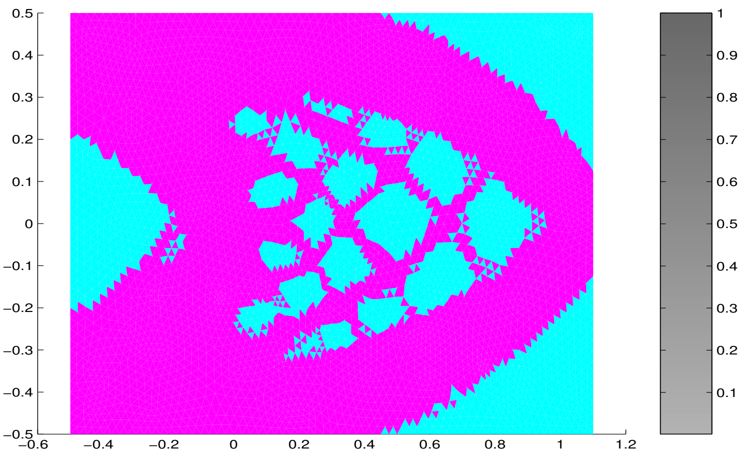
TP CONCEPTION OPTIMALE DE FORME

PHILIPPE GUILLAUME, MOHAMED MASMOUDI

DEA de Mathématiques Appliquées
Ecole Doctorale de Mathématiques de Toulouse
INSA - SUPAERO - UPS - UT1



1. Définition du TP

Le but de ce TP est d'optimiser la forme de quelques structures 2D soumises à un chargement. Le critère à optimiser est l'énergie de déformation. Pour cela, partant d'un rectangle plein, on utilise le gradient topologique pour retirer de la matière progressivement jusqu'à une proportion fixée par l'utilisateur. Nous étudierons trois cas :

1. le cas d'une poutre de longueur 0,4 et de hauteur 1, avec condition de Dirichlet nulle sur le bord gauche. Une force dirigée vers le bas est appliquée au milieu du côté opposé. La solution de ce problème est connue : c'est une équerre,
2. le cas d'une poutre de longueur 1,6 et de hauteur 1, avec les mêmes conditions aux limites. Les solutions de ce problème sont connues : ce sont les treillis de Michell,
3. le cas d'un pont de longueur 2 et de hauteur 1, avec condition de Dirichlet sur le coin inférieur droit, et condition de glissement horizontal sur le coin inférieur gauche. Une force dirigée vers le bas est appliquée au milieu du coté inférieur. Une solution connue est en forme de demie roue avec rayons.

2. Déroulement du TP

Ce TP est effectué sous MATLAB avec utilisation de la PDE TOOLBOX. Les différents fichiers dont vous aurez besoin sont dans le répertoire `/echange/tpcof`. Les fichiers `*.m` sont des fichiers source MATLAB, les fichiers `*.mat` sont des fichiers de données binaires.

Pour chacun des trois cas étudiés, vous aurez à :

- construire le maillage du rectangle via l'interface de la PDE TOOLBOX. Pour cela, lancez la commande MATLAB, puis exécutez l'un des trois fichiers `rect0, poutre0` ou `pont0`. L'interface de la PDE TOOLBOX apparaît. La géométrie, les propriétés de matériau, les coefficients de l'EDP et les conditions aux limites sont prédéfinies. Choisissez alors un pas de maillage dans le menu MESH/PARAMETERS, et lancez le mailleur avec le menu MESH/INITIALZE MESH suivi de quelques MESH/JIGGLE MESH pour améliorer la qualité du maillage,
- exporter les données dans l'environnement MATLAB. Pour cela, choisissez la commande EXPORT dans chacun des menus DRAW, BOUNDARY, PDE, MESH. En utilisant la commande `whos` dans la fenêtre MATLAB, vous devez voir apparaître les noms (par défaut) des variables *a, b, c, d, e, f, g, gd, ns, p, sf, t*. Vous pouvez les sauvegarder avec la commande `save fichier`. Pour les récupérer ultérieurement, vous utiliserez la commande `load fichier`.

Vous exécuterez ensuite votre programme d'optimisation. Un exemple de programme (`main0.m`) se trouve dans le répertoire `/echange/tpcof`. A ne consulter qu'en cas d'extrême nécessité !

3. Description du problème

3.1. Formulation continue.

3.1. Formulation continue. Soit Ω un domaine borné de $\mathbb{R}^2$. Sa frontière Γ est composée de deux parties disjointes Γ_1 et Γ_2. Le champ u des déplacements est solution des équations de l'élasticité linéaire

$$\left\{ \begin{array}{c} -\mathrm{div}\,\sigma(u) = 0 \quad \text{dans } \Omega, \\ u = 0 \quad \text{sur } \Gamma_1, \\ \sigma(u)n = g \quad \text{sur } \Gamma_2, \end{array} \right.$$

avec les notations suivantes :

$$\begin{aligned}\varepsilon(u) &= \frac{1}{2}(Du + Du^T),\\ \sigma(u) &= hH_0\varepsilon(u),\end{aligned}$$

où H_0 est le tenseur de Hooke du matériau, constant sur le domaine, et $h(x) = 1$ ou ϵ, la variable de programmation ϵ étant de l'ordre de 10^{-3}.

