linéarités*. URL : <https://thedatafrog.com/fr/articles/logistic-regression-neural-network/>.

A Approximation des pol. d'Hermite par réseau de neurones (500 points)

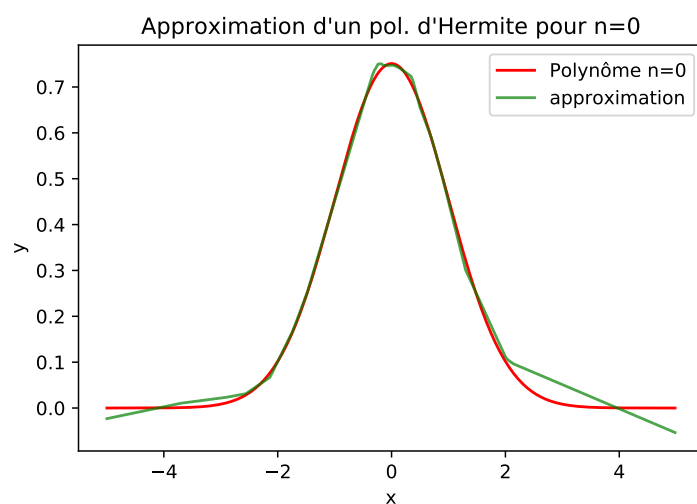


FIGURE 38 – $n=0$, 500 points. outputs : 20 20 1. Params : 481. Trainable : 481.

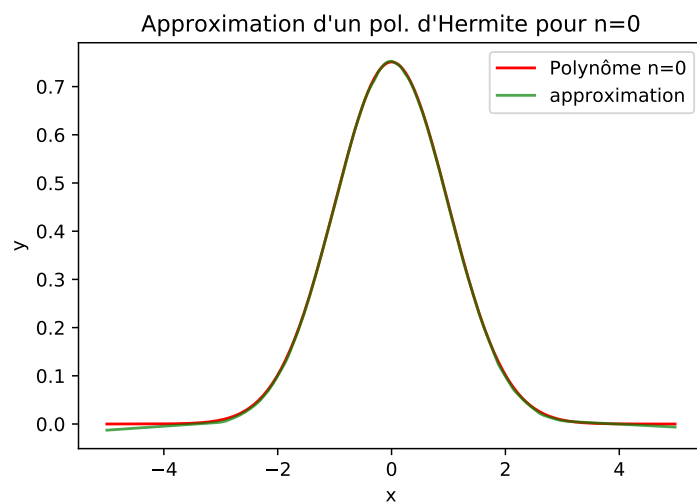


FIGURE 39 – $n=0$, 500 points. Réseau 200 200 1. Params : 40801. Trainable : 40801.

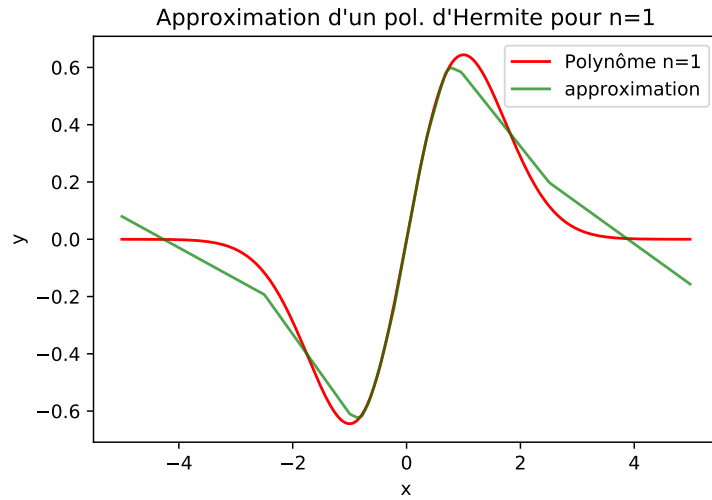


FIGURE 40 – $n=1$, 500 points. outputs : 20 20 1. Params : 481. Trainable : 481.

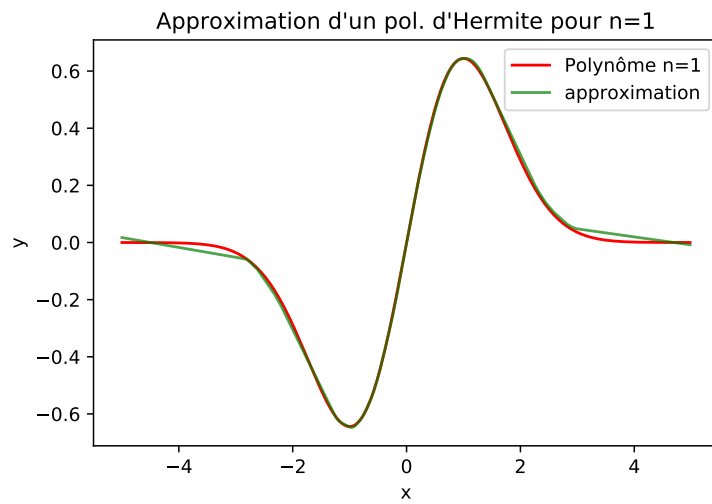


FIGURE 41 – $n=1$, 500 points. outputs : 200 200 1. Params : 40801. Trainable : 40801.

