

Compte Rendu sur la modélisation des équations aux dérivées partielles

Stam Nicolis¹

Institut Denis Poisson (UMR7013)

Université de Tours, Université d'Orléans

37200 Tours, Parc Grandmont

Résumé

On présente la solution de l'équation de Schrödinger pour le cas d'une particule dans un potentiel à plusieurs puits et l'on explore la résurrection de la fonction d'onde dans ce cas.

1. E-Mail : stam.nicolis@lmpt.univ-tours.fr ; stamatios.nicolis@univ-tours.fr

Table des matières

1	Introduction	1
2	Le problème	2
2.1	Choix de coordonnées	2
2.2	Choix d'unités	2
2.3	La méthode spectrale	2
3	La méthode numérique	5
3.1	Tests de cohérence	5
4	Résultats	8
5	Conclusions	13
A	Le paquet d'onde classique et le paquet d'onde quantique dans un domaine infini	13
A.1	La solution pour le cas quantique	14
B	Le code	15

1 Introduction

{intro}

Le but de cet exercice est l'étude de la solution de l'équation de Schrödinger pour la particule libre, soumise à des conditions aux limites particulières :

$$i\hbar \frac{\partial \psi}{\partial t} = H\psi = -\frac{\hbar^2}{2m} \frac{\partial^2 \psi}{\partial x^2} \quad (1) \quad \{\text{Schroed}\}$$

avec les conditions aux limites $\psi(\pm L/2, t) = 0$ (la particule est “piégée” dans une “boîte” à une dimension). Dans un premier temps on étudie l'évolution dans cette boîte, par la suite on introduira des “obstacles”, qui nécessiteront des conditions supplémentaires.

2 Le problème

{prob}

2.1 Choix de coordonnées

{coords}

On peut choisir notre intervalle spatial comme $x \in [-L/2, L/2]$ ou comme $x \in [0, L]$.

Exercice 1. *Etablir la correspondance entre les deux choix.*

On va choisir la convention $x \in [0, L]$.

2.2 Choix d'unités

{units}

On peut choisir nos unités pour mettre la valeur de toute quantité “dimensionnée” égale à 1 ou toute autre valeur commode. Ainsi l'on va poser

$$\begin{array}{ll} \hbar & 1 \\ m & 1/2 \\ L & 1 \end{array} \quad (2) \quad \{\text{unitsq}\}$$

2.3 La méthode spectrale

{spectrale}

La méthode qui sera employée pour résoudre l'éq. (1)–avec le choix d'unités présenté plus haut–est la soi-disant *méthode spectrale* : On va exprimer la fonction, $\psi(x, t)$ comme une combinaison linéaire des fonctions propres du Hamiltonien de la particule libre, qui satisfont les conditions aux limites :

$$\psi(x, t) = \sum_n^{\infty} c_n(t) f_n(x) \quad (3) \quad \{\text{solspectral}\}$$

avec

$$\begin{aligned} -\frac{d^2}{dx^2} f_n(x) &= \lambda_n f_n(x) \\ f_n(0) &= 0 = f_n(1) \end{aligned} \quad (4) \quad \{\text{spectral}\}$$

Théorème 1. *Les valeurs propres, λ_n , sont données par l'expression*

$$\lambda_n = n^2 \pi^2 \quad (5) \quad \{\text{valpropres}\}$$

et les fonctions propres par l'expression

$$f_n(x) = A_n \sin n\pi x \quad (6) \quad \{\text{fonctpropres}\}$$

Démonstration. La solution générale de l'éq. (4) est donnée par l'expression

$$f_n(x) = C_1 \cos x \sqrt{\lambda_n} + C_2 \sin x \sqrt{\lambda_n} \quad (7) \quad \{\text{solutionsp}\}$$

On impose, maintenant, les conditions aux limites, indiquées dans (4), et l'on trouve le système suivant pour les coefficients $C_{1,2}$:

$$\begin{aligned} f_n(0) &= C_1 = 0 \\ f_n(1) &= 0 = C_1 \cos \sqrt{\lambda_n} + C_2 \sin \sqrt{\lambda_n} \end{aligned} \quad (8) \quad \{\text{conditionslim}\}$$

Ces équations impliquent que $C_1 = 0$ et que

$$C_2 \sin \sqrt{\lambda_n} = 0 \quad (9) \quad \{\text{conditionslim1}\}$$

Si $C_2 = 0$, alors $f_n(x) = 0$, donc ce n'est pas une fonction propre ; par conséquent, $C_2 \neq 0$, alors il faut (et suffit) que

$$\sin \sqrt{\lambda_n} = 0 \Leftrightarrow \sqrt{\lambda_n} = n\pi \Leftrightarrow \lambda_n = n^2\pi^2 \quad (10) \quad \{\text{conditionslim2}\}$$

avec $n > 0$. □

Ces fonctions propres peuvent être choisies orthonormées, c.à.d. satisfaire la condition

$$\int_0^1 dx f_n(x) f_m(x) = \delta_{nm} \quad (11) \quad \{\text{orthonormfn}\}$$

Exercice 2. Déterminer le facteur de normalisation, A_n .

