



Université Mohamed Khider de Biskra
Faculté des sciences exactes et des sciences de la nature et de la vie
Département des sciences de la matière

MÉMOIRE DE MASTER

En vue de l'obtention d'un diplôme Startup / Projet innovant régi par l'arrêté ministériel n° 1275

Sciences de la matière
Chimie
Chimie des matériaux

Réf. : /

Présenté et soutenu par :

Bahri Lynda

Le : 15/06/2023

Conception d'une interface interactive dédiée aux calculs de Chimie Quantique

Jury :

M.	Makhloufi Rachid	MCA	Université M. Khider de Biskra	Président
M.	Kenouche Samir	MCA	Université M. Khider de Biskra	Rapporteur
M.	Chadli Abdelhakim	MCA	Université M. Khider de Biskra	Examineur

Année universitaire : 2022/2023

Remerciements

Au nom de Dieu Le Plus Clément et Le Plus Miséricordieux.

Tout d'abord, je remercie ALLAH le Tout Puissant qui m'a accordé la volonté et le courage pour mener à bien ce mémoire dans le cadre d'un projet Start-up. Mes plus sincères remerciements vont au Dr. KENOUCHE Samir, pour sa disponibilité, pour ses compétences en programmation informatique, ses connaissances en chimie quantique et pour m'avoir soutenu tout au long de ce travail.

Merci au Dr. Makhloufi Rachid, pour avoir accepté de présider le jury de ce mémoire. Je tiens également à remercier vivement Dr. Chadli Abdelhakim, pour avoir bien voulu participer au jury de ce mémoire et accepter d'être examinateur. Je voudrais également exprimer toute ma reconnaissance à Mlle Chabbi Anhar pour son aide précieuse dans la conception du BMC. Un grand merci à ceux qui ont été à mes côtés et à ceux qui ont cru en moi.

Avant-propos

Ce mémoire de fin d'étude s'inscrit dans le cadre d'un diplôme universitaire **Start-up/Projet Innovant**. Les modalités de préparation de ce projet de mémoire sont régies par l'Arrêté ministériel n°1275 du 27 septembre 2022. Cette démarche s'articule dans le sillage des efforts des hautes autorités de l'état pour promouvoir ce type de projet afin de renforcer le tissu économique national. A cet égard, le programme **Algeria Disrupt** fixe les aspects juridiques et réglementaires dans la perspective de promouvoir les Start-ups et de soutenir leur croissance. Par ailleurs, ce programme intègre aussi un fonds d'investissement publique destiné à ces Start-ups. Le label **Projet Innovant** est destiné aux porteurs de projets n'ayant pas encore créé leur société. A travers ce présent projet, nous souhaitons concevoir une interface interactive, conviviale et intuitive, couvrant les calculs usuels inhérents à la Chimie Quantique. Cette interface utilisateur/machine offrira un environnement intégré permettant de faciliter la compréhension des concepts théoriques (modèles quantiques usuels, calculs des probabilités de présence d'un électron, calcul et visualisation des orbitales atomiques, simulation des états stationnaires et de la loi de Planck ... etc.) abordés dans cette discipline durant les deux premiers cycles universitaires. La conception de cette interface est motivée par la volonté de mettre à la disposition des étudiants, un certain nombre d'outils pratiques afin de stimuler leurs connaissances opérationnelles. Nous envisageons de développer cette application au moyen du logiciel Matlab[®]. Ce logiciel est très puissant et offre plusieurs possibilités afin de mettre au point ce type d'interface. L'idée de base consiste à créer plusieurs interfaces pour ensuite les regrouper afin de mettre au point l'application. Par conséquent, l'interface graphique constituera la brique élémentaire de ce projet. Cette application, utilisé dans une démarche d'apprentissage, servira comme un outil pédagogique intégré dans un mécanisme de formation, aux seins notamment des établissements universitaires.

Une interface interactive, comme son nom l'indique constitue un affichage graphique qui permet à l'utilisateur de réaliser des opérations interactives en cliquant simplement sur des boutons. Des résultats numériques et/ou graphique sont renvoyés à l'issue de cette interaction utilisateur/interface. L'avantage d'utiliser ce type d'interface est la facilitation de la com-

préhension des différents concepts théoriques sous-adjacents. De plus, l'utilisateur n'est pas astreint de maîtriser un langage de programmation ni d'introduire des commandes plus au moins compliquées pour utiliser cette interface. Cette dernière est entièrement autonome et est conçue afin de simplifier au maximum l'interaction utilisateur/interface, en automatisant toutes les opérations de calcul. Dans le logiciel Matlab[®], l'outil de conception pour ce type d'interface est appelée **GUIDE** (Graphical User Interface Development Environment). Cet environnement graphique offre les outils nécessaires pour concevoir des interfaces utilisateur et créer des applications personnalisées.

Le logiciel Matlab[®] présente plusieurs avantages et est devenu un outil de calcul scientifique incontournable aussi bien en sciences fondamentales qu'en ingénierie. Les programmes (ou scripts) peuvent être utilisés comme une sorte de bloc-notes pour évaluer des expressions saisies sur la ligne de commande, ou pour exécuter des scripts volumineux écrits manuellement. Les interfaces peuvent être écrites et modifiées à l'aide de l'environnement de développement intégré et déboguées à l'aide du débogueur. La simplicité d'utilisation du langage en fait un outil idéal pour développer des calculs de différentes natures (numérique et symbolique). De nombreux outils de développement sont pris en charge pour faciliter l'utilisation du programme. Ils comprennent un Éditeur/Débogueur intégré, une documentation et des manuels en ligne, un navigateur d'espace de travail ainsi que des démonstrations complètes. Ce logiciel est pris en charge par différents systèmes d'exploitation, ce qui lui confère une grande indépendance vis-à-vis des plateformes. Le langage est disponible sous Windows[®], Linux[®], diverses versions d'Unix[®] et Macintosh[®]. Par conséquent, les programmes écrits dans Matlab[®] peuvent être transférés facilement vers de nouvelles plateformes, selon les besoins de l'utilisateur. Matlab[®] dispose d'une myriade de boîtes à outils ayant des fonctions prédéfinies. Ces boîtes à outils servent à résoudre des problèmes complexes dans des domaines particuliers et les fonctions prédéfinies fournissent des solutions testées et prêtes à l'emploi pour de nombreux calculs. Ces dernières sont intégrées directement dans le langage Matlab[®], ce qui rend le travail beaucoup plus convivial. En outre, un compilateur Matlab[®] séparé est disponible. Ce compilateur peut compiler les programmes en un véritable exécutable qui s'exécute plus rapidement que le code interprété. Il s'agit d'une méthode efficace pour convertir une interface en un exécutable adapté à la vente et à la distribution aux utilisateurs. Toutefois, le principal inconvénient de ce logiciel tient au fait qu'il s'agit d'un langage interprété et qu'il peut donc s'exécuter plus lentement qu'un langage compilé. Ce problème peut être résolu en structurant correctement les scripts.

Ce mémoire est scindé en deux parties distinctes. La première partie comporte trois chapitres. Le premier chapitre est dédié à la présentation, par des exemples concrets, des diverses syntaxes de Matlab[®]. Les différentes notions de chimie quantique intégrées dans cette application, sont traitées en détail au deuxième chapitre. Le dernier chapitre est destiné à la présentation et à la description détaillée des diverses interfaces développées dans le cadre de ce projet Start-up. La deuxième partie est consacrée à la description du BMC (Business Model Canvas) adossé au projet en question.

Table des matières

1	Programmer avec Matlab®	2
1.1	Résolution d'équations non-linéaires	3
1.2	Intégration numérique	6
1.3	Interpolation polynomiale	8
1.4	Équations différentielles ordinaires	10
1.4.1	Résolution formelle des équations différentielles	12
1.5	Conclusion	14
2	Notions théoriques intégrées dans l'application	15
2.1	Loi de Planck	15
2.2	Équation de Schrödinger dépendante du temps	18
2.2.1	Particule dans une boîte	20
2.2.2	Oscillateur harmonique	27
2.3	Orbitales atomiques	29
2.4	Représentation des particules en paquets d'onde	32
2.5	Conclusion	37
3	Présentation et description des interfaces graphiques	38
3.1	Description des interfaces graphiques développées	40
3.2	Interface principale	40
3.3	Interface de la loi de Planck	41
3.4	Interface de la particule dans une boîte	47
3.4.1	Cas unidimensionnel	47
3.4.2	Cas bidimensionnel	48
3.5	Interface de l'état stationnaire	53
3.6	Interface de l'oscillateur harmonique	56
3.7	Interface des orbitales atomiques	59
3.8	Interface des fonctions d'ondes radiales	65
3.9	Conclusion générale	66

Sommaire

1.1	Résolution d'équations non-linéaires	3
1.2	Intégration numérique	6
1.3	Interpolation polynomiale	8
1.4	Équations différentielles ordinaires	10
1.4.1	Résolution formelle des équations différentielles	12
1.5	Conclusion	14

Matlab® est l'abréviation de **Matrix laboratory**. Il a été développé par la société MathWorks <https://www.mathworks.com/>. Il s'agit d'un langage de programmation polyvalent particulièrement efficace pour le calcul numérique. Il a été conçu pour réaliser des calculs scientifiques de différentes natures. Il intègre les calculs numérique et symbolique, la visualisation et la programmation dans un environnement convivial où les problématiques sont formulées selon une notation mathématique familière. Ce logiciel est doté d'un environnement interactif dont l'élément de données de base est un tableau qui n'a pas besoin d'être dimensionné. Cette structure de données permet de résoudre de nombreux problèmes de calcul technique, en particulier ceux qui comportent des formulations matricielles et vectorielles. Matlab® a énormément évolué au fil des ans grâce notamment aux contributions de nombreux utilisateurs. En outre, les fonctions intégrées ou prédéfinies (ou encore commandes) du logiciel offrent des ressources de premier ordre pour effectuer des calculs, notamment l'optimisation, l'algèbre linéaire, l'analyse de données, la résolution numérique d'équations différentielles, le traitement du signal et bien d'autres tâches mathématiques. Aux sein des universités, il s'agit de l'outil pédagogique incontournable pour les cours avancés en mathématiques, en ingénierie et en sciences fondamentales. Dans le domaine technologique, il est l'outil de choix pour la recherche et le développement.

Ce chapitre est destiné à la présentation, par des exemples d'application concrets, les diverses syntaxes de Matlab®. La plupart des notions mathématiques décrites ici sont intégrées dans l'application que nous souhaitons développer. Nous avons jugé plus judicieux d'adopter cette approche au lieu de se focaliser sur des notions générales qui ne sont pas pertinentes dans le cas présent. Les fondements théoriques de chaque méthode mathématique sont présentés succinctement, en favorisant ainsi l'aspect opérationnel en écrivant des scripts Matlab® avec des annotations très détaillées.

1.1 Résolution d'équations non-linéaires

Matlab® dispose de fonctions prédéfinies destinées à chercher les zéros d'une équation. On testera dans un premier temps la commande `fzero`. Cette commande permet de trouver un point où la fonction change de signe. Si la fonction est continue, il s'agit également d'un point où la fonction a une valeur proche de zéro. Si la fonction n'est pas continue, `fzero` peut renvoyer des valeurs qui sont des points discontinus au lieu de zéros. La syntaxe de cette commande est :

```
1 [x , fval, exitflag, output] = fzero(fun, x0, options)
```

Cette commande cherche un zéro autour de la valeur initiale `x0`. La racine approchée `x` renvoyée est près d'un point où la fonction `fun` change de signe. Cette commande définit un zéro comme un point où la fonction croise l'axe des x . La commande renvoie NaN (Not a Number) dans le cas où la recherche de la racine a échoué. Les points où la fonction touche l'axe des x sans le croiser ne sont pas des zéros valides. Par exemple, $y = x^2$ est une parabole qui touche l'axe des x à 0. Cependant, comme la fonction ne traverse jamais l'axe des abscisses, aucun zéro n'est trouvé. Pour les fonctions sans zéros valides, `fzero` s'exécute jusqu'à ce que Inf, NaN ou une valeur complexe soit détectée. Par ailleurs, il est aussi possible de spécifier un intervalle dans le quel `fzero` cherchera la racine. Toutefois, il faut s'assurer que l'équation change de signe dans cet intervalle. L'argument `options` est de type `structure` qui recèle les options d'optimisation indiquées dans `optimset`. Ci-dessous, un exemple d'application :

```
1 clear all ;
2 clc ;
3 % Trouver la racine de f ENTRE LES BORNES lB (lowerBound) ET uB (upperBound)
4
5 fun = @(x) x - 0.2.*sin(4.*x) - 1/2 ; % FONCTION
6 lB = -2 ; uB = 4 ; % BORNES
7 opts = optimset('Display','iter','FunValCheck','on','TolX',1e-6) ;
8 % PARAMETRES EN OPTION
9
10 [racine, funEval, exitTest, output] = fzero(fun, [lB uB], opts) ;
11 % LA SORTIE RACINE EST LA SOLUTION RECHERCHEE
```

Ci-dessous, les résultats affichés par ce script :

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AFFICHAGE PAR DEFAUT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  Func-count      x          f(x)
3      2           -2          -2.30213
4      3           0.357245      -0.340747
5      4           0.730969       0.187769
6      5           0.598194      -0.0379606
7      6           0.620523      -0.00202224
8      7           0.621766      8.20089e-06
9      8           0.621761     -6.06744e-09
10     9           0.621761     -6.06744e-09
11
12 Zero found in the interval [-2, 4]
13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14     racine      =
15                 0.6218
16     funEval     =
17                 -6.0674e-09
18     exitTest    =
19                 1
20     output     =
21                 intervaliterations: 0
22                 iterations: 7
23                 funcCount: 9
24                 algorithm: 'bisection, interpolation'
25                 message: [1x34 char]

```

La sortie `exitTest = 1` signifie que l'algorithme a convergé vers la solution approchée. Une valeur de `exitTest < 0` indiquerai plutôt une divergence de l'algorithme. L'argument `funEval = -6.0674e-09` donne l'évaluation de l'équation à la dernière itération, c'est-à-dire à la racine approchée. Les sorties renvoyées par `output` sont `funcCount = 9` représentant le nombre d'évaluations de l'équation et `intervaliterations` est le nombre d'itérations pour trouver un intervalle contenant la racine approchée. Le champ `iterations = 7` renvoie le nombre d'itérations conduisant à la racine approchée et `message` renvoie la chaîne de caractères : 'Zero found in the interval [-2, 4]'

La sortie `algorithm` renvoie l'algorithme utilisé pour résoudre cette équation qui est 'bisection'. Le champ de la structure `opts` spécifié par `optimset('Display', 'iter', ...)` affiche des détails à chaque itération. Le champ `optimset(..., 'FunValCheck', 'on', ...)` contrôle si les valeurs de l'équation sont réelles et affiche dans le cas contraire un avertissement quand `fzero` renvoie une valeur complexe ou NaN. Le champ `optimset(..., 'TolX', 1e-6)` désigne la tolérance avec laquelle est calculée la racine. Comme deuxième application, nous

souhaitons appliquer la commande `fsolve` pour résoudre le système d'équations non-linéaires suivant :

$$\begin{cases} 2x_1 - x_2 = \exp(-x_1) \\ -x_1 + 2x_2 = \exp(-x_2) \end{cases} \quad (1.1)$$

Pour résoudre ce système d'équations, nous commençons d'abord par écrire un script *M-file*, selon :

```
1 function fun = myfunction(x)
2
3 fun = [2*x(1) - x(2) - exp(-x(1));
4       -x(1) + 2*x(2) - exp(-x(2))];
5 return
```

Bien évidemment, ce fichier doit être absolument sauvegardé dans le répertoire courant, avec le nom `myfunction.m`, sinon Matlab® ne reconnaitra pas ce système d'équations. Une fois cette étape achevée, on écrira le script principal comme suit :

```
1 clear all ; clc ;
2
3 xinit = [-5 ; -3];
4 opts = optimset('NonlEqnAlgorithm','lm','LargeScale','off','Display','iter',
5               'TolX',1e-6,'TolFun',1e-6);
6 [x, fval, exitflag, output, jacobian] = fsolve(@myfunction, xinit, opts) ;
```

Le champ `optimset('NonlEqnAlgorithm','lm', ...)` impose l'utilisation de l'algorithme de Levenberg-Marquardt. En remplaçant la chaîne de caractères `'lm'` par `'gn'`, cela signifie qu'on utilisera l'algorithme de Gauss-Newton. Le dernier argument en entrée (`..., 'TolFun', 1e-6`) est la tolérance finale. Il existe des commandes comparables à l'instar de `fminbnd` et `fminsearch`. La syntaxe de ces deux commandes est présentée ci-dessous.

```
1 clc ;
2 clear all ;
3
4 fun = @(x) x.^3-2.*x-5 ; xmin = -2 ; xmax = 2 ;
5
6 [racine, fval, exitflag] = fminbnd(fun, xmin, xmax, ...
7   optimset('TolX',1e-12,'Display','off')) ;
8
9 % LA RACINE OBTENUE EST racine = 0.8165
```

La commande `fminbnd` minimise une fonction d'une seule variable sur un intervalle fixe. La fonction à minimiser doit être continue, et `fminbnd` trouve uniquement des solutions locales. De plus, elle présente souvent une convergence lente lorsque la solution se trouve sur une frontière de l'intervalle, et ne traite que des variables réelles.

```

1 clc ;
2 clear all ;
3
4 xinit = 1.5 ;
5
6 x = fminsearch(@(x)sin(x.^2), xinit, optimset('TolX',1e-8))
7
8 % SOLUTION x = 2.1708

```

La commande `fminsearch` peut également trouver le minimum d'une fonction scalaire de plusieurs variables, à partir d'une estimation initiale. On parle généralement d'optimisation non linéaire sans contrainte. Cette commande peut souvent gérer des discontinuités, en particulier si ces dernières ne se produisent pas à proximité de la solution, et elle ne minimise que sur les nombres réels. Lorsqu'il y a des variables complexes, elles doivent être divisées en parties réelles et imaginaires. Il convient de souligner que les deux commandes `fminbnd` et `fminsearch` sont particulièrement dépendante de la valeur initiale, ce qui rend leur utilisation très sensible.

1.2 Intégration numérique

Il existe des commandes Matlab® permettant de résoudre numériquement des intégrales simple, double et triple. Ces commandes sont les suivantes : `quad`, `quadl`, `quadgk` et `quadv`. La syntaxe usuelle de la commande `quad` est la suivante :

```

1 [q, fcnt] = quad(fun, lowerB, upperB, Tol)

```

Cette commande évalue numériquement l'aire (intégrale) sous la courbe de la *fonction anonyme* `fun`. Cette dernière peut également être écrite comme une *fonction M-file*. L'algorithme d'intégration de cette commande est basé sur la méthode de *Simpson adaptative* (adaptive Simpson quadrature). Les arguments en entrée `lowerB` et `upperB` désignent respectivement les bornes inférieure et supérieure de l'intégrale. L'argument `Tol` désigne la tolérance considérée relative à l'erreur d'intégration. Par défaut, `Tol` vaut `1e-06`. L'argument de sortie `q` représente la valeur de l'intégrale calculée. La sortie `fcnt` exprime le nombre d'évaluation de la fonction à intégrer. Les commandes `quadl` (adaptive Lobatto quadrature) et `quadv` (Vectorized quadrature) présentent une syntaxe similaire à celle de `quad`. Par ailleurs, la commande `quadgk` (adaptive Gauss-Kronrod quadrature) accepte d'autres arguments optionnels selon la syntaxe :

```
1 [q, errbnd] = quadgk(fun, lowerB, upperB, 'param1',val1, 'param2',val2, ...)
```

La sortie `errorbnd` exprime l'erreur absolue d'intégration $|Q - I|$, avec Q et I sont respectivement les valeurs approchée et exacte de l'intégrale. Parmi les arguments optionnels, on citera, `'AbsTol'` (Absolute error tolerance) dont la valeur par défaut vaut $1e-10$ et `'RelTol'` (Relative error tolerance) dont la valeur par défaut vaut $1.e-6$. La commande `dblquad` permet de calculer numériquement une intégrale double. Sa syntaxe usuelle est donnée par :

```
1 [int, evalFun] = dblquad(fun, xmin, xmax, ymin, ymax, tol)
```

L'argument de sortie `int` constitue la valeur de l'intégrale calculée. Le deuxième argument de sortie `evalFun` désigne le nombre d'évaluation de la fonction. La commande `dblquad` évalue l'intégrale double `fun(x,y)` sur le rectangle défini par `xmin <= x <= xmax` et `ymin <= y <= ymax` avec la tolérance considérée `tol`. Par défaut, la valeur de cette tolérance vaut : 10^{-6} . L'argument `fun` est une *fonction handle*. Ci-dessous un exemple d'application :

```
1 clc ;
2 clear all ;
3 % INTEGRALE DOUBLE
4
5 xmin = pi ; xmax = 2*pi ; % DISCRETISATION SELON X
6 ymin = 0 ; ymax = pi ; % DISCRETISATION SELON Y
7 tol = 10e-8 ;
8
9 fun = @(x,y) y.^2.*sin(x) + x.^2.*cos(y) ; % FONCTION A INTEGRER
10
11 int = dblquad(fun, xmin, xmax, ymin, ymax, tol) % CALCUL
12
13 % RESULTAT int = -20.6709
```

De la même façon, la commande `triplequad` sert à résoudre des intégrales triples en utilisant la syntaxe suivante :

```
1 int = triplequad(fun,xmin,xmax,ymin,ymax,zmin,zmax,tol,method)
```

Cette commande évalue numériquement l'intégrale triple `fun(x,y,z)` sur un rectangle tridimensionnel défini par `xmin <= x <= xmax`, `ymin <= y <= ymax` et `zmin <= z <= zmax` avec la tolérance considérée `tol`. L'argument `fun` est une *fonction handle*. Comme pour les intégrales doubles, l'entrée `method` admet comme valeur : `method = @quad` prise par défaut ou bien `method = @quadl`. Ci-dessous un exemple d'application :

```

1 clc ;
2 clear all ;
3 % INTEGRALE TRIPLE
4
5 xmin = 0 ; xmax = pi ; ymin = 0 ; ymax = pi ; zmin = -1 ; zmax = 1 ;
6 tol = 10e-8 ;
7 fun = @(x,y,z) z.^2.*sin(x) + z.^2.*cos(y) ; % FONCTION A INTEGRER
8
9 int = triplequad(fun, xmin, xmax, ymin, ymax, zmin, zmax, tol) % CALCUL
10
11 % RESULTAT int = 4.1888

```

Signalons par ailleurs, que ces intégrales peuvent également être résolues en se servant de la boîte à outil : Symbolic Math Toolbox™ selon le script suivant :

```

1 clc ;
2 clear all ;
3 syms x y z real % DECLARATION DES VARIABLES SYMBOLIQUES
4
5 fun1 = y^2*sin(x) + x^2*cos(y) ; % INTEGRALE DOUBLE
6 int2Val = int(int(fun1, x, pi, 2*pi), y, 0, pi) ;
7 int2Val = double(int2Val)
8
9 fun2 = z^2*sin(x) + z^2*cos(y) ; % INTEGRALE TRIPLE
10 int3Val = int(int(int(fun2, x, 0, pi), y, 0, pi), z, -1, 1) ;
11 int3Val = double(int3Val)

```

Les résultats renvoyés :

```

1 int2Val = -20.6709 % INTEGRALE DOUBLE
2 int3Val = 4.1888 % INTEGRALE TRIPLE

```

1.3 Interpolation polynomiale

Matlab® offre également la possibilité de réaliser des interpolations (polynomiales, trigonométriques ... etc.), au moyen de fonctions préprogrammées (ou commandes). Parmi ces commandes, on citera : `interp1`, `interp1q`, `interpft`, `interp2`, `interp3`, `interp`, `pchip`, `spline`, `griddata`. Nous implémentons la commande `interp1` qui réalise une interpolation en une dimension. Cette commande peut être utilisée avec plusieurs méthodes d'interpolation. Sa syntaxe usuelle est :

```
1 yi = interp1(x, y, xi, method).
```

Les arguments en entrée sont x , représentant les nœuds d'interpolation et y est la fonction évaluée aux nœuds x . Ces deux entrées doivent avoir la même dimension et peuvent être des scalaires, des vecteurs ou des matrices. Les xi sont les abscisses de l'interpolation et la sortie yi renvoie les valeurs d'interpolation aux points xi . L'argument `method` exprime la méthode d'interpolation considérée, à savoir : `'linear'` (Linear interpolation), `'cubic'` (Cubic interpolation), `'spline'` (Cubic spline interpolation) et `'nearest'` (Nearest neighbor interpolation). Le script ci-dessous, met en exergue une comparaison entre ces différentes méthodes d'interpolation. Comme application, au moyen de la commande `interp1`, on désire réaliser l'interpolation des points générés par la fonction suivante :

$$\begin{cases} f(x) = \sin(x) - x^3 \\ \text{Avec } x \in [-10, 10] \end{cases} \quad (1.2)$$

On prendra $n = 20$, ce qui génère $n + 1$ nœuds d'interpolation définis sur l'intervalle $[-10, 10]$. Ci-dessous le script Matlab® :

```
1 clear all ;
2 close all ;
3 clc ;
4 % INTERPOLATION
5
6 lowerBound = -10 ; upperBound = 10 ; n = 20 ;
7 step = (upperBound-lowerBound)/n ;
8 xNoeuds =lowerBound :step: upperBound ;
9
10 fun = @(xNoeuds) sin(xNoeuds) - xNoeuds.^3;
11 xInterp = lowerBound : step/2 : upperBound ;
12
13 interpfun1 = interp1(xNoeuds, fun(xNoeuds), xInterp,'linear') ;
14 interpfun2 = interp1(xNoeuds, fun(xNoeuds), xInterp,'cubic') ;
15 interpfun3 = interp1(xNoeuds, fun(xNoeuds), xInterp,'spline') ;
16 interpfun4 = interp1(xNoeuds, fun(xNoeuds), xInterp,'nearest') ;
17
18 figure('Color', [1 1 1]) ; hold on ; plot(xInterp, interpfun1,'b') ;
19 hold on ; plot(xInterp, interpfun2,'m') ; hold on ;
20 plot(xInterp, interpfun3,'k') ; hold on ; plot(xInterp, interpfun4,'g') ;
21
22 hold on ; plot(xNoeuds, fun(xNoeuds),'rx','MarkerSize',8,'LineWidth',1) ;
23 hold on ; plot(xNoeuds, fun(xNoeuds),'ro','MarkerSize',8,'LineWidth',1)
24 legend('linear','cubic','spline','nearest') ;
```

1.4 Équations différentielles ordinaires

Ce logiciel dispose d'un certain nombre de *solveurs* destinés à la résolution d'équations différentielles. Les plus utilisés sont `ode23` et `ode45`, qui sont basés sur la méthode de *Runge-Kutta* explicite à un pas. Le *solveur* `ode113`, utilise la méthode de *Adams-Bashforth-Moulton* multi-pas. Les autres *solveurs* sont `ode15s`, `ode23s`, `ode23t`, `ode23tb`. Ces *solveurs* présentent des syntaxes similaires :

$$[t, y] = \text{solveur}(\text{eqs}, [t_i; t_f], y_{\text{init}}, \text{opts})$$

Cette syntaxe renvoie la solution y au temps t . L'argument `eqs` est le système d'équations différentielles. Ce système peut être défini de plusieurs manières. Soit à travers un fichier *M-file*, dans ce cas on doit rajouter l'identifiant `@eqs`. Soit à travers une commande `inline` ou bien au moyen de la *fonction anonyme*. Ce système d'équations différentielles est résolu sur l'intervalle $[t_i; t_f]$, avec les conditions initiales $y_{\text{init}} = [y(t_i); y(t_f)]$. L'argument d'entrée `opts`, de type *structure*, compte les options d'optimisation indiquées dans `odeset`, sa syntaxe est donnée par :

$$\text{opts} = \text{odeset}(\text{'Property1'}, \text{value1}, \text{'Property2'}, \text{value2}, \dots)$$

Chaque propriété est suivie de sa valeur. À titre illustratif, la propriété (`'Stats', 'on', ...`) affiche à la fin de l'exécution, des statistiques relatives au calcul effectué. Pour plus d'informations sur les paramètres d'optimisation, taper `help odeset` en ligne de commande. Comme application, nous souhaitons résoudre symboliquement et numériquement au moyen du *solveur* `ode23`, l'équation différentielle du second ordre suivante :

$$\begin{cases} y''(t) + 3y = 4 \sin(t) + \exp(-t) \\ y(0) = 1, \quad y'(0) = 1 \end{cases} \quad (1.3)$$

Ci-dessous le Script :

```

1 clear all ; clc ; close all ;
2 %%%%%%%%%%%%%%% SOLUTION SYMBOLIQUE %%%%%%%%%%%%%%%
3 syms t
4
5 ySym = dsolve('D2y + 3*y = 4*sin(t) + exp(-t)', 'y(0) = 1', 'Dy(0) = 1', 't')
6
7 %%%%%%%%%%%%%%% SOLUTION NUMERIQUE %%%%%%%%%%%%%%%
8 a = 0 ; b = 10 ; n = 200 ; h = (b-a)/n ; tn = a :h: b ; y = [1 ; 1] ;
9
10 eqs = @(tn,y) [y(2); - 3*y(1) + 4*sin(tn) + exp(-tn)] ;
11 opts = odeset('Stats', 'on', 'RelTol', 1e-3) ;
12 [tn, ysol] = ode23(eqs, [tn(1) ; tn(end)], y, opts) ; % SOLVEUR ode23
13
14 %%%%%%%%%%%%%%% AFFICHAGE GRAPHIQUE %%%%%%%%%%%%%%%
15 figure('color', [1 1 1]) ;

```

```

16 plot(tn, ysol(:,1), 'o-', 'LineWidth',1) ; hold on ;
17 plot(tn, double(subs(ySym, t, tn)), 'o-r', 'LineWidth',1) ;
18 xlabel('t', 'FontSize',12) ; ylabel('y(t)', 'FontSize',12) ;
19 ih1 = legend('SOLUTION NUMERIQUE', 'SOLUTION ANALYTIQUE') ;
20 set(ih1, 'Interpreter', 'none', 'Location', 'NorthWest', 'Box', 'off', ...
21     'Color', 'none') ;

```

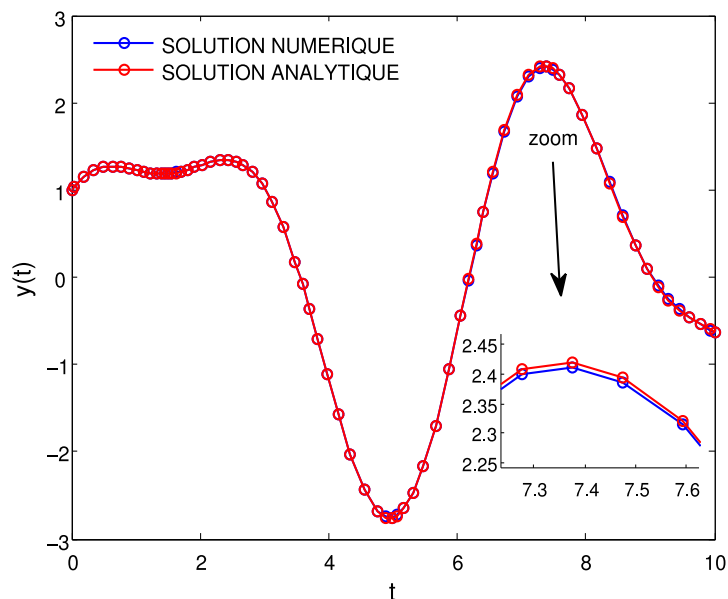


FIGURE 1.1: Solutions analytique et numérique générées par le solveur ode23.

Une autre façon de faire est de créer des fonctions imbriquées dans un fichier *M-file*, selon la syntaxe suivante :

```

1 function [] = eqs_sol(t,y)
2 tsol = [t(1) ; t(end)] ;
3
4 opts = odeset('Stats','on','RelTol', 1e-3) ;
5 [t , ysol] = ode23(@eqs, tsol, y, opts) ; plot(t,ysol(:,1)) ;
6 function dydt = eqs(t, y)
7 dydt = [y(2) ; - 3*y(1) + 4*sin(t) + exp(-t)] ;
8 end
9
10 end

```

Qu'il faudra ensuite appeler, en tapant dans la fenêtre des commandes : `eqs_sol(t, [1 ; 1])`. Cette commande accepte deux arguments en entrée, le vecteur `t` et les conditions initiales `[1 ; 1]`. Le fichier doit être sauvegardé sous le nom `eqs_sol.m` dans le répertoire courant.

1.4.1 Résolution formelle des équations différentielles

Le logiciel dispose de la commande `dsolve` permettant la résolution formelle d'équations différentielles ordinaires et des systèmes d'équations également. Avec cette fonction, la connaissance des conditions initiales n'est pas nécessaire afin d'obtenir les solutions de ces équations. Sa syntaxe usuelle est :

```
sol = dsolve('eq1, eq2,...', 'cond1, cond2,...', 'v')
```

La commande `dsolve` résout symboliquement les équations différentielles ordinaires `eq1`, `eq2` suivant la variable indépendante `v` et tenant compte des conditions initiales indiquées par `cond1`, `cond2`. Nous allons illustrer ce propos avec l'exemple suivant :

