Calcul scientifique 2 - TP2

Dans toute la feuille d'exercice, on supposera, comme dans le cours qu'une matrice A se décompose comme $A = D - E - F$ avec

- D, la partie diagonale de A,
- $-E$ la partie triangulaire inférieure (stricte) de A,
- $-F$ la partie triangulaire supérieure (stricte) de A.

Exercices

Exercice 1 Soit un problème $Ax = b$, avec A *creuse* et inversible et b dense. Ecrire une fonction `exo1` qui implémente un algo itératif de la forme

$$x_{k+1} = M^{-1}Nx_k + M^{-1}b$$

pour calculer une estimation `x` de x et `niter` le nombre d'itérations pour atteindre une tolérance ε donnée. On laisse la possibilité à l'utilisateur de choisir M via le paramètre `prec=` .

Exemple d'utilisation

```
from scipy.sparse import rand as srand
from numpy.random import rand

A = srand(4, 4, density=0.3) # qu'on suppose inversible
b = rand(4, 1)

x, niter = exo1(A, b, tol=1e-10, prec='jacobi')
x, niter = exo1(A, b, tol=1e-8, prec='gs')
x, niter = exo1(A, b, tol=1e-6, prec='ilu')
```

Remarque On pourra utiliser la librairie `scipy.sparse.linalg` , en particulier les routines `spilu` pour construire la décomposition ILU de A. Une fois la décomposition `ilu` faite, on pourra utiliser `.solve()` pour résoudre le système. Enfin, on pourra utiliser `triu` et `tril` pour extraire les parties triangulaires sup/inférieures d'une matrice creuse.

Exercice 2 S'inspirer de l'exercice ci-dessus pour implémenter la méthode SOR, c'est à dire la méthode itérative avec $M_w = \frac{1}{w}D - E$ et $N_w = (\frac{1}{w} - 1)D + F$ avec $w \in (0, 2)$ un paramètre fixé par l'utilisateur.

Exemple d'utilisation

```
x, niter = exo2(A, b, w)
```

Exercice 3 Soit A la matrice bande, creuse, de taille $n \times n$ définie comme :

$$A = \begin{pmatrix} 4 & -2 & 0 & 0 & 0 & \cdots & 0 \\ -1 & 4 & -2 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 4 & -2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & -1 & 4 & -2 \\ 0 & 0 & 0 & 0 & 0 & -1 & 4 \end{pmatrix}$$

Déterminer le paramètre $w \in [0.5, 2.5]$ optimal tel que la méthode itérative avec $M = \frac{1}{w}(D - E)$ converge le plus rapidement possible pour une matrice A donnée (peu importe n). Attention, ce n'est pas la même méthode qu'au-dessus.

Exemple d'utilisation

```
A = ...
wopt = exo3(A)
```

Exercice 4 Soit A SDP, écrire l'algorithme du **gradient conjugué** (avec possibilité de préconditionner par Jacobi ou Gauss-Seidel) pour résoudre $Ax = b$ à une tolérance ε fixée.

Exemple d'utilisation

```
x, niter = exo4(A, b, tol=1e-10, prec='jacobi')
x, niter = exo4(A, b, tol=1e-10, prec='gs')
```

De la même manière qu'au dessus, on souhaite récupérer x et le nombre d'itérations `niter`.

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 `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 listes Python.**
5. Déposez votre fichier `nom.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)

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 :
# (...)

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

import numpy as np
import scipy.sparse sp

# Exercice 1
def exo1(A, b, tol=1e-10, prec='jacobi'):
    # ...
    return x, niter

# Exercice 2
def exo2(A, b, w):
    # ...
    return x, niter

# Exercice 3
def exo3(A):
    # ...
    return wopt

# Exercice 4
def exo4(A, b, tol=1e-10, prec='jacobi'):

    return x, niter
```