Le problème d'optimisation de forme est le suivant :

$$\left\{\begin{array}{l}\min\ J(h)\\ \int_\Omega h(x)\,dx = C,\end{array}\right.$$

$$J(h) = \int_\Omega \sigma(u) : \varepsilon(u)\,dx = \int_{\Gamma_2} g.u\,dx,$$

où C est la proportion de matériau à conserver, fixée par vous.

Le gradient topologique est approché en chaque point par

$$J'_T(x) = 2\,\sigma(u) : \varepsilon(u).$$

Une itération de l'algorithme d'optimisation consiste à retirer une partie P du domaine telle que

$$J'_T(x) \leq J'_T(y), \quad \forall (x,y) \in P \times \Omega\backslash P.$$

La proportion $\mathrm{mes}(P)/\mathrm{mes}(\Omega)$ est une variable de programmation à fixer entre 0 et 0.3. On réinitialise alors Ω par $\Omega \leftarrow \Omega\backslash P$ et h par $h(x) = \epsilon$ si $x \in P$, $h(x) = 1$ sinon. L'algorithme s'arrête lorsque la proportion voulue de matériau est atteinte, la solution obtenue étant le domaine $\Omega\backslash P$.

3.2. **Formulation discrète.** On utilise une méthode P_1 sur un maillage non structuré. Le déplacement discret u est continu et affine par morceaux, et on note U le vecteur des coordonnées de u dans la base des éléments finis.

La matrice de rigidité est notée K, elle dépend de H_0 (constant sur tout le domaine) et de h (constant sur chaque élément). Le chargement g définit le second membre G, et le vecteur $U = U(h)$ est solution du système

$$KU = G.$$

Le problème d'optimisation discret devient

$$\left\{\begin{array}{l}\min\ G.U\\ \int_\Omega h(x)\,dx = C.\end{array}\right.$$

3.3. **Quelques commandes utiles de MATLAB.**

- `help, help help, help topic,...` : aide en ligne,
- `help sparfun` : aide sur les matrices creuses,
- `whos` : affiche vos variables courantes,
- `a.*b` : multiplication terme à terme des vecteurs ou des matrices a et b,
- `M'` : transposée (conjuguée) de la matrice M,
- `[nl,nc] = size(M)` : nombre de lignes et de colonnes de la matrice M,
- `programme` : si vous avez créé un fichier nommé `programme.m` et situé dans votre répertoire courant, la commande `programme` exécute les lignes de votre programme,

- `[x,y,...] = fonction(a,b,...)` : si vous avez créé dans votre répertoire courant un fichier nommé `fonction.m` ayant pour entête
  `function [x,y,...] = fonction(a,b,...)`
  la commande `[x,y,...] = fonction(a,b,...)` vous renvoie les valeurs `[x,y,...]` calculées par votre fonction à partir des valeurs d'entrée `(a,b,...)`. Les variables `x,y,a,b,...` peuvent être de tailles différentes,
- `[y,I] = sort(x)` : ordonne par ordre croissant les éléments du vecteur `x` dans le vecteur `y`. Les indices `I` sont tels que `y = x(I)`.
- `disp(x)` : affiche la valeur de la variable x,
- `disp('chaîne')` : affiche la chaîne de caractères `chaîne`,
- `pdeplot(p,e,t,'xydata',ep,'xystyle','flat','mesh','on')` : visualisation sur le maillage des données contenues dans le vecteur `ep` (cf. `help pdeplot`),
- `pause(1)` : arrêt buffet,
- les boucles :

```
for i = 1:10
  instructions;
end
while val >= 0
  instructions;
end
```

3.4. Concernant la résolution de l'edp.

- Sous MATLAB, le tenseur $\sigma(u)$ s'écrit $c \otimes \nabla u$. La variable c que vous avez exportée au départ est une chaîne de caractères, que vous transformerez en une matrice de dimension 10 $\times$ `nbt` (`nbt` = nombre de triangles) par :

```
c0 = zeros(1,10);
for i = 1:10
  c0(i) = eval(c(i,:));
end
c = diag(c0)*ones(10,nbt);
```

  Chaque colonne de c contient alors les informations relatives au tenseur de Hooke H_0 sur l'élément correspondant, à multiplier ensuite par $h(x)$.
- `[K,M,F,Q,G,H,R] = assempde(b,p,e,t,c,a,f)` : calcul de la matrice de rigidité K et du second membre G.
- `U = assempde(K,M,F,Q,G,H,R)` : calcul de la solution.
- `[ux,uy] = pdegrad(p,t,U)` : sur le triangle numéro j, on a

$$Du(x) = \begin{bmatrix} \mathrm{ux}(1,j) & \mathrm{uy}(1,j) \\ \mathrm{ux}(2,j) & \mathrm{uy}(2,j) \end{bmatrix},$$

- `[cgxu,cgyu] = pdecgrad(p,t,c,U)` : sur le triangle numéro j, on a

$$\sigma(u)(x) = \begin{bmatrix} \mathrm{cgxu}(1,j) & \mathrm{cgyu}(1,j) \\ \mathrm{cgxu}(2,j) & \mathrm{cgyu}(2,j) \end{bmatrix}.$$

  Et maintenant, à vous de jouer ! Vous rendrez un rapport de 4 ou 5 pages au plus avec le résultat de vos expérimentations.