Accélération du code BEM pour l'équation de Helmholtz 2D

TP3 : Lundi 11 Octobre 2021 (à rendre le 29/11/2021)

Rappel de la Philosophie des TPs : Le but de la série de trois TPs est de mettre en oeuvre un solveur BEM rapide pour l'équation de Hemholtz 2D. Nous choisissons le cas de la diffraction d'une onde incidente plane, i.e. $u^{inc} = e^{-i\mathbf{k}.\mathbf{x}}$, par un disque de rayon a centré en $\mathbf{0}$ et de frontière Γ. $\mathbf{k}$ est le vecteur qui permet de déterminer l'angle d'incidence de l'onde. Le nombre d'onde du problème est donné par $k = |\mathbf{k}|$. Le domaine extérieur est noté Ω^+. Nous allons considérer le cas d'une condition à la frontière de type Dirichlet. Ce cas test a l'avantage de présenter une solution analytique simple. Le champ diffracté est donné par :

$$u^+(r,\theta) = -\sum_{n\in\mathbb{Z}} (-i)^n \frac{J_n(ka)}{H_n^{(1)}(ka)} H_n^{(1)}(kr) e^{in\theta}, \quad r \geq a. \tag{1}$$

Les J_n sont les fonctions de Bessel de première espèce et les $H_n^{(1)}$ les fonctions de Hankel du premier type.

Grandes étapes de la BEM :

1. Reformulation de l'EDP sous forme intégrale ;
2. Résolution d'une équation intégrale pour obtenir les traces des champs sur la frontière ;
3. Application de la représentation intégrale pour obtenir le champ dans tout le domaine (non-borné pour les problèmes extérieurs).

Le TP1 s'est intéressé à l'étape 3, le TP2 à l'étape 2. Dans ce troisième TP nous allons accélérer l'étape 2.

Rappel : Les trois TPs sont de difficulté croissante. Chaque TP fera l'objet d'un compte-rendu noté (**limité à 5 pages**, soyez concis).

TP3 : Méthode des matrices hiérarchiques pour accélérer la BEM

Dans ce TP, nous allons mettre en oeuvre la méthode des matrices hiérarchiques pour accélérer la résolution numérique de l'équation intégrale pour les formulations directes. Lors du TP1, nous avons vu que le champ diffracté dans le cas de conditions à la frontière de type Dirichlet est donné par la représentation intégrale

$$u^{+}(\mathbf{x}) = \int_{\Gamma} G(\mathbf{x},\mathbf{y}) p(\mathbf{y}) d\Gamma(\mathbf{y}) \tag{2}$$

où $p = -\partial_{\mathbf{n}} u^{+} - \partial_{\mathbf{n}} u^{inc}$ avec $\mathbf{n}$ la normale extérieure au disque.
Dans le TP2, p a été déterminé numériquement en résolvant l'équation intégrale

$$\begin{aligned} &\text{Trouver } p \in H^{-1/2}(\Gamma) \text{ tel que} \\ &\int_{\Gamma} G(\mathbf{x},\mathbf{y}) p(\mathbf{y}) d\Gamma(\mathbf{y}) = -u^{\text{inc}} \end{aligned} \tag{3}$$

Le but de ce TP3 est d'accélérer la solution de (3) tout en vérifiant quelle est l'erreur que l'on commet par rapport à la solution analytique.

Principales étapes du travail (à détailler dans votre compte-rendu)

1. Coder la méthode ACA avec pivotage complet puis avec pivotage partiel. Comparer les temps de calcul, les complexités et les rangs obtenus (pour des matrices dont vous connaissez le rang exact ou le rang numérique).

2. On cherche maintenant à réduire les coûts de stockage de $\mathbb{A}$ et d'accélérer la résolution du système
$$\mathbb{A}\mathbf{p} = \mathbf{b}.$$
On fera cette résolution avec un solveur itératif du type GMRES de sorte que la seule opération importante est de savoir faire un produit matrice-vecteur. Quel est le rang de la matrice $\mathbb{A}$ complète ?

3. Construire l'arbre binaire pour partitionner la géométrie (en utilisant un critère géométrique).

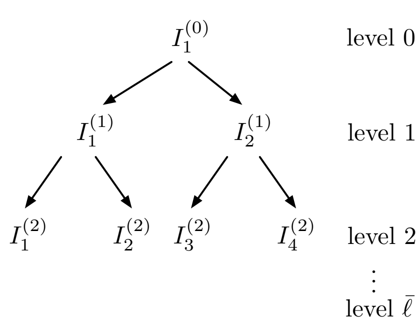
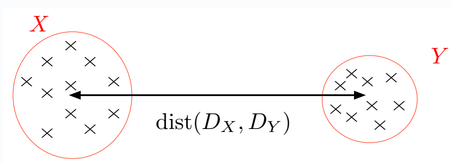


Figure 1 – Partionnement des degrés de liberté avec un arbre binaire.

Le paramètre N_{leaf} permet de déterminer le nombre de niveaux en arrêtant le processus de subdivision quand le nombre de degrés de liberté est inférieur à N_{leaf}. Attention à bien vérifier que les noeuds physiquement consécutifs sont bien numérotés avec des nombres eux aussi consécutifs (sinon il faudra les réordonner au cours de la construction de l'arbre).