Solution : En remplaçant l'expression (6) dans l'intégrale, on trouve

$$\begin{aligned} A_n A_m \int_0^1 dx \sin n\pi x \sin m\pi x &= \frac{A_n A_m}{2} \int_0^1 dx [\cos(n-m)\pi x - \cos(n+m)\pi x] = \\ &= -\frac{A_n A_m}{2} \left[\frac{\sin(n-m)\pi x}{n-m} - \frac{\sin(n+m)\pi x}{n+m} \right] \Big|_0^1 = \delta_{nm} \end{aligned} \quad (12) \quad \{\text{normfn}\}$$

On remarque que

$$-\frac{\sin(n+m)\pi x}{n+m} \Big|_0^1 = 0 \quad (13) \quad \{\text{terme2}\}$$

quelles que soient les valeurs des entiers n et m car $n+m \neq 0$, puisque ce sont des entiers plus grands que 0 ; tandis que le premier terme a besoin de plus d'attention, car le dénominateur, $n-m$, peut s'annuler. Par conséquent, on doit traiter les deux cas à part.

— $n = m$: Alors $\cos(n - m)\pi x = \cos 0 = 1$ et l'on trouve que

$$\int_0^1 dx \cos 0 = 1 \quad (14) \quad \{\mathbf{n=m}\}$$

Par conséquent, si $n = m$,

$$\int_0^1 dx f_n(x)^2 = \frac{A_n^2}{2} = 1 \Leftrightarrow A_n = \sqrt{2} \quad (15) \quad \{\mathbf{normfn=m}\}$$

— $n \neq m$: Alors on trouve que

$$\int_0^1 dx \cos(n - m)\pi x = -\frac{\sin(n - m)\pi x}{(n - m)} \Big|_0^1 = 0 \quad (16) \quad \{\mathbf{normfnneqm}\}$$

Finalement, on trouve que

$$A_n = \sqrt{2} \quad (17) \quad \{\mathbf{Annorm}\}$$

L'expression de la fonction $\psi(x, t)$ comme combinaison linéaire des fonctions propres, $f_n(x)$, du Hamiltonien, qui satisfont les conditions aux limites, implique que la solution de l'équation de Schrödinger se réduit à la solution d'une équation différentielle ordinaire, où le nombre du mode, n , apparaît comme une étiquette.

On trouve l'expression suivante pour cette équation différentielle :

$$\begin{aligned} i \frac{\partial \psi}{\partial t} \psi(x, t) &= i \frac{\partial}{\partial t} \sum_{n=1}^N c_n(t) f_n(x) = i \sum_{n=1}^N i \frac{dc_n}{dt} f_n(x) = \\ -\frac{\partial^2}{\partial x^2} \psi(x, t) &= -\frac{\partial^2}{\partial x^2} \sum_{n=1}^N c_n(t) f_n(x) = \sum_{n=1}^N c_n(t) \left(-\frac{\partial^2}{\partial x^2} f_n(x) \right) = \sum_{n=1}^N c_n(t) \lambda_n f_n(x) \Leftrightarrow \\ \sum_{n=1}^N i \frac{dc_n}{dt} f_n(x) &= \sum_{n=1}^N c_n(t) \lambda_n f_n(x) \Leftrightarrow \\ \sum_{n=1}^N \left\{ i \frac{dc_n}{dt} - \lambda_n c_n \right\} f_n(x) &= 0 \end{aligned}$$

Puisque les fonctions $\{f_n(x)\}$ sont une base, on serait tenté(e)s de déduire que

$$i \frac{dc_n}{dt} - \lambda_n c_n = 0$$

Mais cette déduction est fausse, car les $f_n(x)$ ne sont pas une base pour N fini !

