PHY-3500 Physique Numérique Hiver 2020

TP2: Équations linéaires et non linéaires

Date de remise: au plus tard le 8 mars 2020 à 23h59 Responsable: Antoine Allard

1 Loi du déplacement de Wien [1]

La loi de Planck définit la distribution de l'intensité de radiation (par unité de surface et par unité de longueur d'onde λ) d'un corps noir à la température T selon

$$I(\lambda) = \frac{2\pi hc^2\lambda^{-5}}{\mathrm{e}^{hc/\lambda k_B T} - 1} \tag{1}$$

où h est la constante de Planck, c est la vitesse de la lumière et k_B est la constante de Boltzmann.

a. [6 points] Démontrez que la longueur d'onde λ pour laquelle l'intensité de radiation émise est la plus élevée est une solution de l'équation

$$5\mathrm{e}^{-hc/\lambda k_B T} + \frac{hc}{\lambda k_B T} - 5 = 0 \ . \tag{2}$$

Faites maintenant la substitution $x = hc/\lambda k_B T$ pour démontrer que la longueur d'onde du maximum d'intensité obéit à la loi du déplacement de Wien :

$$\lambda = \frac{b}{T} \ , \tag{3}$$

où la constante de déplacement de Wien est $b = hc/k_B x$ et x est solution de

$$5\mathrm{e}^{-x} + x - 5 = 0 \ . \tag{4}$$

b. [8 points] Écrivez un programme Python pour solutionner l'équation (4) à une précision $\varepsilon = 10^{-6}$ avec la méthode par relaxation, et ainsi trouver la valeur de la constante du déplacement de Wien.

c. [6 points] La solution trouvée numériquement en **b** est-elle l'unique solution de l'équation (4) ? Justifiez votre réponse à l'aide d'un graphique.

d. [6 points] Comment justifiez-vous que la méthode par relaxation ait pu vous permettre d'obtenir une solution à l'équation (4) ? Si vous avez identifié une autre solution, pourquoi la méthode par relaxation n'a-t-elle pas convergé vers cette autre solution ?

e. [4 points] La loi du déplacement de Wien est à la base de la pyrométrie optique, une méthode permettant de mesurer la température d'objets par l'observation de la couleur de la radiation qu'ils émettent. Cette méthode est couramment employée pour estimer la température de surface de corps célestes. La lougueur d'onde correspondant au maximum d'émission du spectre solaire est $\lambda = 502$ nm. Connaissant cette valeur, ainsi que la constante du déplacement calculée précédemment, estimez la température de la surface du soleil et comparez votre réponse avec la valeur généralement acceptée.

1. Inspiré du problème 6.13 de *Computational Physics* [1].

2 Puits de potentiel [2]

Soit un puits de potentiel rectangulaire de largeur w et de hauteur V. Il peut être démontré, par l'entremise de l'équation de Schrödinger, que les énergies permises E pour une particule quantique de masse m piégée dans ce puits sont solutions de

$$\tan\sqrt{\frac{w^2 mE}{2\hbar^2}} = \begin{cases} \sqrt{\dfrac{V-E}{E}} & \text{pour les états pairs} \\ -\sqrt{\dfrac{E}{V-E}} & \text{pour les états impairs} \end{cases} \tag{5}$$

où les états sont numérotés de telle sorte que 0 correspond à l'état fondamental, 1 au premier état excité et ainsi de suite.

a. [6 points] Pour un électron ($m = 9.1094 \times 10^{-31}$ kg) dans un puits de hauteur $V = 20$ eV et de largeur $w = 1$ nm, écrivez un programme Python pour afficher dans un même graphique les quantités

$$y_1 = \tan\sqrt{\frac{w^2 mE}{2\hbar^2}}\ , \quad y_2 = \sqrt{\frac{V-E}{E}}\ , \quad y_3 = -\sqrt{\frac{E}{V-E}}\ , \tag{6}$$

pour E entre 0 et 20 eV. À partir de ce graphique, approximez les énergies des six premiers états de la particule.

b. [10 points] Écrivez un autre programme Python pour calculer plus précisément ($\varepsilon = 0.001$ eV) ces six valeurs avec la méthode de la bissection.

3 Ampoule incandescente [3]

Une ampoule incandescente est un dispositif simple : un filament, habituellement en tungstène, est chauffé par le passage d'un courant électrique pour émettre du rayonnement électromagnétique. À peu près toute la puissance consommée par l'ampoule est émise, bien que pas nécessairement dans la plage du visible.

