Optimisation topologique avec Matlab
Ce n'est pas noté ! faire le travail en 30'
J. Morlier*[1]

Le maillage EF est composé de quadrangle de taille 1*1. La densité appelée x, est définie tq $0.001 \leq x \leq 1$. (La borne inf est non nulle, pour que K ne soit pas singulière). La correspondance entre les éléments de la matrice des densités et le modèle EF est donnée sur la figure 1.1.

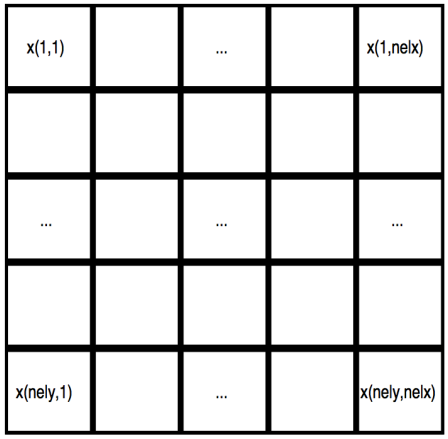

Figure 1: Index de la matrice densité

Nous allons dans cet exercice essayer (en lisant l'article associée au code top.m) d'optimiser la conception d'un cadre de bicyclette.

Le domaine de conception est donné ci-dessous (figure 3)

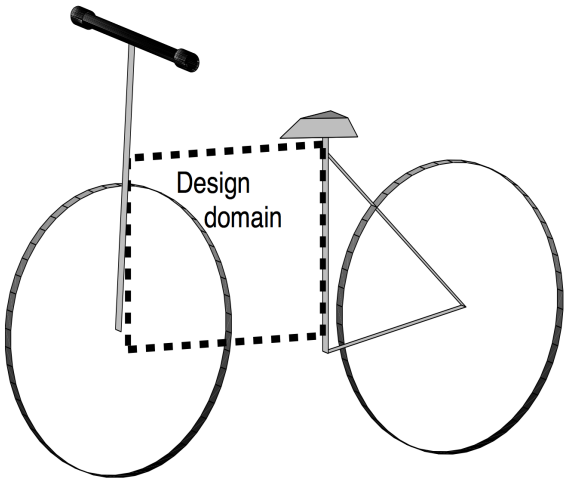

Figure 2: domaine de conception

[1] *d'apres le site:
http://www.topopt.dtu.dk

On schématise le problème de conception optimale sous la forme de la figure 4

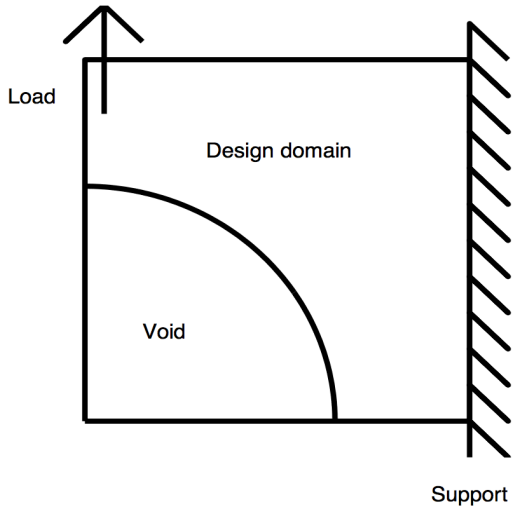

Figure 3: Schéma simplifié

La réaction de la roue produit une force verticale à l'avant, l'espace de la roue est modélisé par le vide (void), enfin la barre sous la selle est encastrée.

>> top(nelx,nely,volfrac,penal,rmin)
where the variables denote the following:
- nelx is the number of finite elements in the horisontal direction.
- nely is the number of finite elements in the vertical direction.
- volfrac is the fraction of volume in the design domain.
- penal is the penalization of intermediate densities. A high penalization will make the solution black and white, that is the finite elements will either be filled or empty. Usually penal = 3. A penalization penal = 1 means that there is no penalty of the intermediate densities.
- rmin is a filter radius for a filter which makes the design mesh-independent.