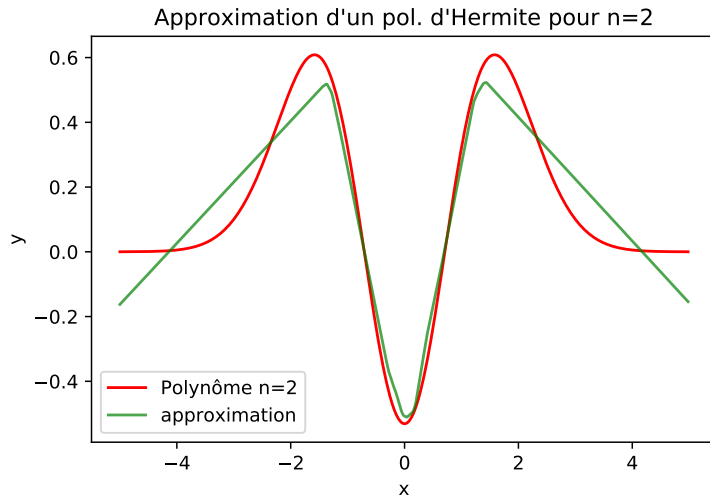


FIGURE 42 – $n=2$, 500 points. outputs : 20 20 1. Params : 481. Trainable : 481.

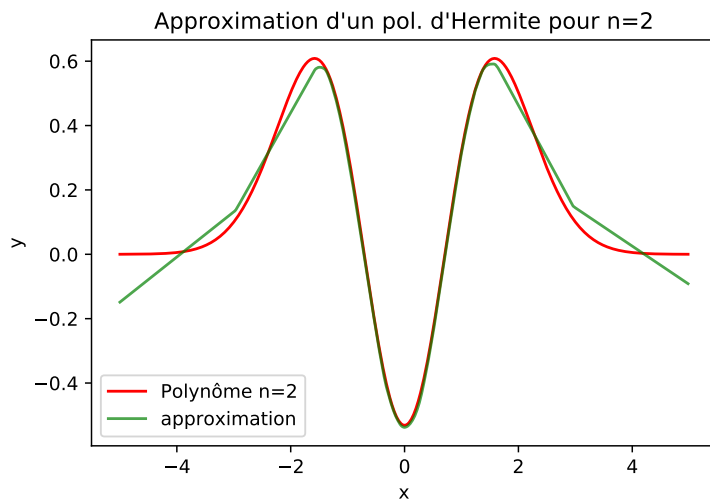


FIGURE 43 – $n=2$, 500 points. outputs : 200 200 1. Params : 40801. Trainable : 40801.

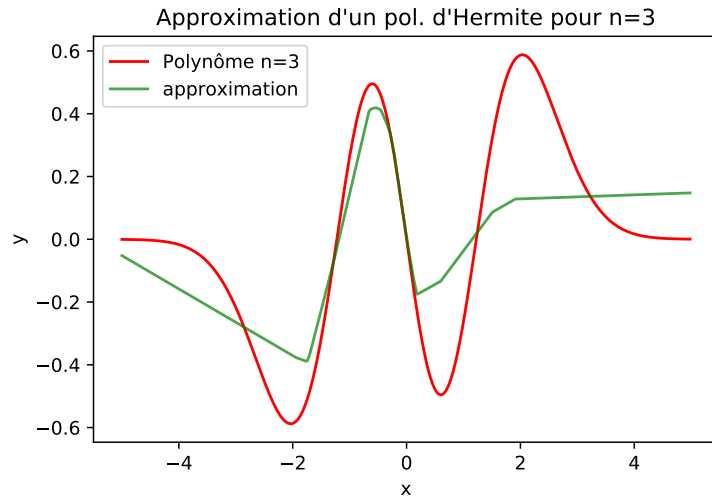


FIGURE 44 – $n=3$, 500 points. outputs : 20 20 1. Params : 481. Trainable : 481.

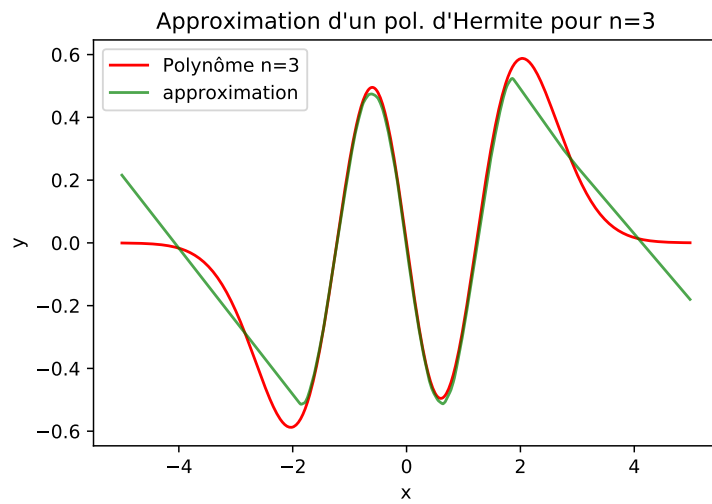


FIGURE 45 – $n=3$, 500 points. outputs : 200 200 1. Params : 40801. Trainable : 40801.