Le raisonnement correct est de dire que

$$\begin{aligned} \sum_{n=1}^N i \frac{dc_n}{dt} f_n(x) &= \sum_{n=1}^N c_n(t) \lambda_n f_n(x) \Leftrightarrow \sum_{n=1}^N i \frac{dc_n}{dt} \int_0^1 dx f_n(x) f_m(x) = \\ \sum_{n=1}^N c_n(t) \lambda_n \int_0^1 dx f_n(x) f_m(x) &\Leftrightarrow \sum_{n=1}^N i \frac{dc_n}{dt} \delta_{nm} = \sum_{n=1}^N c_n(t) \lambda_n \delta_{nm} \Leftrightarrow \\ i \frac{dc_n}{dt} - \lambda_n c_n &= 0 \Leftrightarrow c_n(t) = c_n(0) e^{-i\lambda_n t} \end{aligned}$$

3 La méthode numérique

{mthnum}

L'implémentation à l'ordinateur exige que l'on travaille avec des sommes finies et que les coordonnées spatiales et temporelles soient des multiples entiers de pas fixes. Ces pas et le nombre fini de termes sont une "fiction" - la physique ne peut en dépendre et il faut contrôler que nos résultats n'en dépendent pas.

3.1 Tests de cohérence

{Tests}

Le code, qui est présenté en annexe, passe les tests de cohérence suivants :

- *Reconstruction de la condition initiale* : Un test de base est la représentation de la condition initiale, par la série des fonctions propres :

$$\psi(x, 0) = \sum_{n=1}^N c_n(0) f_n(x) \Leftrightarrow c_n(0) = \int_0^1 dx \psi(x, 0) f_n(x) \quad (18) \quad \{\text{initcondrec}\}$$

vu que l'intégrale est calculée de manière approchée (par la méthode des trapèzes ou Simpson). Les résultats de ces tests, comme fonctions de N , sont affichés dans la fig. 1.

- Un autre test de cohérence est que

$$\int_0^1 dx |\psi(x, t)|^2 = \text{const} < \infty \quad (19) \quad \{\text{unitarity}\}$$

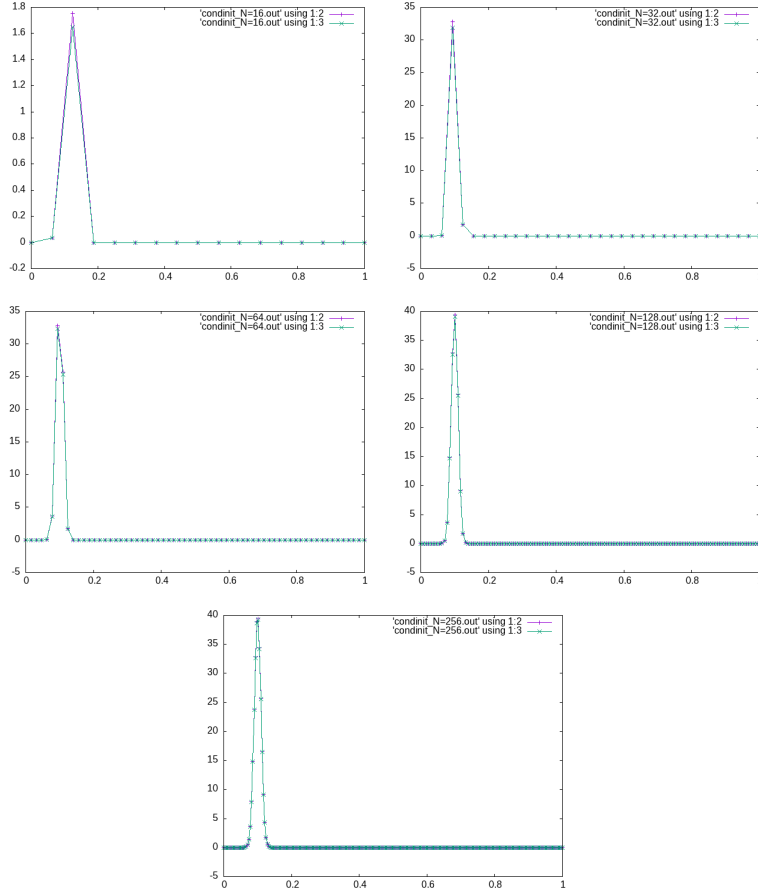


FIGURE 1 – Reconstruction de la condition initiale—une gaussienne—
par la somme des fonctions propres, pour $N = 16, 32, 64, 128$ et 256 . $\{\text{InitCondRec}\}$

pour que l'on puisse définir une densité de probabilité normalisée. La propriété que cette intégrale est une constante exprime le fait que le système est fermé de façon cohérente. En fait ce que l'on cherche à tester est que l'approximation numérique de cette intégrale, où l'intégrande est calculée avec un nombre fini, N , de termes, également, est indépendante du temps à l'approximation à laquelle on travaille, qui consiste à ne prendre qu'un nombre fini de termes dans la somme et de calculer l'intégrale par une méthode approchée dans l'espace.