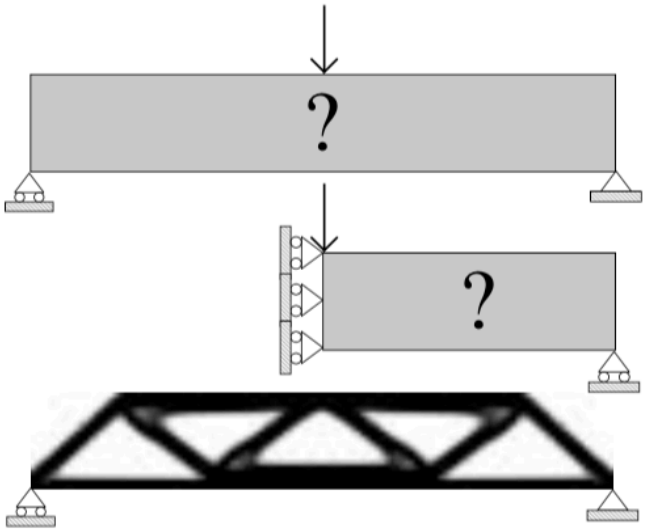

Fig. 1 Topology optimization of the MBB-beam. Top: full design domain, middle: half design domain with symmetry boundary conditions and bottom: resulting topology optimized beam (both halves)

Question 0:

lancer le code :

```
>> top(20,20,0.33,3.0,1.5);
```

le résultat par défaut est pour la poutre MMB

Nous allons devoir changer les CLs/Chargements…pour répondre au problème de la bicyclette.

Question 1 : Sauvegarder le fichier top.m en topB.m
Modifier les conditions aux limites et le chargement

```
78 F(2,1) = 1; % commentez SVP
 79 fixeddofs = 2*nelx*(nely+1)+1:2*(nelx+1)*(nely+1); % commentez SVP
```

Sauvegarder et lancer le code :
>> topB(20,20,0.33,3.0,1.5);

…

copier coller le résultat ici. Commentez

…

Question 2 :
Est ce que le design est dépendant du maillage? Comparer les résultats:

```
>> top(12,12,0.33,3.0,0.9);
>> top(16,16,0.33,3.0,1.2);
>> top(20,20,0.33,3.0,1.5);
```

Notez que le rayon du filtre doit être augmenté de manière proportionnelle à la discrétisation (en x, et en y)

Question 3 :
Comparer les résultats des différentes commandes:
>> top(20,20,0.33,1.0,1.5);
>> top(20,20,0.33,3.0,1.5);

Influence de penal? Vérifier les souplesses associées.

Question 4 :
Analyse du filtre:
Vous pouvez désactiver le filtre en fixant rmin à moins de 1 ou en rendant la ligne 27 inactive:
27 % [dc] = check(nelx,nely,rmin,x,dc);
N'oubliez pas d'enlever le % si vous voulez réutiliser le filtre.

>> top(20,20,0.33,3.0,1.5);
Essayez différentes valeurs de rmin.

Question 5 :
Définition des régions vides: sauvegarder le fichier sous topB2.m
Les résultats précédents ne prennent pas en compte la région vide (roue).

On va devoir classifier les éléments (1 passifs, 0 libres). Ajouter entre les lignes 6 and 7 pour créer le vide. (Passif= pas pris en compte dans l'optimisation comme variable de conception).

```
for ely = 1:nely
  for elx = 1:nelx
   if ((elx)^2+(ely-nely)^2) < (0.65*nelx)^2 passive(ely,elx) = 1; % commentez SVP
   else
passive(ely,elx) = 0; % commentez SVP
   end
  end
end
x(find(passive))=0.001;
```

La dernière commande initialise la région vide à 0.001.
Nous devons aussi mettre à jour la ligne 29 and 39 et ajouter la ligne additionnelle entre 43 et 44:

```
29 [x] = OC(nelx,nely,x,volfrac,dc,passive);
39 function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
43 xnew(find(passive)) = 0.001;
```

Effectuer la commande

```
>> topB2(20,20,0.33,3,1.5)
```

Que constatez vous ?

…

copier coller le résultat ici. Commentez

…

Informations supplémentaires

Les déplacements sont énumérés par colonne de la gauche vers la droite. Le déplacement (variable u) d'un EF quelconque est donné ci dessous.

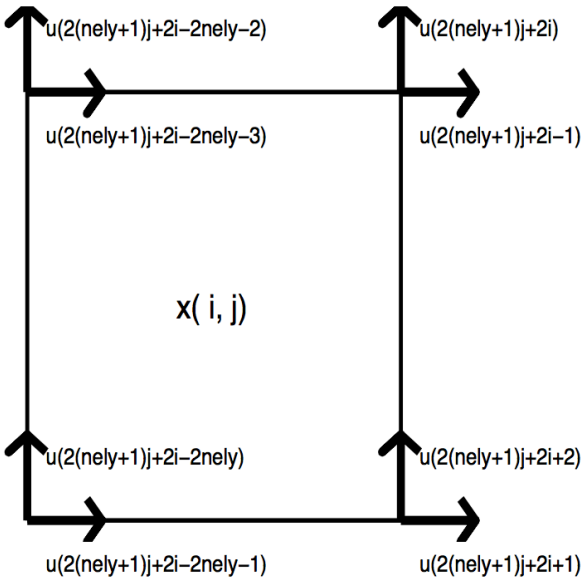

Figure 4: Le déplacement d'un EF (i,j)

