

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	4
3.1	Tests de cohérence	4
4	Résultats	4
5	Conclusions	8
A	Le paquet d'onde classique et le paquet d'onde quantique dans un domaine infini	8
A.1	La solution pour le cas quantique	9
B	Le code	10

1 Introduction

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 :

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

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

2.1 Choix de coordonnées

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

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

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

2.3 La méthode 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 :

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

avec

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

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

$$\lambda_n = \frac{n^2 \pi^2}{L^2} \quad (5)$$

et les fonctions propres par l'expression

$$f_n(x) = A_n \sin n\pi \frac{x}{L} \quad (6)$$

Démonstration. □

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

$$\int_0^1 dx f_n(x) f_m(x) = \delta_{nm} \quad (7)$$

Exercice 2. Déterminer le facteur de normalisation, A_n .

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 :

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

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 fautive, car les $f_n(x)$ ne sont pas une base pour N fini !

Le raisonnement correct est de dire que

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

3 La méthode numérique

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

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 :

$$\psi(x, 0) = \sum_{n=1}^N c_n(0) f_n(x) \Leftrightarrow c_n(0) = \int_0^1 dx \psi(x, 0) f_n(x) \quad (8)$$

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

$$\int_0^1 dx |\psi(x, t)|^2 = \text{const} < \infty \quad (9)$$

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.