B Approximation des modules carrés des pol. d'Hermite par réseau de neurones (500 points)

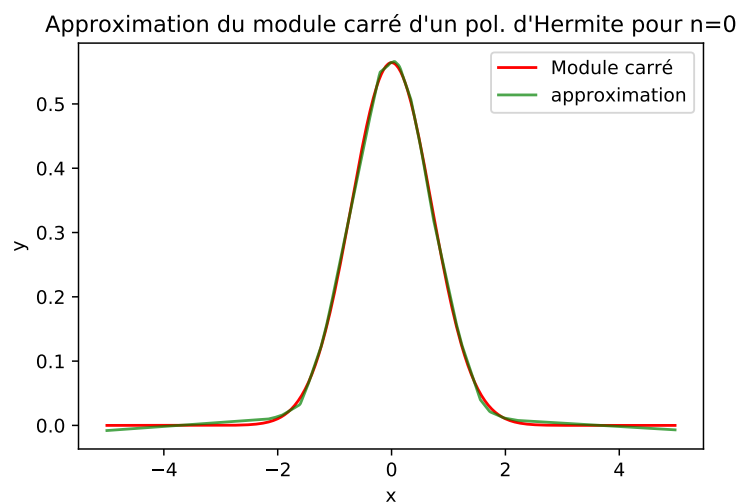


FIGURE 46 – $n=0$, 500 points. outputs : 20 20 1. Params : 481. Trainable : 481.

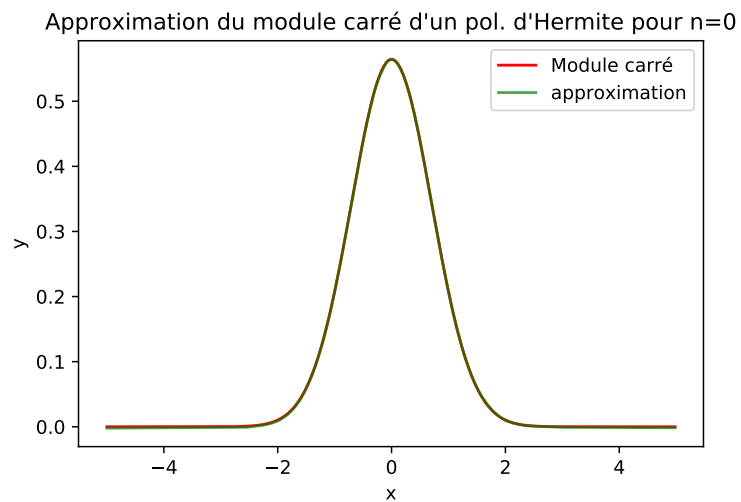


FIGURE 47 – $n=0$, 500 points. outputs : 200 200 1. Params : 40801. Trainable : 40801.

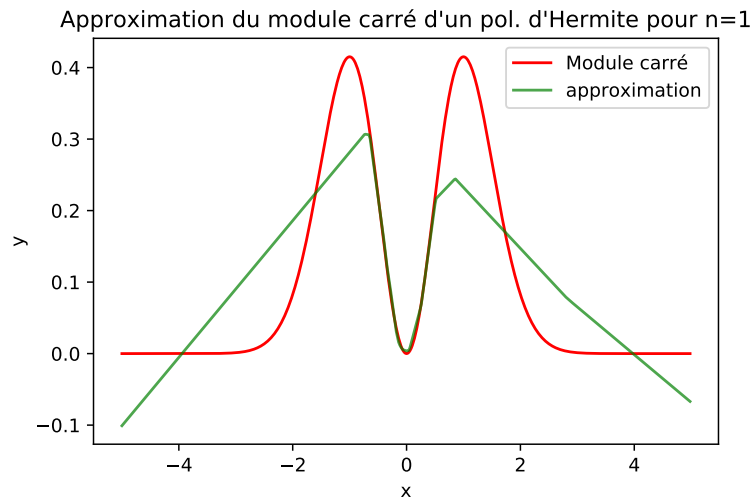


FIGURE 48 – $n=1$, 500 points. outputs : 20 20 1. Params : 481. Trainable : 481.

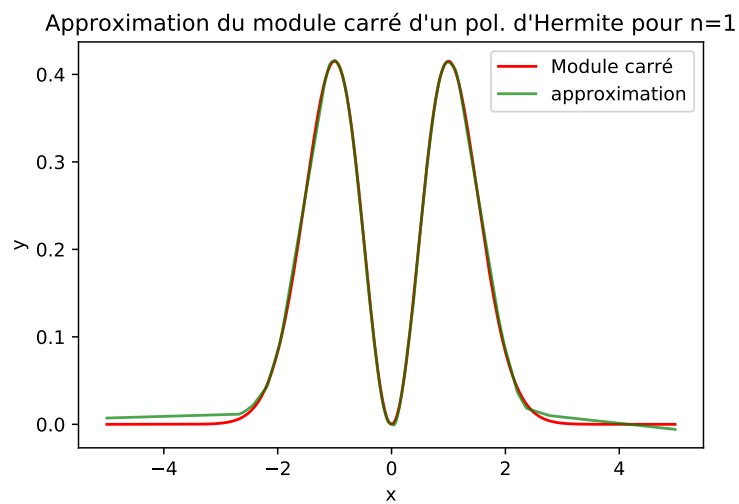


FIGURE 49 – $n=1$, 500 points. outputs : 200 200 1. Params : 40801. Trainable : 40801.