```
1 clc ; clear all ;
2
3 expr = dsolve('Dy = 3*y + 5', 'x') ; % equation du premier ordre
4
5 pretty(expr)
6
7 pretty(expr) % solution renvoyee
8 - 5/3 + exp(3 x) C1
```

La lettre D devant une variable signifie qu'on définit la dérivée de cette variable, c'est l'opérateur *différenciation*. De la même façon, écrire D2 se traduit par $\frac{d^2}{dx^2}$. Les lettres placées juste après cet opérateur, sont considérées comme des variables dépendantes. Ces variables ne sont pas obligatoirement déclarées comme symboliques. Par défaut, la variable indépendante est `t`. Dans l'exemple ci-dessus, nous avons remplacé cette dernière par la variable `x`. Dans le cas où le nombre de conditions initiales est inférieur au nombre de variables dépendantes, la solutions est renvoyée avec une constante arbitraire, notée `C1`, `C2`, ..., etc. Pour un système d'équations, nous écrivons le script suivant :

```
1 clc ; clear all ;
2 [expr1, expr2] = dsolve('Dx = -x + 2', 'Dy = - 5*x + 2') ;
3 % systeme d'equations
4
5 pretty(expr1)
6
7 2 + exp(-t) C2 % solution 1
8
9 pretty(expr2)
10
11 -8 t + 5 exp(-t) C2 + C1 % solution 2
```


En intégrant des conditions initiales, on utilisera la syntaxe :

```

1 clc ; clear all ;
2
3 expr = dsolve('D2y = y', 'y(0) = 1', 'Dy(0) = 0')
4 % equation du second ordre
5
6 pretty(expr)
7      1/2 exp(-t) + 1/2 exp(t)

```

Signalons que les conditions aux limites ne sont pas forcément imposées à l'instant initial, ils peuvent exister pour différentes valeurs temporelles. La commande graphique `ezplot` peut servir à tracer les expressions formelles des solutions des équations différentielles dépourvues de constantes arbitraires C_1, C_2, \dots , etc. En guise d'application, nous souhaitons déterminer les expressions formelles des solutions des équations différentielles suivantes :

$$f'(t) = \frac{1}{4}f(t) + \sin(t - 1), \quad f(0) = 1/2 \quad (1.4)$$

$$\begin{cases} x'(t) = \frac{1}{2} + 4y(t) + 1 \\ y'(t) = -2x(t) + 3y(t) + 2 \\ \text{Avec, } x(0) = 0 \text{ et } y(0) = 1 \end{cases} \quad (1.5)$$

Ci-dessous le script correspondant :

```

1 clc ; clear all ; close all
2 %%%%%%%%%% EQUATIONS ET SYST. D'EQUATIONS DIFFERENTIELLES %%%%%%%%%%
3 syms t
4
5 eq = dsolve('Dfx = 1/4*fx + sin(t - 1)', 'fx(0) = 1/2') ;
6 figure('color', [1 1 1]) ; ezplot(eq) ; xlim([-7 7]) ;
7 xlabel('t', 'FontSize', 12) ; ylabel('f(t)', 'FontSize', 12)
8 fx = simple(eq) ; pretty(fx) ;
9
10 [eq1, eq2] = dsolve('Dx = 1/2*x + 4*y + 1, Dy = -2*x + 3*y + 2', 'x(0) = 0, y
      (0) = 1'); % SYSTEME D'EQUATION DIFFERENTIELLE AVEC CONDITIONS AUX
      LIMITES
11 pretty(eq1)
12 pretty(eq2)
13
14 borneinf = -1/2; bornesup = 6 ; dx = 1/500 ;
15 ti = borneinf :dx: bornesup ; xt = subs(eq1,t, ti) ; yt = subs(eq2,t, ti)
      ;
16

```

```

17 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AFFICHAGE GRAPHIQUE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
18
19 figure('color', [1 1 1])
20 plot(ti, xt, 'LineWidth',1) ; hold on ;
21 plot(ti, yt, 'r', 'LineWidth',1)
22 xlabel('t', 'FontSize',12) ; ylabel('Solutions', 'FontSize',12)
23 ih = legend('x(t)', 'y(t)') ; axis([1 13/2 -30000 65000]) ;
24
25 set(ih, 'Interpreter', 'none', 'Location', 'North', 'Box', 'off', ...
26 'Color', 'none')

```

Parfois `ezplot` utilise peu de valeurs en abscisse pour tracer la fonction en question. Par conséquent, cette dernière paraîtra très peu lissée. Afin de remédier à cela, on peut utiliser la commande `Subs` pour substituer la variable sur l'axe des abscisses. En outre, la commande `ezplot` accepte très peu d'arguments en input contrairement à la commande `plot`. Si l'on souhaite établir une substitution multiple au moyen de `subs`, il faudra procéder comme suit :

```

1 clc ; clear all ;
2 syms x y real
3
4 exp = sin(x^2 + y^2) + cos(x^2 + y^2) ;
5 exp_evaluation = subs(exp1, {x,y}, {pi/3, 2*pi})
6
7 exp_evaluation =
8             -0.7024

```

1.5 Conclusion

Dans ce chapitre, nous avons décrit quelques syntaxes usuelles du logiciel Matlab®. Ces notions peuvent être très utiles pour les débutants et pour les personnes désireuses de se familiariser avec ce logiciel. Nous avons délibérément favorisé le côté opérationnel en écrivant des scripts sur des exemples d'applications. Cette approche est jugée plus pratique au regard d'une présentation plus générale. Dans le prochain chapitre, il sera question de présenter les notions théoriques de Chimie Quantique ayant été intégrées dans l'application développée dans le cadre de ce projet.

Notions théoriques intégrées dans l'application

Sommaire

2.1	Loi de Planck	15
2.2	Équation de Schrödinger dépendante du temps	18
2.2.1	Particule dans une boîte	20
2.2.2	Oscillateur harmonique	27
2.3	Orbitales atomiques	29
2.4	Représentation des particules en paquets d'onde	32
2.5	Conclusion	37

2.1 Loi de Planck

À la fin du XIXe siècle, de nombreux physiciens théoriciens pensaient que la physique était en passe d'expliquer la plupart des phénomènes naturels. Ils pouvaient calculer les mouvements des objets matériels à l'aide des lois de la mécanique classique de Newton et décrire les propriétés de l'énergie rayonnante à l'aide des relations mathématiques de James Clerk Maxwell. L'univers apparaissait comme un lieu simple et ordonné, contenant de la matière, constituée de particules ayant une masse et dont la position et le mouvement pouvaient être décrits avec précision. Le rayonnement électromagnétique, considéré comme n'ayant pas de masse et dont la position exacte dans l'espace ne pouvait être fixée. La matière et l'énergie étaient donc considérées comme des phénomènes distincts et non liés. Rapidement, cependant, les scientifiques ont commencé à s'intéresser de plus près à quelques phénomènes intrigants qui ne pouvaient être expliqués par les théories disponibles à l'époque. Le rayonnement du corps noir est un phénomène expérimental qui ne pouvait être expliqué de manière adéquate par la physique classique. Les tentatives d'explication ou de calcul de

cette distribution spectrale à partir de la théorie classique se sont soldées par des échecs. Une théorie développée par Rayleigh et Jeans prédisait que l'intensité devait atteindre l'infini aux courtes longueurs d'onde. Comme l'intensité tombe en fait à zéro aux courtes longueurs d'onde, le résultat de Rayleigh-Jeans a été appelé la catastrophe ultraviolette. Il n'y avait pas d'accord entre la théorie et l'expérience dans la région ultraviolette du spectre du corps noir.

En 1900, le physicien théoricien allemand Max Planck a expliqué la catastrophe ultraviolette en proposant que l'énergie des ondes électromagnétiques soit quantifiée plutôt que continue. Cela signifie que pour chaque température, il existe une intensité maximale de rayonnement émise dans un corps noir, correspondant aux pics de la distribution spectrale, de sorte que l'intensité ne suit pas une courbe lisse à mesure que la température augmente, comme le prévoit la physique classique. Ainsi, l'énergie ne peut être gagnée ou perdue qu'en multiples d'une unité d'énergie, un quantum (la plus petite unité d'énergie possible). L'énergie ne peut être gagnée ou perdue que par multiples entiers d'un quantum $E = h\nu = \hbar\omega$, avec la constante de proportionnalité $h = 6.626 \times 10^{-34} \text{ J s}$ est appelée constante de Planck, l'une des constantes fondamentales les plus connues de la science. Max Planck explique la distribution spectrale du rayonnement du corps noir comme étant le résultat des oscillations des électrons. Il a découvert qu'il pouvait expliquer la distribution spectrale du corps noir (courbe en cloche asymétrique) observée s'il exigeait que ces oscillateurs ne rayonnent pas de l'énergie de manière continue, comme l'exigerait la théorie classique, mais qu'ils ne puissent perdre ou gagner de l'énergie que par quanta, d'une taille $h\nu$ pour un oscillateur de fréquence ν . La formule obtenue est :

$$d\rho(\nu, T) = \frac{8\pi h\nu^3}{c^3} \frac{d\nu}{\exp\left(\frac{h\nu}{k_B T}\right) - 1} \quad [J/m^3] \quad (2.1)$$

Cette formule pouvant être écrite en fonction de la longueur d'onde sachant que $\lambda\nu = c$ alors $d\nu = -c d\lambda/\lambda^2$:

$$d\rho(\lambda, T) = \frac{8\pi hc}{\lambda^5} \frac{d\lambda}{\exp\left(\frac{hc}{\lambda k_B T}\right) - 1} \quad [J/m^3] \quad (2.2)$$

L'accord entre la théorie de Planck et l'observation expérimentale a fourni une preuve solide que l'énergie du mouvement des électrons dans la matière est quantifiée. C'est une conséquence aussi que l'énergie transportée par la lumière est également quantifiée en unités de $h\nu$. Ces paquets d'énergie sont appelés *photons*. Pour mieux comprendre le comportement de la loi de Planck, le script Matlab[®] ci-dessous, simule cette loi pour n'importe quelle température.

```

1 clear all ; clc ; close all ;
2 % SAMIR KENOUCHE
3 c = 2.99792458*10.^8 ; % VITESSE DE LA LUMIERE
4 h = 6.626176*10.^-34 ; % CONSTANTE DE PLANCK
5 k = 1.380662*10.^-23 ; % CONSTANTE DE BOLTZMANN
6 T = 4000 : 1e3: 8000 ; % TEMPERATURE EN KELVIN

```

```

7 lambda = 0.1:0.002:30 ;
8 lambda = lambda.*1e-6 ; % LONGUEUR D'ONDE EN MIRO METRE (10^-6 m)
9
10 % RELATION DE PLANCK
11 rho_planck = @(lambda, T) (8*pi*h*c)./((lambda.^5).*(exp((h.*c)./(k.*T.*
    lambda))-1));
12
13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% UN GRAHPHE POUR CHAQUE TEMPERATURE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14 for iT = 1:numel(T)
15
16 ah = figure(1) ; set(ah, 'color', 'white') ;
17 plot(1e+6.*lambda, 1e-6*rho_planck(lambda, T(iT)) , 'k', 'linewidth', 2) ;
18 xlabel('\lambda (\mu m)', 'fontsize', 12) ;
19 ylabel('Radiant energy density (J m^{-3})', 'fontsize', 12) ;
20 hold on ; xlim([0 2]) ;
21 end

```

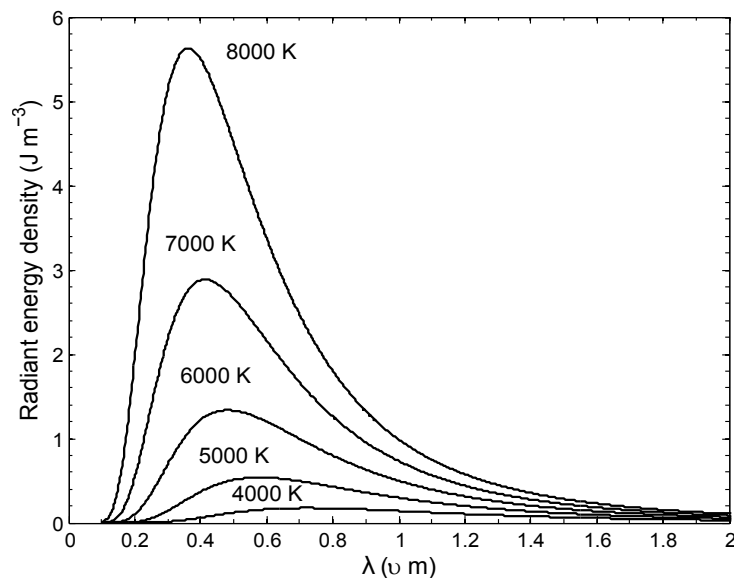


FIGURE 2.1: Profil théorique de la loi de Planck.

Ce graphique représente l'intensité de la lumière émise par un corps noir en fonction de sa température. Lorsque la température augmente, deux phénomènes se produisent. Premièrement, l'intensité de la lumière augmente et deuxièmement, il y a un décalage vers le bleu (λ petit). En d'autres termes, à basse température, la lumière est rouge et, à mesure qu'elle se réchauffe, elle se déplace vers l'ultraviolet (λ grand). On remarquera aisément aussi que la densité spectrale d'énergie diminue pour les plus hautes fréquences après le passage par un maximum. La puissance rayonnée à travers un élément de surface dans le domaine des longueurs d'onde λ et $\lambda + d\lambda$ se calcule au moyen du script ci-dessous :

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% INTEGRALE DE LA DENSITE SPECTRALE D'ENERGIE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % SAMIR KENOUCHE
3 clear all ; clc ; close all ;
4
5 c = 2.99792458*10.^8 ; % VITESSE DE LA LUMIERE
6 h = 6.626176*10.^-34 ; % CONSTANTE DE PLANCK
7 k = 1.380662*10.^-23 ; % CONSTANTE DE BOLTZMANN
8 T = 6000 ; % TEMPERATURE EN KELVIN
9 lB = 0.1 ; uB = 30 ; step = 0.002 ;
10 lambda = lB:step:uB ;
11 lambda = lambda.*1e-6 ; % EN MICROMETRE (10^-6 m)
12
13 rho_planck = @(lambda, T) (8*pi*h*c)/((lambda.^5).*...
14 (exp((h.*c)/(k.*T.*lambda))-1));
15
16 a = lambda(1) ; b = lambda(end)*1e+6 ; n = 9900 ;
17 dx = (b - a)/n ; x = a :dx: b ; int = 0 ;
18
19 for ik = 1:n
20     int = int + 1e-6*rho_planck(lambda(ik), T)*dx ;
21 end
22
23 disp(strcat('DENSITE SPECTRALE (J m^{-3}) :', num2str(int)))

```

Ce script renvoie la densité spectrale calculée à la température $T = 6000 K$, dont le résultat numérique est :

```

1 DENSITE SPECTRALE (J m^{-3}) :1.4855

```

2.2 Équation de Schrödinger dépendante du temps

Il est bien connu que l'équation de Schrödinger indépendante du temps d'une particule a été déterminée à partir de l'équation des ondes classique et de la relation de de Broglie. En revanche, l'équation de Schrödinger dépendante du temps est généralement donnée comme postulat de la mécanique quantique. L'équation de Schrödinger dépendante du temps à une seule particule est la suivante :

$$j\hbar \frac{\partial \psi(\mathbf{r}, t)}{\partial t} = -\frac{\hbar^2}{2m} \nabla^2 \psi(\mathbf{r}, t) + V(\mathbf{r})\psi(\mathbf{r}, t) \quad (2.3)$$

Avec V est une fonction réelle et décrit l'énergie potentielle à laquelle est soumise la particule. Notons que l'équation (2.3) ne tient pas encore compte des effets de spin ou relativistes. Bien entendu, l'équation dépendante du temps peut être utilisée afin d'établir

l'équation indépendante du temps. Si nous écrivons la fonction d'onde comme un produit de deux fonctions spatiale et temporelle, $\psi(\mathbf{r}, t) = \psi(\mathbf{r})f(t)$, alors l'équation (2.3) devient :

$$\frac{j\hbar}{f(t)} \frac{df}{dt} = \frac{1}{\psi(\mathbf{r})} \left[\frac{\hbar^2}{2m} \nabla^2 + V(\mathbf{r}) \right] \psi(\mathbf{r}) \quad (2.4)$$

Puisque le terme de gauche de l'équation est dépendant uniquement du temps et le terme de droite dépend uniquement de l'espace, l'égalité de l'équation (2.4) est satisfaite dans le cas où les deux termes sont égaux à la même constante. Si nous désignons cette constante E (puisque le côté droit doit clairement avoir les dimensions de l'énergie), nous en obtenons deux équations différentielles ordinaires, à savoir :

$$\frac{1}{f(t)} \frac{df(t)}{dt} = -\frac{jE}{\hbar} \quad (2.5)$$

Et,

$$-\frac{\hbar^2}{2m} \nabla^2 \psi(\mathbf{r}) + V(\mathbf{r})\psi(\mathbf{r}) = E\psi(\mathbf{r}) \quad (2.6)$$

Cette dernière équation est celle de Schrödinger indépendante du temps. La solution de (2.5) est :

$$f(t) = e^{-jEt/\hbar} \quad \text{avec} \quad \text{Re} \left[e^{-jEt/\hbar} \right] = \cos(\omega t) \quad (2.7)$$

Nous retrouvons le résultat de $f(t)$ écrit pour le cas de l'équation de Schrödinger indépendante du temps. L'hamiltonien de l'équation (2.6) est un opérateur hermitien et les valeurs propres d'un opérateur hermitien doivent être réelles, donc E est réel. Cela signifie que les solutions $f(t)$ sont purement oscillatoires, rappelons la formule d'Euler $e^{\pm i\theta} = \cos(\theta) \pm i \sin(\theta)$. Par voie de conséquence si :

$$\psi(\mathbf{r}, t) = \psi(\mathbf{r}) e^{-jEt/\hbar} \quad (2.8)$$

Alors la fonction d'onde totale $\psi(\mathbf{r}, t)$ diffère de $\psi(\mathbf{r})$ uniquement par un facteur de phase d'amplitude constante. Cela a des conséquences intéressantes. Tout d'abord, la quantité $|\psi(\mathbf{r}, t)|^2$ est indépendante du temps, puisque :

$$|\psi(\mathbf{r}, t)|^2 = \psi^*(\mathbf{r}, t) \psi(\mathbf{r}, t) = e^{jEt/\hbar} \psi^*(\mathbf{r}, t) e^{-jEt/\hbar} \psi(\mathbf{r}, t) = \psi^*(\mathbf{r}) \psi(\mathbf{r})$$

Deuxièmement, la valeur attendue pour tout opérateur indépendant du temps est également indépendante du temps, si $\psi(\mathbf{r}, t)$ satisfait l'équation (2.8). Par le même raisonnement :

$$\langle \mathcal{O} \rangle = \int \psi^*(\mathbf{r}, t) \hat{\mathcal{O}} \psi(\mathbf{r}, t) d^3r = \int \psi^*(\mathbf{r}) \hat{\mathcal{O}} \psi(\mathbf{r}) d^3r$$

Pour ces raisons, les fonctions d'onde de la forme (2.8) sont appelées *états stationnaires*. L'équation (2.8) représente une solution particulière de l'équation (2.3). La solution générale de l'équation (2.3) serait une combinaison linéaire de ces solutions particulières :

$$\Psi(\mathbf{r}, t) = \sum_i c_i \psi_i(\mathbf{r}) e^{-jE_i t/\hbar}$$

Afin de simuler le comportement de l'électron dans la boîte unidimensionnelle, le script Matlab[®] ci-dessous permet ; de visualiser le puits de potentiel, les fonctions d'onde (solutions exacte et numérique) ainsi que les densités de probabilité de cet électron, suivant n'importe quel niveau d'énergie. De plus, ce script renvoie également le nombre de nœuds obtenus pour le niveau d'énergie considéré.

2.2.1 Particule dans une boîte

La particule dans une boîte est un modèle pouvant illustrer le fonctionnement d'une équation d'onde. Bien qu'il s'agit d'une situation hypothétique, nous pouvons limiter notre modèle à une seule dimension de sorte que l'équation de Schrödinger soit simplifiée. Cette simplification est très utile pour appréhender le comportement de cette équation. Afin de quantifier les différentes propriétés (position, énergie, quantité de mouvement, densité de probabilité de présence ... etc) de l'électron en question, il va falloir résoudre l'équation de Schrödinger en tenant compte des conditions aux limites : $\Psi(x = 0, t) = \Psi(x = a, t) = 0$. Ces conditions expriment l'impossibilité de retrouver l'électron aux extrémités $x = 0$ et $x = a$. Pour l'électron de masse m , l'équation de Schrödinger à une dimension vaut :

$$j \hbar \frac{\partial \Psi(x, t)}{\partial t} = \left(-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} + V(x) \right) \Psi(x, t) \quad (2.9)$$

La position $x = 0$ est interdite car dans ce cas l'électron occupe la position nucléaire. Par conséquent, $\Psi(x = 0, t) = 0$ et la probabilité $|\Psi(x = 0, t)|^2 = 0$ de trouver l'électron est nulle. D'un autre côté, la situation $x = a$ est analogue à la valeur $\Psi(x = a, t) = 0$ d'un électron s'approchant de zéro $\Psi(x = a, t) = 0$ à mesure qu'il s'éloigne du noyau. L'électron est "isolé", ne subissant aucune force extérieure. Dans ce cas de figure, il s'agit des états stationnaires¹ de l'équation de Schrödinger. Par voie de conséquence, la fonction d'onde $\Psi(x, t)$ s'écrit comme un produit d'une fonction "spatiale" par une fonction "temporelle" invariable pour l'ensemble des états quantiques de la particule.

$$\Psi(x, t) = \varphi_n(x) \times f(t) \quad (2.10)$$

Cette équation nous informe que si l'on connaît la fonction d'onde à l'instant t_0 , on la connaîtra aussi à tout instant ultérieur t . L'expression de la fonction $f(t)$ est invariable. Injectons l'équation (2.10) dans (2.9).

$$j \hbar \varphi_n(x) \frac{\partial f(t)}{\partial t} = \left(-\frac{\hbar^2}{2m} \frac{\partial^2 \varphi_n(x)}{\partial x^2} + V(x) \varphi_n(x) \right) f(t) \quad (2.11)$$

$$\underbrace{\Rightarrow j \hbar \frac{1}{f(t)} \frac{\partial f(t)}{\partial t}}_{\text{Temps}} = \underbrace{\frac{1}{\varphi_n(x)} \left(-\frac{\hbar^2}{2m} \frac{\partial^2 \varphi_n(x)}{\partial x^2} + V(x) \varphi_n(x) \right)}_{\text{Espace}} \quad (2.12)$$

1. Une grandeur stationnaire ne signifie pas qu'elle n'évolue pas au cours du temps ou qu'elle est dans un état d'équilibre. C'est plutôt la variation au cours du temps de la grandeur en question qui est constante à tous les instants du temps. Donnons l'exemple d'un mobile qui roule à vitesse constante d'un point A vers un point B. Ce mobile est dans un état stationnaire car sa vitesse (donc son état dynamique), ne change pas au cours du temps, mais cela ne veut aucunement dire que le mobile en question est immobile.

L'égalité exprimée par l'équation (2.12) est satisfaite uniquement dans le cas où les deux termes sont égaux à la même constante. Le deux termes ont clairement la dimension d'une énergie (Joule). Notons cette constante de séparation E_n :

$$\begin{cases} j \hbar \frac{1}{f(t)} \frac{\partial f(t)}{\partial t} = E_n \\ \frac{1}{\varphi_n(x)} \left(-\frac{\hbar^2}{2m} \frac{\partial^2 \varphi_n(x)}{\partial x^2} + V(x) \varphi_n(x) \right) = E_n \end{cases} \quad (2.13)$$

La solution de la première équation différentielle est triviale et donne :

$$f(t) = e^{-j E_n t / \hbar} \quad \text{avec} \quad \text{Re}[e^{-j E_n t / \hbar}] = \cos(\omega t) \quad (2.14)$$

Résolvons la deuxième équation différentielle.

$$\left(-\frac{\hbar^2}{2m} \frac{\partial^2 \varphi_n(x)}{\partial x^2} + V(x) \right) \varphi_n(x) = E_n \varphi_n(x) \quad (2.15)$$

Avec E_n est l'énergie totale de l'électron et m sa masse. Nous avons déjà écrit $V(x) = 0$.

$$\frac{\partial^2 \varphi_n(x)}{\partial x^2} = -\underbrace{\frac{2m E_n}{\hbar^2}}_{\beta^2} \varphi_n(x) \Leftrightarrow \varphi_n''(x) + \beta^2 \varphi_n(x) = 0 \quad (2.16)$$

Écrivons le polynôme caractéristique de l'équation (2.16) qui devient :

$$\lambda^2 + \beta^2 = 0 \Rightarrow \lambda_{1,2}^2 = -\beta^2 \Rightarrow \lambda_{1,2}^2 = j^2 \beta^2 \Rightarrow \lambda_{1,2} = \pm j \sqrt{\beta} \quad (2.17)$$

La solution générale de l'équation (2.16) prend la forme :

$$\varphi_n(x) = c_1 e^{\lambda_1 x} + c_2 e^{\lambda_2 x} = c_1 e^{j \beta x} + c_2 e^{-j \beta x} \quad (2.18)$$

Afin de simplifier les calculs, écrivons l'équation (2.18) sous forme d'une combinaison de fonctions sinusoidales. Utilisons pour cela la formule d'Euler :

$$e^{\pm j \theta} = \cos(\theta) \pm j \sin(\theta) \quad (2.19)$$

A partir de l'équation (2.18) :

$$\begin{aligned} \varphi_n(x) &= c_1 \cos(\beta x) + c_1 j \sin(\beta x) + c_2 \cos(\beta x) - c_2 j \sin(\beta x) \\ &= \underbrace{(c_1 + c_2)}_{c_\alpha \in \mathbb{R}} \cos(\beta x) + \underbrace{(c_1 j - j c_2)}_{c_\beta \in \mathbb{C}} \sin(\beta x) \\ &\Rightarrow \varphi_n(x) = c_\alpha \cos(\beta x) + c_\beta \sin(\beta x) \end{aligned} \quad (2.20)$$

Les constantes c_α et c_β sont déterminées à partir des conditions aux limites :

$$\varphi_n(x=0) = c_\alpha = 0$$

$$\varphi_n(x = a) = 0 = c_\alpha \cos(\beta a) + c_\beta \sin(\beta a)$$

$$\Rightarrow \varphi_n(x = a) = c_\beta \sin(\beta a) = 0 \quad (2.21)$$

Cette équation est nulle dans deux cas de figure. D'abord si $c_\beta = 0$ dans ce cas il en résulte $c_\alpha = c_\beta = 0 \Rightarrow \varphi_n(x) = 0$ c'est une solution triviale qui n'est pas intéressante d'un point de vue physique. Ensuite, la deuxième condition si ² :

$$c_\beta \neq 0 \Rightarrow \sin(\beta a) = 0 \Rightarrow \beta a = n \pi, \quad \text{avec } n \in \mathbb{N}^* \quad (2.22)$$

Par conséquent, la solution de l'équation (2.18) est :

$$\varphi_n(x) = c_\beta \sin\left(\frac{n \pi x}{a}\right) \quad \text{avec } n \in \mathbb{N}^* \quad \text{et} \quad \beta = \sqrt{\frac{2 m E_n}{\hbar^2}} = \frac{n \pi}{a} \quad (2.23)$$

Cette solution décrit l'amplitude spatiale de l'onde de matière associée à l'électron en fonction de la position. L'énergie totale de l'électron est quantifiée par le biais du nombre quantique principal n :

$$E_n = \frac{\hbar^2 \pi^2 n^2}{2 m a^2} \quad (2.24)$$

La fonction d'onde associée à l'électron délocalisée est :

$$\varphi_n(x) = c_\beta \sin\left(\frac{n \pi x}{a}\right) \quad (2.25)$$

La constante c_β est déterminée par la condition de normalisation (ou condition de conservation de l'électron sur le segment de droite) :

$$\int_0^a |\varphi_n(x)|^2 dx = 1 \Rightarrow c_\beta^2 \int_0^a \sin^2\left(\frac{n \pi x}{a}\right) dx = 1 \quad (2.26)$$

Afin de résoudre cette intégrale, nous devons linéariser le sinus en exploitant la relation trigonométrique suivante :

$$\cos 2x = \cos^2 x - \sin^2 x \Rightarrow 1 - \sin^2 x - \sin^2 x = 1 - 2 \sin^2 x \quad (2.27)$$

$$\Rightarrow \sin^2 x = \frac{1 - \cos 2x}{2} \quad (2.28)$$

$$\begin{aligned} \Rightarrow c_\beta^2 \int_0^a \sin^2\left(\frac{n \pi x}{a}\right) dx &= \frac{c_\beta^2}{2} \int_0^a dx - \int_0^a \cos\left(\frac{n \pi x}{a}\right) dx = 1 \\ \Rightarrow \frac{c_\beta^2}{2} [x]_0^a + \frac{c_\beta^2}{2} \left[\frac{a}{n \pi} \sin\left(\frac{n \pi x}{a}\right) \right]_0^a &= 1 \Rightarrow c_\beta = \sqrt{\frac{2}{a}} \end{aligned}$$

L'expression finale de la fonction d'onde $\varphi_n(x)$ associée à la valeur propre E_n est :

$$\varphi_n(x) = \sqrt{\frac{2}{a}} \sin\left(\frac{n \pi x}{a}\right) \quad \text{et} \quad E_n = \frac{\hbar^2 \pi^2 n^2}{2 m a^2} \quad (2.29)$$

2. Les valeurs négatives de n sont négligeables, car elles changent simplement le signe du sinus

L'expression de E_n est utilisée pour calculer l'énergie de la particule dans une boîte unidimensionnelle de longueur a , et cette énergie est quantifiée à travers le nombre quantique $n \in \mathbb{N}^*$. En outre, nous remarquons que plus la longueur de la molécule conjuguée est grande plus l'écart entre deux niveaux successifs est grand, et inversement. En effet, plus a est grand plus nous nous approchons du cas où l'énergie est continue. La fonction d'onde "globale" s'écrit alors :

$$\Psi(x, t) = \left[\sum_n^{\infty} c_n \varphi_n(x) \right] \times e^{-j E_n t/\hbar} \quad (2.30)$$

Le spectre des valeurs propres correspondant est :

$$E_n = \{E_1, E_2, E_3, \dots, \} \quad (2.31)$$

L'écriture (2.30) traduit le principe de superposition des états quantiques de l'électron. Cela signifie concrètement que tant qu'on ne cherche pas à mesurer l'énergie totale, l'électron possède simultanément toutes les valeurs du spectre énergétique (2.31). Si l'on décide de mesurer son énergie totale, on obtiendra aléatoirement l'une des énergies du spectre. La probabilité d'avoir l'état quantique $\{E_i, \varphi_i(x)\}$ est :

$$P_i = |c_i|^2 = \langle \varphi_i | \Psi \rangle^2 = \left[\int_0^a \varphi_i^*(x) \Psi(x) dx \right]^2$$

Supposons que la mesure de l'énergie totale de l'électron donne $E_3 = \frac{\hbar^2 \pi^2 9}{2 m a^2}$. La fonction d'onde décrivant l'électron, faisant objet de la délocalisation est immédiatement :

$$\varphi_3(x) = \sqrt{\frac{2}{a}} \sin\left(\frac{3 \pi x}{a}\right)$$

Une deuxième mesure de l'énergie totale de l'électron, réalisée immédiatement après la première donnera lieu à la même valeur propre E_3 avec une précision infinie. Ceci est vrai car $\varphi_3(x)$ est une fonction propre de l'hamiltonien \hat{H} . Qu'il s'agisse d'une équation d'onde unidimensionnelle ou tridimensionnelle, la fonction d'onde elle-même Ψ décrit les propriétés ondulatoires de la particule. La fonction d'onde n'a pas de signification physique réelle, il est donc difficile d'imaginer à quoi elle "ressemble" autrement qu'en traçant sa fonction. Cependant, la probabilité de trouver la particule à n'importe quelle position le long de l'axe des x entre $x = 0$ et $x = a$ est plus significative physiquement. La probabilité de trouver la particule est proportionnelle au carré de la fonction d'onde, qui est représentée par Ψ^2 .

```

1 clc ;
2 clear all ;
3 close all;
4 % Samir Kenouche
5 n= 2 ; % NIVEAU D'ENERGIE CONSIDERE
6 a= 1; freq = n*pi/a ; npts = 4000; lbond = -0.5 ; ubond = 1.5 ;
7 x = linspace(lbond, ubond, npts) ; psix= sqrt(2./a).*sin(freq.*x) ;

```

```

8
9 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SOLUTION NUMERIQUE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10 psinum = [0 ; 1] ; xn = linspace(0,1, npts) ;
11 eqs = @(xn,psinum) [psinum(2); - (n*pi/a).^2.*psinum(1)] ;
12 opts = odeset('Stats','on','RelTol', 1e-6) ;
13 [xn, psinumsol] = ode45(eqs, [xn(1) ; xn(end)], psinum, opts) ; % SOLVEUR
    ode23
14
15 for k = 1:length(x)
16
17     if x(k) < 0 || x(k) > 1
18         potential(k) = 1 ; psix(k) = 0 ;
19     elseif 0 < x(k) < 1
20         psix = psix ; potential(k) = 0 ;
21     end
22 end
23
24 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PROBABILITES %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
25 plot(x, potential.^2./(max(potential.^2)), 'g', 'LineWidth',1) ; hold on ;
26 plot(x, psix.^2./(max(psix.^2)), 'LineWidth',1) ;
27 plot(xn, psinumsol(:,1).^2./max(psinumsol(:,1).^2), 'ro', 'LineWidth',1) ;
28 xlabel('x', 'FontSize',12) ; ylabel('\psi(x)^2', 'FontSize',12) ;
29 ih1 = legend('Potential', 'Exact solution', 'Numerical solution') ;
30 set(ih1, 'Interpreter', 'none', 'Location', 'NorthWest', 'Box', 'off', ...
31 'Color', 'none') ;
32 xlim([lbond ubond]) ; box off ;
33
34 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% NOMBRE DE NOEUDS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
35 density = psinumsol(:,1).^2 ; indmax = [] ; density_max = [] ;
36 for k = 2:length(xn)-1
37     if (density(k-1) < density(k)) && (density(k) > density(k+1))
38         indmax =[indmax k] ;
39         density_max = [density_max density(k)];
40     end
41 end
42
43 pos = xn(indmax) ;
44 noeuds_nbre = length(pos) % AFFICHAGE DU NOMBRE DE NOEUDS
45
46 plot(pos, density_max./max(density_max), 'kx', 'MarkerSize',10, 'LineWidth',2)
    ; hold on ;
47 plot(pos, density_max./max(density_max), 'ko', 'MarkerSize',10, 'LineWidth',2)

```

Les sorties renvoyées par ce script sont comme suit :

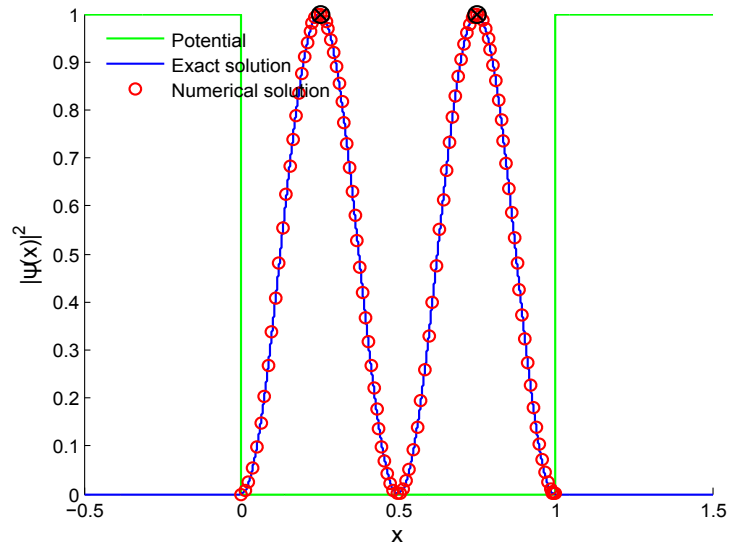


FIGURE 2.2: Densité de probabilité de présence pour le premier état excité ($n = 2$). Les positions privilégiées sont marquées en couleur noire.

