

Devoir 3 : Équation de Schrödinger en dimension 1

PHQ404

Date de complétion suggérée : 23 mars 2024 à 23h45

1 Objectif

L'objectif de ce TP est de se familiariser avec deux techniques pour solutionner des problèmes aux limites en dimension 1 : la méthode du tir et la méthode des éléments finis. Ces méthodes sont présentées au chapitre 12 des notes de David Sénéchal.

2 Comment présenter et remettre votre TP

Vous devez cloner le répertoire github dans l'organisation du cours au lien suivant : <https://classroom.github.com/a/eH84M6i->. Dans ce répertoire se trouvera votre code python, vos tests unitaires ainsi que votre readme décrivant les méthodes utilisés et l'analyse de vos résultats. La structure des fichiers ne doit pas être modifiée, mais vous pouvez ajouter des fichiers si vous le désirez. Voici la structure de fichiers que votre répertoire devra garder :

Root

```
→ src
    ↳ fichier0.py
    ↳ fichier1.py
    ↳ ...
→ tests
    ↳ test_fichier0.py
    ↳ test_fichier1.py
    ↳ ...
↳ .gitignore
↳ requirements.txt
↳ README.md
```

Le fichier `requirements.txt` doit contenir les dépendances de votre projet. Le fichier `README.md` doit contenir les instructions pour installer et utiliser votre projet ainsi qu'une brève description du devoir et des méthodes utilisés dans le code. Voir la section 5 pour plus

de détails. Dans le dossier `src` se trouvera votre code python et dans le dossier `tests` se trouvera vos tests unitaires.

La remise du code se fera à chaque **push** sur le répertoire github. Notez que seul le dernier **push** sera considéré pour la correction.

3 Énoncé

3.1 Méthode du tir

On considère l'équation de Schrödinger indépendante du temps pour une particule de masse unité dans un potentiel quadratique

$$-\frac{1}{2}\ddot{\psi} + \frac{1}{2}x^2\psi = E\psi. \quad (3.1)$$

On cherche les solutions pour x de $-L$ à L pour $L = 5$ avec 200 points avec les valeurs initiales $\psi(-L) = 0$ et $\dot{\psi}(-L) = 0.001$. La valeur initiale de $\dot{\psi}$ n'est pas très importante, mais celle-ci devrait vous donner une échelle raisonnable. Vous devez :

- Implémenter une fonction qui évalue $\psi(x = L|E)$ comme définie dans les notes.
- Implémenter une fonction pour cadrer une racine. C'est-à-dire, une fonction qui à partir d'une valeur x_0 et d'une fonction f , trouve une valeur x_1 telle que $f(x_1) = -f(x_0)$. Pour ce faire, on augmente x par pas de Δ jusqu'à trouver x_1 ou atteindre un nombre limite de pas (20 par exemple).
- Trouver les racines de $\psi(x = L|E)$ à l'aide de l'algorithme de Brent et de la fonction précédente.
- À l'aide de ces fonctions, écrivez un programme qui trouve les 6 premières solutions à l'équation de Schrödinger (les 6 énergies les plus basses) et portez-les sur un même graphique en les décalant par la valeur de leur énergie, comme illustré à la figure 3.1.

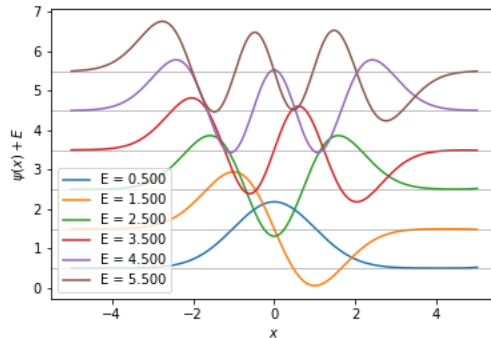


FIGURE 3.1 – Exemple de figure pour les états propres de basses énergies de l'équation 3.1.

3.2 Méthode des éléments finis

Dans cette partie, la méthode des éléments finis sera utilisée. La méthode n'a pas besoin d'être programmée entièrement : un fichier python est fourni et doit être importé dans votre programme. Ce fichier introduit une classe pour vous aider. Appliquez la méthode des éléments finis à la solution de la même équation qu'à la partie précédente. Cette fois, utilisez une grille de 500 points et une borne $L = 6$. Produisez le même type de graphique. Idéalement, vous utiliseriez la même fonction pour le graphique que pour l'énoncé précédent. Les fonctions de la classe ne calculent pas les valeurs aux frontières. Celles-ci ne sont nécessaire et peuvent ajouter du bruit à vos solutions.

Note : Toutes les fonctions qui doivent être implémentées sont déjà définies dans les fichiers et retournent des `NotImplementedError`.

4 Vérification

Il est important de vérifier vos implémentations. En effet, vous devez vous assurer que vos méthodes fonctionnent correctement et pour ce faire, vous devez rouler et implémenter des tests unitaires qui testent chacune de vos classes et fonctions. De plus, vous devriez tester si les résultats obtenus sont logiques. Il serait aussi intéressant de retrouver vos vérifications dans votre rapport. Il faut ajouter des tests unitaires dans le dossier `tests` afin d'augmenter la couverture des tests, mais les tests déjà implémentés ne doivent pas être modifiés.

5 Readme

Vous devez faire un fichier Readme qui explique ce que contient votre répertoire et comment l'utiliser. Le Readme sera divisé en 2 parties : une partie plus courte qui consiste essentiellement à ce qu'on retrouve normalement dans Readme scientifique et une partie plus longue qui consiste en une présentation et analyse des résultats.

La première partie doit contenir les éléments suivants :

- Une brève description du contenu du répertoire ;
- Une figure qui résume le contenu du répertoire ainsi que les résultats principaux ;
- Les instructions pour installer et utiliser votre projet.

Il faut qu'un utilisateur externe soit en mesure de regarder la première partie du Readme et comprendre en quelques secondes le contenu du répertoire, les résultats principaux et comment utiliser le projet. C'est important d'être concis, clair et efficace. Pour la figure, il s'agit d'une image permettant au lecteur de comprendre rapidement le contenu du répertoire. Celle-ci pourrait être, par exemple, un diagramme représentant le pipeline de traitement des données, un graphique comparant les différentes méthodes implémentées, etc.

Le deuxième partie doit contenir les éléments suivants :

- Une plus longue description du contenu du répertoire ;

- Une présentation et explication des méthodes utilisées ;
- Une présentation des résultats obtenus ;
- Une analyse des résultats obtenus ;
- Une conclusion.

Cette seconde partie sert à expliquer en quoi consiste ce dépôt si l'utilisateur décidait que la première partie du Readme était assez intéressante et bien présentée pour qu'il veuille en savoir plus. Il s'agit ici d'un court rapport scientifique. Il faut donc rester concis afin d'être lu en quelques minutes seulement, mais mettre suffisamment d'information pour que l'utilisateur comprenne bien la théorie, les méthodes et les résultats.

6 Critères d'évaluation

70 points Pour le résultat de l'autocorrection du code obtenue à l'aide du module TAC.

30 points Pour la qualité du Readme.