Les résultats de ce calcul par le code sont affichés dans la fig. 2 pour une Gaussienne, centrée au milieu de l'intervalle et de largeur $\sigma = 0.01, 0.1$ et 0.5 .

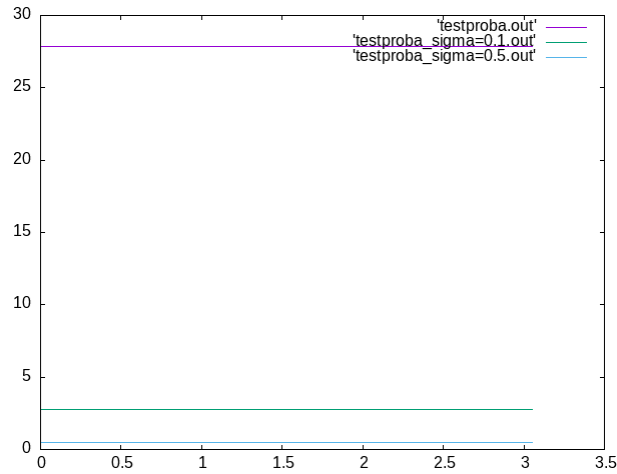


FIGURE 2 – On affiche $\int_0^1 dx |\psi(x, t)|^2$ pour trois conditions initiales qui sont des Gaussiennes, centrées au milieu de l'intervalle et de largeur $\sigma = 0.01, 0.1$ et 0.5 . On constate un résultat indépendant du temps, comme attendu.

{testproba}

Si l'on fait le calcul de façon analytique, on s'attend à trouver

que :

$$\begin{aligned}
\psi(x, t) &= \sum_{n=1}^N c_n(0) e^{-i\lambda_n t} f_n(x) \Rightarrow \\
|\psi(x, t)|^2 &= \sum_{n=1}^N c_n(0) e^{-i\lambda_n t} f_n(x) \sum_{m=1}^N c_m(0) e^{i\lambda_m t} f_m(x) \Rightarrow \\
\int_0^1 dx |\psi(x, t)|^2 &= \sum_{n,m=1}^N c_n(0) c_m(0) e^{i(\lambda_m - \lambda_n)t} \underbrace{\int_0^1 dx f_n(x) f_m(x)}_{\delta_{nm}} = \sum_{n=1}^N c_n(0)^2
\end{aligned} \tag{20} \quad \{\text{testnorme}\}$$

donc un résultat qui est, en fait, indépendant du temps—et ce calcul nous donne la valeur du dénominateur, donc du facteur de normalisation de la densité de probabilité.

4 Résultats

\{\text{res}\}

Le premier résultat est le “recouvrement” entre la condition initiale, $\psi(x, 0)$ et l’évolution dans le temps, $\psi(x, t)$, toutes les deux étant approchées par des sommes finies et par une intégration numérique :

$$\psi(x, t) = \sum_{n=1}^N c_n(0) e^{-i\lambda_n t} f_n(x) \tag{21} \quad \{\text{overlap}\}$$

Le recouvrement, $P(t)$, est donné par l’expression suivante

$$P(t) = \frac{\left| \int_0^1 dx \psi(x, 0)^* \psi(x, t) \right|^2}{\left| \int_0^1 dx |\psi(x, 0)|^2 \right|^2} \tag{22} \quad \{\text{overlap}\}$$

On commence par le cas le plus simple : La condition initiale est une fonction propre :

$$\psi(x, 0) = f_{n_0}(x) \tag{23} \quad \{\text{condinitfn}\}$$

Exercice 3. Montrer que, lorsque $\psi(x, 0) = f_{n_0}(x)$, alors

$$P(t) = 1$$

Démonstration.