Définissons l'efficacité comme la fraction du rayonnement total qui tombe dans la bande visible du spectre électromagnétique. En première approximation, utilisons la loi de Planck pour estimer la distribution spectrale de l'émission (par unité de longueur d'onde) du filament à la température T :

$$I(\lambda) = 2\pi A h c^2 \frac{\lambda^{-5}}{\mathrm{e}^{hc/\lambda k_B T} - 1} \tag{7}$$

où A est la surface du filament, h est la constante de Planck, c est la vitesse de la lumière et k_B est la constante de Boltzmann.

En considérant que la plage visible du spectre électromagnétique va de $\lambda_1 = 390$ nm à $\lambda_2 = 750$ nm, la fraction utile du rayonnement (l'efficacité η) sera donnée par le ratio de $\int_{\lambda_1}^{\lambda_2} I(\lambda)d\lambda$ et de $\int_0^\infty I(\lambda)d\lambda$, soit

$$\eta = \frac{\displaystyle\int_{\lambda_1}^{\lambda_2} \frac{\lambda^{-5}}{\mathrm{e}^{hc/\lambda k_B T} - 1} d\lambda}{\displaystyle\int_0^\infty \frac{\lambda^{-5}}{\mathrm{e}^{hc/\lambda k_B T} - 1} d\lambda}\ , \tag{8}$$

2. Inspiré de l'exercice 6.14 de *Computational Physics* [1].
3. Inspiré de l'exercice 6.18 de *Computational Physics* [1].

où les constantes et la surface A ont été annulées. Avec la substitution $x = hc/\lambda k_B T$, ceci peut aussi être écrit comme

$$\eta = \frac{\displaystyle\int_{hc/\lambda_2 k_B T}^{hc/\lambda_1 k_B T} \frac{x^3}{\mathrm{e}^x - 1} dx}{\displaystyle\int_0^\infty \frac{x^3}{\mathrm{e}^x - 1} dx} = \frac{15}{\pi^4} \int_{hc/\lambda_2 k_B T}^{hc/\lambda_1 k_B T} \frac{x^3}{\mathrm{e}^x - 1} dx \ , \tag{9}$$

où l'intégrale au dénominateur a pu être calculée de façon exacte.

a. [8 points] Écrivez une fonction Python acceptant une température T en argument et calculant l'efficacité η selon l'équation précédente. Utilisez l'implémentation de *Computational Physics* [1] de la quadrature de Gauss pour calculer l'intégrale[4], avec $N = 100$. Utilisez cette fonction pour réaliser un graphique de l'efficacité η en fonction de la température entre 300 K et 10 000 K. Normalement, vous devriez voir une température pour laquelle l'efficacité est maximale.

b. [12 points] Calculez la température correspondant au maximum d'efficacité de l'ampoule, à 1 K près, avec la méthode du ratio doré (*golden ratio search*). Pour atteindre ce niveau d'exactitude, vous devrez utiliser des valeurs de constantes fondamentales précises à plusieurs décimales (utilisez `scipy.constants`).

c. [4 points] Est-il possible d'utiliser un filament de tungstène à la température que vous avez trouvée ? Si non, pourquoi ?

4 Algorithme QR[5]

Dans cet exercice, vous allez développer un algorithme permettant de calculer les vecteurs et valeurs propres d'une matrice réelle symétrique à partir de la décomposition QR.

Soit une matrice carrée et réelle $\mathbf{A}$ que l'on souhaite exprimer comme

$$\mathbf{A} = \mathbf{QR} \tag{10}$$

où $\mathbf{Q}$ est une matrice orthogonale et $\mathbf{R}$ une matrice triangulaire supérieure. Pour y arriver, imaginons la matrice $\mathbf{A}$ de dimensions $N \times N$ comme un ensemble de N vecteurs colonnes $\boldsymbol{a}_0 \dots \boldsymbol{a}_{N-1}$

$$\begin{pmatrix} | & | & | & \dots \\ \boldsymbol{a}_0 & \boldsymbol{a}_1 & \boldsymbol{a}_2 & \dots \\ | & | & | & \dots \end{pmatrix} \ .$$

