

**Projet 2 : Problème stationnaire avec grille non-structurée**

L'objectif de ce projet est d'utiliser un code d'éléments finis  $P^1$  parallélisé, d'en étudier les performances sur le cluster cholesky, et d'analyser des solveurs de systèmes linéaires.

**Description du problème et du code de calcul**• **Problème continu**

Étant donné un domaine  $\Omega = ]0, 1[ \times ]0, 1.5[$ , on cherche la solution  $u(\mathbf{x}) \in H^1(\Omega)$  du système

$$\begin{cases} \alpha u - \Delta u = f, & \forall \mathbf{x} \in \Omega, \\ \partial_n u = 0, & \forall \mathbf{x} \in \partial\Omega, \end{cases}$$

où  $\alpha$  est un scalaire réel constant et  $f(\mathbf{x}) \in L^2(\Omega)$ . Ce problème peut se réécrire

$$u \in H^1(\Omega) : \quad \alpha \int_{\Omega} uv \, d\Omega + \int_{\Omega} \nabla u \cdot \nabla v \, d\Omega = \int_{\Omega} f v \, d\Omega, \quad \forall v \in H^1(\Omega). \quad (1)$$

• **Problème discrétisé**

Pour résoudre ce problème, on utilise un schéma d'éléments finis  $P^1$  basé sur un maillage de triangles  $\mathcal{T}_h$ , dont nœuds sont notés  $(M_i)_{i=1 \dots N_{\text{nœuds}}}$ . Après approximation de la formulation variationnelle et utilisation de fonctions de base nodales, on obtient le système

$$[\alpha \mathbf{M} + \mathbf{K}] \mathbf{u} = [\mathbf{M} \mathbf{f}],$$

où les éléments de la matrice de masse  $\mathbf{M}$ , de la matrice de rigidité  $\mathbf{K}$ , et des vecteurs  $\mathbf{f}$  et  $\mathbf{u}$  sont définis par

$$M_{ij} = \int_{\Omega} w_i w_j \, d\Omega, \quad K_{ij} = \int_{\Omega} \nabla w_i \cdot \nabla w_j \, d\Omega, \quad f_i = f(M_i), \quad u_i = u_h(M_i),$$

pour  $i, j = 1 \dots N_{\text{nœuds}}$ .

La convergence de l'erreur en norme  $\|\cdot\|_{L^2}$  du schéma est d'ordre 2, c'est à dire

$$\mathcal{E} = \|u_h - u_{\text{ref}}\|_{L^2(\Omega)} \leq Ch^2,$$

où  $h$  est le pas du maillage et  $C$  est une constante indépendante de  $h$ . Cette erreur peut être évaluée numériquement en utilisant l'approximation d'une norme  $L^2$ ,

$$\|v\|_{L^2(\Omega)}^2 = \int_{\Omega} |v|^2 \, d\Omega \approx \mathbf{v}^T \mathbf{M} \mathbf{v}.$$

- *Code de calcul parallèle*

On fournit un code de calcul C++ qui résout le problème décrit ci-dessus. Le fichier `main.cpp` de ce code effectue les opérations suivantes :

1. Initialisation de MPI;
2. Lecture du maillage et du partitionnement (fonction `readMsh`) et construction de listes de nœuds pour les communications (fonction `buildListNodesMPI`);
3. Construction de la solution de départ du solveur ( $u_0$ ), de la solution de référence ( $u_{\text{ref}}$ ), du terme source ( $f$ ) et des matrices du système (fonction `buildLinearSystem`);
4. Résolution parallèle du système par la méthode de Jacobi (fonction `jacobi`);
5. Écriture de la solution dans un fichier `gmsh` (fonction `saveToMsh`);
6. Finalisation MPI.

Le code est parallélisé en suivant une stratégie “*par groupes d’éléments*”.

La librairie `eigen` est utilisée pour le stockage des matrices creuses et pour les opérations matricielles (produits matrice-matrice et matrice-vecteur). La compilation du code est réalisée en utilisant le `Makefile` qui se trouve dans le dossier principal. Il suffit alors d’exécuter la commande `make`, qui crée l’exécutable `solver`.