□

Un test de cohérence de notre code est qu'il reproduit ce résultat. On affiche le résultat de l'expérience dans la fig. 3. Un cas plus

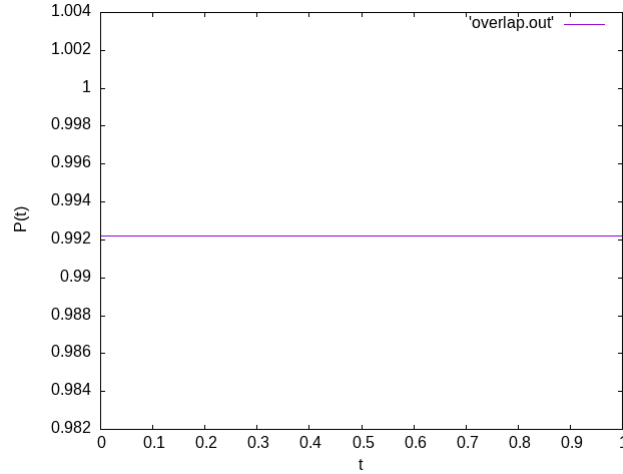


FIGURE 3 – Le recouvrement, $P(t)$, pour le cas, où la condition initiale est une fonction propre, dans ce cas, $\psi(x, 0) = f_1(x)$.

{overlap1}

intéressant est lorsque la condition initiale est une superposition linéaire de plus d'une fonctions propres du Hamiltonien. Une telle condition initiale est un état quantique, qui ne possède pas de limite classique.

Le cas le plus simple est celui d'une superposition de deux fonctions propres :

$$\psi(x, 0) = Af_{n_1}(x) + Bf_{n_2}(x) \quad (24) \quad \{\text{deuxfonpropres}\}$$

On trouve immédiatement que

$$\psi(x, t) = Ae^{-i\lambda_{n_1}t}f_{n_1}(x) + Be^{-i\lambda_{n_2}t}f_{n_2}(x) \quad (25) \quad \{\text{evoldeuxfonpropres}\}$$

Exercice 4. Calculer l'expression de $P(t)$ pour ce cas et montrer qu'elle est de la forme

$$P(t) = 1 - \frac{|A|^2|B|^2}{(|A|^2 + |B|^2)^2} \sin^2 \frac{\lambda_{n_1} - \lambda_{n_2}}{2} t \quad (26) \quad \{\text{Pdet2fpropres}\}$$

Expliquer comment cette expression permet d'expliquer les résultats numériques, affichés dans la fig. 4.

Démonstration. On calcule séparément numérateur et dénominateur pour $P(t)$.

Pour le dénominateur on trouve

$$D(P(t)) \equiv \left| \int_0^1 dx |\psi(x, 0)|^2 \right|^2 = (|A|^2 + |B|^2)^2 \quad (27) \quad \{\text{denom}\}$$

puisque les fonctions propres sont orthonormées.

Le calcul concret :

$$\begin{aligned} |\psi(x, 0)|^2 &= (Af_{n_1}(x) + Bf_{n_2}(x))(A^*f_{n_1}(x) + B^*f_{n_2}(x)) = \\ &|A|^2 f_{n_1}(x)^2 + |B|^2 f_{n_2}(x)^2 + 2(AB^* + A^*B)f_{n_1}(x)f_{n_2}(x) \Rightarrow \\ \int_0^1 dx |\psi(x, 0)|^2 &= |A|^2 \times 1 + |B|^2 \times 1 + 2(AB^* + A^*B) \times 0 = |A|^2 + |B|^2 \end{aligned}$$

Pour le numérateur, on calcule d'abord,

$$\begin{aligned} \psi(x, 0)^* \psi(x, t) &= (A^*f_{n_1}(x) + B^*f_{n_2}(x))(Ae^{-i\lambda_{n_1}t}f_{n_1}(x) + Be^{-i\lambda_{n_2}t}f_{n_2}(x)) = \\ &|A|^2 e^{-i\lambda_{n_1}t} f_{n_1}(x)^2 + |B|^2 e^{-i\lambda_{n_2}t} f_{n_2}(x)^2 + (A^*Be^{-i\lambda_{n_2}t} + AB^*e^{-i\lambda_{n_1}t})f_{n_1}(x)f_{n_2}(x) \end{aligned}$$

En remplaçant cette expression dans le numérateur de $P(t)$, on trouve qu'il prend la forme

$$N(P(t)) = ||A|^2 e^{-i\lambda_{n_1}t} + |B|^2 e^{-i\lambda_{n_2}t}|^2 = \quad (28) \quad \{\text{numerP}\}$$

et l'on en déduit l'expression pour $P(t)$ de l'éq. (26). $\square$

On cherche à comprendre ces résultats à partir des propriétés de l'expression (26).

