OPTIMISATION DE PROFIL (Cuthill-McKee)

RALIJAONA Rina
M1 MISA

15 avril 2024

On se propose d'optimiser le rangement et le nombre d'operations pour la résolution de $Ax = B$ pour A symétrique régulière.

1 Algorithme de Cuthill-McKee

Nous cherchons à restructurer une matrice symétrique, définie positive A, afin de concentrer ses éléments non nuls autour de sa diagonale. Pour ce faire, procedons à un réarrangeant des numéros de ses nœuds, ce qui équivaut à une permutation des lignes de la matrice, nous obtenons une nouvelle matrice qui satisfait à nos exigences. Cette nouvelle matrice est obtenue par le produit $A' = P^T AP$, où P est une matrice de permutation.

1.1 Choix du premier noeud

L'objectif de cet algorithme est de trouver le premier nœud à partir duquel commencer l'exploration du graphe. L'algorithme prend en entrée un nœud initial et explore les voisins de ce nœud pour trouver celui qui offre le meilleur point de départ, en privilégiant ceux ayant une excentricité maximale. Cette approche garantit une exploration efficace du graphe en commençant par les régions les plus éloignées du nœud initial.

Objectif : Trouver le premier nœud optimal pour débuter l'algorithme de Cuthill-Mckee
Entrée : Un nœud initial x.
Étapes :

1. **Initialisation :** $n \leftarrow x$
2. **Récupération des voisins :** Nn est la liste des voisins de n
3. **Ajout à la liste :** Ajouter n à la liste des nœuds visités
4. **Calcul de l'excentricité :** Calculer l'excentricité de n et la stocker dans en
5. **Recherche du prochain nœud :** Pour chaque voisin s de n, si s n'a pas été visité et que son excentricité est supérieure à celle de n, choisir s comme prochain nœud à visiter
6. **Récursion :** Appeler récursivement l'algorithme pour le prochain nœud choisi
7. **Retour :** Retourner le nœud initial n

1.2 Algorithme de Cuthill-McKee

Son rôle est de réorganiser les indices des nœuds du graphe associé à A, ce qui a pour effet d'optimiser le profil de la matrice. En d'autres termes, nous ajustons la disposition des éléments de la matrice, ce qui facilite le stockage et le traitement ultérieur.

1. Prendre le premier nœud calculé dans la fonction `firstNode` pour débuter l'algorithme et le stocker dans le vecteur `sommets`.

2. Chercher les voisins de ce nœud et les stocker, dans l'ordre, des sommets ayant le moins de voisins.

3. Prendre ensuite l'élément suivant dans `sommets` et chercher ses voisins.

4. Stocker dans `sommets` les sommets non encore stockés ayant le moins de voisins, et ainsi de suite.

2 Résolution de $Ax = b$

Nous sommes confrontés à la résolution d'un système linéaire de la forme $Ax = b$, où A est une matrice symétrique et définie positive. Afin d'améliorer l'efficacité de cette résolution, nous faisons appel à l'algorithme de Cuthill McKee. Ce processus de réorganisation génère une matrice de permutation P. En appliquant cette permutation à la matrice d'origine A, nous obtenons une nouvelle matrice A' avec un profil optimisé. Plus précisément, cette transformation s'exprime par $A' = P^T AP$.

Nous procedons ensuite à l'algorithme de Cuthill McKee Inverse(CMKI) en inversant les numeros des noeuds obtenu après l'algorithme de Cuthill McKee (CMK).

Après l'optimisation du profil de la matrice A par l'algorithme de Cuthill McKee Inverse, le système initial $Ax = b$ devient :

$$A'x' = b'$$

avec
$x' = P^t x$ et $b' = P^t b$

$$x = P^t x'$$

Pour aboutir à cela, il faut donc :
. Stocker en profil A'
. Puis après la factorisation LDL^T de A' , résoudre en profil $LDL^T x' = b'$