Le script ci-dessous, renvoie les mêmes sorties que le précédent, en considérant le cas bidimensionnel.

```

1 clear all ; clc ; close all ;
2
3 a = 1 ; b = 1 ; nn = 2 ; freq = nn*pi/a ; npts = 500 ;
4 lbond = 0 ; ubond = 1.0 ; nstep = (ubond-lbond)/npts ;
5 [xn, yn] = meshgrid(lbond:nstep:ubond) ;
6 Z = sqrt(2./a).*sqrt(2./b).*sin(freq.*xn).*sin(freq.*yn) ;
7
8 %%%%%%%%%% AFFICHAGE DE LA FONCTION D'ONDE EN 3 DIMENSION %%%%%%%%%%
9 fig1 = figure('color', [1 1 1]) ;
10 h1 = mesh(Z./max(max(Z))) ;
11 box on ; axis vis3d; rotate3d on ; camlight; lighting phong ;
12 light('Position',[0 0 1],'Style','infinite') ;
13 ih1 = gca ; % RECUPERATION DES IDENTIFIANTS DES AXES
14 set(ih1,'XTickLabel',{'0','0.5','0.75'},'FontSize',12)
15 set(ih1,'YTickLabel',{'0','0.5','0.75'},'FontSize',12)
16 xlabel('x','FontSize',12) ; ylabel('y','FontSize',12) ;
17 zlabel('\psi(x,y)','FontSize',12) ;
18
19 %%%%%%%%%% AFFICHAGE DE LA PROBABILITE DE PRESENCE EN 2 DIMENSION %%%%%%%%%%
20 fig2 = figure('color', [1 1 1]) ;
21 density_norm = Z.^2./max(max(Z.^2)) ;
22 h2 = mesh(density_norm) ;

```

```

23 box on ; axis vis3d; rotate3d on ; camlight; lighting phong ;
24
25 light('Position',[0 0 1],'Style','infinite') ;
26 ih2 = gca ; % RECUPERATION DES IDENTIFIANTS DES AXES
27 set(ih2,'XTickLabel',{'0','0.5','0.75'},'FontSize',12)
28 set(ih2,'YTickLabel',{'0','0.5','0.75'},'FontSize',12)
29
30 xlabel('x','FontSize',12) ; ylabel('y','FontSize',12) ;
31 zlabel('\psi(x,y)^2','FontSize',12) ;

```

Les sorties renvoyées par ce script sont comme suit :

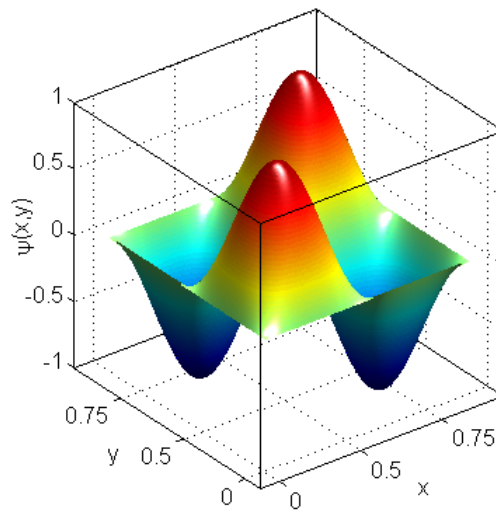


FIGURE 2.3: Fonction d'onde bidimensionnelle pour le premier état excité ($n = 2$).

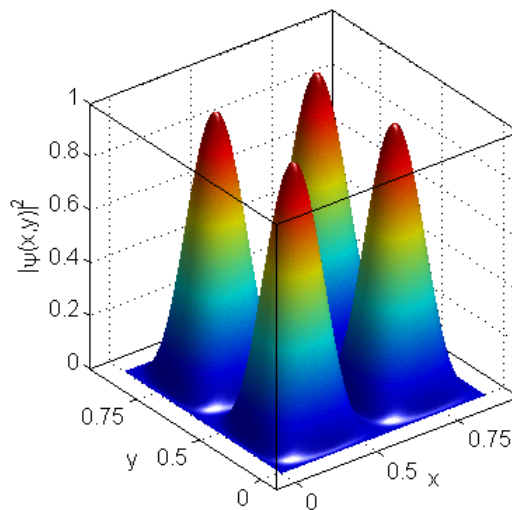


FIGURE 2.4: Densité de probabilité de présence bidimensionnelle pour le premier état excité ($n = 2$).

2.2.2 Oscillateur harmonique

Dans la formulation quantique de l'oscillateur harmonique, la fonction d'énergie potentielle de type quadratique est écrite sous la forme générale suivante :

$$V(x) = \frac{1}{2} m\omega^2 x^2 \quad (2.32)$$

Le déplacement de l'objet de masse m (l'oscillateur) varie périodiquement dans le temps avec une fréquence angulaire $\omega = \sqrt{k/m}$ qui dépend à la fois de la masse et de la constante de force k . En combinant cette expression avec l'équation de Schrodinger indépendante du temps pour le cas unidimensionnel, on obtient :

$$-\frac{\hbar}{2m} \frac{d^2\psi(x)}{dx^2} + \frac{1}{2} m\omega^2 x^2 \psi(x) = E \psi(x) \quad (2.33)$$

Comme pour le modèle quantique de la particule dans un puits de potentiel, la résolution de cette équation fournira les valeurs propres E_v ainsi que leurs fonctions propres correspondantes $\psi(x)$. Les fonctions propre sont symétriques par rapport à $x = 0$ et sont de carrée sommable. Ces exigences sont suffisantes pour que la densité de probabilité $|\psi(x)|^2$ soit finie lorsqu'elle est intégrée sur toute la gamme de x allant de $-\infty$ à $+\infty$. Les valeurs propres sont données par :

$$E_v = \left(v + \frac{1}{2}\right) \hbar\omega, \quad \forall v \in \mathbb{N} \quad (2.34)$$

Les fonctions propres correspondantes :

$$\psi_v(x) = N_v H_v(\alpha^{1/2}x) e^{-\beta^2 x^2/2} \quad \forall v \in \mathbb{N} \quad (2.35)$$

Où $\alpha = \sqrt{k m/\hbar^2}$ et N_v est la constante de normalisation donnée par :

$$N_v = \frac{1}{(2^v v!)^{1/2}} \left(\frac{\alpha}{\pi}\right)^{1/4} \quad (2.36)$$

$H(\alpha^{1/2}x)$ est un polynôme de degré v appelé polynôme d'Hermite. La vibration d'une molécule diatomique peut être décrite par ce modèle. Malgré ses insuffisances par rapport aux effets anharmoniques, ce modèle est très utilisé en spectroscopie infra-rouge pour décrire des systèmes simples. Nous avons résolu numériquement les fonctions propres de l'oscillateur harmonique. Le polynôme d'Hermite est implémenté dans le fichier M-file suivant :

```

1 function polynome_herm = polynome_hermite(v,x)
2 % FICHIER M-FILE POUR CALCULER LE POLYNOME D'HERMITE
3
4 if v== 0
5
6     polynome_herm = 1 ;
7
8 else
9     polynome_herm = 0 ;

```

```

10
11
12 for k = 0:floor(v/2)
13     termes_polynome = ((-1).^k).*(factorial(v)./(2.^k.*factorial(k).*
14     factorial(v-2*k)));
15     polynome_herm = polynome_herm + termes_polynome.*((x).^(v-2*k)) ;
16 end
17
18 return

```

Ci-dessous, le script principal (ou appelant). Afin d'exécuter ces deux fichiers (M-file et principal), ils doivent impérativement être dans le même répertoire.

```

1 clc ; clear all ; close all ;
2 % SAMIR KENOUCHE
3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4 % CE SCRIPT MATLAB PERMET DE CALCULER NUMERIQUEMENT LES FONCTIONS PROPRES
5 % DE L OSCILLATEUR HARMONIQUE
6 addpath('C:\Users\kenouche\Desktop\') % CHANGER LE CHEMIN
7
8 m = 1 ; frequence = 200 ; hbar = 1 ; npts = 1000 ;
9 v = 1 ; % NIVEAU DE VIBRATION
10 alpha = sqrt(frequence*m)/hbar.^2 ;
11 lbond = -1 ; ubond = 1.0 ; % BORNES
12 x = linspace(lbond, ubond, npts) ; % VECTEUR POSITION
13
14 psioscillateur = 1/sqrt(2.^v.*factorial(v)).*...
15     (alpha/pi).^(1/4).*polynome_hermite(v,sqrt(alpha).*x).*...
16     exp(-alpha.*x.^2/2) ; % LA FONCTION D'ONDE
17     % AVCE polynome_hermite EST LE POLYNOME D'HERMITE
18
19 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% AFFICHAGE GRAPHIQUE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
20
21 fig1 = figure('color',[1 1 1]) ; % VISUALISATION DE LA FONCTION D'ONDE
22 plot(x,psioscillateur./max(psioscillateur),'b') ;
23 xlabel('x') ; ylabel('\psi_v(x)') ;
24 hold on ;
25 plot(x,psioscillateur.^2./max(psioscillateur.^2),'r') ;
26 % VISUALISATION DE LA PROBABILITE DE PRESENCE
27 xlabel('x') ; ylabel('\psi_v(x) and |\psi_v(x)|^2') ;
28 legend('\psi_v(x)', '|\psi_v(x)|^2') ;

```

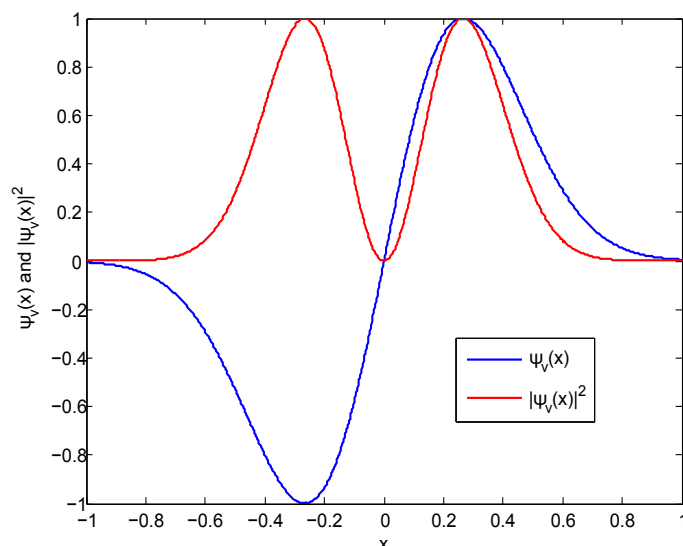



FIGURE 2.5: Fonction d'onde et densité de probabilité de présence pour le deuxième niveau de vibration ($v = 1$).

Ce script permet de calculer la fonction d'onde pour n'importe quelle valeur du niveau d'énergie de vibration (v). Les polynômes d'Hermite sont également calculés $\forall v$.

2.3 Orbitales atomiques

L'atome d'hydrogène et les atomes hydrogéoides ayant un seul électron (semblables à l'hydrogène), constitue le système le plus simple. C'est le seul système atomique pour lequel l'équation de Schrödinger peut être résolue exactement pour les niveaux d'énergie et les fonctions d'onde. À cette fin, nous considérons un noyau de charge $+Ze$ à l'origine et un électron unique repéré par le vecteur position \mathbf{r} . L'équation de Schrödinger pour un tel système est donnée par :

$$\hat{\mathcal{H}}(r, \theta, \varphi)\psi(r, \theta, \varphi) = E\psi(r, \theta, \varphi) \quad (2.37)$$

Les fonctions d'onde de l'atome d'hydrogène, $\psi(r, \theta, \varphi)$, sont appelées *orbitales atomiques*. Une telle orbitale est une fonction au sens mathématique du terme décrivant le comportement ondulatoire de l'électron dans un atome. En outre, une orbitale est une extension du modèle de Bohr où l'électron gravite sur une orbite ayant un rayon fixe. À ce titre, une orbitale est une région de l'espace dont les contours ou les limites sont définis par une fonction mathématique dont le carré donne la probabilité de trouver l'électron en tout point de cet espace. L'équation de Schrödinger d'un tel système s'écrit :

$$\left(-\frac{\hbar^2}{2m}\nabla^2 + V\right)\psi = E\psi \quad (2.38)$$

Où ∇^2 est l'opérateur Laplacien et V est l'énergie potentielle électrostatique ou *Coulombienne* qui est donnée par :

$$V(r) = -\frac{1}{4\pi\epsilon_0} \frac{Ze^2}{r} \quad (2.39)$$

Où ϵ_0 est la permittivité du vide (pas besoin d'une permittivité relative car l'espace à l'intérieur de l'atome est "vide"), les charges e et Ze sont respectivement celles de l'électron et du noyau, pour l'hydrogène et les ions hydrogénoïdes le nombre d'électron $Z = 1$. La distance radiale, r , décrit l'éloignement de l'électron par rapport au noyau. L'énergie potentielle *Coulombienne* est inversement proportionnelle à la distance entre l'électron et le noyau et ne dépend d'aucun angle. Un tel potentiel est appelé *potentiel central*. La solution exacte de l'équation de Schrödinger pour l'atome d'hydrogène et les ions hydrogénoïdes est obtenue sous la forme :

$$\psi(r, \theta, \varphi) = \underbrace{R_{n,l}(r)}_{\text{taille de l'orbitale}} \times \underbrace{Y_l^m(\theta, \varphi)}_{\text{forme de l'orbitale}} \quad (2.40)$$

La solution exacte (valeurs et fonctions propres) de l'équation de Schrödinger pour l'atome d'hydrogène et les ions hydrogénoïdes s'obtient en multipliant les solutions des parties radiale $R_{n,l}$ et angulaire $Y_l^m(\theta, \varphi)$:

$$\begin{aligned} \Rightarrow \psi_{n,l,m}(r, \theta, \varphi) &= \underbrace{\left[\left(\frac{2}{na_0} \right)^3 \frac{(n-l-1)!}{2n[(n+1)!]^3} \right]^{1/2} \left(\frac{2r}{a_0n} \right)^l e^{-\left(\frac{r}{a_0n} \right)} L_{n-l-1}^{2l+1} \left(\frac{2r}{a_0n} \right)}_{R_{n,l}(r)} \\ &\times \underbrace{(-1)^m \sqrt{\frac{(2l+1)!(l-m)!}{4\pi(l+m)!}}}_{Y_l^m(\theta, \varphi)} \times P_{l,m}(\cos \theta) \times e^{jm\varphi} \end{aligned} \quad (2.41)$$

Où L_{n-l-1}^{2l+1} et $P_{l,m}$ sont respectivement les polynômes associés de Laguerre et Legendre. L'équation ci-dessous est réécrite simplement en occultant la partie radiale :

$$\Rightarrow \psi_{n,l,m}(r, \theta, \varphi) = \left[\left(\frac{2}{na_0} \right)^3 \frac{(n-l-1)!}{2n[(n+1)!]^3} \right]^{1/2} \left(\frac{2r}{a_0n} \right)^l e^{-\left(\frac{r}{a_0n} \right)} L_{n-l-1}^{2l+1} \left(\frac{2r}{a_0n} \right) \times Y_l^m(\theta, \varphi) \quad (2.42)$$

Avec $a_0 = \frac{4\pi\epsilon_0\hbar^2}{e^2m_e} = 0.529177 \times 10^{-10} m$ est le rayon de Bohr. Par ailleurs, pour toute fonction d'onde $\psi_{n,l,m}(r, \theta, \varphi)$ le résultat statistique de la mesure de la distance de l'électron par rapport au noyau donne la valeur moyenne (ou l'espérance de r) suivante :

$$\langle r \rangle = \int_0^{2\pi} \int_0^\pi \int_0^\infty |\psi_{nlm}(r, \theta, \varphi)|^2 r^3 \sin \theta dr d\theta d\varphi \quad \Rightarrow \quad \langle r \rangle = \frac{n^2 a_0}{Z} \left[1 + \frac{1}{2} \left(1 - \frac{l(l+1)}{n^2} \right) \right] \quad (2.43)$$

Le nombre quantique principal n indique l'énergie de l'électron ainsi que la distance moyenne d'un électron par rapport au noyau, comme le montre clairement la relation (2.43). Plus un électron est proche du noyau, chargé positivement, plus l'électron est fortement attiré

par le noyau comparativement à un électron plus éloigné dans l'espace. Cela signifie que les électrons ayant une valeur de n plus élevée sont plus faciles à éliminer d'un atome. Les fonctions $Y_l^m(\theta, \phi)$ sont appelées *harmoniques sphériques*, fournissent des informations sur la position de l'électron autour du noyau, et la fonction radiale $R_{n,l}(r)$ décrit l'éloignement de l'électron par rapport au noyau. Les valeurs propres sont obtenues selon :

$$\Rightarrow E_n = -\frac{m}{2\hbar^2} \left[\frac{e^2}{4\pi\epsilon_0} \right]^2 \times \frac{1}{n^2} \quad (2.44)$$

Une orbitale atomique est une fonction mathématique qui décrit le comportement ondulatoire de l'électron dans un atome. A titre d'exemple, la fonction d'onde $\psi_{1,0,0}$ est appelée orbitale $|\psi_{1s}\rangle$. Les contraintes sur n , l et m qui sont imposées lors de la solution de l'équation de Schrödinger de l'atome d'hydrogène explique pourquoi il y a une seule orbitale $|\psi_{1s}\rangle$, trois orbitales $|\psi_{2p}\rangle$, cinq orbitales $|\psi_{3d}\rangle$, ... etc. L'équation de Schrödinger exige trois nombres quantiques (n , l , m) afin de spécifier une fonction d'onde pour l'électron. Les nombres quantiques fournissent des informations sur la distribution spatiale d'un électron. Bien que n puisse prendre n'importe quel nombre entier positif non nul, seules certaines valeurs de l et de m sont autorisées pour une valeur donnée de n . Le nombre quantique l est appelé nombre quantique angulaire, lequel décrit la forme de la région de l'espace occupée par un électron ($n \geq l + 1$). Le troisième nombre quantique, est le nombre quantique magnétique m , lequel est associé à l'orientation de la région de l'espace occupé par un électron par rapport à un champ magnétique appliqué ($-l \leq m \leq +l$). Chaque triplet des trois nombres quantiques fournit une répartition spatiale particulière à l'électron. Les parties radiales des fonctions d'onde pour les premières valeurs de n et de l sont déduites de la solution (2.41) comme suit :

$$\begin{aligned} R_{10}(r) &= 2 \left(\frac{Z}{a_0} \right)^{3/2} e^{-Zr/a_0} \\ R_{20}(r) &= \frac{1}{2\sqrt{2}} \left(\frac{Z}{a_0} \right)^{3/2} \left(2 - \frac{Zr}{a_0} \right) e^{-Zr/2a_0} \\ R_{21}(r) &= \frac{1}{2\sqrt{6}} \left(\frac{Z}{a_0} \right)^{3/2} \frac{Zr}{a_0} e^{-Zr/2a_0} \\ R_{30}(r) &= \frac{2}{81\sqrt{3}} \left(\frac{Z}{a_0} \right)^{3/2} \left[27 - 18 \frac{Zr}{a_0} + 2 \left(\frac{Zr}{a_0} \right)^2 \right] e^{-Zr/3a_0} \end{aligned}$$

Ci-dessous, quelques harmoniques sphériques déduites de la solution (2.41) :

$$\begin{aligned} Y_1^1(\theta, \varphi) &= -\left(\frac{3}{8\pi} \right)^{1/2} \sin \theta e^{i\varphi} \\ Y_1^{-1}(\theta, \varphi) &= \left(\frac{3}{8\pi} \right)^{1/2} \sin \theta e^{-i\varphi} \\ Y_1^0(\theta, \varphi) &= \left(\frac{3}{4\pi} \right)^{1/2} \cos \theta \\ Y_2^2(\theta, \varphi) &= \left(\frac{15}{32\pi} \right)^{1/2} \sin^2 \theta e^{2i\varphi} \\ Y_2^{-2}(\theta, \varphi) &= \left(\frac{15}{32\pi} \right)^{1/2} \sin^2 \theta e^{-2i\varphi} \end{aligned}$$

Pour tout système quantique (atome, molécules, ... etc), les fonctions d'onde $\psi_{n,l,m}(r, \theta, \varphi)$, donnent l'amplitude de la probabilité de trouver l'électron dans une région donnée de l'espace. Ces amplitudes sont utilisées pour calculer les probabilités réelles associées aux mesures de la position de l'électron. En effet, la probabilité de trouver l'électron dans un élément infinitésimal dv repéré par le vecteur position \mathbf{r} est :

$$p = |\psi_{nlm}(r, \theta, \varphi)|^2 dv = R_{nl}^2(r) |Y_l^m(\theta, \varphi)|^2 dv \quad \text{avec} \quad dv = r^2 \sin \theta dr d\theta d\varphi \quad (2.45)$$

Notons que les fonctions radiales $R_{n,l}(r)$ sont propres au potentiel de Coulomb $V(r)$, contrairement aux harmoniques sphériques $Y_l^m(\theta, \varphi)$. Comme il a été souligné précédemment, c'est la solution radiale qui conduit aux niveaux d'énergie autorisés.

2.4 Représentation des particules en paquets d'onde

La fonction d'onde d'une particule massive de quantité de mouvement p et d'énergie E qui se déplace dans la direction des x positifs s'écrit :

$$\psi(x, t) = A e^{i(kx - \omega t)} \quad (2.46)$$

Cette formule désigne une onde plane dont les maxima et minima se propagent dans la direction des x positifs. Où A est l'amplitude de l'onde $k = \frac{p}{\hbar}$, $\omega = \frac{E}{\hbar}$ et $v_p = \frac{\omega}{k}$ étant la vitesse de phase. D'après ce qui précède, l'interprétation physique de la fonction d'onde est liée à la quantité $|\psi(x, t)|^2$, qui définit la densité de probabilité de trouver la particule à la position x à l'instant t . Cependant, le module au carré de la fonction d'onde (2.46) est $|A|^2$ qui ne dépend ni de la variable spatiale ni de la variable temporelle. En d'autres mots, cette fonction d'onde représente une particule qui a autant de chances de se trouver n'importe où sur l'axe des abscisses à tout instant. C'est le comportement d'une onde délocalisée. Cela signifie que l'onde plane décrite par l'expression (2.46) ne peut être localisée ni dans l'espace ni dans le temps. Le challenge consiste à chercher une formulation mathématique qui permet à la fonction d'onde d'une particule d'être localisée selon l'axe considéré. C'est-à-dire une particule qui a plus de chances de se trouver à certaines positions sur cet axe, que dans d'autres. Cet objectif est atteint en construisant une combinaison linéaire d'ondes planes dont le vecteur d'onde varie légèrement autour de \vec{k} :

$$\psi(x, t) = \int_{-\infty}^{\infty} A(k) e^{i(kx - \omega t)} dk \quad (2.47)$$

Avec, $A(k)$ désigne l'amplitude des ondes planes de nombre d'onde k dans cette combinaison. En écrivant l'expression précédente, nous nous basons sur l'hypothèse que les ondes de particules sont superposables : c'est-à-dire qu'il est toujours possible d'ajouter deux solutions d'ondes correctes pour former une troisième solution d'ondes correcte. Le paquet d'onde est la région de l'espace où ces ondes interfèrent de manière constructive. Incidemment, une onde plane qui varie comme $e^{i(kx - \omega t)}$ et qui a un vecteur d'onde négatif (sachant que $\omega > 0$) se propage dans la direction des x négatifs à la vitesse de phase $\frac{\omega}{|k|}$. Par conséquent, la

combinaison linéaire (2.47) inclut les ondes se propageant vers l'avant et vers l'arrière. Le théorème de Fourier, stipule que si :

$$\psi(x, t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} A(k) e^{i(kx - \omega t)} dk \quad (2.48)$$

Alors,

$$A(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \psi(x, t) e^{-i(kx - \omega t)} dx \quad (2.49)$$

Avec, $A(k)$ est connue comme la transformée de Fourier de la fonction $\psi(x, t)$. Les équations (2.48) et (2.49) stipulent que plus la particule est non-localisée, plus la dispersion du vecteur d'onde est faible, et inversement. Nous pouvons utiliser ce théorème de Fourier pour trouver la fonction d'onde dans l'espace k (est donc dans l'espace p) qui génère toute fonction d'onde donnée dans l'espace x à un instant donné. Par exemple, supposons qu'à $t = 0$, la fonction d'onde d'une particule donnée s'écrit comme suit :

$$\psi(x, 0) = \exp \left[i k_0 x - \frac{(x - x_0)^2}{4 (\Delta x)^2} \right] \quad (2.50)$$

La probabilité de présence de la particule à l'instant initial s'obtient :

$$|\psi(x, 0)|^2 = \exp \left[-\frac{(x - x_0)^2}{2 (\Delta x)^2} \right] \quad (2.51)$$

Cette distribution de probabilités (équation (2.51)) produit un profil Gaussien. Le maximum de cette distribution se situe autour de x_0 , c'est la position la plus probable de la particule. L'expression (2.50) est la fonction d'onde d'une particule qui est initialement localisée autour de $x = x_0$ dans une région dont la largeur est de l'ordre de Δx . Ce type de fonction d'onde est connu sous le nom de *paquet d'ondes*. Si $|\psi(x)|^2$ représente la probabilité de présence de la particule à la position x , alors $|A(k)|^2$ exprime la densité de probabilité d'avoir un vecteur d'onde \vec{k} . Comme $p = \hbar k$, $|A(k)|^2$ désigne aussi la densité de probabilité de la quantité de mouvement et par conséquent de la vitesse ($p = m v$). Selon l'équation (2.47) :

$$\psi(x, 0) = \int_{-\infty}^{\infty} A(k) e^{i k x} dk \quad (2.52)$$

Nous pouvons utiliser le théorème de Fourier pour inverser cette expression pour obtenir :

$$A(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \psi(x, 0) e^{-i k x} dx \quad (2.53)$$

En utilisant l'équation (2.50), nous obtenons :

$$A(k) = \frac{1}{\sqrt{2\pi}} e^{-i(k-k_0)x_0} \int_{-\infty}^{\infty} \exp \left[-i(k-k_0)(x-x_0) - \frac{(x-x_0)^2}{4(\Delta x)^2} \right] dx \quad (2.54)$$

Ce qui donne le résultat suivant :

$$A(k) = \frac{1}{\sqrt{2\pi}} \exp \left[-i k x_0 - \frac{(k - k_0)^2}{4 (\Delta k)^2} \right] \quad \text{avec} \quad \Delta k = \frac{1}{2 \Delta x} \quad (2.55)$$

$$\Rightarrow |A(k)|^2 \propto \exp \left[-\frac{(k - k_0)^2}{2(\Delta k)^2} \right] \quad (2.56)$$

Cette distribution de probabilité produit un profil Gaussien³ dans l'espace k . Par conséquent, une mesure de k a toutes les chances de donner la valeur centrale k_0 . Comme on peut le constater, la transformée de Fourier d'une forme Gaussienne dans l'espace des positions produit la même forme mathématique dans l'espace des impulsions. Nous venons de voir qu'une distribution de probabilité Gaussienne de largeur caractéristique Δx dans l'espace des positions se transforme en une distribution de probabilité Gaussienne de largeur caractéristique Δk dans l'espace des impulsions :

$$\Delta x \Delta k = \frac{1}{2} \Leftrightarrow \Delta x \Delta p = \frac{\hbar}{2} \quad \text{puisque} \quad \Delta p = \hbar \Delta k \quad (2.57)$$

Un développement mathématique plus rigoureux mettrait en évidence le fait que le produit $\Delta x \Delta k$ est systématiquement de l'ordre de l'unité et ne peut en aucun cas être nul. Cela nous invite à récrire le produit en question selon :

$$\Delta x \Delta k \geq \frac{1}{2} \Leftrightarrow \Delta x \Delta p \geq \frac{\hbar}{2} \quad (2.58)$$

Cette inégalité est connue sous le nom du principe d'indétermination d'Heisenberg. Ce résultat illustre une propriété fondamentale des paquets d'ondes. En effet, si l'on désire former un paquet qui soit très localisé dans l'espace des positions (Δx petit), nous devons combiner des ondes planes avec une très large gamme de valeurs des vecteurs d'onde (Δk grand). En revanche, si nous ne combinons que des ondes planes dont les vecteurs d'onde ont des valeurs très proches (Δk petit), le paquet d'ondes résultant sera très étendu dans l'espace des positions (Δx grand).

```

1  clc ; clear all ; close all ;
2  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3  % Samir KENOUCHE
4  % TRANSFORMEE DE FOURIER DISCRETE
5
6  x = -50:0.1:50 ; M = 400 ; delta_x = 1.0 ;
7  psi_x = (1/sqrt(2*pi)).*exp(-x.^2/delta_x.^2) ; scale = 10*M ;
8
9  for k = 1:M
10     y = 0 ;
11     for ix = 1:numel(x)
12
13         y = y + (1/sqrt(2*pi)).*exp(-x(ix).^2/delta_x.^2).*...
14             exp(-1i*2*pi*(ix-1)*(k-1)/scale) ;
15     end
16     int(k) = y ;
17

```

3. En référence à Johann Carl Friedrich Gauss, mathématicien et physicien Allemand, 1777 - 1855.

```

18 end
19
20 int = [flipdim(real(int),2) real(int)] ;
21
22 fig1 = figure('color',[1 1 1]) ; plot(x,psi_x./max(psi_x)) ;
23 xlabel('Position') ; ylabel('\psi(x,0)') ;
24
25 k = linspace(-M,M,2*M) ; fig2 = figure('color',[1 1 1]) ;
26 plot(k, int./max(int)) ; xlabel('Impulsion') ;
27 ylabel('A(k)') ;
28
29 %%%%%%%%%%%%% ANIMATION DU PAQUET D'ONDE %%%%%%%%%%%%%
30 % MILIEU NON-DISPERSIF
31 offset = 0:300 ; fig3 = figure('color',[1 1 1])
32
33 for j = 1:numel(offset)
34
35     clf ;
36
37     plot(offset(j)+k, int./max(int),'r-','linewidth',2) ; hold on ;
38     title(['Particule at offset = ', num2str(offset(j)), ' ']) ;
39     axis([-numel(offset) numel(offset)+100 -2 2]); f = getframe ;
40
41 end

```

Ci-dessous la visualisation du paquet d'onde obtenu avec le script ci-dessus. Les animations correspondantes doivent être activées ou exécutées au moyen d'un ordinateur.

