TP 3 : Méthodes itératives de résolution de systèmes linéaires.

1) On cherche à résoudre le système $Ax = b$ avec

$$A = \begin{pmatrix} 1 & 2 & 2 & 1 \\ -1 & 2 & 1 & 0 \\ 0 & 1 & -2 & 2 \\ 1 & 2 & 1 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

a) On définit M_J comme la matrice diagonale qui a la même diagonale que A, puis on pose $N_J = M_J - A$. On consultera la page d'aide de la fonction `diag`

b) On définit la suite $(x_n)_{n\in\mathbb{N}}$ de points de $\mathbb{R}^4$ par

$$\begin{aligned} &x_0 = b, \\ &x_{n+1} = M_J^{-1}(N_J x_n + b) \quad \forall n \in \mathbb{N}. \end{aligned}$$

C'est ce qu'on appelle la méthode de Jacobi.

c) Calculer le rayon spectral de $M_J^{-1}N_J$. Que constate-t-on ?

d) Calculer $x_{100}, x_{200}, x_{300}$. Cette suite converge-t-elle ?

2) Reprendre l'exercice 1) en remplaçant M_J par [1] :
- M_{GS} =`tril(A)`, ce qui donne la méthode de Gauss-Seidel,
- M_{SOR} =`2*tril(A)`, ce qui donne une méthode de surrelaxation successive (Successive Over-Relaxation).

3)

a) Écrire une fonction permettant de résoudre un système linéaire au moyen de l'algorithme de Jacobi.

```
function [x,n]=Jacobi(A,b,iter_max,tol)
...
endfunction
```

Les arguments d'entrée de cette fonction seront la matrice du système A, le second membre b, un seuil de tolérance `tol` et un nombre maximal d'itérations `iter_max`. On demande que l'algorithme s'arrête lorsque

$$\|x_{n+1} - x_n\| \leq \texttt{tol} \text{ ou } n > \texttt{iter\_max}$$

Cette fonction retourne la solution approchée `x`, ainsi que le nombre d'itérations effectuées `n`.

b) Calculer la solution approchée du système $Ax = b$ fournie par cette méthode lorsque A est la matrice du Laplacien en dimension 1 à 20 points, c'est-à-dire la matrice 20×20 définie par

$$A = \frac{1}{h^2}\begin{pmatrix} 2 & -1 & 0 & \cdots & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & \cdots & 0 & -1 & 2 \end{pmatrix}, \qquad h = \frac{1}{21},$$

1. `tril` extrait la matrice triangulaire inférieure (TRIangular Lower).

et en fixant un second membre b aléatoire. On pourra créer A en utilisant une fonction du TP1.

c) Comparer la solution approchée obtenue en b) avec la solution donnée par la commande `A \ b`.

d) Tracer la courbe d'erreur de la solution en fonction du seuil de tolérance ε (en échelle logarithmique).

Si `X` *est un vecteur en abscisse et* `Y` *est un vecteur en ordonnée, on peut tracer le graphe de* `X` *en fonction de* `Y` *en échelle logarithmique à l'aide de la commande*

```
plot2d(X, Y, logflag="..");
```
(1)

où chacun des deux points de l'argument de `logflag` *correspondent à l'échelle que l'on souhaite pour l'abscisse et l'ordonnée. On mettra* `l` *pour l'échelle logarithmique et* `n` *pour l'échelle normale. Par exemple* `logflag="nl"` *donnera une échelle normale en abscisse et logarithmique en ordonnée.*

4)

a) Reprendre l'exercice précédent en utilisant l'algorithme de Gauss-Seidel.

b) Comparer les performances des algorithmes de Jacobi et de Gauss-Seidel en traçant le nombre d'itérations nécessaire pour chaque algorithme en fonction du seuil de tolérance (sur un même graphe).

5) Pour rappel, la méthode SOR (ou méthode de Gauss-Seidel relaxée) consiste à construire la suite $(x_n)_{n\in\mathbb{N}}$ définie pour tout $n \in \mathbb{N}$ par :

$$\begin{aligned} &x_0 = b, \\ &x_{n+1} = M_{SOR}^{-1}(N_{SOR}x_n + b), \end{aligned}$$

où $M_{SOR} = \frac{1}{\omega}\mathrm{diag}(A) + L$ et $N_{SOR} = M_{SOR} - A$, L étant la matrice strictement triangulaire inférieure de A. Ici $\omega \in]0, 2[$ est le paramètre de relaxation de la méthode ; on retrouve la méthode de Gauss-Seidel lorsque $\omega = 1$. Noter que la méthode SOR peut s'écrire de façon équivalente :

$$\begin{aligned} &x_0 = b, \\ &\hat{x}_{n+1} = M_{GS}^{-1}(N_{GS}x_n + b), \\ &x_{n+1} = (1-\omega)\ x_n + \omega\ \hat{x}_{n+1}. \end{aligned}$$

a) Modifier la fonction créée à la question précédente pour mettre en œuvre la méthode SOR.

b) Tester cette méthode pour différentes valeurs du paramètre de relaxation ω, et essayer d'estimer (expérimentalement) la valeur du paramètre de relaxation qui permet la convergence la plus rapide. On pourra tracer un graphe d'erreur à seuil de tolérance fixé et en fonction de ω.

c) Comparer cette valeur à la valeur théorique du paramètre de relaxation optimal [2].

d) Reprendre la question 4.b) en ajoutant au comparatif l'algorithme SOR.

2. Pour des matrices tridiagonales symétriques définies positives, la vitesse de convergence de la méthode SOR est optimale pour $\omega = \frac{2}{1+\sqrt{1-\rho(M_J^{-1}N_J)}}$, où M_J et N_J sont les matrices utilisées dans la méthode de Jacobi.