On remarque que

$$\begin{aligned} P(t) = 1 &\Leftrightarrow \sin^2 \frac{\lambda_{n_1} - \lambda_{n_2}}{2} t = 0 \Leftrightarrow \frac{\lambda_{n_1} - \lambda_{n_2}}{2} t = 2m\pi \quad (m \in \mathbb{N}) \Leftrightarrow \\ T &= \frac{4\pi}{\lambda_{n_1} - \lambda_{n_2}} = \frac{4}{|n_1^2 - n_2^2|} \end{aligned} \quad (29) \quad \{\text{Period}\}$$

Car $P(t) = 1$ aussi bien aux périodes (m pairs) qu'aux demi-périodes (m impairs).

Cette quantité est calculée dans les “conventions” de notre travail analytique et de notre code numérique—par conséquent on devra pouvoir comparer sa valeur à celle calculée par le code.

La correspondance est le sujet de l'exercice suivant :

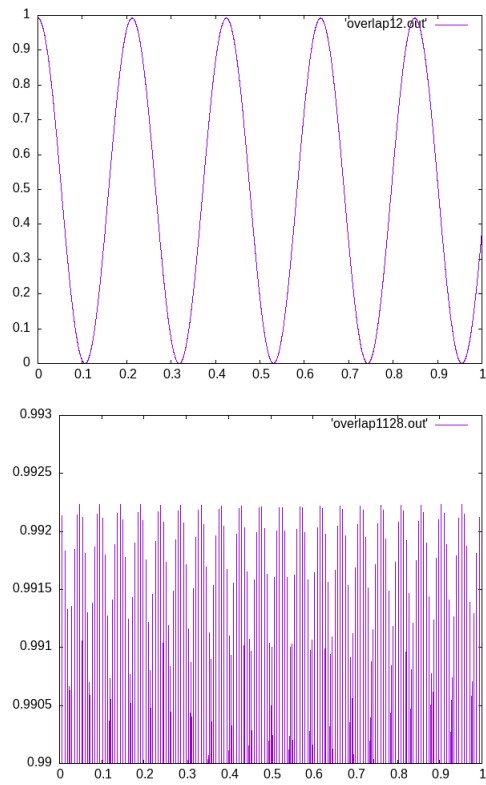


FIGURE 4 – Les cas $n_1 = 1, n_2 = 2$ et $n_1 = 1, n_2 = 128$.

{overlap1many}

{periodcode}

- Exercice 5.** 1. Dans le code on a pris $\Delta\tau = 1/N^2$ et $\tau = j\Delta\tau, j = 0, \dots, j + \delta j, \dots, T_{\max}$. Déterminer le pas (un entier), δj , pour que $j\Delta\tau$ soit le plus proche possible, à $2T$, avec T définie par l'éq. (29).
2. Discuter pourquoi le résultat de la question précédente donne la motivation de choisir le pas de temps de façon plus générale. Montrer que la relation entre le pas de temps, $\Delta\tau$, et le nombre d'itérations $T_{\max}$, utilisé dans le code, pour que l'on puisse afficher deux périodes, est la suivante

$$T_{\max} = \frac{2T}{\Delta\tau}$$

et afficher les résultats pour différentes valeurs de $\Delta\tau$.

Solution :

1.

$$2T = j_{\max}\Delta\tau = T_{\max}\Delta\tau$$

Pour les cas concrets, mentionnés plus haut :

$n_0 = 1, n_1 = 2$, on trouve $T = 4/3$ ce qui implique que $T_{\max} = [8N^2/3] + 1 = [2^{19}/3] + 1$ pas.

$n_0 = 1, n_1 = 128$, on trouve que $T = 4/(128^2 - 1) \approx 4/128^2 = 2^{2-14} = 2^{-12} = 1/4096$ ce qui implique que $T_{\max} = \text{pas}$.

2. Un avantage des méthodes pseudeospectrales est que les pas de temps ou de l'espace ne servent pas pour l'approximation des dérivées ; par conséquent il n'y a plus d'exigence qu'ils soient "petits" devant les valeurs de temps ou de l'espace.