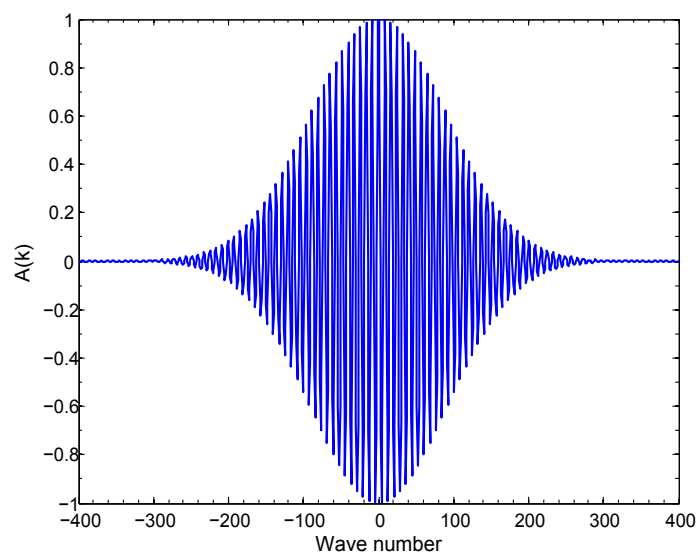


FIGURE 2.6: Visualisation du paquet d'onde.

Notons aussi que le paquet d'onde constitue une solution mathématique de l'équation de Schrödinger. A partir de l'expression de la fonction d'onde (2.48) il en résulte :

$$\frac{\partial \psi(x, t)}{\partial x} = \int_{-\infty}^{\infty} (ik) A(k) e^{i(kx - \omega t)} dk \quad \Rightarrow \quad \frac{\partial^2 \psi(x, t)}{\partial x^2} = \int_{-\infty}^{\infty} (-k^2) A(k) e^{i(kx - \omega t)} dk \quad (2.59)$$

D'un autre côté, la dérivation par rapport à la variable temporelle, donne :

$$\frac{\partial \psi(x, t)}{\partial t} = \int_{-\infty}^{\infty} (-i\omega) A(k) e^{i(kx - \omega t)} dk \quad (2.60)$$

En tenant compte de la formule de dispersion⁴ suivante :

$$\omega(k) = \frac{\hbar k^2}{2m} \quad (2.61)$$

En combinant les équations (2.59), (2.60) et (2.61), nous obtenons :

$$\frac{\hbar}{2m} \frac{\partial^2 \psi(x, t)}{\partial x^2} + i \frac{\partial \psi(x, t)}{\partial t} = \int_{-\infty}^{\infty} \left(\omega - \underbrace{\frac{\hbar k^2}{2m}}_{\omega} \right) A(k) e^{i(kx - \omega t)} dk = 0 \quad (2.62)$$

$$\Rightarrow \quad \frac{\hbar}{2m} \frac{\partial^2 \psi(x, t)}{\partial x^2} = -i \frac{\partial \psi(x, t)}{\partial t} \quad (2.63)$$

En multipliant les deux termes de cette équation par \hbar :

$$\frac{\hbar^2}{2m} \frac{\partial^2 \psi(x, t)}{\partial x^2} = -i \hbar \frac{\partial \psi(x, t)}{\partial t} \quad (2.64)$$

Nous retrouvons l'équation de Schrödinger⁵. Naturellement, l'équation (2.64) ne s'applique qu'aux particules se déplaçant librement. Autrement dit, des particules ne subissant aucune force extérieure. Néanmoins, la généralisation de cette équation pour des particules se déplaçant dans un certain potentiel $V(x)$ est triviale. En présence d'un potentiel $V(x)$ l'énergie⁶ de la particule s'écrit :

$$E = \frac{p^2}{2m} + V \quad \text{sachant que} \quad \frac{p^2}{2m} \equiv -\frac{\hbar^2}{2m} \frac{\partial^2}{\partial x^2} \quad (2.65)$$

Cela conduit à la forme générale (unidimensionnelle) de l'équation de Schrödinger :

$$i \hbar \frac{\partial \psi(x, t)}{\partial t} = -\frac{\hbar^2}{2m} \frac{\partial^2 \psi(x, t)}{\partial x^2} + V(x) \psi(x, t) \quad \equiv \quad i \hbar \frac{\partial \psi(x, t)}{\partial t} = \hat{H} \psi(x, t) \quad (2.66)$$

4. L'enveloppe du paquet d'ondes se déplace à la *vitesse de groupe* $v_g = x/t = d\omega(k)/dk$, qui est concrètement la vitesse de déplacement du maximum central de paquet d'ondes. La célérité c d'une onde unique de vecteur d'onde k est $v_g = c = \omega/k$, qui est nommée *vitesse de phase* et caractérise la vitesse de chaque onde. Lorsque la vitesse de groupe n'est pas constante, il y a dispersion. Dans le cas d'une fonction d'onde d'un électron libre, la relation de dispersion présente une quadratique de type : $\omega(k) = \alpha k^2$. Pour une particule relativiste : $v_g \times v_p = c^2$.

5. Cette équation a été introduite pour la première fois par Erwin Schrödinger en 1926. Il s'agit d'une équation aux dérivées partielles linéaire du second ordre qui régit l'évolution temporelle de la fonction d'onde d'une particule.

6. Rappelons que $p = \hbar k$ et $k = 2\pi/\lambda \Rightarrow E = \hbar \omega = p^2/2m = (\hbar^2 k^2)/(2m)$.

2.5 Conclusion

Dans ce chapitre, nous avons présenté des concepts fondamentaux de la Chimie Quantique abordés lors des deux premiers cycles universitaires. Les fondements théoriques derrière chaque concept sont présentés succinctement afin de ne pas trop alourdir ce chapitre. Néanmoins, cela demeure amplement suffisant pour comprendre les notions de base indispensables à la manipulation des différentes interfaces présentées dans le troisième chapitre. Par ailleurs, quand cela est nécessaire des scripts Matlab[®] sont également écrits afin de consolider le processus de transfert de l'information en privilégiant l'aspect opérationnel.

Présentation et description des interfaces graphiques

Sommaire

3.1	Description des interfaces graphiques développées	40
3.2	Interface principale	40
3.3	Interface de la loi de Planck	41
3.4	Interface de la particule dans une boîte	47
3.4.1	Cas unidimensionnel	47
3.4.2	Cas bidimensionnel	48
3.5	Interface de l'état stationnaire	53
3.6	Interface de l'oscillateur harmonique	56
3.7	Interface des orbitales atomiques	59
3.8	Interface des fonctions d'ondes radiales	65
3.9	Conclusion générale	66

Matlab[®] offre plusieurs possibilités pour créer une interface graphique. Il est possible de convertir directement un script en une interface simple, cette option est pertinente si l'on souhaite partager un script avec des étudiants et leur permettre de modifier des variables à l'aide de commandes prédéfinies. On peut aussi concevoir une interface de manière interactive, il est judicieux de choisir cette option si l'on désire créer une application plus sophistiquée en utilisant un environnement de type glisser/déposer pour construire l'interface utilisateur. Une interface peut également être créée de manière programmatique. Néanmoins, cette option exige des connaissances très avancées en programmation. Dans le cadre de ce projet, nous avons opté pour l'utilisation du module GUIDE (Graphical User Interface Development Environment) pour la conception de toutes les interfaces graphiques. Ce module fournit un excellent environnement pour la création de ce type d'interface de manière interactive, en générant automatiquement le code correspondant. Le code généré contient également des commentaires qui permettent au programmeur de comprendre plus facilement ce que le code

signifie et de mieux organiser les fonctions de rappel. Cette approche facilite le travail du programmeur et lui épargne les codes volumineux. Matlab[®] est donc un outil efficace pour créer des interfaces graphiques. L'objectif de chaque interface est de créer un environnement utilisable, même par des utilisateurs ne maîtrisant pas forcément la programmation. Cela en fait un outil incontournable pour les étudiants ainsi que les enseignants qui le mettent en œuvre dans leurs projets. Le module GUIDE intégré dans Matlab[®] permet de créer des interfaces graphiques de façon conviviale et intuitive. Ce module dispose de tous les outils nécessaires à la création de ce type d'interface. Les commandes sont les boutons figurant sur l'interface sur lesquels l'utilisateur clique pour exécuter une tâche donnée. L'interface graphique est un programme piloté par les scripts implémentés. En effet, il acquiert des données à tout moment et utilise les fonctions de rappel pour exécuter le programme principal et obtenir ainsi des résultats. Dans ce chapitre, nous examinerons les composants utilisés pour concevoir une telle interface. Nous verrons également comment écrire les fonctions de rappel des composants permettant d'exécuter les différentes commandes.

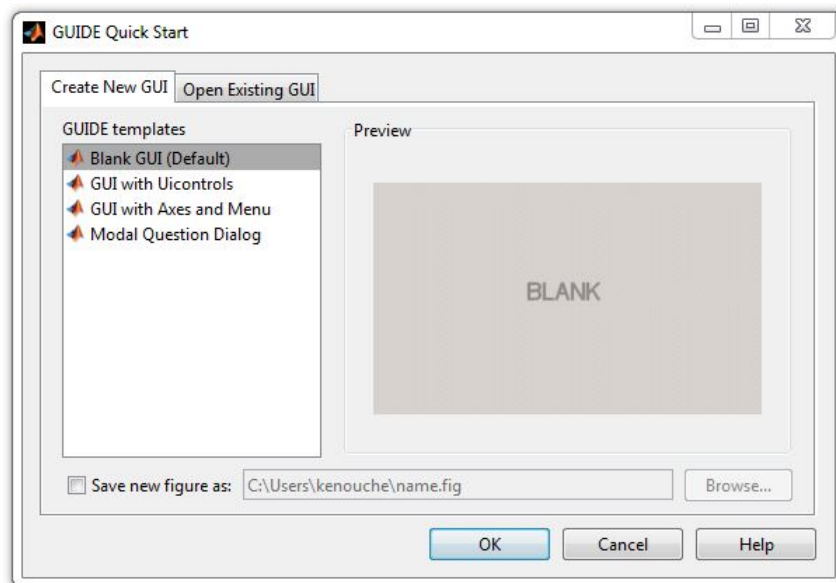


FIGURE 3.1: Lancement du GUIDE.

La disposition des divers objets graphiques est effectuée par sélection dans la palette des composants. Le choix de l'emplacement et le dimensionnement de ces objets sont réalisés au moyen de la souris. Un double-clic sur un objet permet d'activer la fenêtre **Property Inspector** dont les propriétés des objets sont aisément éditables. Les modifications ainsi que les visualisations correspondantes sont effectuées instantanément. Une fois la construction de l'interface graphique est achevée, l'action enregistrement génère deux fichiers ayant le même nom avec deux extensions différentes `*.fig` et `*.m`. Dans le fichier portant l'extension `*.fig`, on trouve des informations sur la définition des objets graphiques (positions et propriétés). Ce fichier peut être ouvert ultérieurement avec le GUIDE pour modifier les objets graphiques. En revanche, dans le fichier `*.m`, on trouve l'algorithme qui contrôle le fonctionnement de l'interface graphique. Ce fichier peut être édité dans l'éditeur du logiciel, pour y intégrer des actions manuellement.

3.1 Description des interfaces graphiques développées

Dans cette section, toutes les tâches des différentes interfaces graphiques sont décrites en détail. Les scripts Matlab[®] sont prodigieusement annotés pour plus de lisibilité et pour simplifier également leur compréhension. A partir du menu et des sous menus correspondants affichés sur la page principale (voir la figure ci-dessous), il est possible d'accéder à toutes les interfaces constituant cette application. Il convient aussi de souligner qu'à partir de l'ensemble des fichiers Matlab[®] (*.m et *.fig), nous avons généré un fichier exécutable (*.exe) de cette application afin qu'elle puisse être utilisée sans avoir besoin d'installer le logiciel Matlab[®].

3.2 Interface principale

La page principale ou la page d'accueil de l'application (ou plus généralement l'interface interactive) est illustrée sur la figure (3.2).

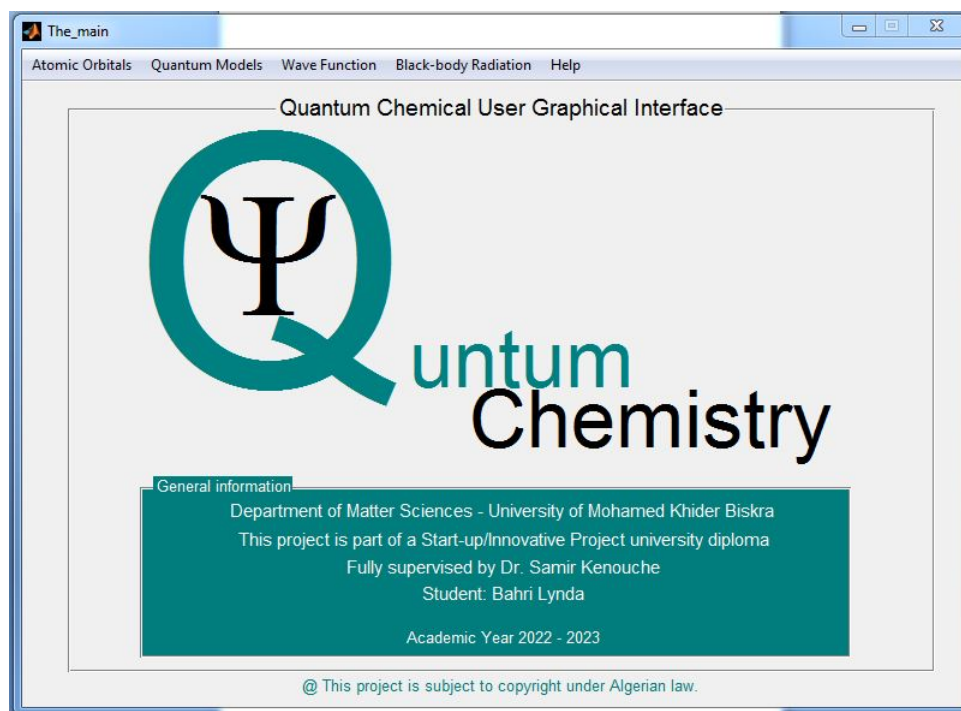


FIGURE 3.2: GUI de l'interface principale.

Désormais nous décrivons le menu principal de cette interface. L'onglet **Atomic Orbitals** contient le menu déroulant suivant : **Atomic Orbitals Visualization**, **Importe Image** et **Quit**. Le deuxième onglet **Quantum Models** contient le menu déroulant suivant : **Particule One Dimension**, **Particule Two Dimension** et **Harmonic Oscilator**. Le troisième onglet **Wave Function** contient le menu déroulant suivant : **Radial Wave Functions** et **Stationnary State**. L'onglet **Black-body Radiation** est formé d'un seul item du menu déroulant qui est **Planck Law**. Le dernier onglet **Help** contient le menu déroulant suivant : **User's Guide** ainsi que **Contact Us**.

3.3 Interface de la loi de Planck

Cette interface (voir figure (3.3)) permet de réaliser plusieurs tâches concernant l'exploitation de la loi de Planck. En introduisant une valeur de la température dans le champ **Temperature in Kelvin** et en cliquant sur **Display**, le profil théorique de cette loi s'affiche sur la figure. De plus, il est possible de choisir l'échelle avec laquelle le graphe en question sera affiché, pour cela il suffit de sélectionner un des champs suivants : **Linear**, **Semi-logarithmic** et **Logarithmic**.

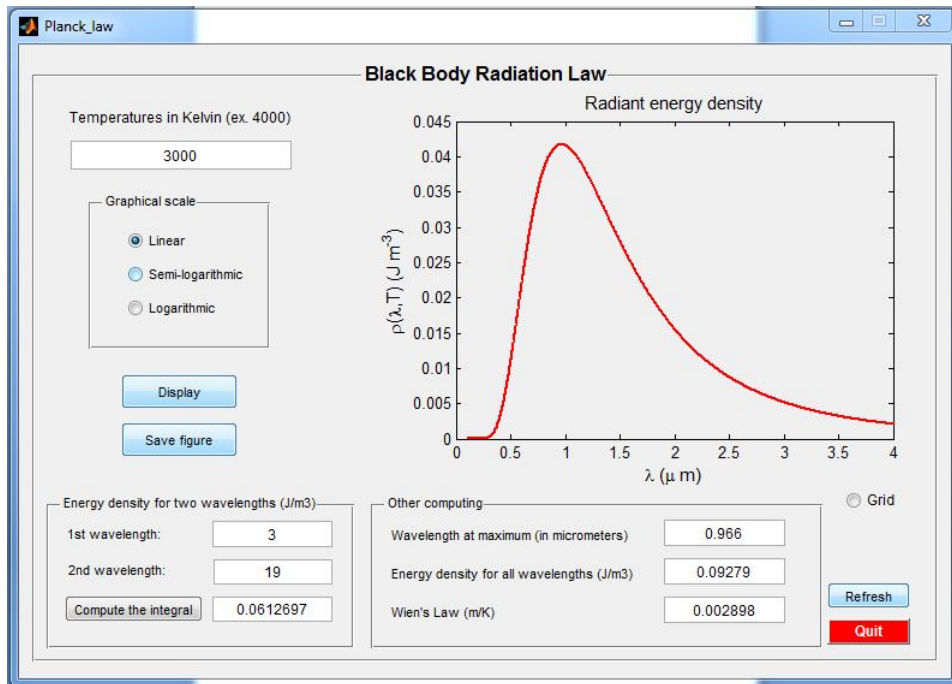


FIGURE 3.3: GUI de la loi de Planck.

En sélectionnant l'une de ces échelles, la boîte de dialogue **Choose the color** s'ouvre immédiatement, invitant l'utilisateur à choisir l'une des couleurs avec laquelle le graphe de la loi de Planck sera tracé (voir figure (3.4)). Le script Matlab[®] ci-dessous, permet l'implémentation de la boîte de dialogue pour la sélection de la couleur.

```

1  s = {'black', 'cyan', 'green', 'red', 'magenta', 'yellow'} ;
2  [ind_oui, ind_non] = listdlg('ListString',s,'promptstring',...
3  'Choose the color','selectionmode','single') ;
4
5  if ind_non == 1 ;
6
7      plot(1e+6*lambda, 1e-6*rho_planck(lambda, T) , 'Color', s{ind_oui}, ...
8      'linewidth',2) ; else plot(1e+6*lambda, 1e-6*rho_planck(lambda, T),...
9      'b' , 'linewidth',2) ;
10 end

```

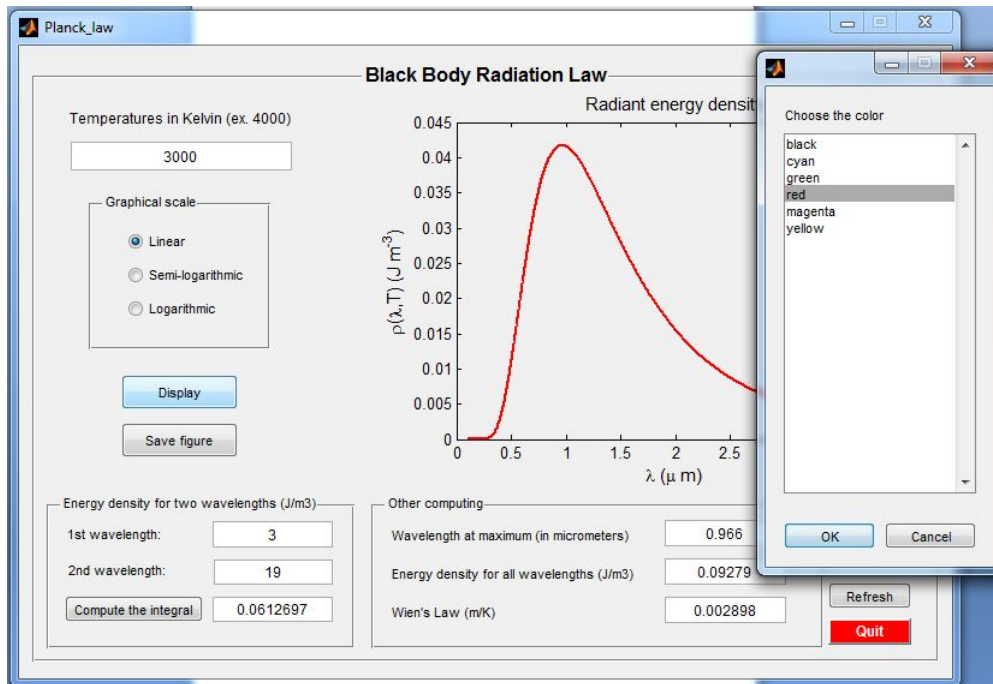


FIGURE 3.4: Sélection de la couleur

La figure ainsi obtenue peut être enregistrée au format *.gif dans le répertoire courant, en cliquant simplement sur **Save figure**. Le script permettant d'effectuer cette tâche est :

```

1  img = getframe(handles.axes3) ; myimage = frame2im(img) ;
2  [myimage_ind, mymap] = rgb2ind(myimage, 256) ;
3  imwrite(myimage_ind, mymap, 'my_figure.gif', 'gif') ;

```

Plusieurs tests de contrôle sont implémentés afin d'automatiser l'interface (3.3), d'augmenter sa robustesse ainsi que son autonomie. Si vous cliquez sur **Display** sans introduire la valeur de la température, le message suivant apparaît sur l'écran :

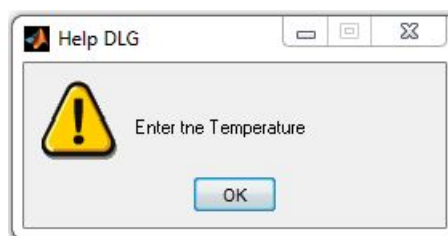


FIGURE 3.5: Message d'aide relatif à l'introduction de la température.

En cliquant sur **OK**, le message se referme. Ci-dessous le script correspondant :

```

1 T = str2num(get(handles.edit1,'String')) ; % TEMPERATURE INPUT
2
3 if (isempty(T) == 1)
4     msgbox('Enter tne Temperature','Help DLG','help') ;
5     uicontrol(hObject) ;
6     return
7 else
8
9 % .... SUITE DES INSTRUCTIONS
10
11 end

```

D'un autre côté, si vous introduisez une valeur alphabétique dans le champ `Temperature in Kelvin`, le message d'erreur suivant s'affiche sur l'écran :



FIGURE 3.6: Message d'erreur relatif à l'introduction d'une valeur numérique.

Ce message d'erreur invite l'utilisateur à introduire une valeur numérique. Ce message est implémenté avec le script suivant :

```

1 input = str2double(get(hObject,'String'));
2
3 if isnan(input)
4
5     errordlg('You must enter a numeric value','Invalid Input','modal') ;
6     uicontrol(hObject) ;
7
8     return
9 else
10     display(input);
11 end

```

Le Panel: `Energy Density for two wavelengths` sert à calculer la densité d'énergie rayonnante en introduisant deux longueurs d'onde dans la gamme $0 - 30 \mu m$. Pour calculer

cette intégrale, il faut absolument introduire les deux bornes de l'intégrale, la borne inférieure λ_1 ainsi que la borne supérieure λ_2 . Si l'un des champs `1st wavelength` et `2nd wavelength` est vide, l'interface renvoie le message d'erreur suivant :



FIGURE 3.7: Message d'erreur relatif aux bornes d'intégration.

Ce message d'erreur est implémenté au moyen du script ci-dessous :

```

1 lb = str2num(get(handles.edit5, 'String')) ; % LOWER BOUND
2 ub = str2num(get(handles.edit6, 'String')) ; % UPPER BOUND
3
4
5 if (isempty(lb) == 1) || (isempty(ub) == 1)
6     errordlg('You must enter the lower and upper bounds to compute', ...
7         'Invalid Input', 'modal') ;
8     uicontrol(hObject) ;
9     return
10 else
11
12 % .... SUITE DES INSTRUCTIONS
13
14 end

```

Le Panel `Other computing` sert à identifier la longueur d'onde maximale du rayonnement, à calculer l'intégrale de la densité d'énergie totale et la loi de Wien ($\lambda_{max} \times T$). En cliquant sur `Refresh` toutes les valeurs des deux Panels `Energy Density for two wavelengths` et `Other computing` sont actualisées afin de lancer de nouveaux calculs. En sélectionnant `Grid`, une grille est ajoutée à la figure. Cette action est réalisée par le biais du script suivant :

```

1 if get(handles.radiobutton7, 'Value') == 1 ;
2
3 get(handles.axes3) ; grid on ; else grid off ;
4
5 end

```


Finalement, un clique sur **Quit**, l'interface renvoie le message de confirmation suivant :

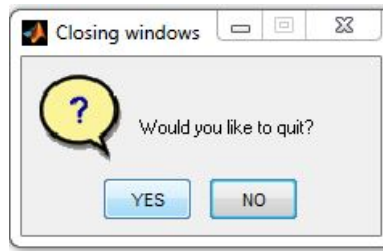


FIGURE 3.8: Fermeture de l'interface.

Ce message invite l'utilisateur à confirmer sa volonté de fermer l'interface. Le script suivant est utilisé à cette fin.

```

1 choice = questdlg('Would you like to quit?', ...
2     'Closing windows', ...
3     'YES', 'NO', 'NO');
4 switch choice
5     case 'YES'
6         close(Planck_law) ; % FERMER L'INTERFACE
7     case 'NO'
8 end
9

```

En cliquant sur OK l'interface se referme instantanément, en revanche, en cliquant sur l'option NO l'interface demeurera active.

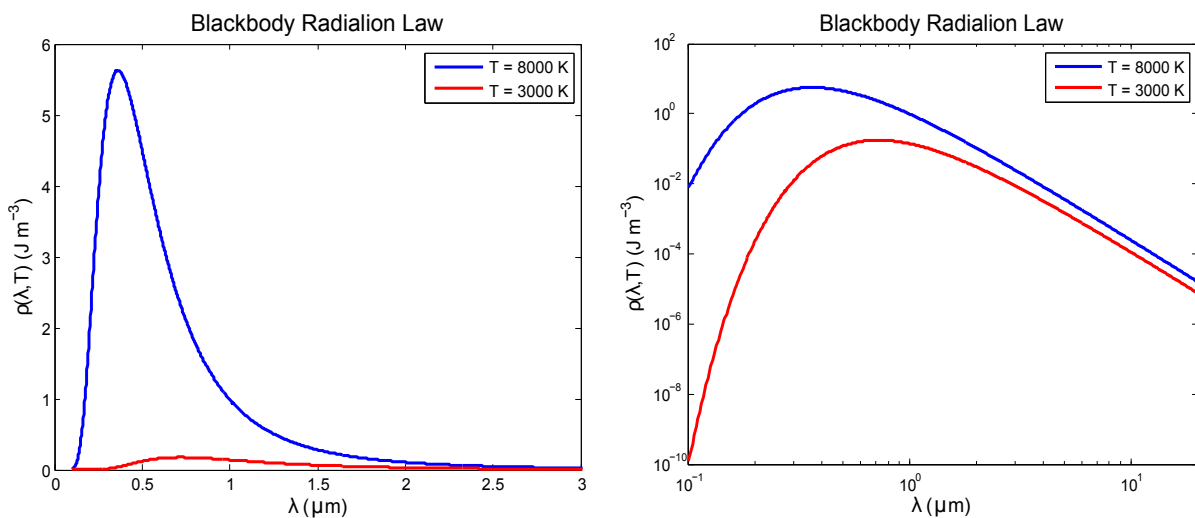


FIGURE 3.9: Échelle linéaire (à gauche) vs échelle logarithmique (à droite).

Dans cette interface, nous avons représenté les deux échelles les plus importantes, à savoir linéaire (espacement par addition) et logarithmique (espacement par multiplication). Une échelle est définie comme un système de graduation d'une droite. L'échelle linéaire est préférable lorsque les écarts entre les données à représenter ne sont pas larges. En revanche, pour des phénomènes représentés par une gamme étendue de valeurs, l'échelle linéaire est mal adaptée. La figure (3.9) ci-dessous, illustre clairement cet aspect. Dans ce cas de figure, l'échelle logarithmique semble plus adaptée de sorte qu'on peut comparer les deux profils. Chaque intervalle est constant selon :

$$\log_{10}(10) - \log_{10}(1) = \log_{10}(100) - \log_{10}(10) = \log_{10}(1000) - \log_{10}(100) = 1 \quad (3.1)$$

La figure ci-dessous est obtenue en faisant un zoom sur l'échelle logarithmique exprimée sur l'axe des abscisses :

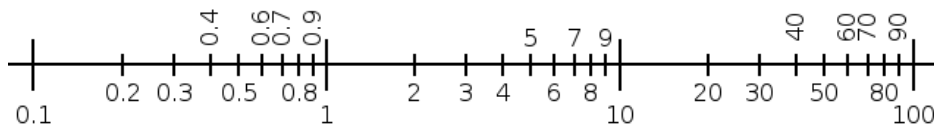


FIGURE 3.10: Échelle logarithmique exprimée sur l'axe des abscisses.

On constate sur cette échelle que les écarts $\log_{10}(x+1) - \log_{10}(x)$ sont décroissants selon :

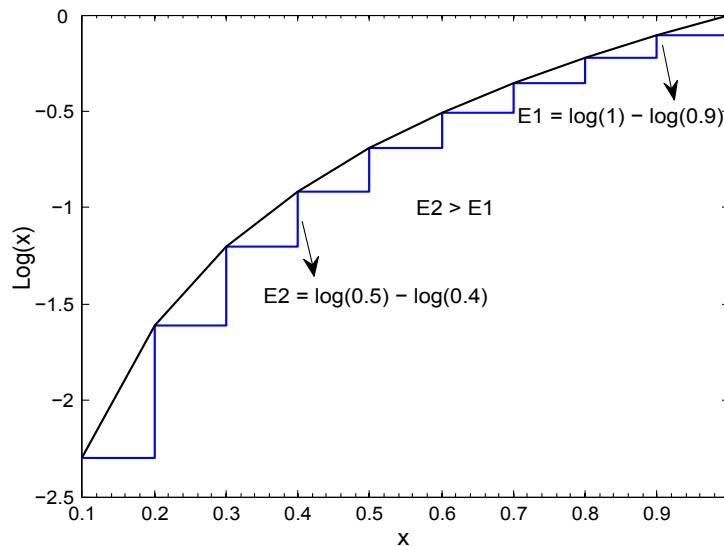


FIGURE 3.11: Illustration de la décroissance des écarts $\log_{10}(x+1) - \log_{10}(x)$.

Selon cette illustration, il apparaît clairement que l'échelle logarithmique permet de séparer les valeurs faibles et de rapprocher les valeurs fortes. Une échelle semi-logarithmique est pertinente pour représenter des phénomènes ayant une évolution selon une ou des exponentielles.

3.4 Interface de la particule dans une boîte

Une particule dans une boîte à une dimension (ou sur un segment de droite) est une approximation fondamentale de la chimie quantique décrivant le mouvement de translation d'une particule confinée à l'intérieur d'un puits de potentiel infiniment profond dont elle ne peut s'échapper. Les solutions du problème donnent les valeurs possibles de E_n (valeurs propres) et $\psi_n(x)$ (vecteurs propres) que la particule peut posséder. Les valeurs propres représentent les énergies autorisées et $\psi_n(x)$ est une fonction d'onde qui, élevée au carré, donne la probabilité de localiser la particule à une certaine position dans la boîte à un niveau d'énergie donné. Nous avons considéré les cas uni- et bidimensionnels.

3.4.1 Cas unidimensionnel

Dans l'interface (3.12) sont implémentés les calculs du modèle quantique de la particule dans une boîte à une dimension. Afin d'exploiter cette interface, il faut d'abord introduire le nombre quantique principal ($n \in \mathbb{N}^*$) dans le champ **Principal quantum number**. En suite, en cliquant sur **Plot the wave function**, la fonction d'onde correspondant au niveau d'énergie E_n s'affiche à gauche de l'interface. Le profil en vert correspond à l'énergie potentielle. Les profils en couleurs bleu et rouge correspondent respectivement aux solutions exacte et numérique de la fonction d'onde. La densité de probabilité de présence est affichée à droite de l'interface en cliquant sur **Plot the probability density**. Les marqueurs \otimes désignent les positions les plus probables de la particule.

