INTRODUCTION - OPTIMISATION DE FORMES ET TOPOLOGIQUE

BENIAMIN BOGOSEL

2. Mise en pratique - algorithmes d'optimisation de formes

Nous avons vu comment calculer les dérivées de forme associées aux problèmes d'optimisation de formes liées aux EDP. Dans ce qui suit on illustrera les résultats de cours dans des divers cas d'optimisation géométrique et topologique.

L'objectif de la séance pratique est de comprendre le fonctionnement des algorithmes numériques et de pouvoir changer les codes fournis pour construire des nouvels cas d'optimisation.

2.1. **Optimisation géométrique - déplacement de maillage.** Une première manière de faire de l'optimisation géométrique est de considérer un maillage $\mathcal{T}^n$ du domaine Ω^n et de trouver la fonction d'état u_{Ω^n} et l'adjoint p_{Ω^n} en résolvant les EDP correspondantes sur un espace éléments finis sur $\mathcal{T}^n$. Les résultats théoriques permettent ensuite de trouver une formule pour la dérivée de forme comme une intégrale sur $\partial\Omega^n$ et d'identifier une direction de descente θ^n.

Une fois la direction de descente θ^n est identifiée, elle doit être étendue sur tout le domaine Ω^n, pour les raisons suivantes :

- si un déplace uniquement les nœuds du bord $\partial\Omega^n$ il y aura presque sûrement des inversions d'éléments;
- comme dans l'exemple d'optimisation paramétrique présenté la dernière fois, la direction de descente donnée par la théorie est souvent moins régulière si on la considère sur l'espace des éléments finis. Il est donc souhaitable de régulariser la direction de descente en même temps que la procédure d'extension.

Vu que le champ θ^n a deux composantes on peut utiliser une équation d'élasticité pour effectuer la régularisation et l'extension.

Une fois la direction θ^n régularisée et étendue sur tout le domaine Ω^n (représenté par le maillage $\mathcal{T}^n$), il est possible d'utiliser la commande `movemesh` de `FreeFem++` pour déplacer le maillage $\mathcal{T}^n$ et de définir le nouvel domaine comme le maillage déplacé avec le champ $\tau^n\theta^n$, avec τ^n un pas de descente assez petit. Avant d'effectuer la commande `movemesh` il est possible de vérifier si le nouvel maillage ne va pas contenir des éléments inverses: la commande `checkmovemesh` permet de faire cette vérification. Si la vérification ne donne pas le résultat attendu, on diminue le pas de descente τ^n.

Souvent le maillage déplacé n'a pas les mêmes qualités que le maillage initial. On peut utiliser la commande adaptmesh pour récupérer un maillage de bonne qualité.

Ce procédé fonctionne bien en dimension deux, car les routines d'adaptation de maillage sont rapides et robustes dans cette situation. En dimension trois la méthode devient beaucoup plus pénible et lente.

Question 1. *Créez votre propre exemple d'optimisation géométrique en changeant la géométrie du maillage initial et en considérant des chargements différents.*

2.2. **Optimisation topologique - fonctions "level-set".** Une autre manière de faire de l'optimisation géométrique est de représenter les ensembles Ω^n comme des niveaux $\{\phi^n < 0\}$ où les fonctions ϕ^n ont un domaine commun de définition et prennent des valeurs réelles. Ceci revient à considérer un domaine D fixe qui englobe toutes les formes intermédiaires Ω^n et de considérer un maillage de calcul fixe. On peut observer que déjà, le fait de garder le maillage fixe, évite tout le coût de calcul supplémentaire pour recalculer les maillages après chaque itération (coût non-négligeable en dimension trois). Du point de vue géométrique, une fonction level-set permet de récupérer assez facilement les éléments géométriques, pourvu que la fonction level-set ait une forme particulière.

La fonction level-set qui se comporte bien est la distance signée au $\partial\Omega$:

$$d_\Omega(x) = \begin{cases} -d(x, \partial\Omega) & \text{si } x \in \Omega \\ \ \ d(x, \partial\Omega) & \text{si } x \notin \Omega \end{cases}$$

Heureusement qu'on peut trouver des algorithmes standard de transformation d'une fonction level-set en une distance signée : par exemple la fonction `distance` en `FreeFem++` . Une fois que la fonction level-set ϕ^n est une distance signée, les quantités géométriques peuvent êtres récupérées assez facilement, mais il faut faire attention toujours à l'aspect régularisation, si ϕ^n est représentée comme une fonction éléments finis.

