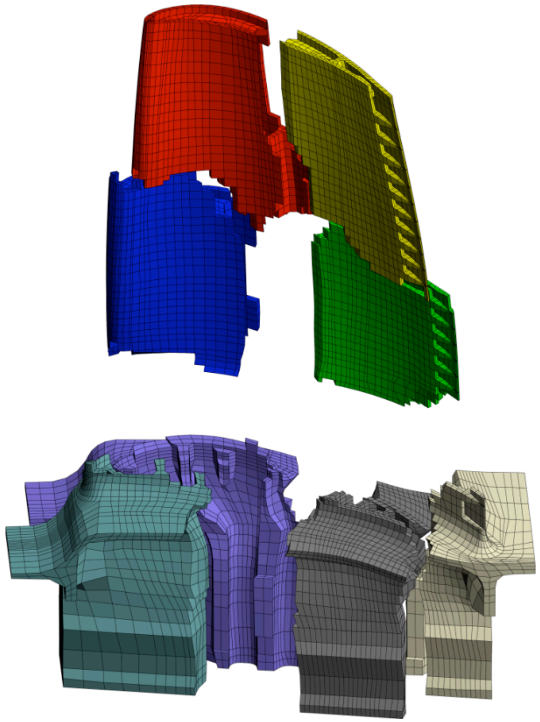
ENSMP 2ème/3ème année, Cours d'éléments finis, 24 au 28 novembre 2008

Introduction aux méthodes de décomposition de domaine

Tous les développements de processeurs récents sont basés sur l'utilisation d'architectures multicoeurs ; 8 coeurs sur le Cell d'IBM dès 2005, 2 coeurs chez intel et AMD en 2006, puis 4 et 6 en 2009... Sans oublier les GPU qui utilisent depuis presque 10 ans des unités de calcul parallélisées. Ce type de conception permet de proposer des systèmes toujours plus puissants tout en limitant la consommation liée aux montées en fréquence, néanmoins elle requiert le développement d'algorithmes spécifiques capables de procéder à d'efficaces parallélisations.

Dans le cadre de la résolution implicite de problèmes régits par des équations aux dérivées partielles (EDP), il existe depuis plus d'une dizaine d'années des stratégies de résolution permettant d'utiliser des architectures massivement parallèles.

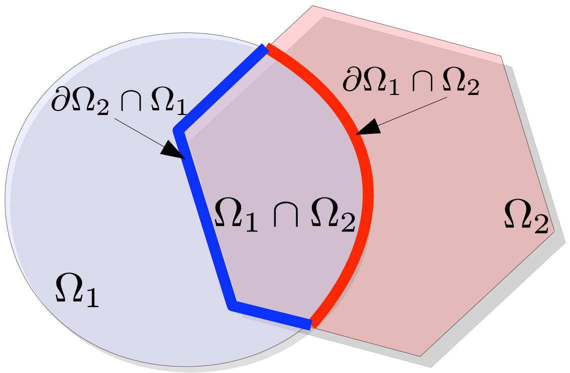
Ce mini-projet constitue une introduction à différentes approches envisagées pour la résolution partitionnée de problèmes de mécanique des milieux continues utilisant une modélisation à base d'éléments finis.



Code utilisé : *ZéBuloN*

Mots-clés : *résolution partitionnée, décomposition de domaine, résolution itérative*

Méthodes de Schwarz



Présentation

Dès 1890, Schwarz propose une méthode permettant la résolution partitionnée d'EDP sur des domaines recouvrants. Dans le cas de la décomposition de Ω en deux domaines recouvrants Ω_1 et Ω_2, la formulation du problème complet conduit à vérifier :

$$\begin{cases} L(u_1) = f_1 \qquad L(u_2) = f_2 \qquad u_1 = u_2 \text{ sur } \Omega_1 \cap \Omega_2 \\ u_1 = u_{1d} \text{ sur } \partial_U \Omega_1,\ u_2 = u_{2d} \text{ sur } \partial_U \Omega_2, \end{cases} \tag{1}$$

La résolution est menée à l'aide du schéma itératif suivant :