On trouve, ainsi

5 Conclusions

{concl}

A Le paquet d'onde classique et le paquet d'onde quantique dans un domaine infini

{paquetsdominf}

Pour valider les résultats du code numérique, on doit pouvoir les comparer avec des calculs analytiques. L'exemple est celui d'un domaine infini, où la condition initiale est un paquet d'ondes Gaussien, c.à.d.

$$\psi(x, 0) \equiv F(x) = \frac{e^{\frac{(x-m)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} \quad (30) \quad \{\text{condinitG}\}$$

pour le cas quantique ; pour le cas classique, on a besoin d'une condition initiale supplémentaire, puisque l'équation d'ondes classique contient la deuxième dérivée par rapport au temps ; on choisit, alors,

$$\frac{\partial\psi}{\partial t}(x, 0) = 0 \quad (31) \quad \{\text{condinitG1}\}$$

Exercice 6. *Faire les calculs pour l'ze cas général, lorsque*

$$\frac{\partial\psi}{\partial t} = G(x) \quad (32) \quad \{\text{condinitG1nonzero}\}$$

On doit, bien entendu, imposer des conditions aux limites ; l'analogue des conditions aux limites Dirichlet, pour un domaine infini, est que

$$\lim_{|x| \rightarrow \infty} \psi(x, t) = 0 \quad (33) \quad \{\text{CauxBordsDirichlet}\}$$

On cherche, alors, à résoudre, en parallèle, l'équation de Schrödinger, pour la particule libre,

$$i \frac{\partial\psi}{\partial t} = - \frac{\partial^2\psi}{\partial x^2} \quad (34) \quad \{\text{Schroedlibre}\}$$

et l'équation d'ondes classiques,

$$\frac{\partial^2\psi}{\partial t^2} = \frac{\partial^2\psi}{\partial x^2} \quad (35) \quad \{\text{Ondeslibre}\}$$

L'approche sera la même que pour le domaine fini : On exprimera $\psi(x, t)$ comme une combinaison linéaire des fonctions propres de l'opérateur spatial, plus précisément, de l'opérateur $-\partial^2/\partial x^2$; dans le cas présent ce sont les fonctions $f_k(x) \equiv e^{ikx}$

Exercice 7. Démontrer que les $f_k(x)$ sont fonctions propres de $-\partial^2/\partial x^2$, c.à.d. que

$$-\frac{\partial^2}{\partial x^2} f_k(x) = \lambda_k f_k(x) \quad (36)$$

et trouver que la valeur de la valeur propre $\lambda_k = -k^2$.

Démonstration. □

et, par conséquent, on emploie la transformée de Fourier :

$$\psi(x, t) = \int_{-\infty}^{+\infty} \frac{dk}{\sqrt{2\pi}} e^{ikx} \hat{\psi}(k, t) \quad (37) \quad \{\text{FourierT}\}$$

On trouve deux équations, assez similaires, pour les coefficients-fonctions, $\hat{\psi}(k, t)$:

$$\frac{\partial \hat{\psi}(k, t)}{\partial t} = -ik^2 \hat{\psi}(k, t) \quad (38) \quad \{\text{Schroedpsihat}\}$$

pour le cas quantique et

$$\frac{\partial^2 \hat{\psi}(k, t)}{\partial t^2} = -k^2 \hat{\psi}(k, t) \quad (39) \quad \{\text{Ondespsihat}\}$$

pour le cas classique.

A.1 La solution pour le cas quantique

{\solquantique}

La solution du système infini d'équations (38) est donnée par l'expression

$$\hat{\psi}(k, t) = e^{-ik^2 t} \hat{\psi}(k, 0) \quad (40) \quad \{\text{solpsihat}\}$$

La transformée de Fourier de la condition initiale est donnée par l'expression

$$\hat{\psi}(k, 0) = \int_{-\infty}^{+\infty} \frac{dx}{\sqrt{2\pi}} e^{-ikx} F(x) \quad (41) \quad \{\text{psihatcondinit}\}$$

Exercice 8. Trouver l'expression concrète de $\widehat{\psi}(k, 0)$.

On peut, alors, déduire, après un calcul direct, sans difficultés conceptuelles, que

$$\psi(x, t) = C(t) e^{-\frac{(x-m(t))^2}{2(\sigma(t))^2}} \quad (42) \quad \{\text{solquantique}\}$$

Exercice 9. où $C(t)$ et $\sigma(t)$ sont des fonctions à valeurs complexes de t . Trouver leurs expressions.

Exercice 10. Contrôler que

$$\int_{-\infty}^{+\infty} dx |\psi(x, t)|^2 \quad (43) \quad \{\text{probtest}\}$$

qui, en principe, pouvait dépendre du temps, est, en fait, indépendante du temps et que sa valeur-finie-définit la normalisation de la densité de probabilité pour trouver la particule entre x et $x + dx$ à l'instant t .