Définissons maintenant deux nouveaux ensembles de vecteurs $\boldsymbol{u}_0 \dots \boldsymbol{u}_{N-1}$ et $\boldsymbol{q}_0 \dots \boldsymbol{q}_{N-1}$ tels que

$$\begin{aligned} \boldsymbol{u}_0 &= \boldsymbol{a}_0 \ , & \boldsymbol{q}_0 &= \frac{\boldsymbol{u}_0}{|\boldsymbol{u}_0|} \ , \\ \boldsymbol{u}_1 &= \boldsymbol{a}_1 - (\boldsymbol{q}_0 \cdot \boldsymbol{a}_1)\boldsymbol{q}_0 \ , & \boldsymbol{q}_1 &= \frac{\boldsymbol{u}_1}{|\boldsymbol{u}_1|} \ , \\ \boldsymbol{u}_2 &= \boldsymbol{a}_2 - (\boldsymbol{q}_0 \cdot \boldsymbol{a}_2)\boldsymbol{q}_0 - (\boldsymbol{q}_1 \cdot \boldsymbol{a}_2)\boldsymbol{q}_1 \ , & \boldsymbol{q}_2 &= \frac{\boldsymbol{u}_2}{|\boldsymbol{u}_2|} \ , \end{aligned}$$

4. Voir http://www-personal.umich.edu/~mejn/cp/programs/gaussxw.py et http://www-personal.umich.edu/~mejn/cp/programs/gaussint.py.

5. Inspiré de l'exercice 6.8 de *Computational Physics* [1].

et ainsi de suite. Les formules générales pour calculer $\boldsymbol{u}_i$ et $\boldsymbol{q}_i$ sont

$$\boldsymbol{u}_i = \boldsymbol{a}_i - \sum_{j=0}^{i-1} (\boldsymbol{q}_j \cdot \boldsymbol{a}_i)\boldsymbol{q}_j \ , \qquad\qquad \boldsymbol{q}_i = \frac{\boldsymbol{u}_i}{|\boldsymbol{u}_i|} \ .$$

a. [6 points] Démontrez, par induction ou autrement, que les vecteurs $\boldsymbol{q}_i$ sont orthonormaux.

En réarrangeant la définition des vecteurs, on a

$$\begin{aligned}
\boldsymbol{a}_0 &= |\boldsymbol{u}_0|\boldsymbol{q}_0 \\
\boldsymbol{a}_1 &= |\boldsymbol{u}_1|\boldsymbol{q}_1 + (\boldsymbol{q}_0 \cdot \boldsymbol{a}_1)\boldsymbol{q}_0 \\
\boldsymbol{a}_2 &= |\boldsymbol{u}_2|\boldsymbol{q}_2 + (\boldsymbol{q}_0 \cdot \boldsymbol{a}_2)\boldsymbol{q}_0 + (\boldsymbol{q}_1 \cdot \boldsymbol{a}_2)\boldsymbol{q}_1
\end{aligned}$$

et ainsi de suite. Les vecteurs $\boldsymbol{q}_i$ peuvent être groupés pour former les colonnes d'une matrice qui permet d'exprimer les équations précédentes comme une seule opération matricielle :

$$\mathbf{A} = \begin{pmatrix} | & | & | & \dots \\ \boldsymbol{a}_0 & \boldsymbol{a}_1 & \boldsymbol{a}_2 & \dots \\ | & | & | & \dots \end{pmatrix} = \begin{pmatrix} | & | & | & \dots \\ \boldsymbol{q}_0 & \boldsymbol{q}_1 & \boldsymbol{q}_2 & \dots \\ | & | & | & \dots \end{pmatrix} \begin{pmatrix} |\boldsymbol{u}_0| & \boldsymbol{q}_0 \cdot \boldsymbol{a}_1 & \boldsymbol{q}_0 \cdot \boldsymbol{a}_2 & \dots \\ 0 & |\boldsymbol{u}_1| & \boldsymbol{q}_1 \cdot \boldsymbol{a}_2 & \dots \\ 0 & 0 & |\boldsymbol{u}_2| & \dots \end{pmatrix} . \tag{11}$$

Vous pouvez vous en convaincre en effectuant la multiplication matricielle à droite pour retrouver les vecteurs $\boldsymbol{a}_i$.