1. initialisation : $k = 0$, $u_1^0 = 0$, $u_2^0 = 0$
2. $k = k + 1$
3. initialisation d'itération : $u_2^k = u_2^{k-1}$
4. calculer u_1^k sur le domaine 1 tel que :
   $L(u_1^k) = f_1, \quad u_1^k = u_{1d} \text{ sur } \partial_U \Omega_1, \quad u_1^k = u_2^k \text{ sur } \partial\Omega_1 \cap \Omega_2$
5. calculer u_1^k sur le domaine 2 tel que :
   $L(u_2^k) = f_2, \quad u_2^k = u_{1d} \text{ sur } \partial_U \Omega_1, \quad u_2^k = u_1^k \text{ sur } \partial\Omega_2 \cap \Omega_1$
6. calculer $err = \|u_1^k - u_2^k\|$ sur $\Omega_1 \cap \Omega_2$
7. si $err > \varepsilon$ retour en 2.
8. solution $u_1 = u_1^k$ et $u_2 = u_2^k$

Le critère de convergence porte sur la distance entre les solutions u_1^k et u_2^k sur la zone de recouvrement $\Omega_1 \cap \Omega_2$.

Travail proposé

On se propose de résoudre un problème de mécanique à l'aide de l'approche introduite précédemment. Pour cela nous allons utiliser le code ZéBuloN sur la résolution d'un problème multimatériau (acier/caoutchouc) et mener une démarche de résolution par couplage entre deux processus de résolution éléments finis.

Aller dans le répertoire `Schwarz`. Vous trouverez le fichiers suivants :

- `DD_Schwarz.inp`, mise en donnée de la résolution découplée avec un critère en saut de déplacement sur la zone de recouvrement ;
- `meca_gauche.inp`, mise en donnée du sous-problème représentant la partie gauche de la structure ;
- `Poutre_gauche_rec.geof`, maillage de la partie gauche ;
- `meca_droite.inp`, mise en donnée du sous-problème représentant la partie droite de la structure ;
- `Poutre_droite_rec.geof`, maillage de la partie droite ;
- `../acier.mat`, mise en donnée des paramètres caractérisant le matériau "acier" ;
- `../caoutchouc.mat`, mise en donnée des paramètres caractérisant le matériau "caoutchouc".

En étudiant les fichiers de mise en donnée ".inp" et ".mat", à l'aide de la documentation ZéBuloN et de la visualisation des maillages (avec la commande `Zmaster fichier.geof`), formuler le problème de mécanique des milieux continus dont on cherche la résolution EF.

La commande `Zrun DD_Schwarz.inp` permet d'effectuer la résolution du problème partitionné. Lancer la résolution et observer le résultat obtenu, détailler comment se comporte les solutions gauche et droite si l'on effectue qu'une seule itération (paramètre `**iteration`).

Expliquer pourquoi l'approche proposée n'est pas directement parallélisable (au sens de la résolution simultanée des deux problèmes sur chaque sous-domaine) et suggérer une modification permettant d'effectuer les deux résolutions simultanément.

La mise en donnée permettant une résolution parallèle avec la méthode de Schwarz est réalisée dans le répertoire `Schwarz_para`. Lancer la résolution avec ZéBuloN et comparer les performances avec celle de la méthode habituelle.

Un algorithme sans recouvrement

Dans cette dernière partie, on cherche à utiliser une méthode de résolution itérative partitionnée, sur une structure séparée en deux sous-domaines qui ne se recouvrent pas.

Développement de l'algorithme

On demande d'établir un tel algorithme pour la résolution d'un problème de mécanique.

Quelle sont les conditions à respecter à l'interface entre les deux sous-domaines ?

Déterminer une stratégie itérative permettant la résolution du problème couplé en assurant des communications uniquement sur l'interface entre les deux domaines.

Mise en œuvre dans ZéBuloN

Il est demandé de réaliser la mise en œuvre de la démarche de résolution, développée précédemment, à la résolution du problème précédent en séparant les domaines au niveau de l'interface acier/caoutchouc.

Indiquer si la démarche est parallélisable, et si tel n'est pas le cas comment la transformer pour qu'elle le devienne.

Cette stratégie peut-elle être mise en défaut si l'un des deux domaines ne possède pas suffisamment de conditions de déplacement imposé ?