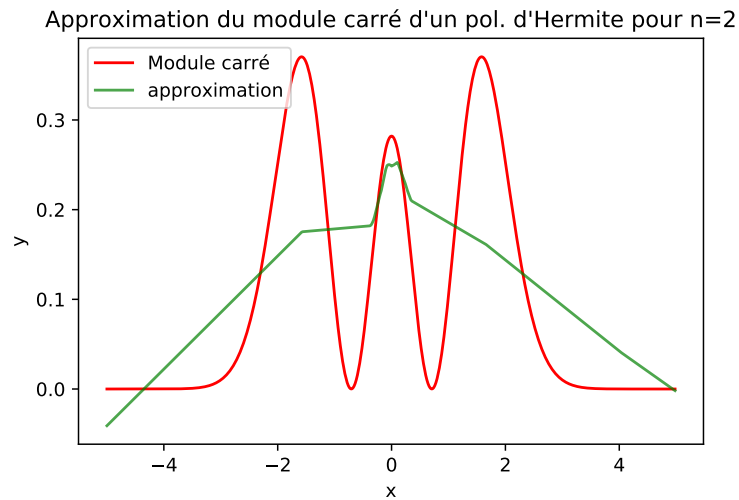


FIGURE 50 – $n=2$, 500 points. outputs : 20 20 1. Params : 481. Trainable : 481.

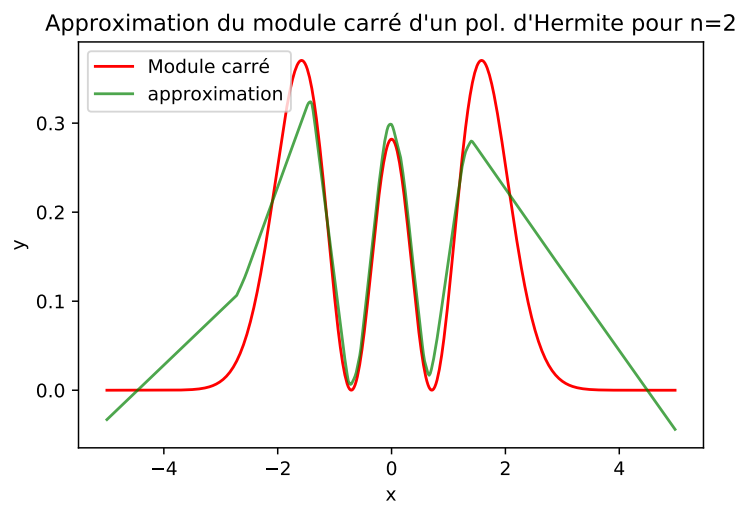


FIGURE 51 – $n=2$, 500 points. outputs : 200 200 1. Params : 40801. Trainable : 40801.

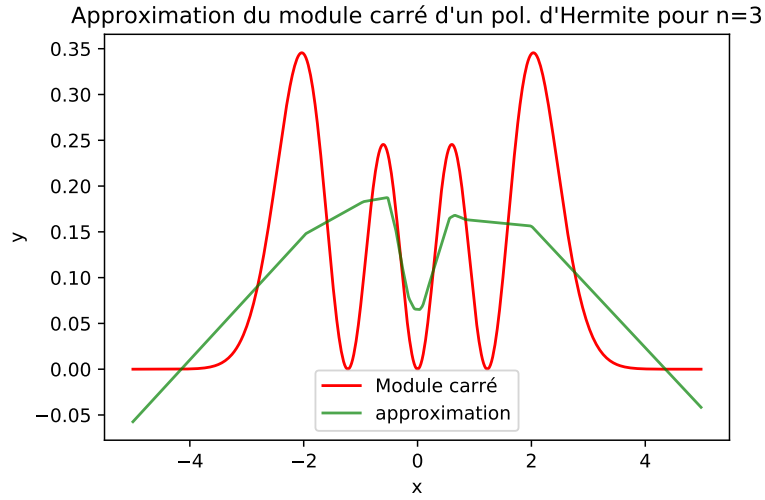


FIGURE 52 – $n=3$, 500 points. outputs : 20 20 1. Params : 481. Trainable : 481.

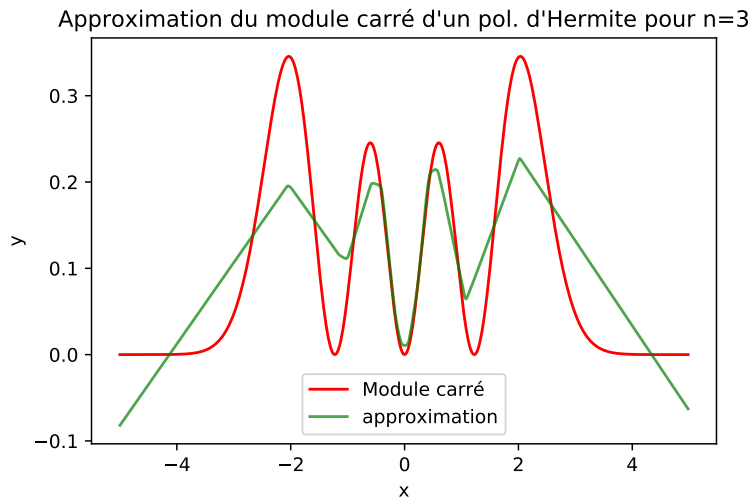


FIGURE 53 – $n=3$, 500 points. outputs : 200 200 1. Params : 40801. Trainable : 40801.