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

$$\psi(x, t) = \sum_{n=1}^N c_n(0) e^{-i\lambda_n t} f_n(x) \quad (10)$$

4 Résultats

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

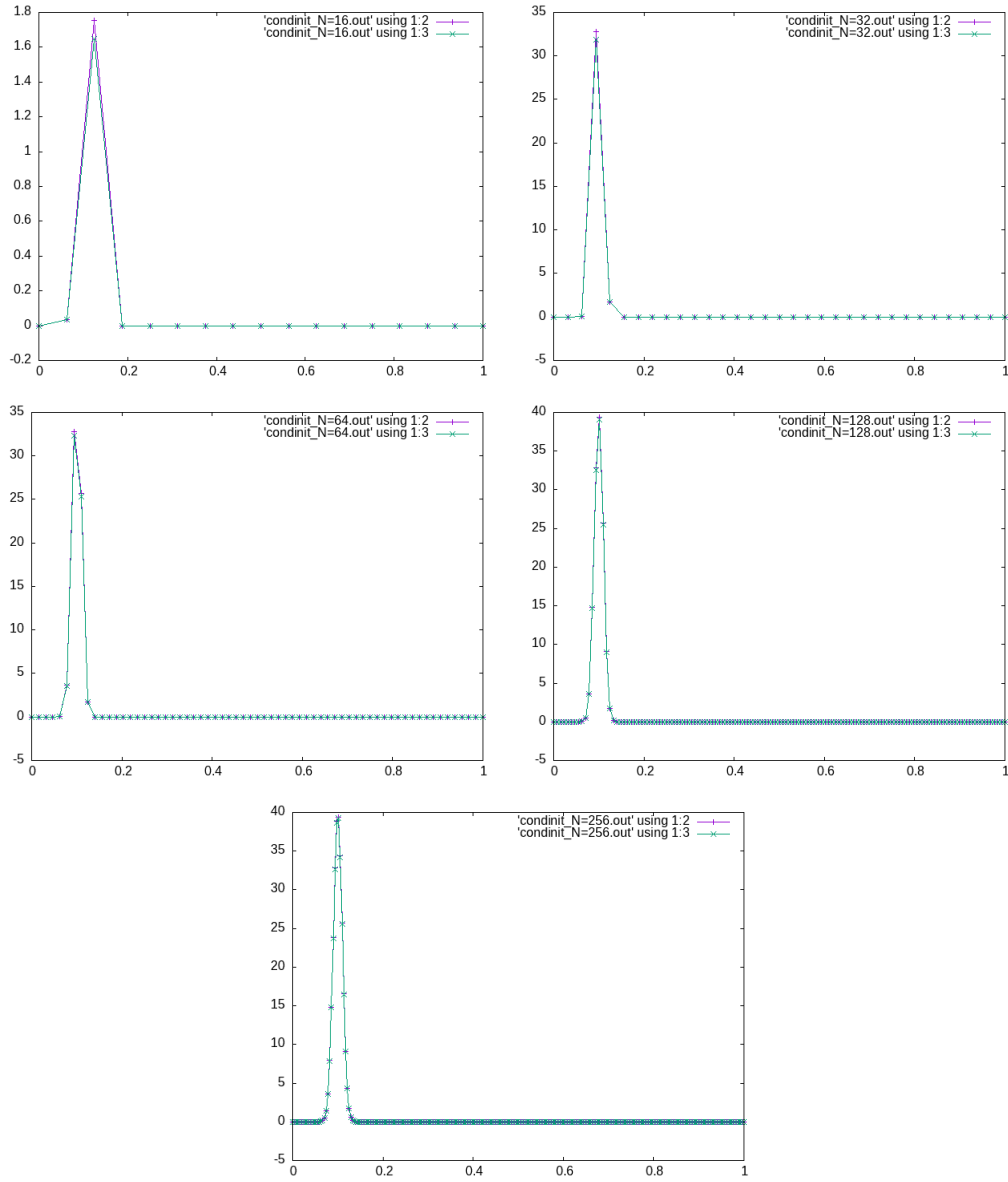


FIGURE 1 – Reconstruction de la condition initiale—une gaussienne—par la somme des fonctions propres, pour $N = 16, 32, 64, 128$ et 256 .

par des sommes finies et par une intégration numérique :

$$\psi(x, t) = \sum_{n=1}^N c_n(0) e^{-i\lambda_n t} f_n(x) \quad (11)$$

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

$$P(t) = \frac{\left| \int_0^1 dx, \psi(x, 0)^* \psi(x, t) \right|^2}{\left| \int_0^1 dx |\psi(x, 0)|^2 \right|^2} \quad (12)$$

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

$$\psi(x, 0) = f_{n_0}(x) \quad (13)$$

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. 2. Un cas plus 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 :

$$\psi(x, 0) = A f_{n_1}(x) + B f_{n_2}(x) \quad (14)$$

On trouve immédiatement que

$$\psi(x, t) = A e^{-i\lambda_{n_1} t} f_{n_1}(x) + B e^{-i\lambda_{n_2} t} f_{n_2}(x) \quad (15)$$

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

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

Expliquer comment cette expression permet d'expliquer les résultats numériques, affichés dans 3.

Démonstration.

□

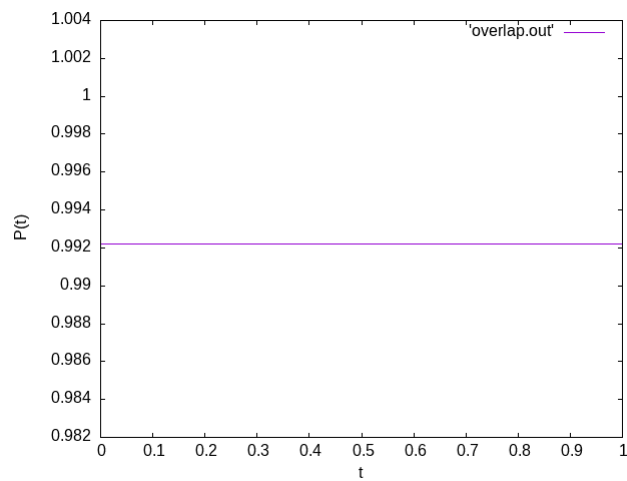


FIGURE 2 – Le recouvrement, $P(t)$, pour le cas, où la condition initiale est une fonction propre, dans ce cas, $\psi(x, 0) = f_1(x)$.