Le calcul du résidu et de l’erreur  $L^2$  de la solution n’est pas encore implémentée.

- *Générer un maillage, gérer le partitionnement, et visualiser la solution*

Les données d’un premier cas test sont contenues dans le fichier `main.cpp` et dans le fichier de géométrie `benchmark/domain.geo`. Le logiciel libre `gmsh`<sup>1</sup> est utilisé pour générer et partitionner le maillage (*avant l’exécution du programme*), et pour visualiser la solution.

- Pour charger une version récente de `gmsh` sur une machine d’une salle de l’ENSTA, vous devez exécuter la commande `useuma gmsh` à l’ouverture du terminal (*la version par défaut est trop ancienne*). Consulter le tutoriel d’utilisation du cluster Cholesky pour voir en particulier les modules à charger avant de compiler votre code.
- Pour générer et partitionner le maillage, on utilise la commande suivante dans le dossier `benchmark/` :

```
>> gmsh -2 -part 4 domain.geo -o domain.msh
```

où 4 est le nombre de sous-domaines souhaité (*à modifier*). La commande génère le fichier de maillage `domain.msh` dans le dossier `benchmark/`. Dans la procédure de résolution, le nombre de sous-domaine est choisi lors de la génération du maillage, alors que le nombre de processus MPI est choisi au moment de l’exécution du programme avec `mpirun`. Par défaut, dans le code, le sous-domaine numéro  $n_{\text{dom}}$  sera pris en charge par le processus de rang  $(n_{\text{dom}} \bmod N_{\text{proc}})$ , où  $N_{\text{proc}}$  est le nombre total de processus MPI.

- Après l’exécution du code de calcul, on utilise la commande `gmsh domain.msh solNum.msh_*` dans le dossier `benchmark/` pour visualiser la solution.

---

<sup>1</sup>Site Internet : <http://gmsh.info/>

## Consignes

1. Ajoutez et parallélisez :
  - le calcul de la norme quadratique du résidu pour le critère d'arrêt de la méthode de Jacobi;
  - le calcul de l'erreur  $L^2$  à la fin de la résolution du système.

Pour ce faire, vous devrez implémenter une fonction pour calculer le produit scalaire de deux vecteurs en parallèle.
2. Validez la version parallèle du code en vérifiant la convergence de l'erreur  $L^2$  pour un problème manufacturé (prenez par exemple  $\alpha = 1$  et  $u(\mathbf{x}) = \cos(\pi x) \cos(2\pi y)$ ). Vérifiez si les erreurs sont les mêmes pour les cas où le code est utilisé en séquentiel ou en parallèle.
3. Implémentez la méthode du gradient conjugué en parallèle. Étudiez la convergence de la méthode, et comparez avec celle de la méthode de Jacobi.

Choisissez une extension ...

- A. [Scalabilité] Étudiez la scalabilité faible et la scalabilité forte du code parallélisé sur le cluster cholecky et comparez les performances des 2 méthodes (Jacobi et Gradient Conjugué) Pour ce faire, prenez des maillages suffisamment grands.
- B. [Analyse numérique] Prolongez la comparaison des méthodes de Jacobi et du Gradient Conjugué en prenant d'autres valeurs de  $\alpha$  ( $-1, 0, \dots$ ) et des problèmes différents.

La version finale de vos **codes** et un **rapport écrit préliminaire** sur les questions 1, 2 et 3 du projet (*4 pages maximum, ne pas inclure les extensions*) doivent être envoyés **le vendredi 12 novembre 2021** (au plus tard) aux deux adresses :

- [nicolas.kielbasiewicz@ensta-paris.fr](mailto:nicolas.kielbasiewicz@ensta-paris.fr) (encadrant du projet)
- [axel.modave@ensta-paris.fr](mailto:axel.modave@ensta-paris.fr) (responsable du cours)

Une **présentation orale** (sur les questions 1, 2 et 3, ainsi que sur l'extension choisie) est prévue pendant la semaine des examens, c'est-à-dire la semaine du 15 au 19 novembre 2021.