C Approximation des pol. d'Hermite par réseau de neurones (10001 pts)

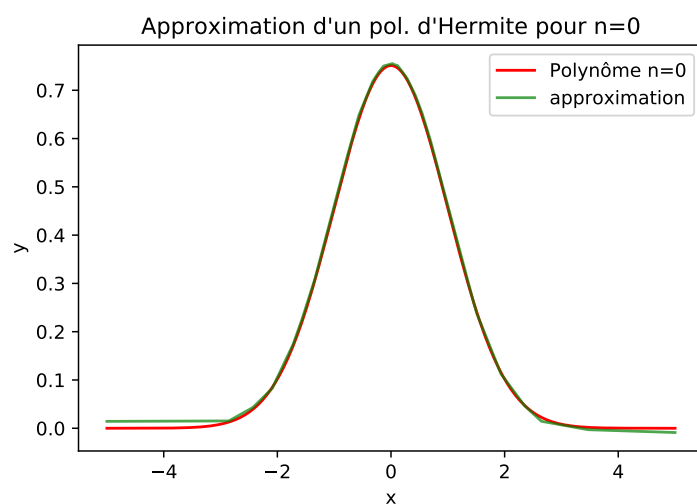


FIGURE 54 – $n=0$, 10001 pts. outputs : 20 20 1. Params : 481. Trainable : 481.

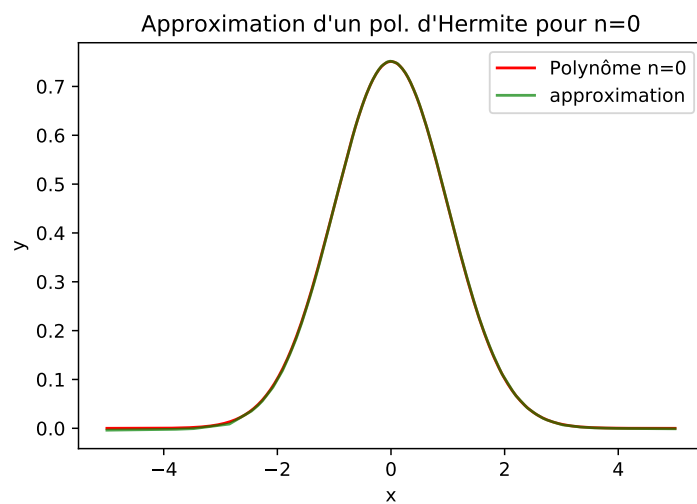


FIGURE 55 – $n=0$, 10001 pts. outputs : 200 200 1. Params : 40801. Trainable : 40801.

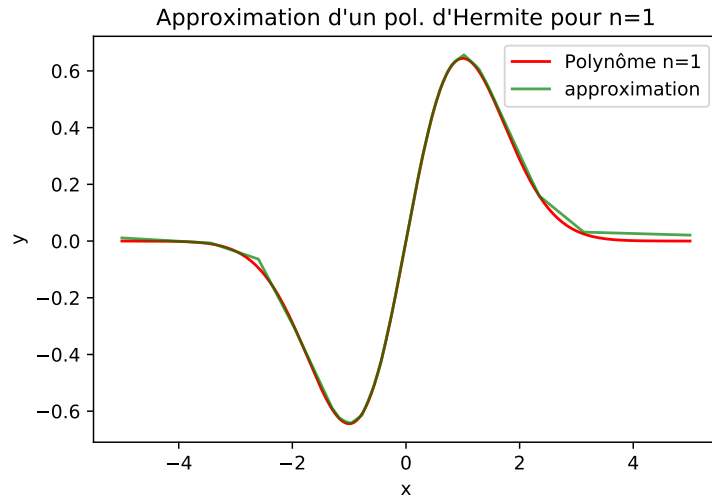


FIGURE 56 – $n=1$, 10001 pts. outputs : 20 20 1. Params : 481. Trainable : 481.

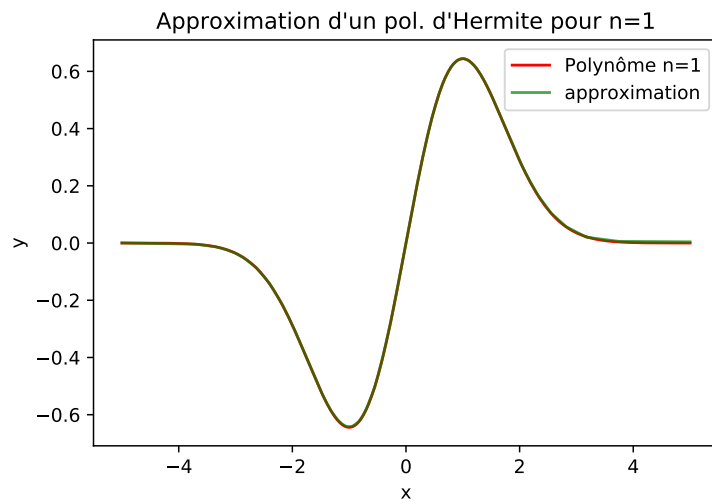


FIGURE 57 – $n=1$, 10001 pts. outputs : 200 200 1. Params : 40801. Trainable : 40801.