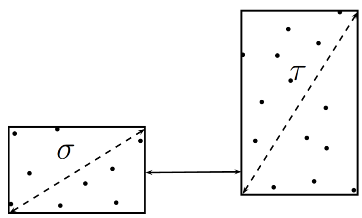
4. Il faut maintenant effectuer le découpage récursif de la matrice en utilisant le critère d'admissibilité

X et Y sont η-*admissibles* ($\eta > 0$) si



$$\min\Big(\text{diam}\, X, \text{diam}\, Y\Big) < \eta \times \text{dist}(X, Y)$$

Commencer par calculer les diamètres de toutes les boîtes :



Calculer ensuite les distances quand c'est nécessaire. Pour que ce calcul soit rapide on utilise le diamètre de la boîte englobante

$$\text{diam}(X) \leq \Big(\sum_{\alpha=1}^{2} (\max_{\mathbf{x}\in\mathbf{X}} \mathbf{x}_\alpha - \min_{\mathbf{x}\in\mathbf{X}} \mathbf{x}_\alpha)^{\mathbf{2}}\Big)^{\mathbf{1/2}} =: \text{diam}_{\text{box}}(\mathbf{X})$$

qui est une borne supérieure. De la même manière la distance entre 2 ensembles est replacée par la distance entre les faces les plus proches des boîtes englobantes

$$\text{dist}(X, Y) \geq \Big(\sum_{\alpha=1}^{2} (\max(0, \min_{\mathbf{x}\in\mathbf{X}} \mathbf{x}_\alpha - \max_{\mathbf{y}\in\mathbf{Y}} \mathbf{y}_\alpha))^{\mathbf{2}} + (\max(\mathbf{0}, \min_{\mathbf{y}\in\mathbf{Y}} \mathbf{y}_\alpha - \max_{\mathbf{x}\in\mathbf{X}} \mathbf{x}_\alpha))^{\mathbf{2}}\Big)^{\mathbf{1/2}} =: \text{dist}_{\text{box}}(\mathbf{X}, \mathbf{Y})$$

Si le critère est satisfait, garder en mémoire que ces deux boites sont admissibles. Si le critère n'est pas satisfait, il faut subdiviser la matrice en 4 sous-matrices et continuer.

Comme le critère d'admissibilité ne dépend que de la géométrie, regarder quelle est la structure de la matrice hiérarchique.

Vérifier en utilisant la fonction `rank` de Matlab que les blocs admissibles sont bien de rang faible. Vous pouvez (et c'est plus que recommandé) faire une illustration visuelle des rangs en utilisant les fonctions Matlab `patch` et `text`. A défaut faites un dessin manuel ! Quel est l'effet du paramètre η (en pratique on prend souvent $\eta = 3$) ?

5. Pour voir le gain mémoire, vous allez utiliser la matrice construite lors du TP2 et en construire sa représentation sous forme de matrice hiérarchique en utilisant la méthode ACA avec pivotage complet. Quel est le gain obtenu ? Calculer le taux de compression

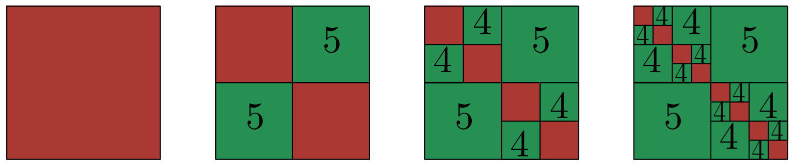


FIGURE 2 – Illustration de la structure de la matrice dans le cas d'un segment 1D. Les blocs verts sont de rang faible et les rouges de rang plein.

τ qui correspond au ratio entre le coût de stockage sous forme hiérarchique et le coût de stockage sous forme classique. Comment évolue ce taux de compression si la fréquence augmente, à nombre de degrés de liberté fixé ? à fréquence fixée et nombre de degrés de liberté qui augmente ? à densité de point fixée ? Est-ce que c'est normal ?

6. Coder le produit matrice-vecteur avec une matrice stockée sous forme hiérarchique. Quelle est la complexité de cette opération si la fréquence augmente, à nombre de degrés de liberté fixé ? à fréquence fixée et nombre de degrés de liberté qui augmente ? à densité de point fixée ? Est-ce que c'est normal ?

7. Il ne reste qu'une étape pour avoir un code efficace. Il faut maintenant éviter de construire la matrice complète. Pour cela, dans un premier temps, coder la fonction qui permet de calculer une ligne ou une colonne de la matrice $\mathbb{A}$. Puis construire la matrice directement sous forme hiérarchique en utilisant la méthode ACA avec pivotage partiel. Vous allez maintenant pouvoir traiter des problèmes plus gros que lorsque vous construisiez la matrice complète. Quel est le gain en terme de taille de problème accessible ?

8. Pour aller plus loin : si vous avancez vite vous pouvez également regarder comment se comporte le code avec d'autres géométries d'obstacle (cube, ellipse, ...).

BON COURAGE !