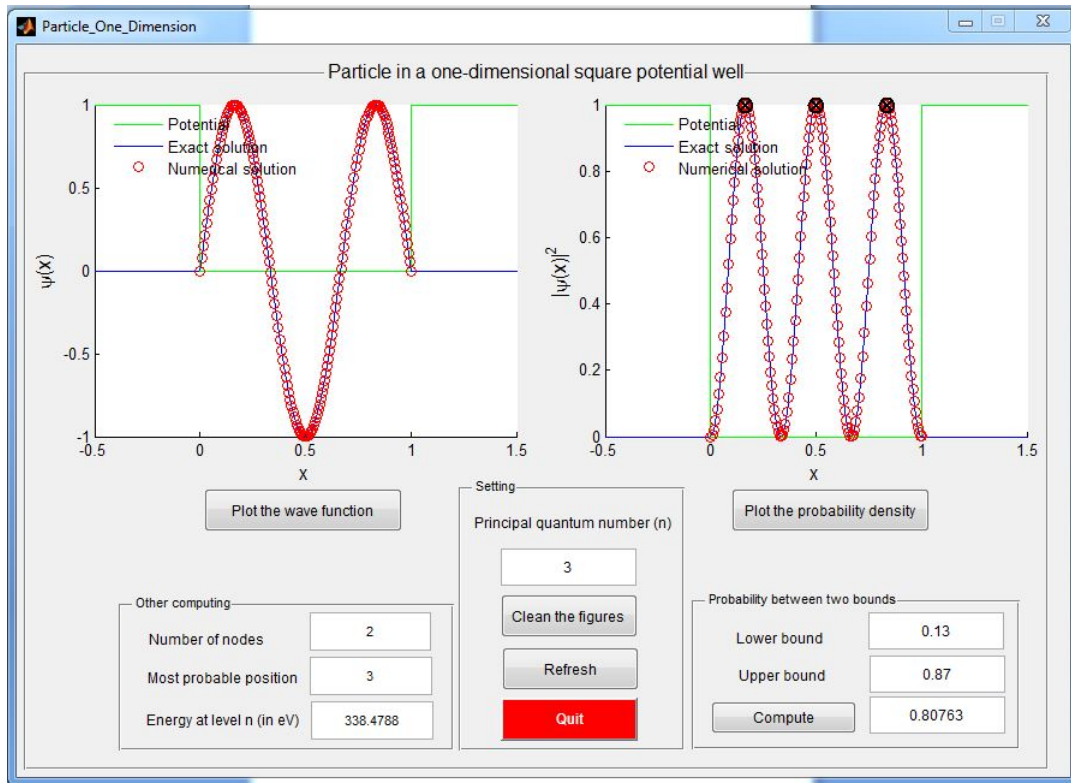


FIGURE 3.12: GUI de la particule dans une boîte unidimensionnelle.

Le Panel **Probability between two bounds** sert à calculer l'intégrale de la densité de probabilité de présence en choisissant deux bornes d'intégration dans la gamme 0 – 1 (en unité de longueur). Dans le Panel **Other computing**, sont affichés le nombre de nœuds (position interdites), les positions les plus probables ainsi que la valeur de l'énergie totale (en électron volt) du niveau considéré. En cliquant sur **Clean the figure**, les deux figures sont effacées pour lancer de nouvelles représentations graphiques. Par ailleurs, si une valeur zéro ou un caractère alphabétique est introduit dans le champ **Principal quantum number**, le message d'erreur suivant s'affiche sur l'écran :

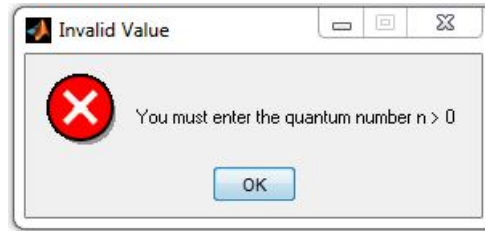


FIGURE 3.13: Message d'erreur par rapport au nombre quantique principal.

Ce message invite l'utilisateur à introduire une entrée valide, soit un entier naturel non nul.

3.4.2 Cas bidimensionnel

Dan ce cas de figure, l'électron est confiné dans une boîte à deux dimensions ($0 < x < a$ et $0 < y < b$). En d'autres termes, dans ce rectangle, la fonction d'onde de l'électron se comporte comme une particule libre ($V(x, y) = 0$), mais les murs sont infranchissables, puisque $\psi(x, y, t) = 0$ aux bords. Étant donné que l'hamiltonien de ce système quantique est la somme de deux termes avec des variables séparées $\hat{H} = \hat{h}_x + \hat{h}_y$, nous exprimons la fonction d'onde bidimensionnelle comme un produit de deux composantes unidimensionnelles indépendantes :

$$\psi_{n_x, n_y}(x, y) = \left(\frac{2}{a}\right)^{1/2} \left(\frac{2}{b}\right)^{1/2} \sin\left(\frac{n_x \pi x}{a}\right) \sin\left(\frac{n_y \pi y}{b}\right) \quad (3.2)$$

Nous avons besoin de deux entiers différents n_x et n_y car les conditions aux bords sont totalement indépendantes et peuvent être satisfaites par deux valeurs différentes (ou similaires) de ces entiers. Les valeurs autorisées de l'énergie totale sont maintenant données par :

$$E_{n_x, n_y} = \frac{\hbar^2 \pi^2}{2mL^2} \left(\frac{n_x^2}{a^2} + \frac{n_y^2}{b^2} \right) \quad (3.3)$$

Notez que les énergies permises dépendent maintenant de deux nombres entiers n_x et n_y au lieu d'un seul. Cela provient des deux conditions aux limites indépendantes dans les zones indépendantes n_x et n_y . Comme dans la boîte unidimensionnelle, les valeurs de n_x et n_y sont des entiers naturels non nuls. Contrairement à l'analogie unidimensionnelle, où les nœuds de la fonction d'onde sont des points où $\psi_n(x) = 0$, des lignes entières peuvent

être nodales (appelées lignes nodales). Par exemple, dans l'état $\psi_{2,1}(x, y)$, il existe une ligne nodale à $\psi_{2,1}(x = a/2, y)$. Le long de la ligne entière $x = a/2$ la fonction d'onde est nulle et indépendante de la valeur de y . La fonction d'onde $\psi_{2,2}(x = a/2, y = b/2)$ possède deux lignes nodales. Le fait que la fonction d'onde $\psi_{n_x, n_y}(x, y)$ est un produit de fonctions d'onde unidimensionnelles :

$$\psi_{n_x n_y}(x, y) = \psi_{n_x}(x) \times \psi_{n_y}(y) \quad (3.4)$$

La densité de probabilité de présence correspondante s'obtient simplement selon :

$$\begin{aligned} P(x \in [0, a], y \in [0, b]) &= \int_0^a dx \int_0^b dy |\psi_{n_x n_y}(x, y)|^2 \\ &= \int_0^a dx \int_0^b dy \psi_{n_x}^2(x) \psi_{n_y}^2(y) \\ &= \left[\int_0^a \psi_{n_x}^2(x) dx \right] \left[\int_0^b \psi_{n_y}^2(y) dy \right] \end{aligned}$$

En guise d'application pour $a = b = 1/2$, alors :

$$\begin{aligned} P(x \in [0, 1/2], y \in [0, 1/2]) &= \left[\int_0^a \psi_1^2(x) dx \right] \left[\int_0^a \psi_2^2(y) dy \right] \\ &= \left[\frac{2}{a} \int_0^a \sin^2\left(\frac{\pi x}{a}\right) dx \right] \left[\frac{2}{a} \int_0^a \sin^2\left(\frac{2\pi y}{a}\right) dy \right] \\ &= \left[\frac{2}{a} \frac{a}{4} \right] \left[\frac{2}{a} \frac{a}{4} \right] \\ &= \frac{1}{4} \end{aligned}$$

Deux fonctions d'onde distinctes sont dites dégénérées si elles correspondent à la même énergie. Si les côtés a et b du rectangle sont tels que a/b est irrationnel (cas général), il n'y aura pas de dégénérescence. Le cas le plus dégénéré est $n_x = n_y$ pour lequel il est clair que $E_{n_x, n_y} = E_{n_y, n_x}$. En théorie quantique, les dégénérescences sont le plus souvent associées à des symétries moléculaires. A titre d'exemple, le premier état excité ($E_{1,2} = E_{2,1}$) est dit doublement dégénéré. Le nombre de fonctions d'onde indépendantes pour les états stationnaires avec une énergie partagée est appelé le *degré de dégénérescence* du niveau d'énergie qui se déduit de : $n_x^2 + n_y^2$. Dans l'interface (3.14) sont implémentés les calculs du modèle quantique de la particule dans une boîte à deux dimensions.

Cette interface exécute les mêmes tâches que dans le cas à une dimension, avec tous les messages d'aide et d'erreur correspondants. La fonction d'onde bidimensionnelle, le nombre de nœuds et l'énergie totale sont affichées en utilisant le script ci-dessous :

```

1 nx = str2num(get(handles.edit1, 'String')) ;
2 ny = str2num(get(handles.edit18, 'String')) ;
3
```

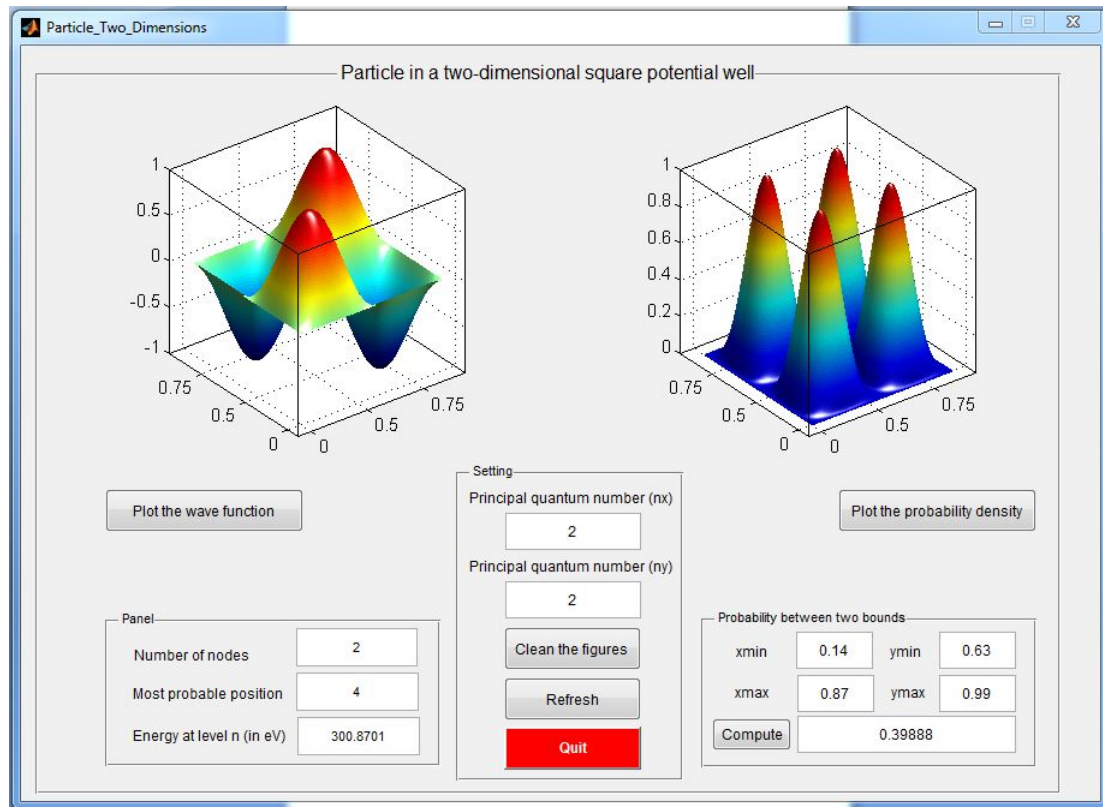


FIGURE 3.14: GUI de la particule dans une boîte bidimensionnelle.

```

4 if (isempty(nx) == 1) || (isempty(ny) == 1)
5     errordlg('You must enter the quantum number n','Invalid Value','modal')
6     uicontrol(hObject)
7     return
8 else
9
10
11 set(gcf,'CurrentAxes', handles.axes1) ;
12
13 a = 1 ; b = 1 ; freqx = nx*pi/a ; freqy = ny*pi/a ; npts = 500 ;
14 lbond = 0 ; ubond = 1.0 ; nstep = (ubond-lbond)/npts ;
15 [xn, yn] = meshgrid(lbond:nstep:ubond) ;
16 Z = sqrt(2./a).*sqrt(2./b).*sin(freqx.*xn).*sin(freqy.*yn) ;
17
18 h1 = mesh(Z./max(max(Z))) ;
19 box on ; axis vis3d; rotate3d on ; camlight; lighting phong ;
20 light('Position',[0 0 1],'Style','infinite') ;
21 ih1 = gca ; % RECUPERATION DES IDENTIFIANTS DES AXES
22 set(ih1,'XTickLabel',{'0','0.5','0.75'},'FontSize',11)
23 set(ih1,'YTickLabel',{'0','0.5','0.75'},'FontSize',11)
24

```

```

25 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% noeuds number %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
26
27 set(handles.edit8,'String', num2str(2*(nx - 1))) ;
28 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Total energy %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
29 me = 9.109*10^(-31) ;      % masss on electron kg
30 h = 6.626176*10.^(-34) ;  % Planck constant
31 a = 1*10^(-10) ;         % convert to m
32 b = 1*10^(-10) ;         % convert to m
33
34 En = (0.125*(h.^2/me)*(nx.^2/a.^2 + ny.^2/b.^2))*6.242*10^18 ;
35
36 set(handles.edit13,'String', num2str(En)) ;
37
38 end

```

La densité de probabilité est affichée en utilisant un script similaire en élevant au carré la fonction d'onde :

```

1 nx = str2num(get(handles.edit1,'String')) ;
2 ny = str2num(get(handles.edit1,'String')) ;
3
4 if (isempty(nx) == 1) || (isempty(ny) == 1)
5     errordlg('You must enter the quantum number n','Invalid Value','modal')
6     uicontrol(hObject)
7     return
8 else
9
10 set(gcf,'CurrentAxes', handles.axes2) ;
11
12 a = 1 ; b = 1 ; freqx = nx*pi/a ; freqy = ny*pi/b ; npts = 500 ;
13 lbond = 0 ; ubond = 1.0 ; nstep = (ubond-lbond)/npts ;
14 [xn, yn] = meshgrid(lbond:nstep:ubond) ;
15 Z = sqrt(2./a).*sqrt(2./b).*sin(freqx.*xn).*sin(freqy.*yn) ;
16
17 density_norm = Z.^2./max(max(Z.^2)) ;
18 h2 = mesh(density_norm) ;
19 box on ; axis vis3d; rotate3d on ; camlight; lighting phong ;
20 light('Position',[0 0 1],'Style','infinite') ;
21 ih2 = gca ; % RECUPERATION DES IDENTIFIANTS DES AXES
22 set(ih2,'XTickLabel',{'0','0.5','0.75'},'FontSize',11)
23 set(ih2,'YTickLabel',{'0','0.5','0.75'},'FontSize',11)
24
25 index = find(density_norm == max(max(density_norm))) ;

```

```

26 pos = xn(index) ; % POSITIONS PROBABLES
27 set(handles.edit9,'String', num2str(length(pos))) ;
28
29 end

```

Le panel Probability between two bounds permet de calculer et d'afficher le résultat de l'intégrale double selon le script ci-dessous :

```

1
2 a1 = str2num(get(handles.edit10,'String')) ;
3 a2 = str2num(get(handles.edit15,'String')) ;
4 b1 = str2num(get(handles.edit14,'String')) ;
5 b2 = str2num(get(handles.edit16,'String')) ;
6
7 if (isempty(a1) == 1) && (isempty(a2) == 1) (isempty(b1) == 1) && (isempty(
   b2) == 1)
8
9     errordlg('You must enter a numeric value ([0 to 1]) to compute','Invalid
   Input','modal')
10
11     uicontrol(hObject)
12
13     return
14
15 else
16
17 a = 1 ; % a = b EN UNITE DE LONGUER
18
19 npts = 4000; lbond = 0 ; ubond = 1 ;
20
21 x = linspace(lbond, ubond, npts) ;
22
23 y = linspace(lbond, ubond, npts) ;
24
25 psi = @(x,y) (4./a*a).*((sin((nn*pi/a).*x)).^2.*(sin((nn*pi/a).*y)).^2);
26
27 int = integral2(psi,a1,a2,b1,b2,'Method','iterated',...
28 'AbsTol',0,'RelTol',1e-10) ;
29
30 set(handles.edit17,'String',num2str(int)) ;
31
32 end

```


3.5 Interface de l'état stationnaire

Comme les fonctions d'onde peuvent très souvent être des fonctions complexes, la signification physique ne peut pas être trouvée à partir de la fonction elle-même puisque la quantité $\sqrt{-1}$ ne représente pas une réalité physique. La signification physique se trouve plutôt dans le produit de la fonction d'onde et de son conjugué complexe, autrement dit, le carré absolu de la fonction d'onde (encore appelé le carré du module) :

$$\psi^*(\mathbf{r}, t)\psi(\mathbf{r}, t) = |\psi(\mathbf{r}, t)|^2 \quad (3.5)$$

Où $\mathbf{r} = (x, y, z)$ est un vecteur spécifiant un point dans l'espace tridimensionnel. Le carré est utilisé, plutôt que le module lui-même, tout comme l'intensité d'une onde lumineuse dépend du carré du champ électrique.

L'interprétation de Max Born, qui est largement acceptée aujourd'hui, est que $\psi^*(\mathbf{r}_i, t)\psi(\mathbf{r}_i, t)dv$ indique la probabilité que l'électron se trouve dans le volume infinitésimal dv situé à \mathbf{r}_i . Selon cette interprétation, la fonction d'onde représente l'amplitude de probabilité, le carré absolu de la fonction d'onde est appelé densité de probabilité, et la densité de probabilité multipliée par un élément de volume dans l'espace tridimensionnel désigne la probabilité. Il existe deux types d'équations de Schrödinger : la version dépendante du temps et la version indépendante du temps. L'équation de Schrödinger dépendante du temps prédit que les fonctions d'onde peuvent former des ondes stationnaires (appelées états stationnaires) qui permettent de résoudre plus facilement l'équation de Schrödinger dépendante du temps pour n'importe quel état.

Les états stationnaires peuvent également être décrits par l'équation de Schrödinger indépendante du temps¹. Il convient toutefois de noter que les solutions de l'équation de Schrödinger indépendante du temps présentent toujours des dépendances temporelles. Les états stationnaires sont déterminés par des solutions de la forme :

$$\Psi(\mathbf{r}, t) = \psi_n(\mathbf{r}) \exp\left(\frac{-jE_n t}{\hbar}\right) \quad (3.6)$$

Ce type de solutions satisfait l'équation de Schrödinger indépendante du temps. Dans ce cas de figure, la densité de probabilité de présence est indépendante du temps :

$$p(\mathbf{r}, t) = \Psi(\mathbf{r}, t) = \left[\psi_n(\mathbf{r}) \exp\left(\frac{-jE_n t}{\hbar}\right) \right] \left[\psi_n(\mathbf{r}) \exp\left(\frac{jE_n t}{\hbar}\right) \right] = |\psi_n(\mathbf{r})|^2 = p(\mathbf{r}, 0) \quad (3.7)$$

La probabilité ne change pas au cours du temps et, pour cette raison, $\psi_n(\mathbf{r})$ est appelé un état stationnaire. Dans un tel état, l'énergie reste constante à la valeur bien définie E_n . L'interface représenté dans (3.15) permet d'illustrer cette notion de l'état stationnaire. Afin d'activer cette interface, il suffit d'introduire le nombre quantique principal dans le champ **Principal quantum number**. Ce nombre doit être un entier naturel non nul.

1. Utilisée uniquement lorsque l'hamiltonien ne dépend pas explicitement du temps

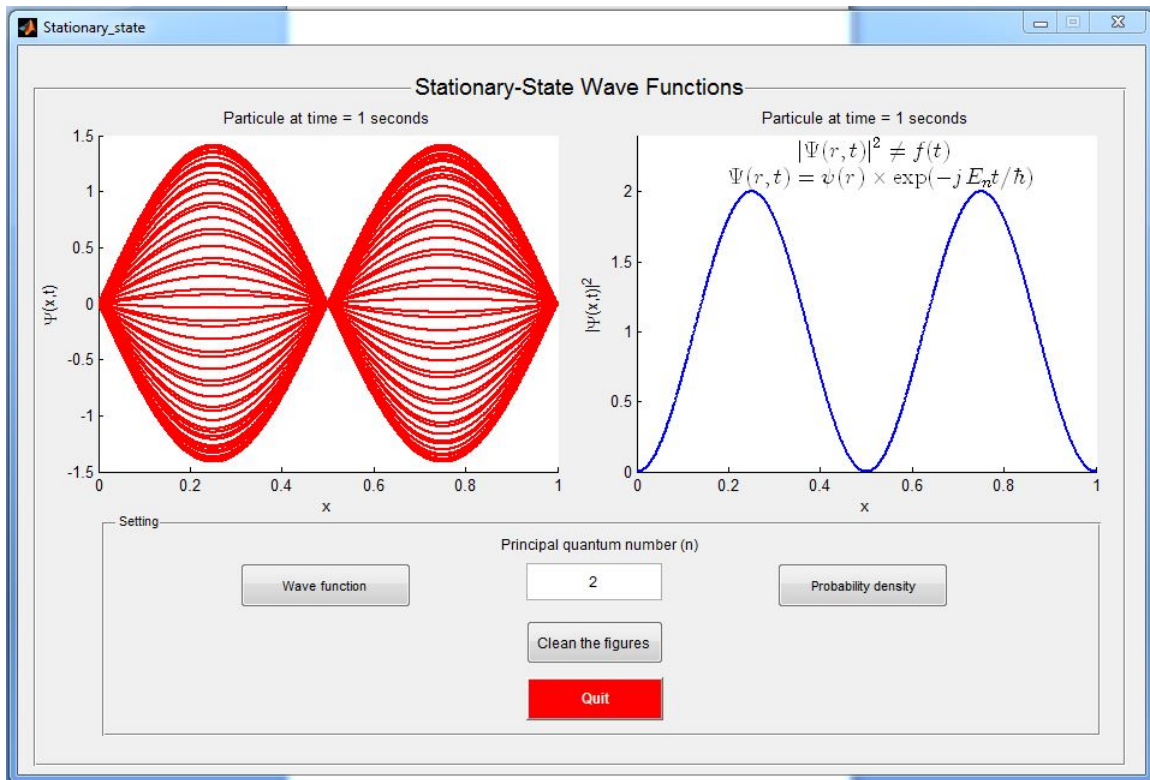


FIGURE 3.15: GUI des états stationnaires.

La solution (3.6) est implémentée par le biais du script ci-dessous :

```

1 n = str2num(get(handles.edit1,'String')) ;
2
3 if (isempty(n) == 1) || n == 0
4     errordlg('You must enter the quantum number n > 0','Invalid Value','modal'
5         )
6     uicontrol(hObject)
7     return
8 else
9     x = 0:0.001:1 ; k = 10.0 ; t = 0:0.02:1 ;
10    a = 1 ;
11
12    wave_fct = @(x,t) sqrt(2/a).*sin((n*pi/a).*x).*exp(1i.*k.*t) ;
13
14    for it = 1:numel(t)
15
16
17        set(gcf,'CurrentAxes', handles.axes4) ;
18
19        hold on ;

```

```

20 plot(x, real(wave_fct(x,t(it))), 'r-', 'LineWidth',2) ;
21 title(['Particule at time = ', num2str(t(it)), ' seconds']) ;
22
23 myanimation1 = getframe ;
24
25 end
26
27 xlabel('x') ; ylabel('\Psi(x,t)') ;
28 end

```

D'un autre côté, l'équation (3.7) a été implémentée en utilisant le script suivant :

```

1 n = str2num(get(handles.edit1, 'String')) ;
2
3 if (isempty(n) == 1) || n == 0
4
5     errordlg('You must enter the quantum number n > 0', 'Invalid Value', 'modal'
6         )
7     uicontrol(hObject)
8     return
9
10 else
11
12 x = 0:0.001:1 ; k = 10.0 ; t = 0:0.02:1 ;
13 a = 1 ;
14
15 wave_fct = @(x,t) sqrt(2/a).*sin((n*pi/a).*x).*exp(1i.*k.*t) ;
16
17 for ik = 1:numel(t)
18
19     set(gcf, 'CurrentAxes', handles.axes6) ; hold on ;
20
21     plot(x, abs(wave_fct(x,t(ik))).^2, 'b-', 'LineWidth',2) ;
22     title(['Particule at time = ', num2str(t(ik)), ' seconds']) ;
23
24     myanimation2 = getframe ;
25
26 end
27
28 text('Interpreter', 'latex', ...
29     'String', '$$ |\Psi(r,t)|^2 \neq f(t)$$', ...
30     'Position', [.35 2.3], ...

```

```

31     'FontSize',14)
32
33 text('Interpreter','latex',...
34     'String','$$ \Psi(r,t) = \psi(r) \times \exp(-jE_n t/\hbar)$$',...
35     'Position',[.2 2.1],...
36     'FontSize',14)
37
38 ylim([0 2.4]) ; xlabel('r \equiv x') ; ylabel('|Psi(x,t)|^2') ;
39 xlabel('x') ; ylabel('|Psi(x,t)|^2') ;
40
41 end

```

Les champs `Clean the figures` et `Quit` sont analogues à ceux décrits dans les interfaces précédentes y compris les messages d'aide et d'erreur.

3.6 Interface de l'oscillateur harmonique

L'oscillateur harmonique est un modèle qui a plusieurs applications importantes en mécanique classique et quantique. Il sert de prototype dans le traitement mathématique de phénomènes aussi divers que l'élasticité, les vibrations moléculaires et les propriétés optiques de la matière. Pour un oscillateur classique, il est possible de déterminer exactement à chaque instant, la position et la quantité de mouvement. Il n'y a aucune restriction sur l'énergie de l'oscillateur, et les variations de l'énergie de l'oscillateur produisent des changements dans l'amplitude des vibrations subies par l'oscillateur. Pour l'oscillateur de la mécanique quantique, l'énergie de l'oscillateur est limitée à certaines valeurs. Les niveaux d'énergie quantifiés autorisés sont également espacés et sont liés aux fréquences de l'oscillateur selon :

$$E_v = \left(v + \frac{1}{2}\right) \hbar\omega = \left(v + \frac{1}{2}\right) h\nu, \quad v \in \mathbb{N} \quad (3.8)$$

L'énergie du point zéro (E_0) désigne l'énergie la plus basse, il s'agit donc de l'énergie de son état fondamental. Seules des valeurs moyennes de la position $\langle \hat{r} \rangle$ et de la quantité de mouvement $\langle \hat{p} \rangle$ qui sont calculables. Les fonctions propres normalisées correspondantes sont données par :

$$\psi_v(x) = \frac{1}{(2^v v!)^{1/2}} \left(\frac{\alpha}{\pi}\right)^{1/4} H_v(\alpha^{1/2}x) e^{-\beta^2 x^2/2} \quad \forall v \in \mathbb{N} \quad (3.9)$$

Où $\alpha = \sqrt{k m/\hbar^2}$ et $H(\alpha^{1/2}x)$ est un polynôme de degré v appelé polynôme d'Hermite. Dans l'interface (3.16) sont implémentés les calculs de ce modèle quantique. Afin d'activer cette interface, il va falloir introduire le nombre quantique de vibration ($v \in \mathbb{N}$) dans le champ `Vibrational quantum number`. En suite, en cliquant sur `Harmonic-oscillator wave function`, la fonction d'onde correspondant au niveau d'énergie E_v s'affiche à gauche de l'interface. Le profil en rouge correspond à l'énergie potentielle quadratique. En revanche, le profil en couleur bleu est associé à la fonction d'onde. La densité de probabilité de présence est

affichée à droite de l'interface en cliquant sur **Harmonic-oscillator probability density**. Les marqueurs \otimes indiquent les positions les plus probables de la masse oscillante.

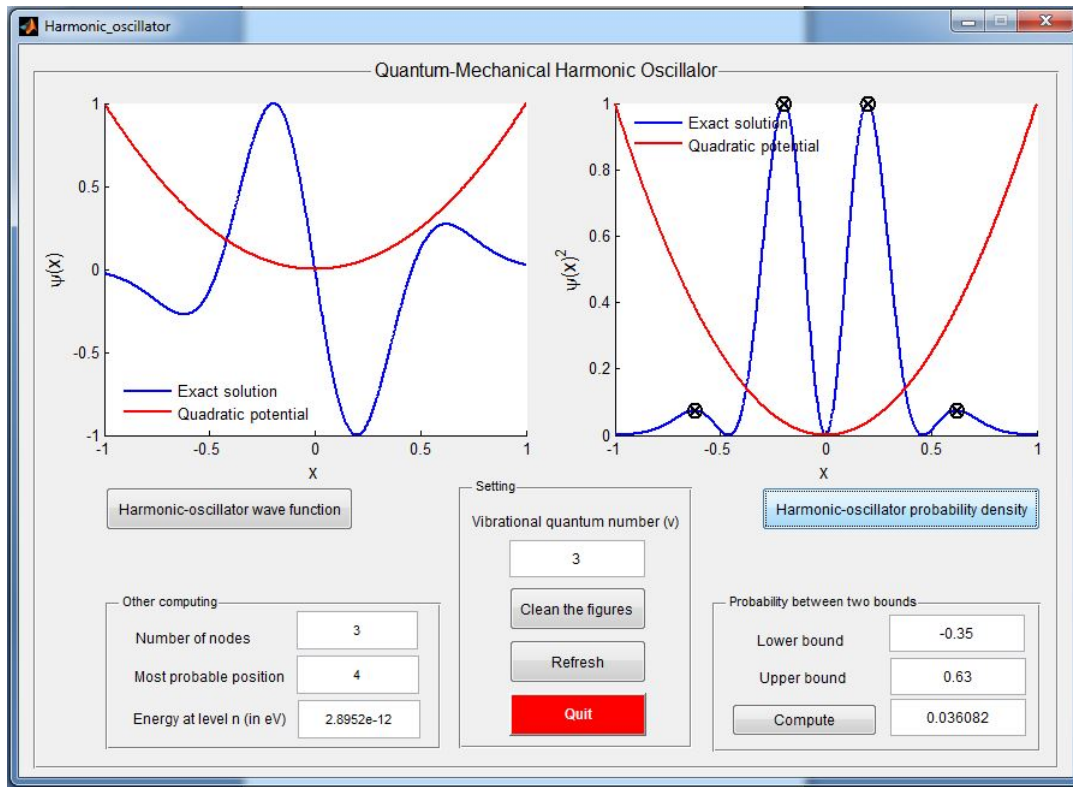


FIGURE 3.16: Gui de l'oscillateur harmonique.

