Calcul scientifique 2 - TP1

Exercices

Exercice 1 Écrire une fonction `exo1` qui calcule le stockage CSR d'une matrice pleine M et renvoie le triplet V, C, R comme indiqués dans le cours.

Exercice 2 Écrire une fonction `exo2` qui calcule le stockage Skyline (*ligne de ciel*) d'une matrice pleine M et renvoie I, V comme dans le cours.

Exercice 3 Écrire une fonction `exo3` qui calcule le produit d'une matrice carrée symétrique A au format Skyline (I, V) avec un vecteur b.

Exercice 4 Écrire une fonction `exo4` qui calcule la factorisation LU (avec pivot partiel) d'une matrice pleine M et renvoie L, U et P

Exercice 5 Écrire une fonction `exo5(L,U,b,P=None)` qui résout le système linéaire $LUx = b$ en utilisant la méthode de descente-remontée de factorisation LU.

- Si P n'est pas spécifié, alors on applique l'algorithme LU sans pivot.
- Si P est spécifié, alors on résout $PAx = Pb$.

Exercice 6 Écrire une fonction `exo6` appliquant l'algorithme de Cuthill-McKee. Celle ci prend une matrice carrée pleine A, de taille n, et renvoie la matrice (pleine) de permutation P.

Indication : Vous pourrez utiliser la librairie Python `networkx` pour manipuler des graphes. Utilisez en particulier la fonction `nx.from_numpy_array()` pour créer un graphe à partir d'une matrice numpy. Pour plus d'informations, consultez la documentation : https://networkx.org/documentation/stable/tutorial.html.

Exercice 7 Écrire une fonction `exo7` renvoyant le *profile* d'une matrice Skyline $A = (I, V)$ par la formule : $\text{profile}(A) = n + \sum_{i=1}^{n}(i - f_i(A))$ avec $f_i(A) = \min\{j : 1 \leq j \leq i, \quad a_{i,j} \neq 0\}$ pour $1 \leq i \leq n$.

Consignes

1. Créer un seul fichier `nom.py` (en utilisant le modèle) sur la base du modèle donné en bas de ce document.

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`), évitez donc les liste Python via `[]` ou `list()`.
5. Déposez votre fichier `nom.py` dans Moodle **AVANT LE 12 OCTOBRE 2023**. Pas d'envoi par Slack, courriel, ou autre.

La note finale tiendra compte du respect de ces consignes. Pour toute question, me demander en message privé sur **Slack** ou `#csmi-2024`.

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 soumets, 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 :
# (...)
```

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

import numpy as np
import networkx as nx

# Exercice 1 - CSR
def exo1(M):
    # ...
    return V,C,R

# Exercice 2 - Skyline
def exo2(M):
    # ...
    return I,V

# Exercice 3 - multiplication A (Skyline) * b
def exo3(I,V,b):
    # ...
    return Ab

# Exercice 4 - factorisation LU avec pivot partiel
def exo4(M):
    # ...
    return L,U,P

# Exercice 5 - résolution d'un système linéaire avec/sans pivot
def exo5(L, U, b, P=None):
    # ...
    return x

# Exercice 6 - Algo de Cuthill-McKee
def exo6(M):
    # ...
    return P

# Exercice 7 - Profile
def exo7(I,V):
    # ...
    return p
```