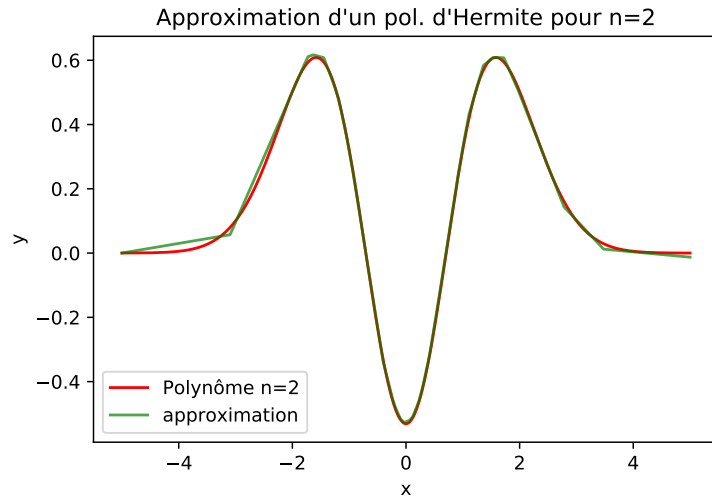


FIGURE 58 – $n=2$, 10001 pts. outputs : 20 20 1. Params : 481. Trainable : 481.

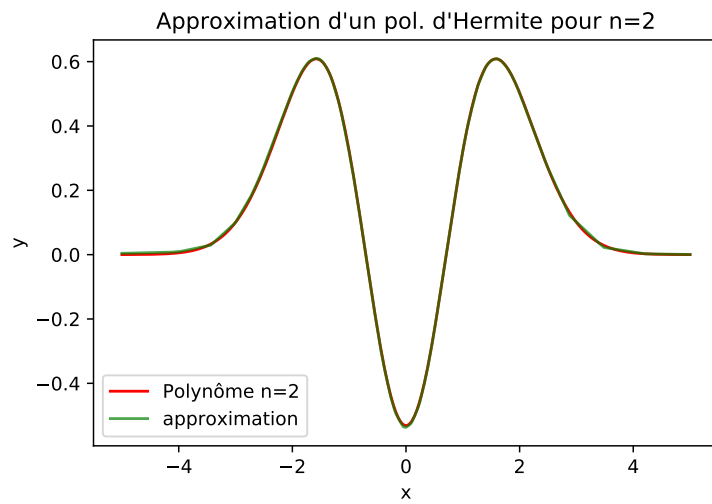


FIGURE 59 – $n=2$, 10001 pts. outputs : 200 200 1. Params : 40801. Trainable : 40801.

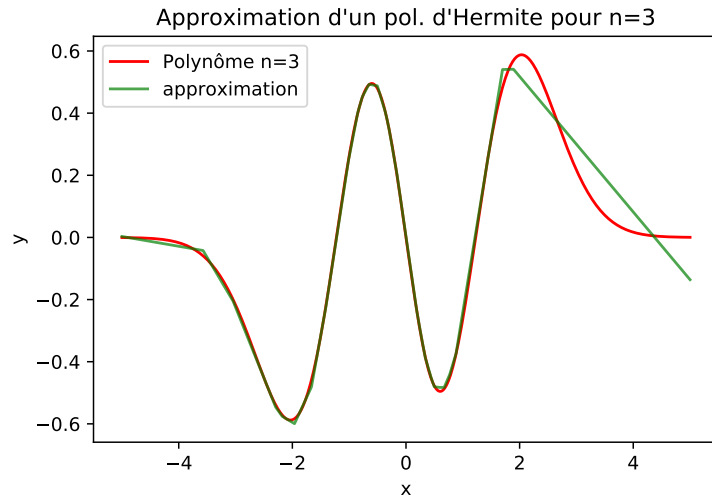


FIGURE 60 – $n=3$, 10001 pts. outputs : 20 20 1. Params : 481. Trainable : 481.

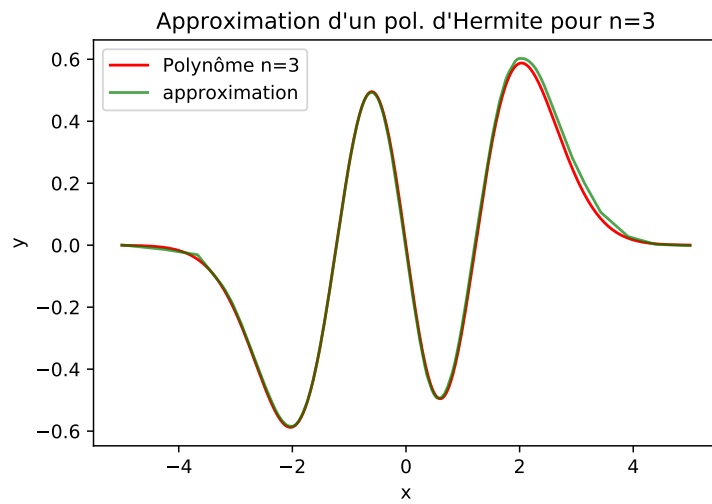


FIGURE 61 – $n=3$, 10001 pts. outputs : 200 200 1. Params : 40801. Trainable : 40801.

