

Calcul scientifique 2 - TP3

Consignes

La note finale tiendra compte du respect des consignes suivantes. Pour toute question, me demander sur [Slack](#), sur le canal ou en message privé.

1. Créer un seul fichier `identifiant.py` où `identifiant` peut être **soit votre nom ou bien votre no. étudiant**.
2. Remplir **obligatoirement** l'attestation de recherche personnelle dans l'en-tête de votre fichier.
3. Respecter **scrupuleusement** les noms et les formats d'entrée-sortie des fonctions, par exemple `exo1` et non `Exo1` ni `Exercice1`.
4. Tout vecteur/matrice doit être un tableau Numpy (càd un `numpy.ndarray`), **évités donc les listes Python**.
5. Déposez votre fichier `identifiant.py` dans Moodle **AVANT LA DATE BUTOIR (voir Moodle)**. Pas d'envoi par Slack, courriel, ou autre.
6. Merci de compléter le fichier modèle (ci-dessous)

Exercice 1

Ecrire une fonction `exo2` qui transforme une matrice A symétrique en une matrice de Hessenberg H avec la méthode de Givens et de Householder. Le choix de la méthode pourra se faire via un paramètre `method='householder'` ou `method='givens'`.

Exercice 2

Ecrire une fonction `exo2` qui, dans le contexte du Chapitre 5 § *Lien avec la constante de coercivité*, estime la constante C_P à deux décimales bonnes après la virgule. Cette fonction aura pour seul argument n , le nombre de point de discrétisation.

Suggestions :

- On pourra considérer le problème aux valeurs propres $-u'' = \lambda u$ avec $u(0) = u(1) = 0$ aux bords comme dans le cours,
- considérer l'approximation éléments finis 1D de ce problème menant à un problème aux valeurs propres **généralisés** $A_h U = \lambda M_h U$
- Faire une décomposition de Cholesky de M_h (on pourra utiliser `scipy.chol`) et ainsi avoir $M_h = LL^T$,
- Remarquer alors qu'on a $LA_h(L^{-1})^T L^T U = \lambda L^T U$, qui peut se réécrire comme un **problème aux vp classiques** $CW = \lambda W$ avec $C = L^{-1}A_h(L^{-1})^T$ et $W = L^T U$.
- On pourra utiliser la routine `scipy.linalg.lapack.dtrtri` pour inverser plus rapidement une matrice dense triangulaire supérieure (ou inférieure). Il existe aussi `scipy.linalg.lapack.strtri` pour les creuses.
- Calculer la plus petite valeur propre de la matrice C (voir Chap 4) et en déduire C_P à deux décimales bonnes après la virgule.

Exercice 3

Ecrire une fonction `exo3` qui résoud le problème

$$\int_I k(x) u'(x) v'(x) dx = \int_I f(x) v(x) dx, \quad \forall v \in H_0^1(I), \quad x \in I = (0, 1).$$

avec une méthode élément fini P^1 . On prendra $f = 1$ et $k(x) = k_0 + k_1 x$ où $k_0, k_1 > 0$ sont deux paramètres de la fonction `exo3` à renseigner. On notera n le nombre de points de discrétisation de I .

Suggestions:

- On commencer par exprimer la forme bilinéaire $a(u, v)$ du problème comme une somme de deux formes plus simples dépendant de k_0 et k_1 : $a(u, v) = k_0 a_0(u, v) + k_1 a_1(u, v)$.
- Calculer à la main les deux matrices A_h et B_h associées à ces deux formes bilinéaires, voir Chap 5. On pourra s'aider de la librairie `sympy` (voir docs.sympy.org) pour vérifier les intégrales.
- On pourra résoudre le système avec la méthode de son choix.

Modèle de fichier à rendre

Merci d'utiliser le fichier modèle ci-après, que vous renommerez en `nom.py` (où `nom` est votre nom en **minuscules**):

```
# Déclaration de recherche personnelle

# Je soussigné(e), [VOTRE NOM], étudiant(e) en Calcul Scientifique 2, déclare par
la présente que le travail que je sou mets, reflète une recherche personnelle
effectuée dans le cadre de ce cours. Je certifie que toutes les informations,
analyses, idées et conclusions présentées dans ce travail sont le résultat de mon
propre effort intellectuel et de mes propres recherches.

# Je reconnais avoir respecté les principes éthiques de l'intégrité académique
tout au long du processus de recherche et de rédaction de ce travail. Je m'engage
à fournir ci-après toutes les sources d'information, qu'elles soient imprimées,
électroniques ou orales, conformément aux normes de citation académique prévues
par le cours.

# Je comprends que la falsification, le plagiat ou toute autre forme de tricherie
académique sont des violations graves du code de conduite académique de notre
institution et peuvent entraîner des sanctions sévères, y compris l'échec du
cours.

# Je m'engage à assumer la responsabilité totale du contenu de ce travail et à
accepter les conséquences de tout acte contraire à l'intégrité académique.

# Je déclare avoir utilisé les sources suivantes :
# (...)

# Exemples :
# - Article Wikipédia ...
# - ChatGPT avec le prompt "..."
# - Livre de Machin et Bidule, 4e édition
# - Discussions avec un collègue

import numpy as np
#...

def exo1(A, method='givens'):
    #... construction de H
    return H

def exo2(n=100):
    # ..
```

```
    return CP
```

```
def exo3(k0, k1, n=100):
```

```
    # ...
```

```
    return uh
```