

Forme des comptes rendus des TPs

Les comptes rendus des étudiants doivent comporter les éléments suivants :

1. Partie théorique de la méthode étudiée
2. Programme Matlab
3. Résultats obtenus
4. Conclusion

TP N° 02

Résolution des systèmes linéaires

par les méthodes itératives de Jacobi et de Gauss Seidel

1. Objectif du TP

Le but de ce TP est l'implémentation des méthodes itératives de Jacobi et Gauss Seidel pour la résolution d'un système d'équations linéaires.

2. Rappel de la méthode

Soit une matrice A dont tous les éléments diagonaux sont non nuls. Pour une matrice creuse on doit pouvoir mettre en évidence une diagonale non nulle par des permutations des lignes et des colonnes. On appelle normalisation des équations la mise en évidence d'une telle diagonale et son existence constitue une condition nécessaire pour la non-singularité de la matrice. Considérons alors le système linéaire $Ax=b$ d'ordre 3 ci-après :

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases}$$

Explicitons l'élément diagonal de chaque ligne et réécrivons le système de la façon suivante :

$$\begin{cases} x_1 = (b_1 - a_{12}x_2 - a_{13}x_3)/a_{11} \\ x_2 = (b_2 - a_{21}x_1 - a_{23}x_3)/a_{22} \\ x_3 = (b_3 - a_{31}x_1 - a_{32}x_2)/a_{33} \end{cases}$$

Les x_i dans le membre droit des équations ne sont pas connus, mais supposons que $x^{(k)}$ est une approximation pour $x = A^{-1}b$, alors on peut imaginer d'obtenir une nouvelle approximation en calculant :

$$\begin{cases} x_1^{(k+1)} = (b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)})/a_{11} \\ x_2^{(k+1)} = (b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)})/a_{22} \\ x_3^{(k+1)} = (b_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)})/a_{33} \end{cases}$$

Ceci définit l'itération de Jacobi qui est formalisée avec l'algorithme ci-dessous. L'itération de Jacobi est particulière dans le sens qu'on n'utilise pas les résultats les plus récents et on procède à n résolutions unidimensionnelles indépendantes. Ainsi lorsqu'on calcule, par exemple, $x_2^{(k+1)}$, on utilise $x_1^{(k)}$ et non $x_1^{(k+1)}$ qui est déjà connu.

$$x_i^{(k+1)} = \dots \quad \text{pour } i=1 \text{ jusqu'à } n \quad (*)$$

Algorithme de Jacobi

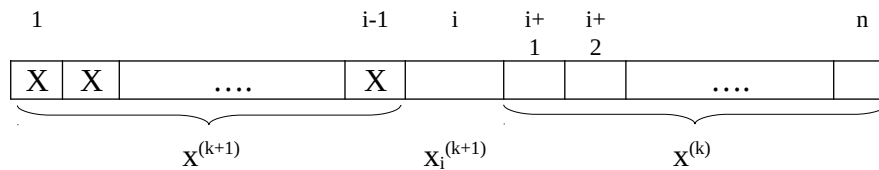
1. Donner un point de départ $x^{(0)} \in \mathbb{R}^n$
2. for $k = 0, 1, 2, \dots$ jusqu'à la tolérance
3. for $i = 1 : n$
4. Equation (*)
5. end
6. end

Pour la tolérance on aura : $|x_i^{(k+1)} - x_i^{(k)}| < \varepsilon$ avec ε : erreur maximal pour $i = 1$ jusqu'à n .

Si l'on modifie l'itération de Jacobi de sorte à tenir compte des résultats les plus récents on obtient l'itération de Gauss-Seidel:

$$x_i^{(k+1)} = \dots \quad \text{pour } i=1 \text{ jusqu'à } n \quad (**)$$

Remarquons qu'il est dans ce cas possible de remplacer au fur et à mesure l'élément $x_i^{(k)}$ par l'élément $x_i^{(k+1)}$ dans le vecteur x .



Les éléments 1 à $i-1$ du vecteur x contiennent déjà les valeurs de l'itération $k+1$ et les éléments $i+1$ à n encore les valeurs de l'itération précédente. On obtient alors l'algorithme suivant :

Algorithme de Gauss-Seidel

1. Donner un point de départ $x^{(0)} \in \mathbb{R}^n$
2. for $k = 0, 1, 2, \dots$ jusqu'à tolérance
3. for $i = 1 : n$
4. Equation (**)
5. end
6. end

3. Travail demandé

* Soit le système $Ax=b$ suivant :

$$\begin{bmatrix} 5 & 1 & 2 & -1 \\ 0 & 4 & 1 & -1 \\ -1 & 1 & 5 & 1 \\ 1 & 2 & 0 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 7 \\ -2 \\ 2 \\ -7 \end{bmatrix}$$

1. Calculer les inconnus x_i en le calculant directement avec : $x = \text{inv}(A) * b$
2. Etudier théoriquement la convergence en utilisant la méthode de Jacobi.
3. Ecrire un script Matlab qui utilise la méthode de Jacobi pour trouver la solution de ce système avec $\varepsilon = 0.001$.
4. Ecrire un script Matlab qui utilise la méthode de Gauss Seidel pour trouver la solution du système $Ax=b$ suivant avec $\varepsilon = 0.001$.
5. Comparer les vitesses de calcul des deux méthodes. Quelle est la méthode la plus rapide?