Le Panel **Probability between two bounds** sert à calculer l'intégrale de la densité de probabilité de présence en choisissant deux bornes d'intégration dans la gamme $-1 < x < 1$ (en unité de longueur). Dans le Panel **Other computing**, sont affichés le nombre de nœuds (position interdites), les positions les plus probables ainsi que la valeur de l'énergie totale (en électron volt) du niveau considéré. En cliquant sur **Clean the figure**, les deux figures sont effacées pour lancer de nouvelles représentations graphiques. Comme précédemment, des boites de dialogue sont affichées pour les messages d'erreur. Le calcul de la fonction d'onde (3.9), le nombre de nœuds ainsi que l'énergie de vibration (équation (3.8)) sont obtenus avec le script ci-dessous :

```

1 % Samir Kenouche
2 v = str2num(get(handles.edit1,'String')) ; % NIVEAU DE VIBRATION
3
4 if (isempty(v) == 1)
5     errordlg('You must enter the quantum number v','Invalid Value','modal')
6     uicontrol(hObject)
7     return
8 else

```

```

9
10 set(gcf,'CurrentAxes', handles.axes1) ;      % EMPLACEMENT DE LA FIGURE
11
12
13 m = 1 ; frequence = 200 ; hbar = 1 ; npts = 1000 ;
14 alpha = sqrt(frequence*m)/hbar.^2 ;
15 lbond = -1 ; ubond = 1.0 ; x = linspace(lbond, ubond, npts) ;
16
17 psioscillateur = 1/sqrt(2.^v.*factorial(v)).*...
18     (alpha/pi).^(1/4).*polynome_hermite(v,sqrt(alpha).*x).*...
19     exp(-alpha.*x.^2/2) ;                    % FONCTION D'ONDE
20
21
22 quadratic_pot = 0.5*frequence.*x.^2 ; offset = 0 ;
23
24 quadratic_pot = (quadratic_pot - min(quadratic_pot))./...
25     (max(quadratic_pot) - min(quadratic_pot)) - offset ;
26
27 plot(x,psioscillateur./max(psioscillateur),'LineWidth',2) ;
28 hold on ; plot(x,quadratic_pot,'g','LineWidth',2) ;
29
30
31 xlabel('x','FontSize',12) ; ylabel('\psi(x)','FontSize',12) ;
32 ih1 = legend('Exact solution','Quadratic potential') ;
33 set(ih1,'Interpreter','none','Location','SouthEast','Box','off',...
34     'Color','none')
35 xlim([lbond ubond]) ; box off ;
36
37
38 h = 6.626176*10.^-34 ; % Planck constant
39
40 Ev = ((h*frequence)*(v + 0.5))*6.242*10^18 ; % ENERGIE DE VIBRATION
41
42 set(handles.edit13,'String', num2str(Ev)) ;
43 end

```

Le calcul de la densité de probabilité est mené avec le même script en élevant au carré la fonction d'onde. En outre, le calcul de l'intégrale dans le Panel Probability between two bounds, est mené à travers le script suivant :

```

1 v = str2num(get(handles.edit1,'String')) ; % NIVEAU DE VIBRATION
2 a1 = str2num(get(handles.edit10,'String')) ; % BORNE INFERIEURE
3 a2 = str2num(get(handles.edit11,'String')) ; % BORNE SUPERIEURE

```

```

4
5 if (isempty(a1) == 1) && (isempty(a2) == 1) % TEST DE VALIDITE DES BORNES
6   errordlg('You must enter a numeric value ([-1 to 1]) to compute', 'Invalid
7     Input', 'modal')
8   uicontrol(hObject)
9   return
10 else
11   m = 1 ; frequence = 200 ; hbar = 1 ; npts = 5000 ;
12   alpha = sqrt(frequence*m)/hbar.^2 ; x = linspace(a1, a2, npts) ;
13
14   psioscillateur = 1/sqrt(2.^v.*factorial(v)).*...
15     (alpha/pi).^(1/4).*polynome_hermite(v,sqrt(alpha).*x).*...
16     exp(-alpha.*x.^2/2) ; % FONCTION D'ONDE
17
18     intv = 0 ; % INITIAMISATION DE L'INTEGRALE
19
20 for ik = 1:numel(x)- 1
21   intv = intv + psioscillateur(ik)*((a2-a1)/npts) ;
22 end
23 set(handles.edit12, 'String', num2str(intv)) ;
24 end

```

3.7 Interface des orbitales atomiques

D'après Schrödinger, l'électron dans l'atome d'hydrogène est décrit par l'onde stationnaire $\psi_{n,l,m}(r, \theta, \varphi)$. En effet, selon la théorie quantique, les électrons en orbite autour d'un noyau ne pouvaient pas être entièrement décrits comme des particules, mais doivent satisfaire au concept de dualité onde-particule. Par conséquent, les électrons doivent aussi avoir des propriétés ondulatoires. Les électrons ne gravitent pas autour du noyau à l'image du modèle planétaire, mais existent comme des ondes stationnaires. L'énergie la plus basse qu'un électron puisse prendre est analogue à la fréquence fondamentale d'une onde au sens classique du terme (vibration d'une corde, par exemple). Les états excités (énergies plus élevées) sont semblables aux harmoniques de cette fréquence fondamentale. Les électrons ne peuvent se trouver en un seul point, bien que la probabilité d'interagir avec l'électron en un seul point puisse être déterminée à partir de la fonction d'onde de l'électron. La charge de l'électron se comporte comme si elle était étalée dans l'espace selon une distribution continue, proportionnelle en tout point au carré du module de la fonction d'onde de l'électron. La probabilité élémentaire de trouver l'électron dans une couche de rayon compris entre r et $r + dr$ à différentes directions (θ et ϕ) et distances du noyau (r) est donnée par :

$$dP(r, \theta, \phi) = \psi_{n,l,m}(r, \theta, \phi)^2 dv = \psi_{n,l,m}(r, \theta, \phi)^2 r^2 \sin \theta dr d\theta d\phi \quad (3.10)$$

Cette probabilité donne la répartition du nuage électronique. Afin de trouver l'électron à

une distance r , nous devons intégrer toutes les directions (θ et ϕ) possibles que peut prendre l'électron en se "déplaçant" autour du noyau. Cela se traduit par l'écriture mathématique ci-dessous :

$$dP(r, \theta, \phi) = \{R_{n,l}(r)\}^2 r^2 dr \times \underbrace{\int_{\theta=0}^{\theta=\pi} \int_{\phi=0}^{\phi=2\pi} |Y_l^m(\theta, \phi)|^2 \sin \theta d\theta d\phi}_{=1} \quad (3.11)$$

Qui se réduit juste à la variable radiale. Comme les harmoniques sphériques sont normées, il en ressort :

$$dP(r) = \{R_{n,l}(r)\}^2 r^2 dr \quad \Rightarrow \quad D(r) = \frac{dP(r)}{dr} = \{R_{n,l}(r)\}^2 r^2 \quad (3.12)$$

Donc cette quantité exprime la densité (probabilité divisée par un volume) de probabilité radiale. Notons aussi que la probabilité de trouver un électron à un endroit donné est également fonction de la quantité de mouvement moyenne $\langle \hat{p} \rangle$ de l'électron à cet endroit. Le principe d'incertitude d'Heisenberg, nous enseigne que plus la quantité de mouvement est élevée ($\Delta \hat{p} \rightarrow \infty$) plus la position de l'électron à cet endroit est localisée ($\Delta \hat{x} \rightarrow 0$) (cf. chapitre 2 : propriétés des paquets d'ondes). Le principe d'incertitude de Werner Heisenberg stipule qu'il est impossible de décrire avec une précision arbitraire la position et la quantité de mouvement (donc vitesse) des particules ayant un comportement ondulatoire. En conséquence, les représentations graphiques des orbitales atomiques ne peuvent pas montrer l'ensemble de la région où un électron peut être trouvé, puisque pour les électrons loin du noyau ont des quantités de mouvement relativement faibles donc faiblement localisables en vertu du principe d'Heisenberg. En comprend ainsi que ces représentations graphiques constituent des approximations de surfaces limites où la densité de probabilité $|\psi_{n,l,m}(r, \theta, \varphi)|^2$ a une valeur constante, choisie de manière à ce qu'il y ait une certaine probabilité (par exemple 90 % ou 95 %) de trouver l'électron à l'intérieur de ce *volume de localisation*. L'interface (3.17) est destinée à la visualisations des orbitales atomiques de l'atome d'hydrogène et des ions hydrogénoïdes. Ces orbitales constituent les solutions exactes de l'équation de Schrödinger exprimées en coordonnées sphériques, soit $|\psi_{n,l,m}(r, \theta, \varphi)|^2$ qui désigne la probabilité de trouver l'électron à la position (r, θ, φ) .

Bien que la *distribution de la densité électronique* $|\psi_{n,l,m}(r, \theta, \varphi)|^2$ soit une quantité positive en tout points de l'espace, le signe de la fonction d'onde $\psi_{n,l,m}(r, \theta, \varphi)$ est souvent indiqué dans chaque sous-région de l'orbitale. En effet, la fonction $\psi_{n,l,m}(r, \theta, \varphi)$ est représentée graphiquement pour montrer ses phases, plutôt que $|\psi_{n,l,m}(r, \theta, \varphi)|^2$ qui montre la densité de probabilité mais n'a pas de phase (qui est perdue lorsque l'on prend la valeur absolue, puisque $\psi_{n,l,m}(r, \theta, \varphi)$ est un nombre complexe). Les graphes orbitaux $|\psi_{n,l,m}(r, \theta, \varphi)|^2$ ont tendance à avoir des lobes moins sphériques et plus fins que les graphes $\psi_{n,l,m}(r, \theta, \varphi)$, mais ont le même nombre de lobes aux mêmes endroits. En introduisant les trois nombres quantiques dans le Panel **Setting**, un clique sur **Display the atomic orbital** conduit à la visualisation de l'orbitale atomique en question. Le champ **Refresh the setting** permet l'actualisation des trois nombres quantiques afin de lancer une nouvelle visualisation. Par ailleurs, la fonction d'onde $\psi_{n,l,m}(r, \theta, \varphi)$ en fonction des trois nombres quantiques est calculée au moyen du script

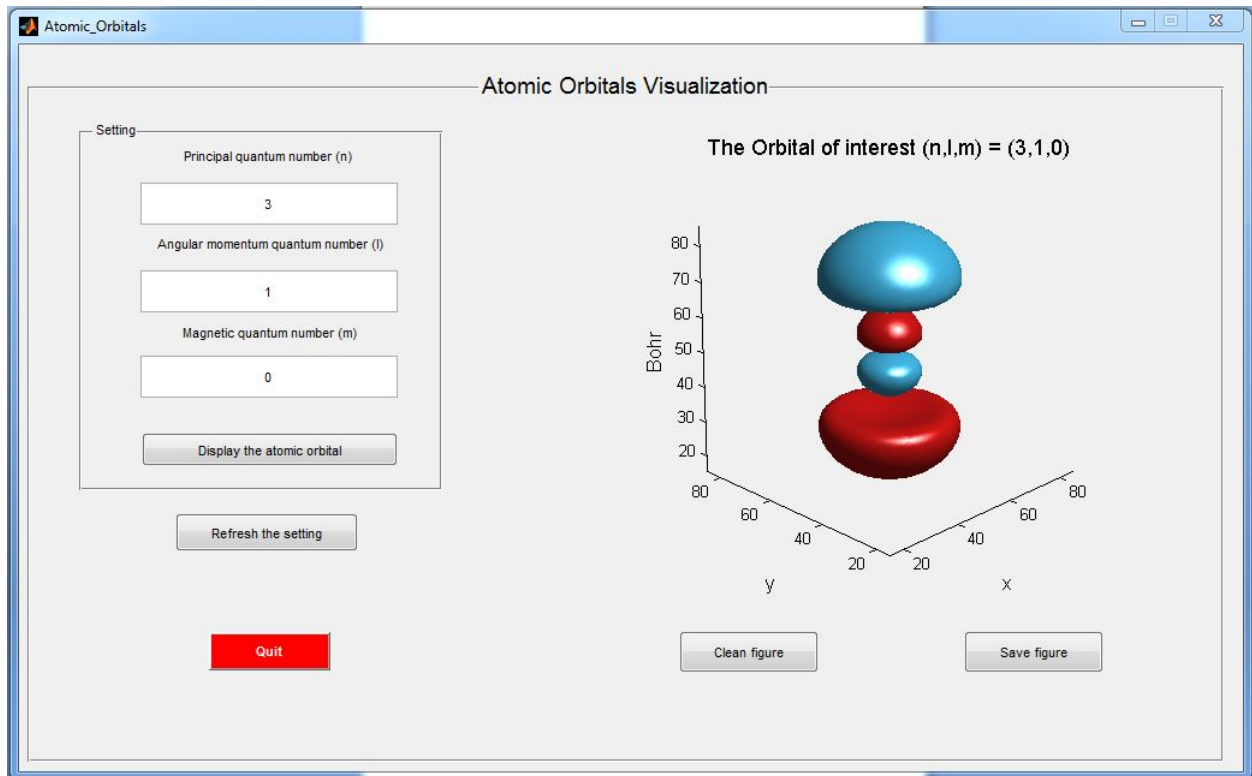


FIGURE 3.17: GUI des orbitales atomiques.

ci-dessous :

```

1 % Samir Kenouche
2 %%%%%%%%%% LES TROIS NOMBRES QUANTIQUES %%%%%%%%%%
3 n = str2num(get(handles.edit4,'String')) ;
4 l = str2num(get(handles.edit5,'String')) ;
5 m = str2num(get(handles.edit6,'String')) ;
6
7 %%%%%%%%%% TEST DE VALIDITE DES ENTREES %%%%%%%%%%
8 if (isempty(n) == 1) || (isempty(l) == 1) || (isempty(m) == 1)
9     msgbox('Enter the valid quantum numbers','Help DLG','help') ;
10    uicontrol(hObject)
11    return
12 else
13
14 set(gcf,'CurrentAxes', handles.axes3) ;      % EMLACEMENT DE LA FIGURE
15
16 %%%%%%%%%% LES HARMONIQUES SPHERIQUES
17 fonct_angulaire = @(l, m, ang_theta, ang_varphi) (-1)^m*...
18     sqrt((2*l + 1)/(4*pi)*factorial(l - abs(m))/factorial(l + abs(m))) * ...
19     polynome_legendre(l,m,cos(ang_theta)).*exp(1i*m*ang_varphi) ;
20

```

```

21 switch m
22     case 0
23         fonct_angulaire = @(l, m, ang_theta, ang_varphi) fonct_angulaire(l, m,...
24             ang_theta, ang_varphi) ;
25     case m > 0
26         fonct_angulaire = @(l, m, ang_theta, ang_varphi) sqrt(2) * (-1)^m *...
27             real(fonct_angulaire(l, m, ang_theta, ang_varphi)) ;
28     otherwise
29         fonct_angulaire = @(l, m, ang_theta, ang_varphi) sqrt(2) * (-1)^m *...
30             imag(fonct_angulaire(l, abs(m), ang_theta, ang_varphi)) ;
31 end
32
33 %%%%%%%%%% FONCTION RADIALE
34
35 fonct_radiale = @(n,l,r) sqrt((2/n).^3.*...
36     factorial(n-l-1)/(2*n*factorial(n + l).^3)).*exp(-r /n).*(2*r/n).^l.*...
37     polynome_laguerre(n-l-1,2*l+1,2*r/n) ;
38
39 %%%%%%%%%% FONCTION D'ONDE
40 fonct_onde = @(n, l, m, r, theta, varphi) fonct_radiale(n,l,r) .*...
41     fonct_angulaire(l, m, theta, varphi) ;
42
43 %%%%%%%%%% DENSITE ELECTRONIQUE
44 densite_electronique = fonct_onde(n, l, m, my_r, my_theta, my_varphi).*...
45     conj(fonct_onde(n, l, m, my_r, my_theta, my_varphi)) ;
46
47 % ..... SUITES DES INSTRUCTIONS
48
49 end

```

Une introduction erronée des nombres quantiques conduit à l'activation de la boîte de dialogue suivante :

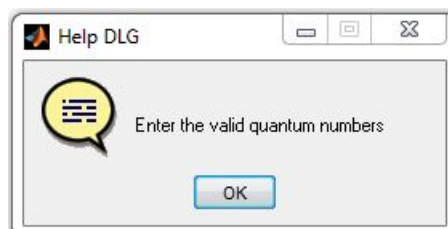


FIGURE 3.18: Vérification des nombres quantiques.

Ce message invite l'utilisateur à introduire des valeurs correctes des trois nombres quan-

tiques. Ci-dessous, le script correspondant :

```

1
2 %%%%%%%%%%%%%%% LES TROIS NOMBRES QUANTIQUES %%%%%%%%%%%%%%%
3
4 n = str2num(get(handles.edit4,'String')) ;
5 l = str2num(get(handles.edit5,'String')) ;
6 m = str2num(get(handles.edit6,'String')) ;
7
8 %%%%%%%%%%%%%%% TEST DE VALIDITE DES ENTREES %%%%%%%%%%%%%%%
9
10 if (isempty(n) == 1) || (isempty(l) == 1) || (isempty(m) == 1)
11     msgbox('Enter the valid quantum numbers','Help DLG',...
12         'help') ;
13
14     uicontrol(hObject)
15
16     return
17
18 else
19
20 % .... SUITES DES INSTRUCTIONS
21
22 end

```

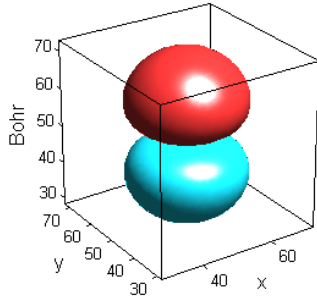
En cliquant sur **Save figure**, une boîte de dialogue s'ouvre pour choisir le répertoire dans lequel la figure est enregistrée. Cette tâche est réalisée par le biais du script suivant :

```

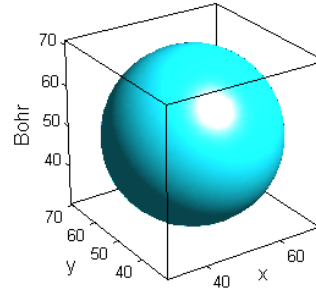
1 marg = 20 ;
2 rect = [marg/17 marg/17 17*marg 19*marg] ;
3
4 set(gca,'Color','none') ; axis off ;
5
6 [file, mypath]=uiputfile({'*.jpeg','JPEG'},...
7 'Save Image As') ;
8
9 im = getframe(handles.axes3, rect) ; [myimage, map]=frame2im(im) ;
10
11 imwrite(myimage,fullfile(mypath, file),...
12 'jpeg') ;

```

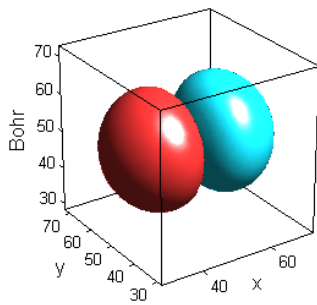
Orbitale de type $(n,l,m) = (2,1,0)$



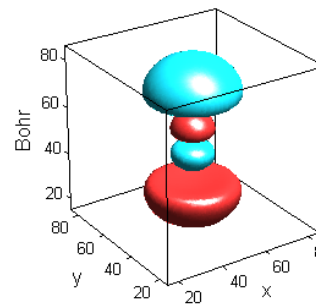
Orbitale de type $(n,l,m) = (2,0,0)$



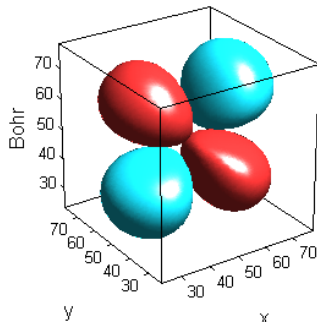
Orbitale de type $(n,l,m) = (2,1,1)$



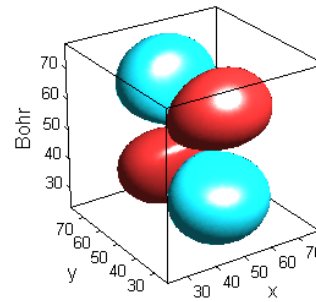
Orbitale de type $(n,l,m) = (3,1,0)$



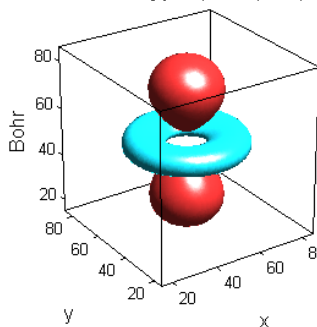
Orbitale de type $(n,l,m) = (3,2,1)$



Orbitale de type $(n,l,m) = (3,2,-1)$



Orbitale de type $(n,l,m) = (3,2,0)$



Orbitale de type $(n,l,m) = (4,1,0)$

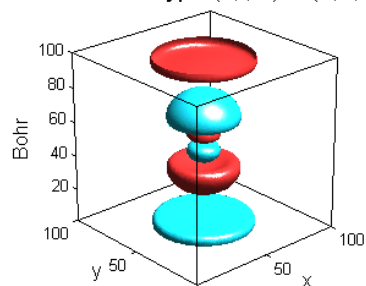


FIGURE 3.19: Représentation de quelques orbitales atomiques.

3.8 Interface des fonctions d'ondes radiales

L'interface (3.20) est destinée à l'exploitation de la partie radiale des fonctions d'ondes. Il suffit d'introduire les nombres quantiques n et l sachant que $0 \leq l \leq n - 1$. En cliquant sur **Plot the wave function**, la fonction d'onde radiale est affichée à gauche de l'interface. En revanche, la densité de probabilité de présence est affichée à droite en cliquant sur **Plot the probability density**. De plus, les nœuds (les positions interdites) sont également affichés.

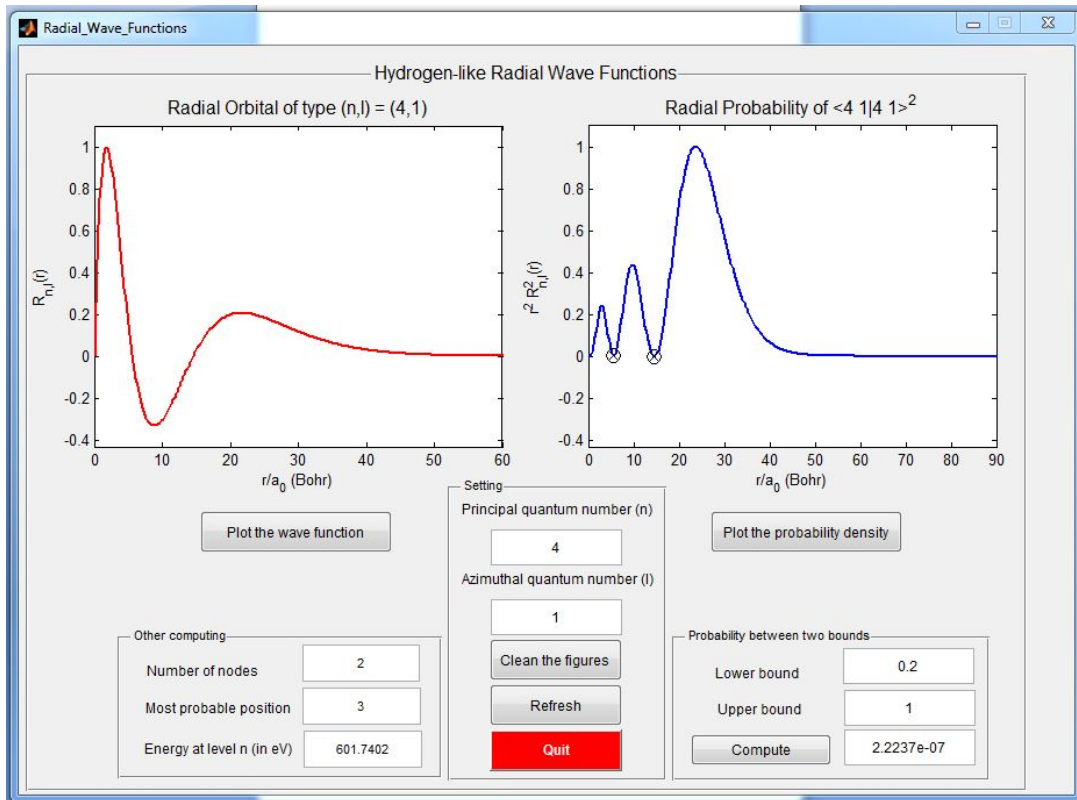


FIGURE 3.20: GUI des fonctions d'ondes radiales.

Le panel **Probability between two bounds** permet de calculer la densité de probabilité de présence en spécifiant les bornes inférieure **Lower bound** et supérieure **Upper bound**. Dans le panel **Other computing**, nous calculons le nombre de nœuds **Number of nodes**, les positions les plus probables **Most probable position** ainsi que l'énergie totale **Energy at level n**.

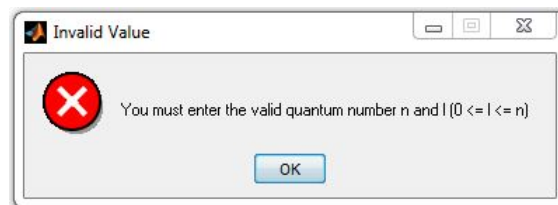


FIGURE 3.21: Message de vérification des nombres quantiques.

L'interface renvoie le message (3.21) dans le cas d'une mauvaise introduction des valeurs des deux nombres quantiques.

3.9 Conclusion générale

La chimie quantique est une discipline théorique pouvant aider les enseignants et particulièrement les étudiants à mieux comprendre la structure et les propriétés des molécules, ainsi que la modélisation des réactions chimiques. En utilisant ses outils conceptuels, les étudiants peuvent mieux comprendre les interactions entre les atomes et les molécules, ce qui peut aider à prédire les propriétés des composés, y compris leur réactivité, leur polarité et leur stabilité. La chimie quantique peut également aider à comprendre comment les molécules interagissent avec la lumière et comment les réactions chimiques se produisent à différentes échelles. Les concepts théoriques de cette discipline peuvent également offrir une perspective plus profonde et plus précise sur les processus chimiques qui se produisent dans le monde qui nous entoure. Développer des interfaces interactives en chimie quantique présente plusieurs avantages. Tout d'abord, cela permet d'effectuer des calculs de manière plus efficace et plus précise. Les interfaces interactives peuvent aider les étudiants à comprendre les données complexes générées par la chimie quantique et à les visualiser plus clairement. Cela peut simplifier les processus d'apprentissage des différentes notions théoriques inhérentes à cette discipline. Ces interfaces peuvent également aider à identifier les erreurs dans les simulations et à ajuster les paramètres pour obtenir des résultats plus précis. En outre, ces interfaces contribuent à rendre la chimie quantique plus accessible et plus compréhensible pour les non-spécialistes, en offrant des outils visuels et interactifs pour explorer les différents concepts théorique.

Cette présente application inscrite dans le cadre d'un projet Start-up est conçue entièrement dans cette perspective. Les étudiants en sciences fondamentales rencontrent souvent des difficultés pour assimiler et appréhender les concepts théoriques inhérents à la Chimie Quantique. C'est un module fondamental pour le cursus universitaire et la formation des étudiants dans les domaines théoriques. Cette discipline théorique fait appel à des notions mathématiques relativement difficiles à saisir pour des étudiants n'ayant pas une ossature solide en mathématiques. C'est à partir de ce constat que l'idée de la conception d'une application a émergé. Cette dernière formée à partir de multiples interfaces graphiques aura pour objectif de faciliter certains calculs usuels de Chimie Quantique abordés lors des deux premiers cycles universitaires. Cette application offrira un environnement intégré permettant une meilleure compréhension des fondements théoriques liés à cette théorie quantique, abordés dans cette discipline durant les deux premiers cycles universitaires. La conception de cette application a été motivée par la volonté de mettre à la disposition des étudiants, un certain nombre d'outils pratiques afin de stimuler leurs connaissances opérationnelles. Nous avons fait le choix de développer cette application au moyen de Matlab[®]. Ce logiciel est très puissant et offre plusieurs possibilités afin de mettre au point ce type d'interface. Cette application utilisée dans une démarche d'apprentissage, servira comme un outil pédagogique intégré dans un mécanisme de formation, aux seins des établissements universitaires.

En guise de perspective, nous ambitionnons d'intégrer d'autres concepts théoriques afin de couvrir de manière exhaustive toutes les notions fondamentales liées à ce module.

Références

- S. Kenouche, *Méthodes numériques et programmation*, Office des Publications Universitaires (OPU), Alger, 2022.
- J. Louis Merrien, *Analyse numérique avec Matlab*. Edition Dunod, 2007.
- S. Kenouche, *Méthodes mathématiques pour la physique*, Office des Publications Universitaires (OPU), Alger, 2023.
- D. A. McQuarrie, *Quantum Chemistry*. Second edition. University Science Books, 2008.
- D. J. Griffiths, *Introduction to Quantum Mechanics*. Prentice Hall, Englewood Cliffs, New Jersey, 1995.
- A. J. Austin, *Studies in Computational Quantum Chemistry*. MedCrave Group LLC, 2016.
- K. I. Ramachandran, G. Deepa, K. Namboori, *Computational Chemistry and Molecular Modeling*. Springer 2008.
- W. Koch and M.C. Holthausen, *A Chemist's Guide to Density Functional Theory*. WILEY-VCH, 2001.
- A. J. Austin, *Studies in Computational Quantum Chemistry*. MedCrave Group LLC, 2016.
- K. I. Ramachandran, G. Deepa, K. Namboori, *Computational Chemistry and Molecular Modeling*. Springer 2008.
- C. Brezinski, M. Redivo-Zaglia, *Méthodes numériques itératives*. Édition Ellipses, 2006.

Résumé : A travers ce mémoire, inscrit dans le cadre d'un diplôme universitaire Start-up/Projet Innovant, nous souhaitons concevoir une interface interactive couvrant les calculs usuels inhérents à la Chimie Quantique. Cette interface offrira un environnement intégré permettant de faciliter la compréhension des concepts théoriques abordés dans cette discipline durant les deux premiers cycles universitaires. La conception de cette interface est motivée par la volonté de mettre à la disposition de la communauté universitaire, d'un outil pédagogique et pratique afin de stimuler les connaissances opérationnelles. Nous envisageons de développer cette application au moyen du logiciel Matlab. Ce logiciel est très robuste et offre plusieurs possibilités afin de mettre au point ce type d'interface. L'utilisation de cette application dans une démarche d'apprentissage, servira comme un outil pédagogique intégré dans un mécanisme de formation aux seins notamment des établissements universitaires.

Abstract: Through this dissertation, registered within the framework of a Start-up/Innovative Project university diploma, we wish to design an interactive interface covering the usual calculations inherent to Quantum Chemistry. This interface will provide an integrated environment to facilitate the understanding of theoretical concepts discussed in this field during the first two undergraduate cycles. The design of this interface is motivated by the intention of providing the university community with a pedagogical and practical tool to stimulate operational knowledge. We intend to develop this application using Matlab software. This program is very robust and offers several possibilities to design this type of interface. The use of this application in a learning process, will serve as a pedagogical tool integrated in a training mechanism especially within academic institutions.

ملخص: من خلال هذه الأطروحة المسجلة وفقاً لمشروع مبتدئ / مبتكر بدرجة جامعية، نرغب في تصميم واجهة تفاعلية تغطي الحسابات المعتادة لكيمياء الكم. ستوفر هذه الواجهة بيئة متكاملة لتسهيل فهم المفاهيم النظرية التي تم تناولها في هذا التخصص خلال الدورات الجامعية الأولى. يتيح تصميم هذه الواجهة إمكانية تزويد الجامعات بأداة تعليمية وعملية لتحفيز المعرفة التشغيلية. نحن نخطط لتطوير هذا التطبيق باستخدام برنامج ماتلاب. هذا البرنامج فعال للغاية ويوفر العديد من الاحتمالات لتطوير هذا النوع من الواجهة. استخدام هذا التطبيق في عملية التعلم، سوف يخدم كأداة تعليمية مدمجة في آلية التدريب خاصة في الجامعات.

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة محمد خيضر-بسكرة

عنوان المشروع:

تصميم واجهة تفاعلية مخصصة لحسابات كيمياء الكم

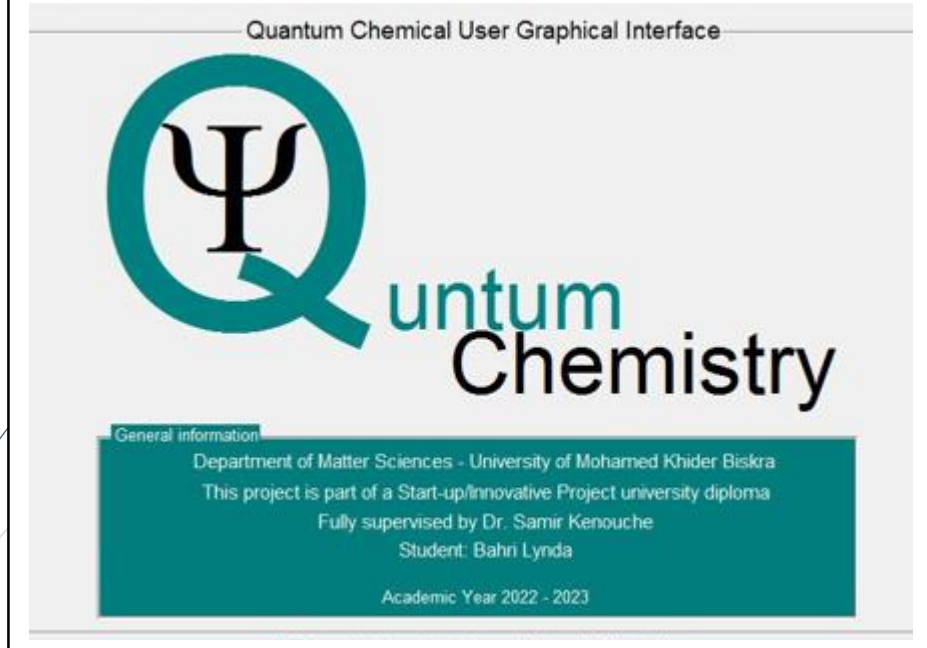
**Conception d'une interface interactive dédiée aux calculs de
chimie quantique**

مشروع لنيل شهادة مؤسسة ناشئة في إطار القرار الوزاري 1275

الاسم التجاري

PsiQuantumChem

العلامة التجارية:



السنة الجامعية

2023 _ 2022

بطاقة معلومات:

حول فريق الاشراف وفريق العمل

1- فريق الاشراف:

فريق الاشراف	
المشرف الرئيسي (01): كنوش سمير	التخصص: كيمياء الكم والفيزياء

2- فريق العمل:

فريق المشروع	التخصص	الكلية
الطالبة: بحري ليندة	كيمياء المواد	علوم الدقيقة وعلوم الطبيعة والحياة

فهرس المحتويات

المحور الأول: تقديم المشروع

المحور الثاني: الجوانب الابتكارية

المحور الثالث: التحليل الإستراتيجي للسوق

المحور الرابع: خطة الإنتاج والتنظيم

المحور الخامس: الخطة المالية

المحور السادس: النموذج الاولي التجريبي

المقدمة:

يعد مشروع المؤسسة الناشئة مكونا مهما في التطوير في مختلف المجالات، حيث يوفر فرصا للإفراد المبدعين والشركات الناشئة لتحقيق أفكارهم وتطوير منتجات وخدمات جديدة، في ظل المشاكل والصعوبات التي يعاني منها الطلبة في مقياس كيمياء الكم ونظرا إلى تطور الأجهزة الذكية التي تعد اليوم رفيفا للطلبة الجامعيين، فهي توفر على الطلبة الكثير من الجهد والوقت في البحث والتعليم. ونظرا إلى هذه الظروف قمنا بإنشاء برنامج تعليمي لحسابات كيمياء الكم يتكون من عدة واجهة تفاعلية لتسهيل المفاهيم النظرية للمقياس في أقل مجهود وأسرع وقت، إن هذا لبرنامج سيكون أداة تعليمية متكاملة سهلة الإستخدام في الجامعات و مراكز البحث العلمي و التعليم العالي، نتمنا أن يلقا مشروعنا الدعم لينمو ويتطور أكثر في المستقبل.