Sous cette forme, la matrice contenant les vecteurs $\boldsymbol{q}_i$ est orthogonale car il a été démontré à la question **a** que les $\boldsymbol{q}_i$ sont orthonormaux. De toute évidence, la matrice multipliant celles des $\boldsymbol{q}_i$ plus haut est triangulaire supérieure. Nous avons donc trouvé la décomposition QR de $\mathbf{A}$, soit

$$\mathbf{Q} = \begin{pmatrix} | & | & | & \dots \\ \boldsymbol{q}_0 & \boldsymbol{q}_1 & \boldsymbol{q}_2 & \dots \\ | & | & | & \dots \end{pmatrix} , \qquad \mathbf{R} = \begin{pmatrix} |\boldsymbol{u}_0| & \boldsymbol{q}_0 \cdot \boldsymbol{a}_1 & \boldsymbol{q}_0 \cdot \boldsymbol{a}_2 & \dots \\ 0 & |\boldsymbol{u}_1| & \boldsymbol{q}_1 \cdot \boldsymbol{a}_2 & \dots \\ 0 & 0 & |\boldsymbol{u}_2| & \dots \end{pmatrix} . \tag{12}$$

b. [12 points] Écrivez un programme Python prenant en argument une matrice carrée réelle $\mathbf{A}$ et qui retourne deux matrices $\mathbf{Q}$ et $\mathbf{R}$ telles que définies plus haut. À titre de test, obtenez la décomposition QR de

$$\mathbf{B} = \begin{pmatrix} 1 & 4 & 8 & 4 \\ 4 & 2 & 3 & 7 \\ 8 & 3 & 6 & 9 \\ 4 & 7 & 9 & 2 \end{pmatrix} \tag{13}$$

et assurez-vous du résultat en vérifiant que le produit de $\mathbf{Q}$ et de $\mathbf{R}$ donne bien $\mathbf{A}$.

c. [12 points] À l'aide de cette fonction, écrivez un programme permettant de calculer les vecteurs et valeurs propres d'une matrice carrée réelle[6]. La nouvelle fonction prendra un deuxième argument, soit la valeur ε que devra atteindre tous les éléments non-diagonaux (i.e. tous plus petits que ε en valeur absolue). Testez votre programme sur la matrice précédente et une valeur $\varepsilon = 10^{-6}$. Que se passe-t-il avec $\varepsilon = 10^{-12}$ ou $\varepsilon = 10^{-18}$?

6. Voir la section 6.2 de *Computational Physics* [1] pour les détails de l'algorithme.

Instructions pour la remise

1. Le travail devra être complété individuellement ou en binôme.
2. Les codes associés à chaque question devront être présentés dans un fichier `.py` distinct. Vous devrez les organiser tel qu'il sera possible de reproduire vos résultats, vos figures, etc. en exécutant votre script (ex. : `python votre_script.py`). Utilisez des noms de fichiers clairs et explicites. Notez que vous pouvez placer certaines fonctions dans leur propre fichier à part (qui sera importé dans votre script) si vous le jugez nécessaire. Voici un exemple :

```python
import numpy as np
from mon_fichier import ma_fonction_compliquee


def la_fonction(x):
    # [votre code ici]


def solution_methode_par_relaxation(param):
    # [votre code ici]
    ma_fonction_compliquee(param)


def tracer_graphique(param):
    # [votre code ici]
    fig.savefig("TP2_numero2_<nom de la figure>.pdf")

if __name__ == "__main__":

    point_fixe = solution_methode_par_relaxation(param1)

    tracer_graphique(param2)

    print("Le point fixe se situe à x = {:.6f}".format(point_fixe))
```

3. Toutes vos réponses devront être colligées dans un document de format `pdf` produit avec l'éditeur de texte de votre choix (LaTeX, Microsoft Word, Google Docs, etc.). Ce document contiendra **toutes** informations pertinentes permettant au lecteur d'apprécier vos résultats et conclusions (ex. développements mathématiques, figures, explications, interprétations), fera référence au script Python que vous avez utilisé de même qu'à d'éventuelles références bibliographiques.
4. La qualité de la présentation est très importante (utilisation de sections, de graphiques appropriés, de mise en contexte, etc.). Vous pouvez vous référez au document `ConsignesTP.ipynb` fourni sur le portail pour plus de détails sur les façons adéquates de présenter vos résultats.
5. Vous devrez rassembler et compresser tous vos fichiers dans un fichier `zip` et le remettre dans la boîte de dépôt créée à cette fin. Prenez soin de bien indiquer votre (ou vos) nom(s) dans le document `pdf` et dans chacun de vos scripts. Pour faciliter la tâche de classification, utilisez la nomenclature suivante pour le fichier transmis (un seul) : `TPn_nom1.zip` ou `TPn_nom1_nom2.zip` .

Références

[1] M. E. J. Newman, *Computational Physics* (CreateSpace Independent Publishing Platform, 2012), p. 562, ISBN : 978-1480145511.