Exercice 11. Calculer la quantité

$$P(t) \equiv \frac{\left| \int_{-\infty}^{+\infty} dx \psi(x, 0)^* \psi(x, t) \right|^2}{\left| \int_{-\infty}^{+\infty} |\psi(x, 0)|^2 \right|^2} \quad (44) \quad \{\text{Overlap}\}$$

B Le code

{code}

Le code est le suivant :

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <complex.h>

#define sqr(x) ((x)*(x))
#define x0 0.1
#define sigma 0.01
#define PI acos(-1.0)
```

```
#define L 1.0
```

```
void simpson(double s[], int N){//set the coefficients of the numerical integration
    int i;
    s[0] = 1./3.;
    for(i=1;i<=N;i=i+2){
        s[i]=4./3.;
    }
    for(i=2;i<N;i=i+2){
        s[i]=2./3.;
    }
    s[N+1]=1./3.;
}
```

```
void trapezes(double s[], int N){// set the coefficients of the numerical integration
    int i;
    s[0]=0.5;
    for(i=1;i<(N+1);i++){
        s[i]=1.0;
    }
    s[N+1]=0.5;
}
```

```
double integrate(double f[], double s[], double A, double B, int N){//numerical integration
    int i; double somme=s[0]*f[0];

    for(i=1;i<=N;i++){
        somme = somme + s[i]*f[i];
    }
    somme = somme + s[N+1]*f[N+1];
    return( ((B-A)/(N+1))*somme);
}
```

```
double valpropre(int n){// Valeurs propres de la seconde dérivée spatiale
```

```

    return(sqr(n*PI));}

double fonpropre(int n, double x){// fonctions propres de la seconde dérivée spatiale
    return(sqrt(2.0)*sin(n*PI*x)); // eigenfunctions for Dirichlet boundary conditions
}

double initialcondition(double x){// condition initiale
    return(exp(-sqr(x-x0)/(2.0*sqr(sigma)))/sqrt(2.0*PI*sqr(sigma))); //une gaussienne
    // return(eigenfunction(1,x)); //testing an eigenfunction as initial condition
}

main(){
    double complex somme;
    int n, l, j;
        int N=128;//pour trapèzes
    //    int N=257;//pour Simpson

double s[N+1],c[N+1],f[N+1],freal[N+1],fimag[N+1];

    //appel à la méthode d'intégration numérique
        trapezes(s,N);
    //simpson(s,N);

    //initialisation des coefficients
    for(n=0;n<=N;n++){
        c[n]=0.0; f[n]=0.0;
    }

    //on se donne la condition initiale, en donnant des valeurs non-nulles à certains

```

```

//Exercice 0: Tester la représentation de la condition initiale

double u, tau;
double Du = 1.0/N;
for(n=0;n<=N;n++){
    u=n*Du;
    f[n] = initialcondition(u);
    //      printf("%g\t%g\n",u,f[n]);
}

/* Exercice 1: Tester la représentation de la condition initiale par la série
*/

for(n=1;n<=N;n++){
    //Définir la fonction à intégrer
    for(j=0;j<=N;j++){
u = j*Du;
f[j] = initialcondition(u)*fonpropre(n,u);
    }
    //Calculer les coefficients
    c[n] = integrate(f,s,0.0,1.0,N);
}

for(j=0;j<=N;j++){
    u = j*Du;
    somme = 0.0;
    for(n=1;n<=N;n++){
somme = somme + c[n]*fonpropre(n,u);
    }
    //      printf("%g\t%g\t%g\n",u,initialcondition(u),somme);
}

double Dtau = 1.0/N;

```

```

    int Tmax=80000;

    for(j=0;j<=Tmax; j = j+10){ //boucle sur le temps

        tau = j*Dtau;

        for(l=0;l<=N;l++){ //boucle sur l'espace
u = l*Du;

        somme = 0.0;

        for(n=1;n<=N;n++){// la boucle sur la somme sur les modes
            somme = somme + c[n]*cexp(-I*valpropre(n)*tau)*fonpropre(n,u);

        }// on ferme la boucle sur la somme sur les modes

        freal[l] = creal(somme*initialcondition(u));
        fimag[l] = cimag(somme*initialcondition(u));

        // printf("%g\t%g\t%g\n",tau,u,sqr(cabs(somme)));

        } // on ferme la boucle sur l'espace
        printf("%g\t%g\n",tau,sqr(integrate(freal,s,0.0,1.0,N))+sqr(integrate(fimag,s,0.0,

        } // on ferme la boucle sur le temps

    }

```