Le déplacement de $\partial\Omega^n$ dans la direction de descente θ^n identifiée à l'aide de la dérivée de forme se fait en résolvant une équation d'advection. En `FreeFem++` ceci est possible, pour les cas simples considérées dans ce cours, en utilisant la commande `convect`.

Des solutions plus poussées peuvent être utilisées quand les fonctions `FreeFem++` ne fonctionnent pas. Par exemples, les routines `MshDist` et `Advect` du toolbox ISCD :
`https://github.com/ISCDtoolbox`.

Les solutions obtenues avec des techniques d'optimisation topologique avec la méthode des lignes de niveaux dépendent fortement de la condition initiale et de la structure du maillage. La méthode des lignes de niveaux, contrairement aux méthodes de mouvement de maillage, permet de changer la topologie des solutions :

- renfermer des trous en dimension 2. La méthode level-set ne permet pas la création naturelle des trous en dimension deux. Pour résoudre cet aspect, la notion de *dérivée topologique* a été introduite. Cette notion n'a pas été traitée dans le cours, mais vous avez des codes qui montre la méthode en pratique.
- renfermer des trous et en créer d'autres en dimension supérieure. En dimension trois, même si la structure initiale ne contient aucun trou, la structure peut tourner autour d'elle même et créer des structures topologiques plus complexes.

Question 2. *Utilisez le code fourni pour créer votre propre exemple d'optimisation topologique en utilisant la méthode level-set. Modifiez les conditions initiales et observez comment la solution change.*

2.3. **Un exemple en dimension trois.** Pour illustrer les difficultés supplémentaires concernant les calculs en dimension trois, un exemple est fourni. Voici quelques différences par rapport à la dimension deux :

- la partie maillage n'est pas immédiate :
  (i) construire un maillage est plus coûteux
  (ii) les maillages de qualité peuvent contenir trop d'éléments
  (iii) il n'est pas si facile de faire une adaptation de maillage qu'en 2D
  (iv) pour avoir des maillages qui se comportent bien, il ne faudrait pas avoir des différences importantes en taille de mailles
- la résolution des EDP est plus coûteuse en 3D ce qui rend les algorithmes d'optimisation plus lents. Le calcul parallèle peut aider, mais nécessite des efforts supplémentaires de codage.

  Notez que si des solveurs directs sont employés il est possible d'avoir des problèmes de mémoire. Dans ce cas, il faut ajouter l'option `solver=CG` quand on définit les formulation variationnelles. Ce solveur est itératif et il est plus efficace en termes de mémoire RAM.

- la visualisation des solutions est aussi un defi, car il faut reconstruire la surface $\{\phi^n = 0\}$ à partir de la fonction level-set. Des solutions freeware existent : `xd3d` et ParaView.

Question 3. *Créer votre propre exemple d'optimisation en dimension trois en modifiant les régions de chargement et les conditions initiales.*

2.4. **Questions pour le prochain cours.**

(1) Plusieurs parmi vous êtes impliqués dans un projet collectif à l'ENSTA concernant la construction d'une voiture solaire. Un enjeu important est d'améliorer l'aero-dynamique de la voiture. Avant de chercher comment utiliser des algorithmes d'optimisation pour atteindre ce objectif, plusieurs aspects doivent êtres traités :
   (i) Quels paramètres peuvent êtres envisagés pour décrire la géométrie de la voiture ? Quelle sont les contraintes éventuelles (position et taille boite moteur, etc...) et quelles sont les régions optimisables ?
   (ii) Pouvez-vous identifier quelques régions locales de la voiture qui pourraient êtres traitées de manière indépendante ? (pour réduire le coût de calcul)
   (iii) Y a-t-il manière d'étudier certaines régions en 2D ?
   (iv) Pour identifier le régime dans lequel il faudrait faire les simulations, documentez-vous sur le nombre de Reynolds et identifiez quel est le nombre de Reynolds correspondant aux diverses simulations possibles.

(2) Cherchez un logiciel disponible à l'ENSTA capable de faire de l'optimisation topologique (autre que `FreeFem++`) et construisez un exemple simple d'optimisation topologique (en regardant la documentation).

(3) Toujours avec un logiciel disponible à l'ENSTA, effectuez des simulations du mouvement d'air autour d'un objet 3D. Cherchez des informations concernant le calcul de la *trainee* et d'autres quantités importantes en aero-dynamique.

Le prochain cours va porter sur des aspects liés à la mécanique des fluides. Nous allons regarder plusieurs aspects concernant l'optimisation de formes sous divers modèles présentés dans la référence [Mohammadi,Pironneau] que vous pouvez trouver dans la liste de références fournie.