Université de Marseille
Licence de Mathématiques, analyse numérique
Travaux Pratiques 3, en python

Pour $A \in \mathcal{M}_n(\mathbb{R})$ $(n > 1)$ inversible et $b \in \mathbb{R}^n$ donnés, on s'intéresse dans cet exercice aux méthodes itératives permettant de calculer la solution du système linéaire $Ax = b$.
On va considère des méthodes qui s'écrivent sous le formalisme suivant :

$$Mx^{(k+1)} = Nx^{(k)} + b, \text{ pour tout } k \geq 0.$$

On rappelle que pour que la suite $(x^{(k)})_{k \in N}$ converge (quelquesoit $x^{(0)}$ et quelquesoit b) il faut et il suffit que $\rho(M^{-1}N) < 1$.
On met A sous la forme $A = D - E - F$, où D est la diagonale de A, E est la partie triangulaire inférieure de A (diagonale exclue) et F est la partie triangulaire supérieure de A (diagonale exclue).
Avec python, on peut définir les matrices D ,E et F à partir de A en utilisant les commandes **numpy.diag**, **numpy.tril** et **numpy.triu**.
Pour M, N et ε donnés avec $\varepsilon > 0$, la méthode s'écrit :
Initialisation Choisir $x^{(0}$ dans $\mathbb{R}^n$,
Itération Pour $k \geq 0$, si $|r^{(k}| > \varepsilon$, avec $r^{(k)} = b - Ax^{(k)}$, calculer $x^{(k+1)}$ solution de $Mx^{(k+1)} = Nx^{(k)} + b$.
Pour tester et comparer les méthodes de Jacobi, Gauss-Seidel et SOR, on va s'intéresser à des problèmes dont on connait la solution exacte, notée x. Les deux problèmes considérés sont :
Premier problème

$$A = \begin{bmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{bmatrix} \text{ et } b = \begin{bmatrix} 32 \\ 23 \\ 33 \\ 31 \end{bmatrix}.$$

Second problème
Pour ce second probème, on prend pour A la matrice de l'exercice 1 du tp1. Pour $n \in \mathbb{N}^\star$, on note A_n cette matrice (qui était notée A_h au tp1, avec $h = 1/(n+1)$). Le second membre b correspond à $f(x) = \pi^2 \sin(2\pi x)$ pour $x \in [0, 1]$ (il est différent de celui des tp 1 et 2). Pour $n \in \mathbb{N}^\star$, on note b_n ce second membre. Le problème discrétisé s'écrit donc $A_n x_n = b_n$.
On pourra se limiter à considérer le cas $n = 10$.

1. On utilise ici la méthode de Jacobi ($M = D$, $N = E + F$).

(a) Construire une fonction dont les paramètres sont A, b, ε et $x^{(0)}$ et qui calcule, avec la méthode de Jacobi, la solution approchée du système linéaire $Ax = b$ et retourne le résidu ($|r^{(k}|$) et l'erreur ($|x - x^{(k)}|$) à chaque itération de la méthode.

(b) Pour les deux problèmes considérés, calculer $\rho(M^{-1}N)$.
Si $\rho(M^{-1}N) < 1$, construire pour $\varepsilon = 10^{-6}$ et deux choix différents de $x^{(0)}$ (par exemple, $x^{(0)} = 0$ et $x^{(0)}$ dont toutes les composantes sont égales à 1) un graphique, avec une échelle semilog, donnant l'évolution de l'erreur en fonction de k et l'évolution de la quantité $\rho(M^{-1}N)^k$ [Utiliser **matplotlib.pyplot.semilogy**].
L'erreur se comporte-t-elle comme $\rho(M^{-1}N)^k$? Donner aussi le nombre d'iérations utilisées.

2. Reprendre les questions 1a et 1b avec la méthode de Gauss-Seidel ($M = D - E$, $N = F$).

3. On s'intéresse maintenant à la méthode SOR, c'est-à-dire $M = \frac{1}{\omega}D - E$, $N = F + (\frac{1}{\omega} - 1)D$ avec $0 < \omega < 2$.

   (a) Reprendre la question 1a en ajoutant le paramètre ω.

   (b) Calculer une approximation de la valeur de ω minimisant $\rho(M^{-1}N)$.

   (c) Reprendre la question 1b avec la valeur de ω trouvée dans la question précédente.

4. Comparer l'efficacité des trois méthodes (Jacobi, Gauss-Seidel et SOR) sur ces deux exemples.

   Pourquoi dans certains cas la méthode de Jacobi donne la solution (approchée) avec un nombre d'itérations inférieur au nombre d'itérations nécessaire pour la méthode de Gauss-Seidel ?