D Approximation des modules carrés des pol. d'Hermite par réseau de neurones (10001 pts)

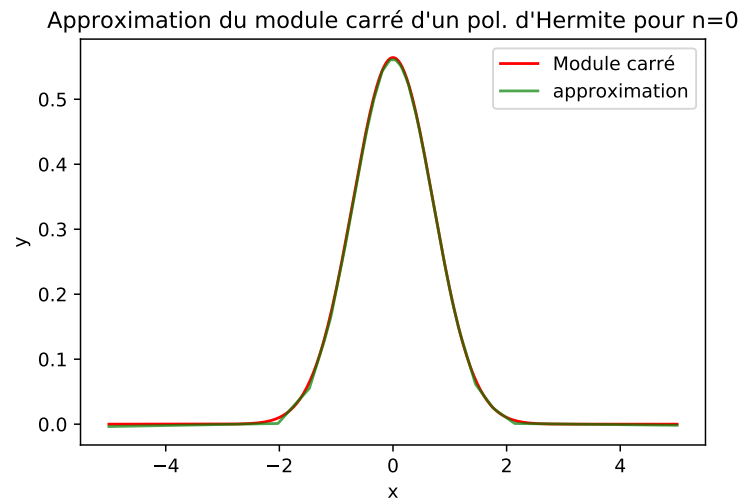


FIGURE 62 – $n=0$, 10001 pts. outputs : 20 20 1. Params : 481. Trainable : 481.

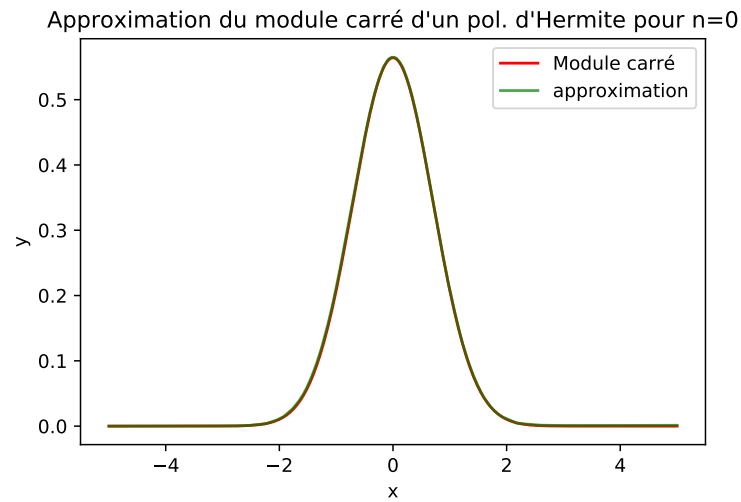


FIGURE 63 – $n=0$, 10001 pts. outputs : 200 200 1. Params : 40801. Trainable : 40801.

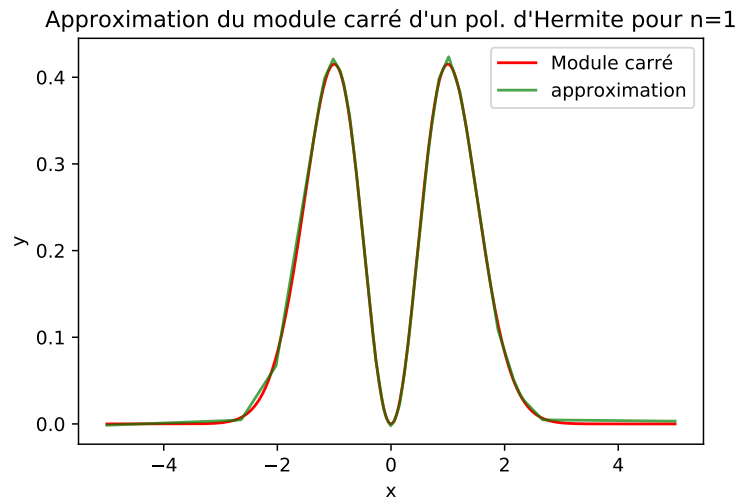


FIGURE 64 – $n=1$, 10001 pts. outputs : 20 20 1. Params : 481. Trainable : 481.

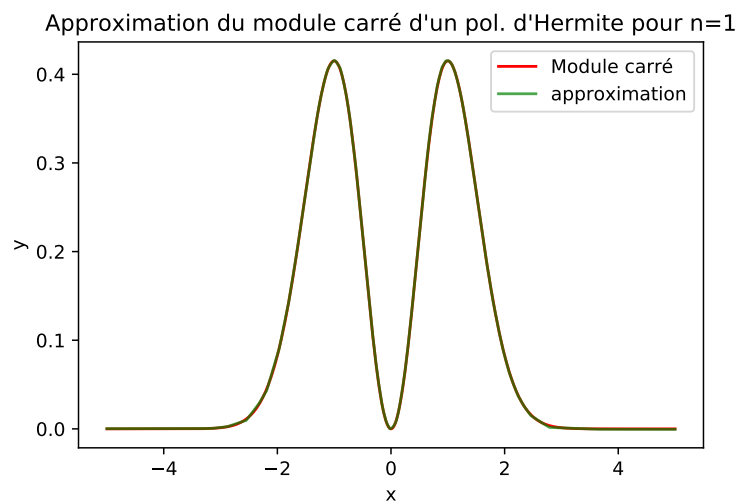


FIGURE 65 – $n=1$, 10001 pts. outputs : 200 200 1. Params : 40801. Trainable : 40801.