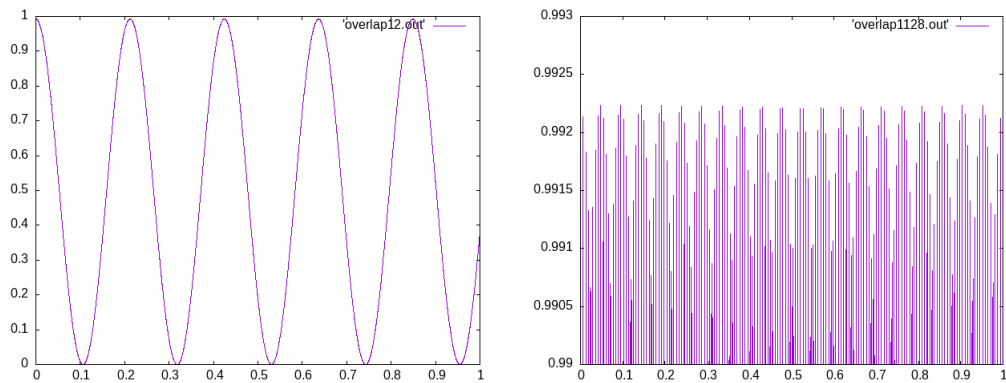


FIGURE 3 – Les cas $n_1 = 1, n_2 = 2$ et $n_1 = 1, n_2 = 128$.

5 Conclusions

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

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.

$$\psi(x, 0) \equiv F(x) = \frac{e^{-\frac{(x-m)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}} \quad (16)$$

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,

$$\frac{\partial\psi}{\partial t}(x, 0) = 0 \quad (17)$$

Exercice 5. *Faire les calculs pour l'cas général, lorsque*

$$\frac{\partial\psi}{\partial t} = G(x) \quad (18)$$

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

$$\lim_{|x| \rightarrow \infty} \psi(x, t) = 0 \quad (19)$$

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

$$i \frac{\partial\psi}{\partial t} = -\frac{\partial^2\psi}{\partial x^2} \quad (20)$$

et l'équation d'ondes classiques,

$$\frac{\partial^2\psi}{\partial t^2} = \frac{\partial^2\psi}{\partial x^2} \quad (21)$$

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

$$-\frac{\partial^2}{\partial x^2} f_k(x) = \lambda_k f_k(x) \quad (22)$$

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

Démonstration. □

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

$$\psi(x, t) = \int_{-\infty}^{+\infty} \frac{dk}{\sqrt{2\pi}} e^{ikx} \hat{\psi}(k, t) \quad (23)$$

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

$$\frac{\partial \hat{\psi}(k, t)}{\partial t} = -ik^2 \hat{\psi}(k, t) \quad (24)$$

pour le cas quantique et

$$\frac{\partial^2 \hat{\psi}(k, t)}{\partial t^2} = -k^2 \hat{\psi}(k, t) \quad (25)$$

pour le cas classique.

A.1 La solution pour le cas quantique

La solution du système infini d'équations (24) est donnée par l'expression

$$\hat{\psi}(k, t) = e^{-ik^2 t} \hat{\psi}(k, 0) \quad (26)$$

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

$$\hat{\psi}(k, 0) = \int_{-\infty}^{+\infty} \frac{dx}{\sqrt{2\pi}} e^{-ikx} F(x) \quad (27)$$

Exercice 7. Trouver l'expression concrète de $\hat{\psi}(k, 0)$.

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

$$\psi(x, t) = C(t) e^{-\frac{(x-m(t))^2}{2(\sigma(t))^2}} \quad (28)$$

Exercice 8. où $C(t)$ et $\sigma(t)$ sont des fonctions à valeurs complexes de t . Trouver leurs expressions.

Exercice 9. Contrôler que

$$\int_{-\infty}^{+\infty} dx |\psi(x, t)|^2 \quad (29)$$

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 10. Calculer la quantité

$$P(t) \equiv \frac{\left| \int_{-\infty}^{+\infty} dx \psi(x, 0)^* \psi(x, t) \right|^2}{\left| \int_{-\infty}^{+\infty} |\psi(x, 0)|^2 \right|^2} \quad (30)$$

B Le 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 integrat
    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 in
    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 spati
    return(sqrt(2.0)*sin(n*PI*x)); // eigenfunctions for Dirichlet boundary conditio
}

```

```

double initialcondition(double x){// condition initiale
    return(exp(-sqr(x-x0)/(2.0*sqr(sigma)))/sqrt(2.0*PI*sqr(sigma))); //une gaussien
    // 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 certa

    //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,0.0,1.0,N)));

} // on ferme la boucle sur le temps

}

```