المحور الأول: تقديم المشروع " تصميم واجهة تفاعلية مخصصة لحسابات كيمياء الكم "

1. فكرة المشروع

2. القيم المقترحة

3. فريق العمل

4. أهداف المشروع

5. جدول زمني لتحقيق المشروع

1. فكرة المشروع (الحل المقترح): يواجه طلاب العلوم الأساسية صعوبات في إستيعاب وفهم المفاهيم النظرية في

كيمياء الكم التي هي مقياس أساسي لمنهاج الجامعة وتدريب الطلاب في المجالات النظرية. يشمل هذا المقياس مفاهيم رياضية يصعب إستيعابها بنسبة لطلاب غير الرياضيين. من هنا ظهرت فكرة المشروع وهي تصميم برنامج مكون من واجهات رسومية متعددة بهدف تسهيل بعض العمليات الحسابية المعتادة في كيمياء الكم وتزويد الطلاب بعدد من الأدوات العلمية لتحفيز معرفتهم التطبيقية في المقياس الذي تنوله خلال الدورات الجامعية الأولى. سيوفر هذا البرنامج بيئة تكاملية لتبسيط المفاهيم النظرية (النماذج الكمية المعتادة وحسابات احتمالات وجود الإلكترون، وحساب وتصوير المدارات الذرية، الدقة العددية للمعادلات التفاضلية، والتكامل العددي، وما إلى ذلك). ونحن نخطط إلى تطوير البرنامج بإستخدام المطالب الذي يوفر العديد من الأليات التي تساعدنا في تطوير هذا النوع من الواجهات التفاعلية.

يتمتع مشرفي بخبرة عدة سنوات في إستخدام هذا البرنامج، ونريد الإستفادة من هذه التجربة للإكمال هذا المشروع (هذا البرنامج) الذي سيكون أداة تعليمية متكاملة في آلية التدريب داخل المؤسسات الجامعية و البحث العلمي والتعليم العالي.

بعد تصميم هذا البرنامج للجامعات ومراكز البحث العلمي فكرة مبتكرة بدأنا بالبحث عن إحتياجات المؤسسات الأكاديمية لتخصص كيمياء الفيزياء، ثم نطمح لإيجاد التمويل والمستثمرين للشركة من أجل ضمان نموها ونجحها.

2. القيم المقترحة:

كيمياء الكم هي مجال علمي واسع، ونحن على يقين من أن برنامجنا سيكون مفيد جدا في العديد من المجالات، مثل الأبحاث الصيدلانية وتصميم المواد الجديدة وخاصة كأداة تعليمية تفاعلية. وعلى حد علمنا هذه المرة الأولى التي يتم في تقديم هذه الأداة التعليمية إلى الجامعات ومراكز البحث العلمي والتعليم العالي ولقد تم تصميم هذا التطبيق بكامله بواسطة المهارات الوطنية، وهذه الميزة لا يمكن إنكارها للحماية من التبعية الأجنبية. ومن مزايا هذا البرنامج أنه متاح بسعر تنافسي، وملائم للاحتياجات المحلية، وكذا سهولة الوصول إليه واستخدامه من دون الحاجة إلى إتقان برمجة الكمبيوتر. مما يؤدي ذلك إلى تبسيط كيمياء الكم لطلاب والمدرسين. فكرتنا مبتكرة من حيث تبسيط عملية المحاكاة وتحليل البيانات والتفاعل لمساعدة الطلاب على تطبيق ما تعلموه في الفصل بطريقة ملموسة.

- **الحدثة:** برنامج جديد لم يسبق إنجازه من قبل في مجال كيمياء الكم.
- **الأداء:** لنا طموح في أن يساعد هذا البرنامج في تسهيل وتبسيط المفاهيم وإعطاء قيم حسابية دقيقة ويقدم حلول لمشاكل الطلبة على مستوى مقياس كيمياء الكم، كذا سهولة في التنزيل واستخدام الجهاز، والمساعدة في تطور الكيمياء عموما و كيمياء الكم خصوصا التي هي مقياس أساسي ومكمل لمقاييس أخرى في المجال.
- **التصميم:** بسيط وسهل الاستعمال والاستيعاب.
- **سهولة الوصول:** البرنامج متاح على منصة تفاعلية حيث يمكن تنزيله بسهولة بعد دفع مستحقاته المالية.

3. فريق العمل:

فريق المشروع	التخصص	الكلية
الطالبة: بحري ليندة	كيمياء المواد	علوم الدقيقة وعلوم الطبيعة والحياة
المبرمج: الدكتور كنوش سمير	كيمياء الكم والفيزياء	علوم الدقيقة وعلوم الطبيعة والحياة

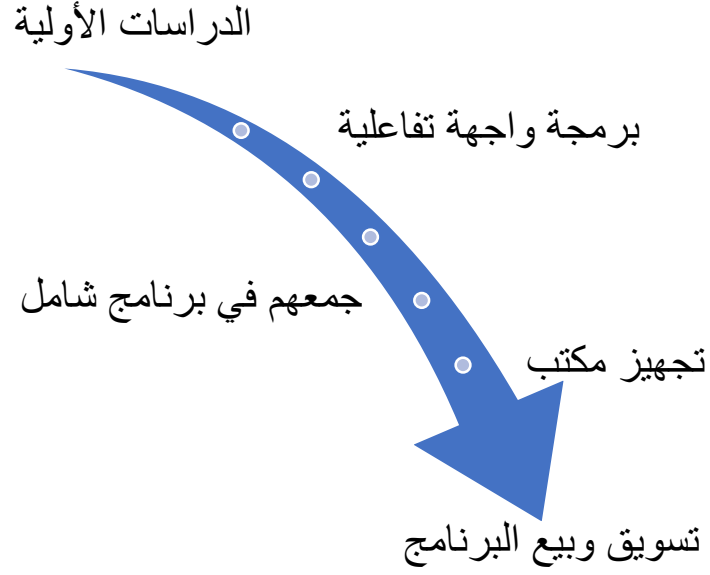
4. أهداف المشروع:

نسعى إلى حل مشاكل الطلبة في مقياس كيمياء الكم وتقديم إضافة في مجال الكيمياء على مستوى الجامعات الجزائرية، وكذا الحصول على وسم لبال وبرأة الإختراع والحصول على التمويل المادي، نجاح هذا المشروع يكون بداية لمشاريع وبرامج أخرى في المستقبل.

5. جدول زمني لتحقيق المشروع:

الأعمال	1 جانفي	2 فيفري	3 مارس	4 أبريل	5 ماي	6 جوان	7 جويلية
1 الدراسات الأولية							
2 برمجة واجهة تفاعلية							
3 جمعهم في برنامج شامل							
4 تجهيز مكتب							
5 تسويق وبيع البرنامج							

الانتقال الزمني لتحقيق المشروع:



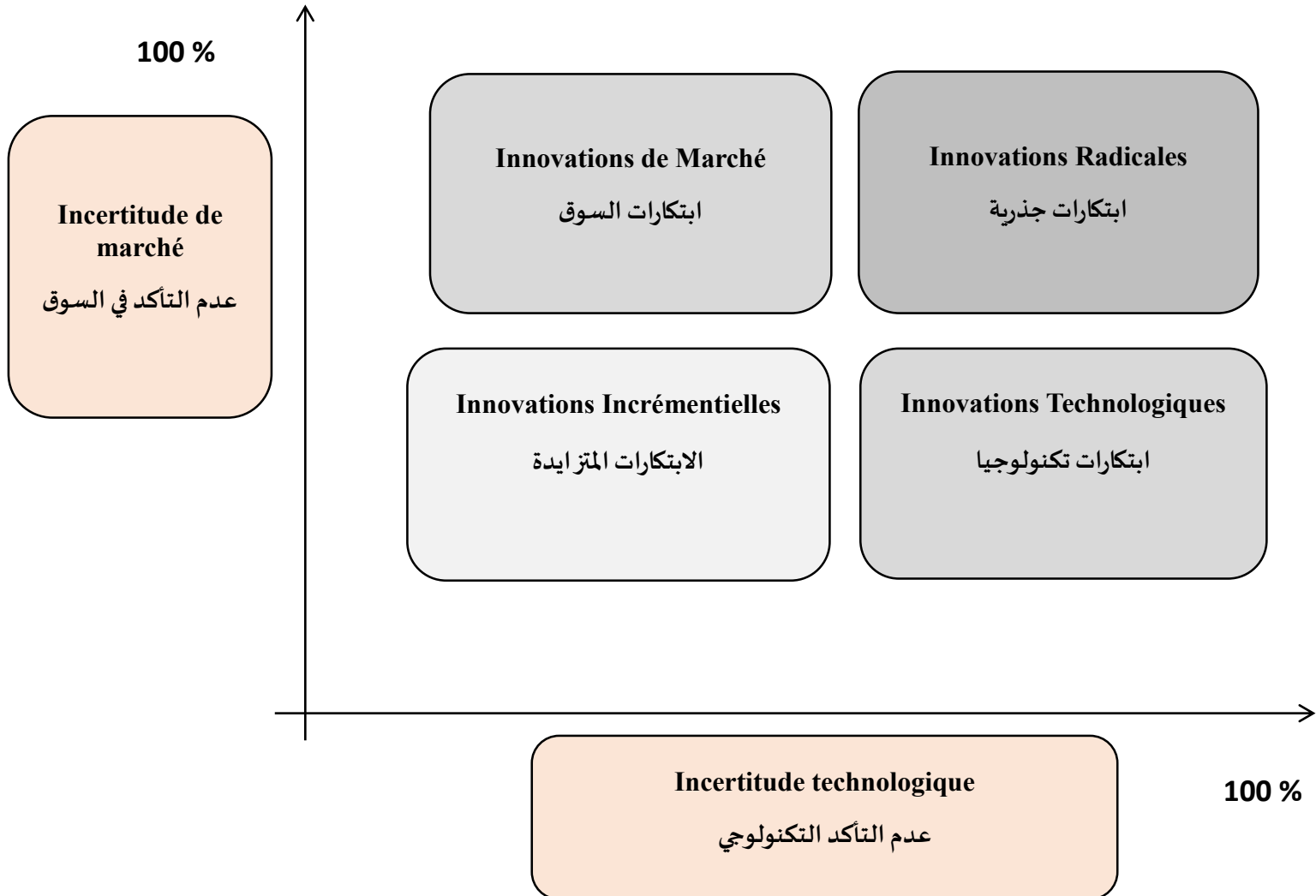
المحور الثاني: الجوانب الابتكارية

1. طبيعة الابتكار

2. مجالات الابتكار

1. طبيعة الابتكار: واجهات تفاعلية تساعد الطلبة في فهم مقياس كيمياء الكم وتجسد قوانينها وإعطاء الحاسبات والمنحنيات البيانية التي تتغير بتغير المعطيات وتسهيل المفاهيم النظرية، وهو ابتكار الأول من نوعه في الجزائر.

-ينتمي مشروعنا إلى نطاق الابتكارات المتزايدة نظرا إلى أن عدم التأكد في السوق منخفض لأنه لا يوجد هذا النوع من البرامج وهو موجه إلى فئة متعلمة (الجامعات ومراكز لبحث العلمي والتعليم العالي).



2. مجالات الإبتكار: مجال البرمجة. تصميم واجهة تفاعلية لحسابات كيمياء الكم وهو الأول من نوعه في الجزائر وبرنامج جديد في السوق.

المحور الثالث: التحليل الإستراتيجي للسوق

1. مخطط PESTEL

2. مخطط SWOT

3. مخطط porter

4. عرض القطاع التسويقي

1.4. السوق المحتمل

2.4. السوق المستهدف

3.4. تحليل السوق

4.4. مزيج التسويقي

5.4. جدول تقدير الطلب لسنة 2024

6.4. تقدير المبيعات

5. قياس شدة المنافسة

6. الاستراتيجية التسويقية

1. مخطط PESTEL :

القانوني	البيئي	التقني	المجتمع	الإقتصادي	السياسي
- يستوفي بالشروط القانونية لا يوجد يمنع هذا النوع من البرامج.+	- ليس لديه تأثيرات على البيئة.+	- يساعد في تطوير التكنولوجيا.+ - إضافة في مجال البرمجة الوطنية.+	- برنامج تعليمي بيداغوجي موجه للطلبة الفئة متعلمة في المجتمع يساعد في إيصال	- الإستثمار في تطوير البرنامج.+ - تقديم الدعم المالي لمشروع.+	- بعد صدور قرار الوزاري 1275 تحظى المؤسسات الناشئة الدعم السياسي.+

حق الملكية يحمى البرنامج+.			المعلومة وتبسيطها+.		
----------------------------------	--	--	------------------------	--	--

2. مخطط SWOT:

نقاط الضعف: -نحن جدد في مجال التسويق ولا نملك خبرة.	نقاط القوة: -إبتكار جديد في السوق . -سهولة الإستعمال . -موجه لفئة متعلمة تستقبل هذا النوع من الإبتكارات.
الفرص: -انتقال من السوق الوطني إلى السوق العالمي. -برمجة برامج جديدة أخرى .	التحديات: -ظهور منافسين في السوق, -القرصنة,

2. مخطط porter:



4. عرض قطاع السوق:

1.4. السوق المحتمل: هو أي شخص يستفيد من البرنامج سواء دكتور، طالب، باحث.....

2.4. السوق المستهدف: هم الجامعات ومراكز البحث العلمي والتعليم العالي (مختلف تخصصات الكيمياء في الجامعات، المعاهد، المدارس التعليم العالي).



3.4. تحليل السوق:

✓ نطاق السوق: وطني

✓ تحليل الزبائن:

العدد	نوع الزبون
66	• جامعات ومراكز البحث العلمي والتعليم العالي

الطلاب •	10000 سنويا
----------	-------------

✓ تحليل المنافسين: لا يوجد منافسين

✓ تحليل الموردين:

المادة	الموقع	إسم المورد
تجهيزات المكتبية أجهزة الحاسوب. المكيف طابعة.	الدار البيضاء. الجزائر	محل بيع تجهيزات المكتبية Mersatrade Mobilier

4.4. مزيج التسويق:

المنتج	السعر DA	الترويج	التوزيع	العمليات	الافراد	المكان
تصميم واجهة تفاعلية تغطي الحسابات المعتادة في كيمياء الكم وتعالج مشاكل الطالبة في فهم المقياس وستوفر هذه الواجهة بيئة متكاملة لتسهيل المفاهيم النظرية التي تم تناولها في	- التسعير سيكون على أساس مدى الإشتراك شهري أو سنوي . 20.000	- الملتقيات - الأيام الدراسية - والمعارض الإبتكارية - صفحات الفايسبوك - عرضها على الجامعات ومراكز البحث العلمي .	- التحميل البرنامج من Pag web عبر رابط الإللكتروني، أو تحميل من Play Store بعد إدخال إسم البرنامج.	بعد الدخول إلى رابط الإللكتروني و إدخال البيانات و دفع المستحقات يمكنه تحميل البرنامج (سنوفر إشتراك شهري أو إشتراك سنوي) - سنوفر فيديو يكون	- ليندة بحري طالبة ثانية ماستر كيمياء المواد. - المبرمج الدكتور كنوش سمير.	المكتب قريب من الجامعة بسكرة مزود بخدمة الأترنت

		فيه شرح تفصيلي لمستعملين البرنامج.				هذا التخصص.
--	--	---	--	--	--	----------------

5.4. جدول تقدير الطلب لسنة 2024 :

بنا على المعلومات المجمعة من الوسط الجامعي فإن فكرة المشروع أي البرنامج لقت ترحيب وتشجيع بين الطلبة ومدرسين.

Min	moyen	max	الثلاثي
14	22	30	الأول
25	40	55	الثاني
36	53	70	الثالث
65	80	95	الرابع
140	195	250	المجموع السنوي

6.4. تقدير المبيعات:

بما أن مشروعنا عبارة عن برنامج فنحن نستطيع تغطية كافة الطلبات.

5. قياس شدة المنافسة: مع تطور الحاصل في مجال البرمجة والتصميم سيظهر لنا عدة منافسين برغم أنه برنامج جديد في سوق الجزائرية.

6. الإستراتيجية التسويقية: إتمدنا على إستراتيجية التميز بتقديم خدمة متكاملة ومميزة وجديدة في السوق، وسنعمد على الملتقيات وأيام الدراسية في التخصص والمعارض الإبتكارية وصفحات الفايسبوك لترويج والتعريف ببرنامجنا. ونوفر منصة تفاعلية لتواصل مع الزبون والإجابة على الإستفسارات ولتحميل البرنامج منها بأقل تكلفة ممكنة.

المحور الرابع: خطة الإنتاج والتنظيم

1. الموقع

2. إحتياجات المشروع

1.2. إحتياجات المعدات والألات

2.2. إحتياجات المعدات المكتبية

3.2. إحتياجات الكهرباء والماء والإنترنت

4.2. إحتياجات الموارد الأولية

3. النظام الإنتاجي

4. المخطط التنظيمي

1.4. إحتياجات اليد العاملة

2.4. الهيكل التنظيمي

3.4. شكل القانوني للمؤسسة

1.1. الموقع:

-الموقع الافتراضي: سنوفر موقع إلكتروني page web حيث يستطيع الزبون تحميله بكل سهولة بمجرد الدخول إلى الرابط الإلكتروني، وكذا فيديو يكون فيه شرح تفصيلي لكيفية إستعمال البرنامج.

- الموقع الواقعي: مكتب قريب من جامعة محمد خيضر في ولاية بسكرة.

- سبب إختيار الموقع: يرجع سبب إختيارنا هذا الموقع لنكون أقرب إلى الجامعة لتسهيل تنقل الطلاب والإجابة على إستفساراتهم، وكذا نكون أكثر إحتكاكا مع الوسط الجامعي والمشاركة في المنتقيات والأيام الدراسية لترويج وتعريف بمشروعنا.

2. إحتياجات المشروع:

1.2. إحتياجات المعدات والألات: لا يوجد

2.2. إحتياجات المعدات المكتبية:

الإحتياج	العدد	سعر الوحدوي (DA)	سعر الإجمالي (DA)
مكتب	1	132.000	132.000
طاوولات	2	8.000	16.000
كراسي	2	8.500	17.000
أجهزة الحاسوب	2	60.000	120.000
المكيف	1	50.000	50.000
طابعة	1	38.900	38.900
معدات المكتبية	1	10.000	10.000
مصاريق النقل	1	10.000	10.000
السعر الإجمالي الكلي			393.900

3.2. إحتياجات الكهرباء والماء والإنترنت:

الإحتياج	سعر الإجمالي في لسنة (DA)
الكهرباء والماء	80.000
الإنترنت والهاتف	57.588
المجموع	137.588

4.2. إحتياجات الموارد الأولية:

الإحتياج	سعر الإجمالي لسنة (DA)
الإستضافة في الموقع الإلكتروني	48.000
التحميل على Play store	3.419
المجموع	51.419

3. النظام الإنتاجي:

مرت عملية البرمجة بعدة المراحل:

- الدراسات الأولية
- برمجة واجهه تفاعلية

- جمعهم في برنامج شامل
- تجهيز مكتب
- تسويق وبيع البرنامج

مخطط يشرح مراحل الإنتاج (البرمجة):

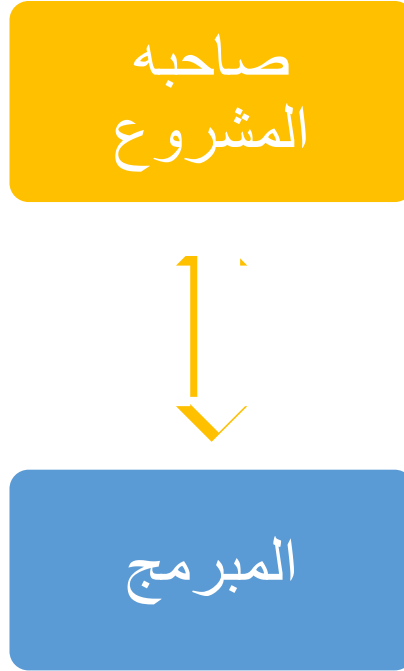


4.المخطط التنظيمي:

1.4.إحتياجات اليد العاملة:

الأجر السنوي DA	الأجر DA	المهمة	المواصفات	المنصب
600.000	50.000	- صيانة البرنامج. - المشاركة في تطوير البرنامج.	-خبرة في مجال البرمجة -خبرة في برنامج المطلاب. -خبرة في مجال كيمياء الكم	المبرمج
480.000	40.000	التنظيمات	صاحبة المشروع	المسؤول
1.080.000				التكلفة يد العاملة

2.4. الهيكل التنظيمي:



3.4. شكل القانوني للمؤسسة: هو EURL المؤسسة ذات المسؤول الوحيد وذات المسؤولية المحدودة (م.ذ.ش.و.م.م)

هناك عدة أشكال قانونية للمؤسسات حسب القانون الجزائري، مكيفة لشخص الواحد او مجموعة الشركاء. يتوافق كل وضع قانوني مع نظام للخضوع للضرائب والمسؤوليات والالتزامات.

شكل مؤسستنا هو المؤسسة ذات المسؤول الوحيد وذات المسؤولية المحدودة حيث يسمح هذا الشكل لشخص ان تكون الأموال الخاصة منفصلة عن أملاك المؤسسة ويكون مسؤولاً فقط عن ديون المؤسسة في حدود مبلغ رأس المال الذي يجب ألا يقل عن 100000 دينار. ويمنح القيد في السجل التجاري الشخصية المعنوية للمؤسسة وصفة التاجر للمسير.

المحور الخامس: الخطة المالية PLAN FINANCIER

1. التكاليف والأعباء

1.1. تكلفة الاستثمارية

2.1. تكلفة التشغيلية

2. الهيكلة التمويلية

3. جدول حسابات النتائج المتوقع

1. التكاليف والأعباء:

1.1. تكلفة الاستثمارية:

الأصول	التكليف
الأثاث	393.900
دفع حقوق برنامج المطالب	120.000
رأس المال العامل	/
المجموع	513.900

2.1. تكلفة التشغيلية:

الأصول	التكليف
مواد الأولية	51.419
أجور	1.080.000
كراء المكتب	240.000
الهاتف والانترنت	57.588
الكهرباء والماء	80.000
المجموع	1.509.007

2. الهيكل التمويلي:

أما بالنسبة إلى مشروعنا سيكون التمويل الذاتي بالاعتماد الكلي على الأموال الخاصة

التكلفة الإجمالية = التكلفة الاستثمارية + التكلفة التشغيلية

التكلفة الإجمالية = 1.509.007 + 513.900 = 2.022.907

البيان	النسبة	القيمة DA
أموال الخاصة	100	2.022.907

3. جدول حسابات النتائج المتوقع:

	REALISATION			PREVISION		
	N	N+1	N+2	N+3	N+4	N+5
Produit A destiné Client						
Quantité produit A	250	300	350	500	600	900
Prix HT produit A	20.000	20.000	20.000	20.000	20.000	20.000
<u>Ventes</u> <u>produit A</u>	5.000.000	6.000.000	7.000.000	10.000.000	12.000.000	18.000.000
CHIFFRE D'AFFAIRES GLOBAL	5.000.000	6.000.000	7.000.000	10.000.000	12.000.000	18.000.000

المحور السادس: النموذج الاولي التجريبي

1. الواجهة الأساسية

1.1. واجهة قانون بلانك

2.1. حسابات النموذج الكمي للجسم

1.2.1. قناة أحادية البعد

2.2.1. علبة ثنائية الابعاد

3.1. واجهة حالة ثابتة

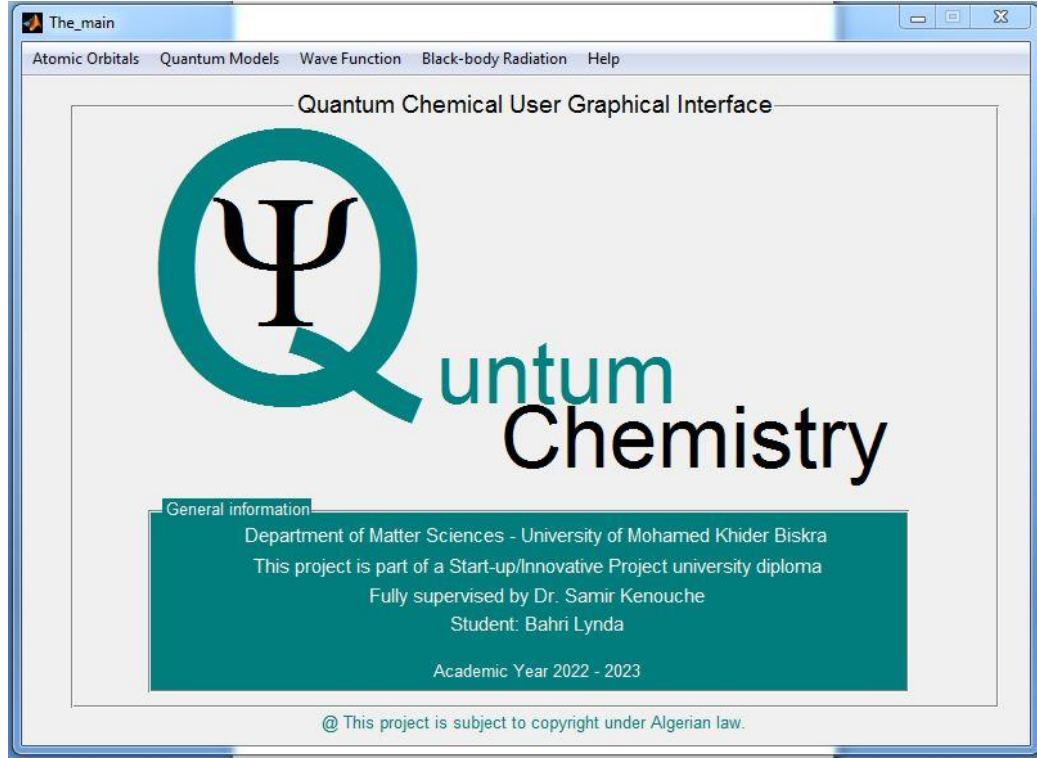
4.1. واجهة مذبذب توافقي

5.1. واجهة المدرات الذرية

6.1. واجه وظائف الموجة الشعاعية

1. الواجهة الأساسية:

الصفحة الرئيسية للبرنامج:

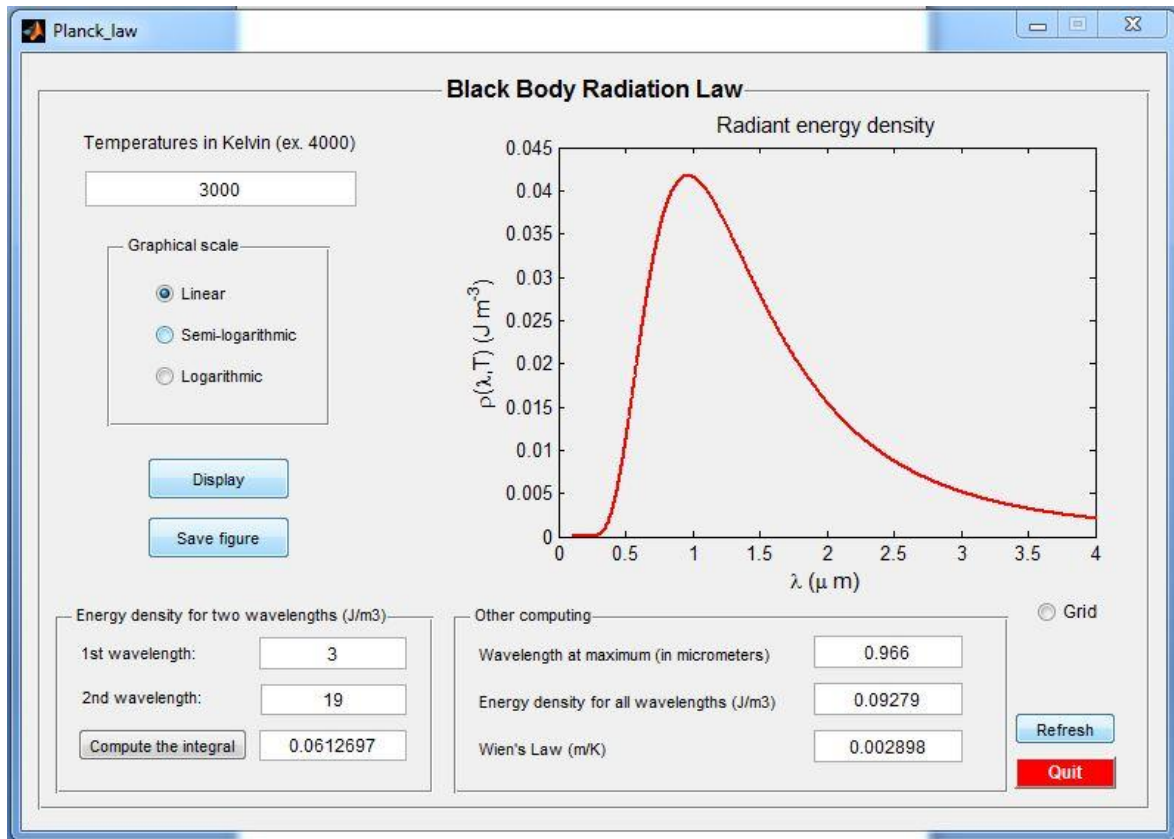


الشكل 1: الواجهة الرئيسية

وصف القائمة الرئيسية لهذه الواجهة، تحتوي على النافذة Atomic Orbitals على القائمة التالية: القائمة التالية: Particle One Dimension و Particle Two Dimension و Harmonic Oscillator ، وتحتوي النافذة الثالثة Wave Functions على القائمة التالية: Radial Wave Function و Stationary State و نافذة Black-body Radiation من قائمة واحدة وهي Planck Law . والنافذة الأخيرة Help على القائمة التالية: User's Guide و Contact Us ، تم وصف تشغيل هذه الواجهات بإيجاز في الأقسام التالية أما التفاصيل التقنية المتعلقة فهي متوفرة في الفصل الثالث من الاطروحة .

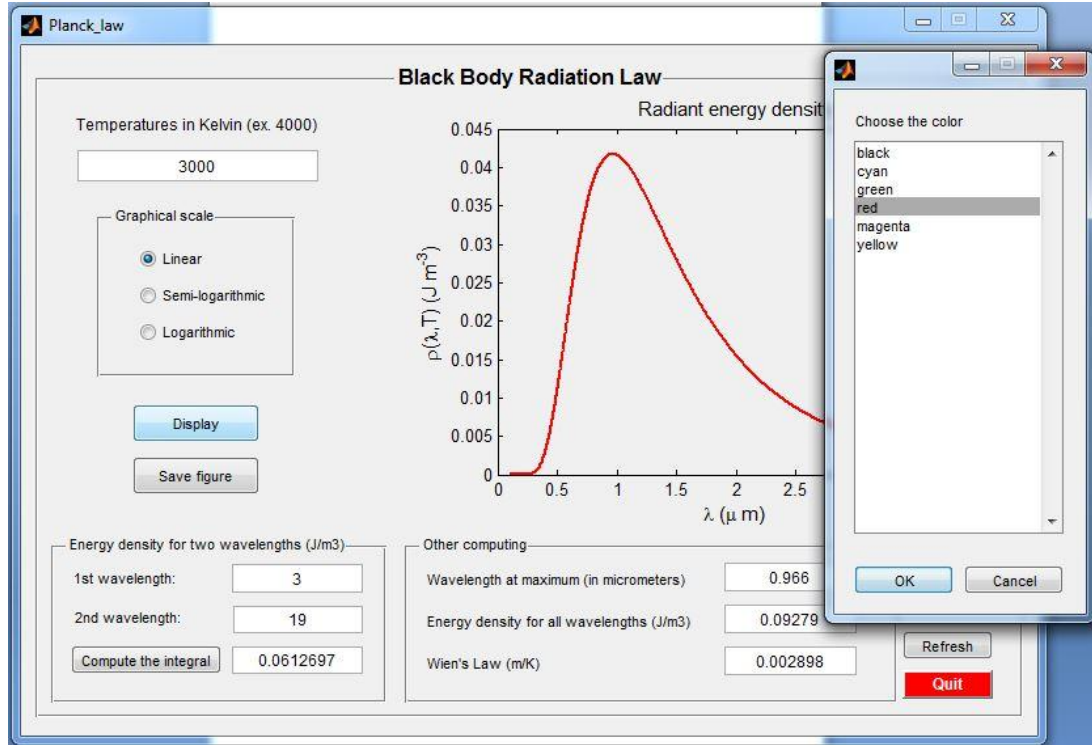
11. واجهة قانون بلانك :

تتيح هذه الواجهة (انظر الشكل أدناه) تنفيذ العديد من المهام المتعلقة بإستغلال قانون بلانك. من خلال إدخال قيمة درجة الحرارة في خانة Temperatures in Kelvin و النقر على Display ، ليتم عرض الشكل النظري لهذا القانون في الشكل. بالإضافة إلى ذلك من الممكن إختيار المقياس الذي سيتم عرض الرسم البياني الخاص بها يكفي تحديد أحد الخيارات ونقر عليها : Linear, Semi-logarithmic, logarithmic.



الشكل 1.1: واجهة المستخدم الرسومية لقانون بلانك

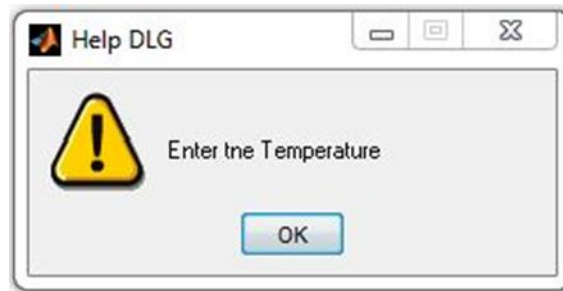
من خلال تحديد أحد هذه المقاييس، يتم فتح choose the color على الفور، مما يدفع المستخدم إلى إختيار أحد الألوان التي سيتم بها رسم الرسم البياني لقانون بلانك (أنظر الشكل أدناه).