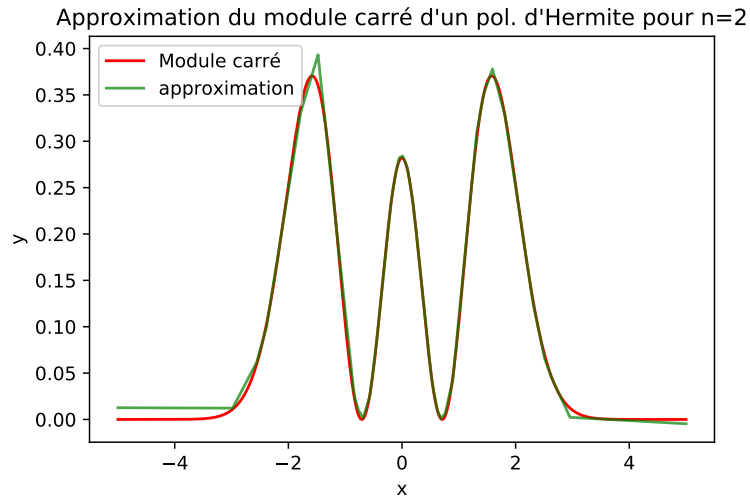


FIGURE 66 – $n=2$, 10001 pts. outputs : 20 20 1. Params : 481. Trainable : 481.

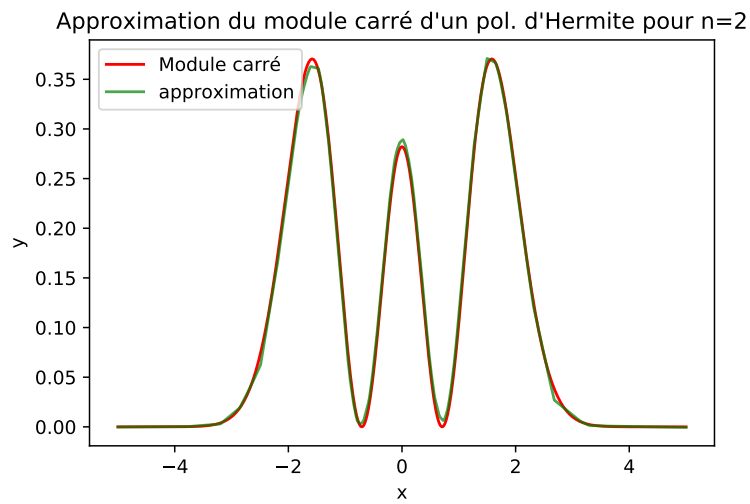


FIGURE 67 – $n=2$, 10001 pts. outputs : 200 200 1. Params : 40801. Trainable : 40801.

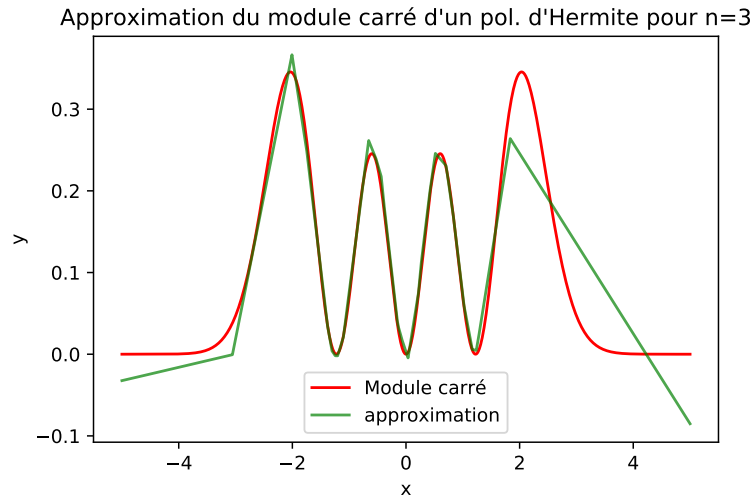


FIGURE 68 – $n=3$, 10001 pts. outputs : 20 20 1. Params : 481. Trainable : 481.

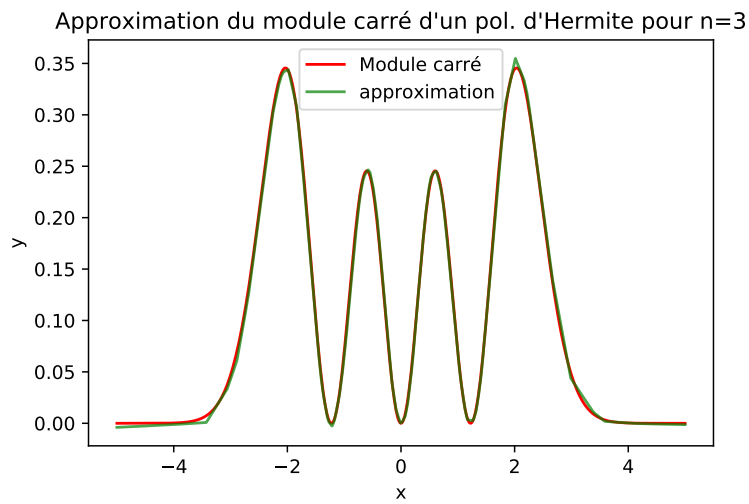


FIGURE 69 – $n=3$, 10001 pts. outputs : 200 200 1. Params : 40801. Trainable : 40801.