الشكل 1.1.1: إختيار اللون

يمكن حفظ الشكل الذي تم الحصول عليه على هذا النحو بتنسيق gif في السجل (le répertoire)، بمجرد النقر على Save figure. يتم تنفيذ العديد من الإختبارات التحكم من أجل الأتمتة الواجهة، لزيادة متانتها وكذلك إستقلاليتها.

إذا قمت بالنقر على Display دون إدخال قيمة درجة الحرارة فستظهر الرسالة الآتية:



الشكل 1.1.2: رسالة تعليمات لإدخال درجة الحرارة

بالضغط على OK، يتم إغلاق الرسالة، من ناحية أخرى إذا أدخلت قيمة أجنبية في خانة Temperature in Kelvin، سيتم عرض رسالة الخطأ على الشاشة:



الشكل 1.1.3: رسالة خطأ تتعلق

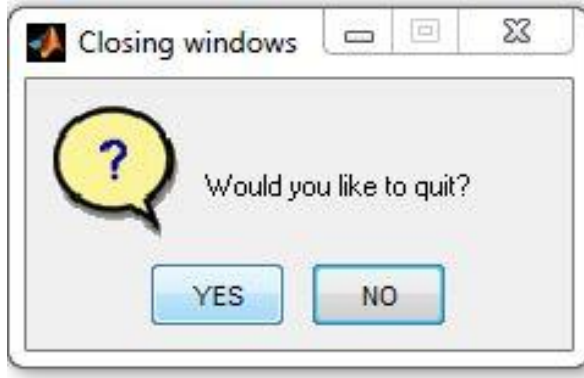
بإدخال قيمة عددية

رسالة الخطأ هذه تطلب من المستخدم بإدخال قيمة رقمية في لوحة Energy Density for two wavelengths التي تستخدم لحساب كثافة الطاقة المشعة عن طريق إدخال طولين موجيين في نطاق 0-30 ميكرومتر. لحساب هذا التكامل من الضروري للغاية إدخال حدي التكامل، الحد الأدنى والحد الأعلى. إذا كان أحد حقول 1st wavelength و 2nd wavelength فارغاً، ترجع الواجهة رسالة خطأ التالية:



الشكل 1.1.4: رسالة خطأ تتعلق بأطراف التكامل

لوحة Other computing تستخدم لتحديد الطول الموجي الأقصى للإشعاع، لحساب تكامل الإجمالي I_a و la loi de Wien و la من خلال النقر على Refresh، يتم تحديث جميع قيم لوحتي Energy Density for two wavelengths و Other computing من أجل بدء عمليات حسابية جديدة. عن طريق تحديد Grid، يتم الضافة شبكة إلى الشكل. أخيراً عند النقر على Quit ترجع الواجهة رسالة التأكيد التالية:



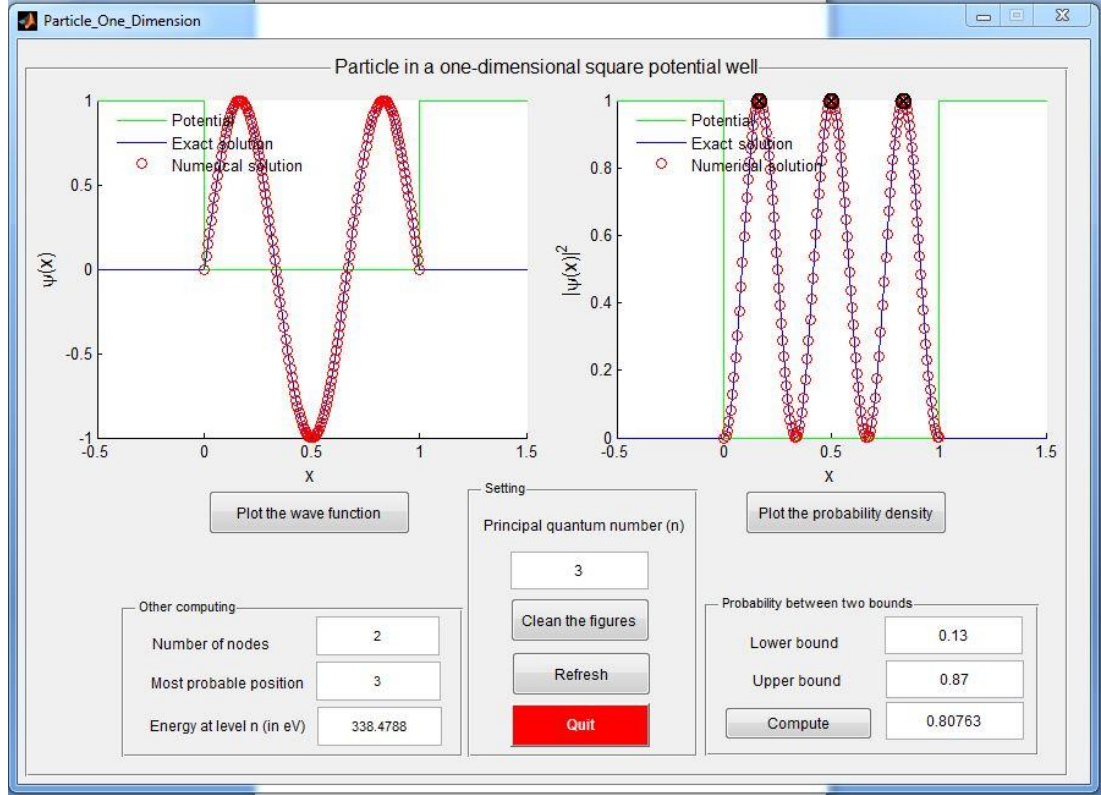
الشكل 1.1.5: اغلاق الواجهة

هذه الرسالة تدعو المستخدم لتأكيد رغبته في إغلاق الواجهة. من خلال النقر على YES، يتم إغلاق الواجهة على الفور، من ناحية أخرى بالنقر على NO ستظل الواجهة نشطة.

2.1. حسابات النموذج الكمي للجسم :

1.2.1. حالة أحادية البعد :

في هذه الواجهة يتم تنفيذ حسابات النموذج الكمي للجسم في مربع احادي البعد. من أجل إستغلال هذه الواجهة، يجب على المستخدم أولاً إدخال الرقم الكمي الرئيسي في حقل Principal quantum number، ثم بالنقر على Plot the wave function، يتم عرض الدالة الموجية المقابلة لمستوى الطاقة على يسار الواجهة، المظهر الجانبي باللون الأخضر يتوافق مع الطاقة الكامنة. تتوافق المنحنيات تعريف اللون الأزرق والأحمر على التوالي مع الحلول الدقيقة والرقمية للوظيفة الموجة. يتم عرض كثافة احتمال الوجود على يمين الواجهة بالنقر على Plot the probability density. تحديد العلامات المتقاطعة أكثر المواضيع احتمالية للجسم .

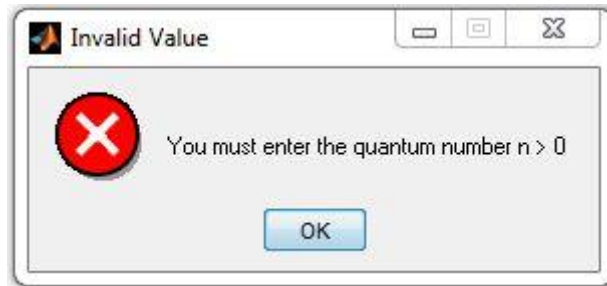


الشكل 1.2.1: واجهة المستخدم الرسومية للجسم في صندوق احادي البعد

يتم استخدام Probability between two bounds لحساب تكامل كثافة احتمال الوجود بإختيار حدي تكامل في النطاق 0-1 (بوحداث الطول). في لوحة Other computing، يتم عرض عدد العقد (المواضيع المحظورة) و

المواضيع الأكثر احتمالية إلى قيمة إجمالي الطاقة (بالإلكترون فولت) للمستوى الذي تم النظر فيه. بالضبط على Clean the figure، يتم مسح الشكليات لبدء تمثيلات بيانية جديدة. علاوة على ذلك اذا تم إدخال قيمة صفرية أو حرف أبجدي في حقل Principal quantum number، فسيتم عرض رسالة الخطأ التالية على

الشاشة:

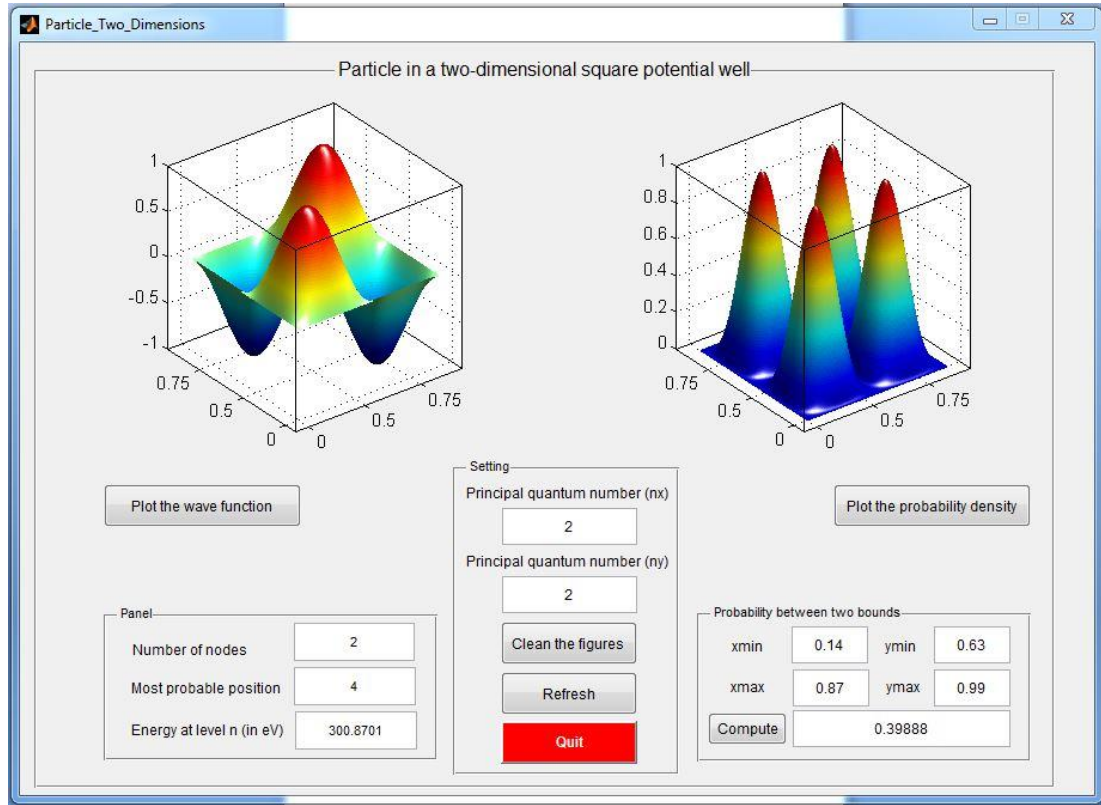


الشكل 1.2.2: رسالة الخطأ تتعلق بالرقم الكمي الرئيسي،

تطالب هذه الرسالة المستخدم بالتصحيح وإدخال عدد صحيح طبيعي غير صفري.

2.2.1. حالة ثنائية الابعاد:

في هذه الحالة ينحصر الالكترن في صندوق ثنائي الابعاد ($0 < x < a$ و $0 < y < b$). تؤدي هذه الواجهة نفس مهام حالة أحادية البعد، مع جميع رسائل المساعدة في الخطأ المقابلة.

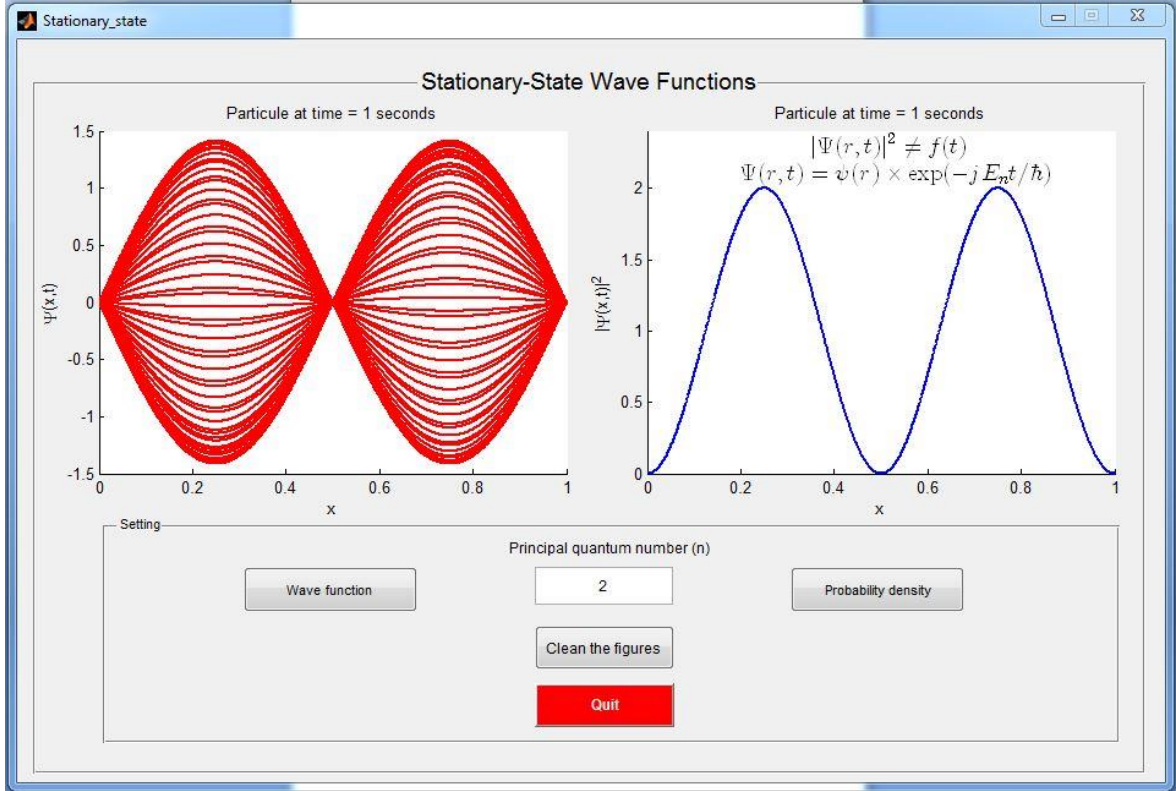


الشكل 1.2.2.1: واجهة المستخدم الرسومية للجسم في صندوق ثنائي البعد

يتم استخدام لوحة Probability between two bounds لحساب وعرض نتيجة التكامل المزدوج.

3.1. واجهة حالة ثابتة :

توضح الواجهة ادناه فكرة الحالة الثابتة و لتنشيط هذه الواجهة، ما عليك سوى إدخال الرقم الكمي الرئيسي في خانة Principal quantum number. ويجب أن يكون هذا الرقم عدد صحيح طبيعي غير صفري.



الشكل 1.3.1: واجهة المستخدم الرسومية الثابتة.

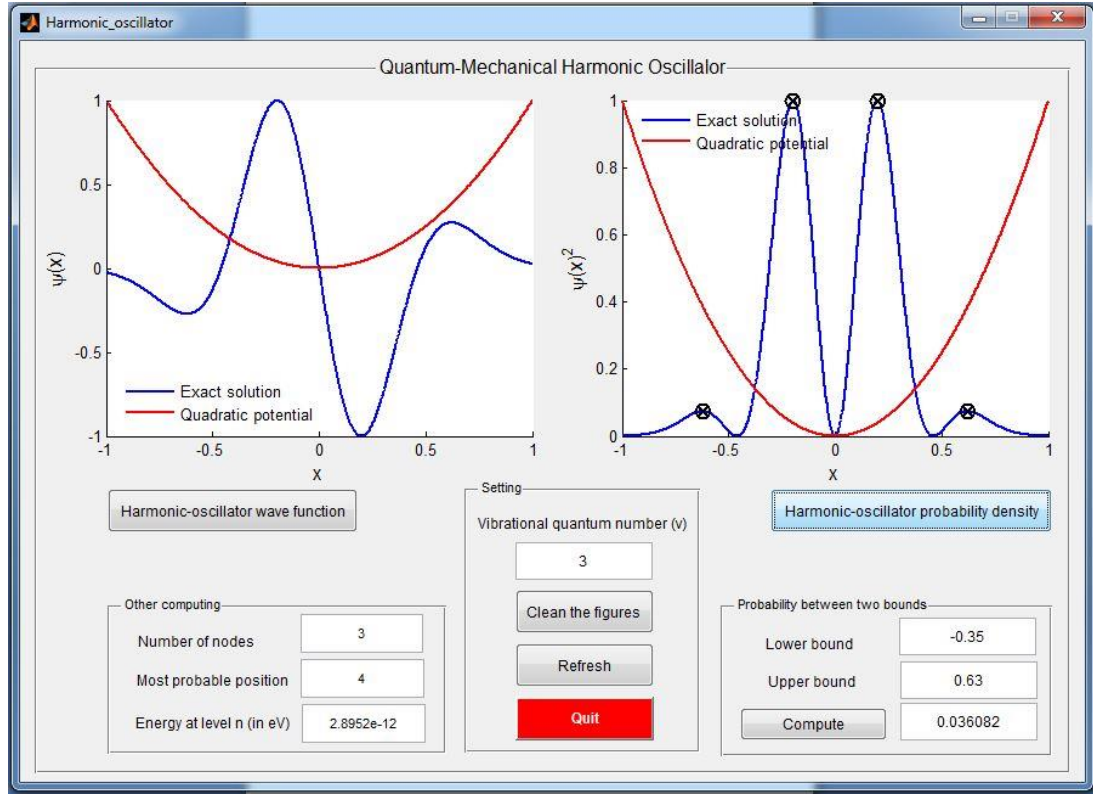
تشابهه خاني Clean the figures و Quit الموضحتين في الواجهات السابقة بما في ذلك رسائل التعليمات و الخطأ.

4.1 واجهة مذنب تو افقي :

المذنب التوافقي هو نموذج له العديد من التطبيقات المهمة في الميكانيك الكلاسيكية و الكمية، وانه بمثابة نموذج اولي في المعالجة الرياضية لظواهر متنوعة مثل المرونة والاهتزازات الجزئية و الخصائص البصرية للمادة، بالنسبة لمذنب كلاسيكي من الممكن تحديد موضع وكمية الحركة بدقة في كل لحظة، لا توجد قيود على طاقة المذنب، وتؤدي الاختلافات في طاقة المذنب الى تغييرات في سعة الاهتزازات التي يتعرض لها المذنب الى تغييرات في سعة الاهتزازات التي يتعرض لها المذنب في هذه الواجهة (انظر الشكل ادناه) يتم تنفيذ حسابات هذا النموذج الكمي .

لتفعيل هذه الواجهة سيكون من الضروري ادخال العدد الكمي لاهتزاز في حقل Vibrational quantum number، ومن خلال النقر على Harmonic-oscillator wavefunction يتم عرض الوظيفة الموجة المقابلة لمستوى الطاقة Ev على يسار الواجهة، المظهر الجاني باللون الأحمر يتوافق مع الطاقة الكامنة التربيعية،

ويرتبط منحى تعريف اللون الأزرق بوظيفة الموجة. يتم عرض كثافة احتمالية التواجد على يمين الواجهة بالنقر على Harmonic-oscillator probability density، وتشير العلامات الى المواضيع الأكثر احتمالا للكثافة المتذبذبة.



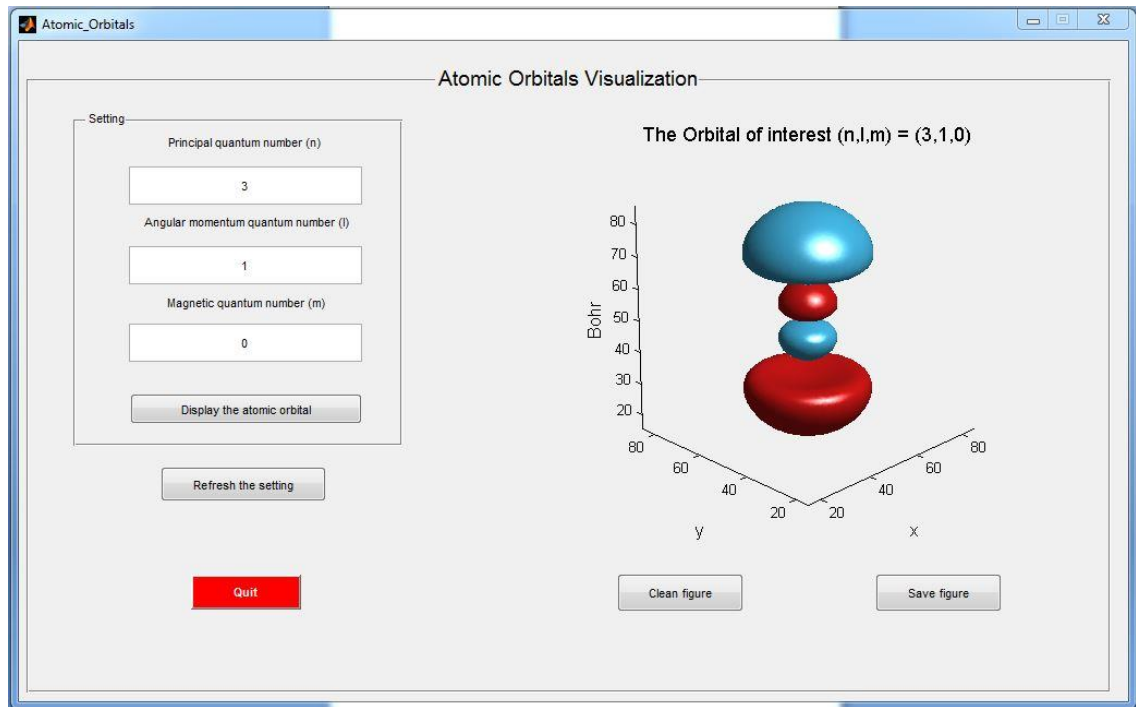
الشكل 1.4.1: واجهة رسومية للمذبذب التوافقي

يتم استخدام اللوحة Probability between two bounds لحساب تكامل كثافة احتمال الوجود باختبار حدي تكامل في النطاق $-1 < x < 1$ (في طول الوحدة). في لوحة Other computing، يتم عرض عدد العقد (المواضيع المحضورة) والمواضيع الأكثر احتمالية بالإضافة الى قيمة إجمالي الطاقة (بالإلكترون فولت) للمستوى الذي تم النظر فيه. بالضغط على Clean the figure، يتم مسح الشكلين لبدء تمثيلات بيانية جديدة كما كان من قبل، يتم عرض مربعات حوار لرسائل الخطأ. يتم حساب كثافة الإحتمال بنفس البرنامج عن طريق تربيع دالة الموجة.

5.1. واجهة المدرات الذرية:

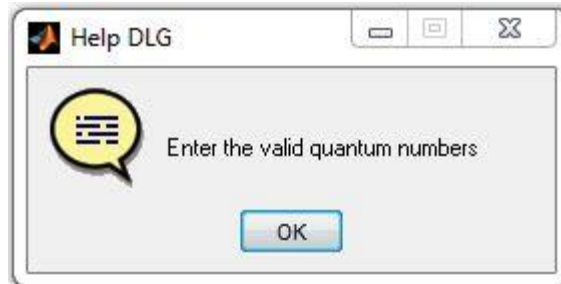
وفقا Schrödinger يتم وصف الإلكترون الموجود في ذرة الهيدروجين بالموجة الثابتة، في الواقع وفقا لنظرية الكم، لا يمكن وصف الإلكترونات التي تدور حول النواة بشكل كامل على أنها جسيمات ولكن يجب أن تفي بمفهوم ازدواجية الموجة والجسيم، لذلك يجب أن يكون للإلكترون أيضا خصائص موجية، لا تدور

الإلكترونات حول النواة مثل *modèle planétaire*، ولكنها موجودة كموجات ثابتة، أدنى طاقة يمكن أن يأخذها الإلكترون هي مماثلة للتردد الأساسي للموجة بالمعنى الكلاسيكي للمصطلح (إهتزاز الحبال على سبيل المثال)، الحالات المثارة (طاقة أعلى) تشبه التوافقات الأساسي. لا يمكن أن تكون الإلكترونات في نقطة واحدة على الرغم من أنه يمكن تحديد احتمال التفاعل مع الإلكترون عند نقطة واحدة، على الرغم من أنه يمكن تحديد احتمال التفاعل مع الإلكترون عند نقطة واحدة من دالة الموجة للإلكترون تتصرف شحنة الإلكترون كما لو كانت منتشرة في الفضاء وفقا لتوزيع مستمر، متناسبة في جميع النقاط مع مربع معامل الدالة الموجية للإلكترون.



الشكل 1.5.1: واجهة المستخدم الرسومية للمدارات الذرية.

من خلال إدخال الأرقام الكمية الثلاثة في لوحة Setting، يؤدي النقر على *Probability between two bounds* إلى تصوير المدار الذري المعني. يسمح خانة Refresh the setting بتحديد الأرقام الكمية الثلاثة من أجل إطلاق جديد. يؤدي الإدخال الخاطئ للأرقام الكمية إلى تنشيط مربع الحوار التالي:

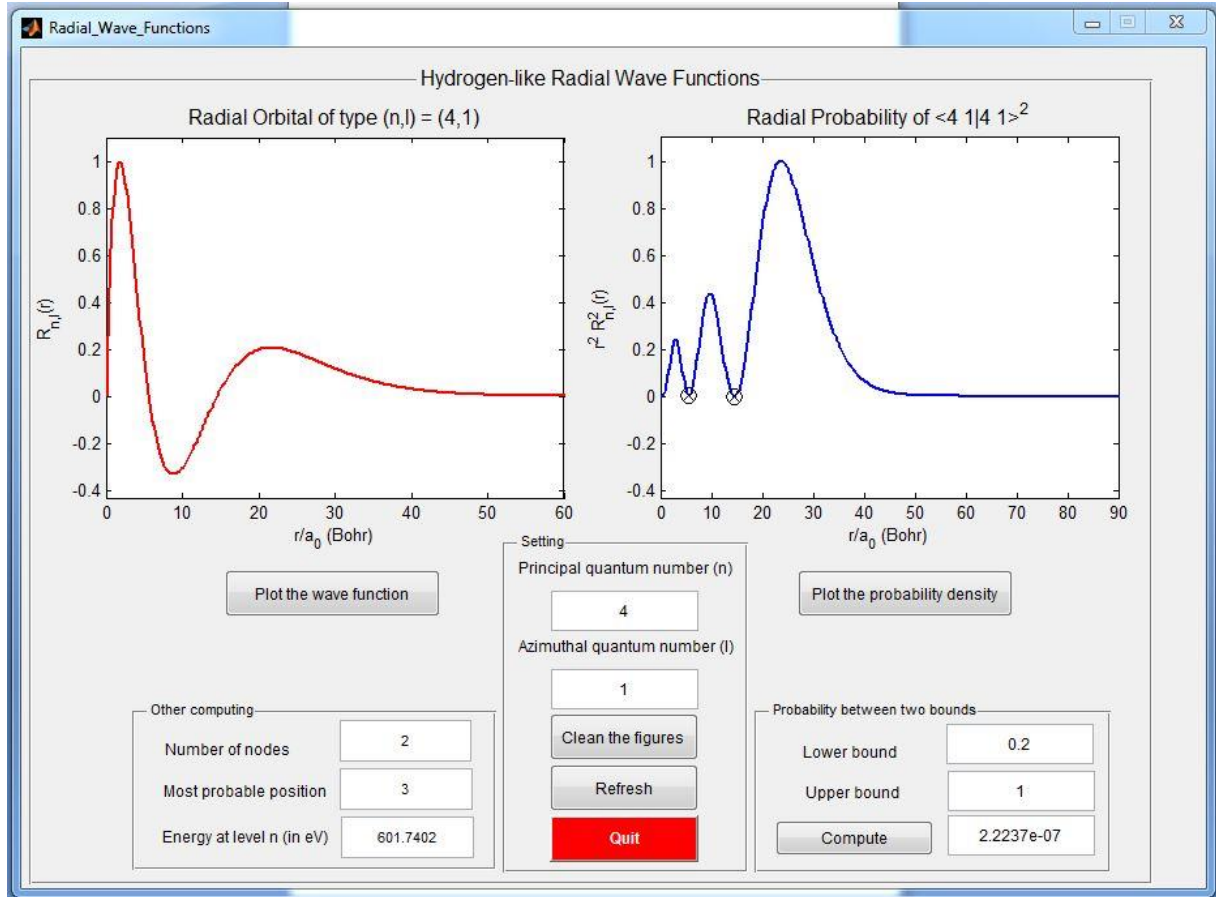


الشكل 1.5.2: التحقق من الأرقام الكمية

تطالب هذه الرسالة المستخدم بإدخال القيمة الصحيحة للأرقام الكمية الثلاثة بالنقر على Save figure، مربع حوار لإختيار الدليل الذي تم حفظ الشكل فيه.

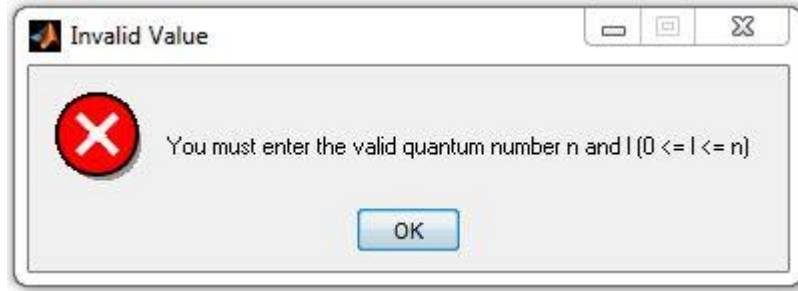
6.1. واجهه وظائف الموجة الشعاعية:

هذه الواجهة مخصصة لإستغلال الجزء الشعاعي من وظائف الموجة. يكفي إدخال الاعداد الكمية n و l . بالنقر على Plot the wave function، فيتم عرض كثافة احتمال التواجد على يسار الواجهة. من الناحية الأخرى يتم عرض كثافة احتمال التواجد على اليمين بالنقر على Plot the probability density، بالإضافة الى عرض العقد (المواضع المحصورة) أيضا.



الشكل 1.6.1: واجهه المستخدم الرسومية لوظائف الموجة الشعاعية

يتم استخدام لوحة Probability between two bounds لحساب كثافة احتمالية التواجد من خلال تحديد Upper bound, Lower bound (الحد الأدنى والحد الأعلى)، في لوحة Other computing نحسب عدد العقد و المواضع الأكثر احتمالية وكذلك الطاقة الكلية.



الشكل 1.2.6: رسالة التحقق من الرقم الكمي

يتم إرجاع هذه الرسالة في حالة وجود خطأ في قيم العددين الكميين.

أخيراً تم تجميع جميع هذه الواجهات معا لتشكيل البرنامج وتم أيضاً إنشاء ملف *.exe. يسمح بإستخدام هذا البرنامج دون الحاجة إلى تثبيت المطالب وهي ميزة حقيقة.

قائمة الملاحق:

نموذج الاعمال BMC:

<p>الشركات:</p> <p>-الجامعات ومراكز البحث العلمي و التعليم العالي . - المبرمج (الدكتور سمير كنوش)</p>	<p>الأنشطة الرئيسية:</p> <p>- البرمجة - تطوير البرنامج -تثبيت. - صيانة البرنامج,</p>	<p>القيمة المقدمة:</p> <p>تصميم واجهة تفاعلية تغطي الحسابات المعتادة في كمياء الكم وتعالج مشاكل الطلبة في فهم المقياس وستوفر هذه الواجهة بيئة</p>	<p>العلاقة مع العملاء:</p> <p>- علاقة غير مباشرة عن طريق منصة في Page web . - تكوينات مجاني لأول عشر مشترين للبرنامج</p>	<p>العملاء:</p> <p>السوق المحتمل: هو أي شخص يستفيد من البرنامج سواء دكتور، طالب، باحث..... السوق المستهدف الجامعات ومراكز البحث العلمي والتعليم العالي (مختلف تخصصات الكمياء و الفزياء في الجامعات، المعاهد، المدارس العليا).</p>
	<p>الموارد الرئيسية: - الحاسوب - الملكية الفكرية - برنامج المطالب - موارد بشرية (المبرمج)</p>	<p>متكاملة لتسهيل المفاهيم النظرية التي تم تناولها في هذا التخصص.</p>	<p>قنوات التواصل:</p> <p>1-ترويج للبرنامج في المنتديات وأيام الدراسية والمعارض الابتكارية ومواقع التواصل . 2-توفير منصة تفاعلية لتواصل مع الزبون والإجابة على استفسارات. 3-توصيل عبر تحميله من منصة تفاعلية بعد دفع مستحققاته المالية.</p>	<p>مصادر الإيرادات:</p> <p>1-بيع البرنامج (مدخل ارادات مالية بعد تحميله من المنصة التفاعلية). 2-الإعلانات لبرامج أخرى.</p>
	<p>هيكل التكاليف:</p> <p>1. المكتب وتجهيزاته و المعدات الازمة, 2. أجهزة الحاسوب. جهاز مكيف. تكاليف حقوق برنامج المطالب.</p>			