Pour avoir les deux outputs graphiques de convergence et sauvegarde des images à chaque itération rajouter apres ligne 85

```
%% PRINT RESULTS

figure(2)
hold on
plot(loop,c,'bo','MarkerFaceColor','b')
plot(loop,mean(xPhys(:))*100,'ro','MarkerFaceColor','r')
% plot(outeriter,(1+GKSl)*VMl,'ko','MarkerFaceColor','k')
title(['Convergence volfrac = ',num2str(mean(xPhys(:))*100),', Compliance
=',num2str(c),', iter = ', num2str(loop)])
grid on
legend('Compliance','Volume Fraction %')
xlabel('iter')
  %% PLOT DENSITIES
   figure(1)
  colormap(gray); imagesc(1-xPhys); caxis([0 1]); axis equal; axis off;
drawnow;
  print(['DZ_it',num2str(loop,'%3d')],'-dpng')
end
```

Plotting displacements

Insert the following lines in your program instead of the current plotting line:

```
% colormap(gray); imagesc(-x); axis equal; axis tight; axis off;pause(1e-6);
colormap(gray); axis equal;
for ely = 1:nely
  for elx = 1:nelx
    n1 = (nely+1)*(elx-1)+ely;
    n2 = (nely+1)* elx +ely;
    Ue = 0.005*U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2; 2*n1+1;2*n1+2],1);
    ly = ely-1; lx = elx-1;
    xx = [Ue(1,1)+lx Ue(3,1)+lx+1 Ue(5,1)+lx+1 Ue(7,1)+lx ]';
    yy = [-Ue(2,1)-ly -Ue(4,1)-ly -Ue(6,1)-ly-1 -Ue(8,1)-ly-1]';
    patch(xx,yy,-x(ely,elx))
  end
end
drawnow; clf;
```

Efficient matrix assembly for large problems

To handle large problems, you may substitute the transparent and easy-to-read stiffness matrix assembly:

```
K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
F = sparse(2*(nely+1)*(nelx+1),1); U = zeros(2*(nely+1)*(nelx+1),1);
for elx = 1:nelx
  for ely = 1:nely
    n1 = (nely+1)*(elx-1)+ely;
    n2 = (nely+1)* elx +ely;
    edof = [2*n1-1; 2*n1; 2*n2-1; 2*n2; 2*n2+1; 2*n2+2; 2*n1+1; 2*n1+2];
    K(edof,edof) = K(edof,edof) + x(ely,elx)^penal*KE;
  end
end
```

with the much more efficient assembly using a list of triplets:

```
I = zeros(nelx*nely*64,1); J = zeros(nelx*nely*64,1); X = zeros(nelx*nely*64,1);
```

```
F = sparse(2*(nely+1)*(nelx+1),1); U = zeros(2*(nely+1)*(nelx+1),1);
ntriplets = 0;
for elx = 1:nelx
   for ely = 1:nely
     n1 = (nely+1)*(elx-1)+ely;
     n2 = (nely+1)* elx +ely;
     edof = [2*n1-1 2*n1 2*n2-1 2*n2 2*n2+1 2*n2+2 2*n1+1 2*n1+2];
     xval = x(ely,elx)^penal;
     for krow = 1:8
       for kcol = 1:8
         ntriplets = ntriplets+1;
         I(ntriplets) = edof(krow);
         J(ntriplets) = edof(kcol);
         X(ntriplets) = xval*KE(krow,kcol);
       end
     end
   end
end
K = sparse(I,J,X,2*(nelx+1)*(nely+1),2*(nelx+1)*(nely+1));
```

Link to more details on creating sparse FE matrices.

The element mass matrix

```
m0 = [4/9 0 2/9 0 1/9 0 2/9 0
   0 4/9 0 2/9 0 1/9 0 2/9
   2/9 0 4/9 0 2/9 0 1/9 0
   0 2/9 0 4/9 0 2/9 0 1/9
   1/9 0 2/9 0 4/9 0 2/9 0
   0 1/9 0 2/9 0 4/9 0 2/9
   2/9 0 1/9 0 2/9 0 4/9 0
   0 2/9 0 1/9 0 2/9 0 4/9]/4/(nelx*nely);
```

The strain displacement matrix

```
bmat = [-1/2 0 1/2 0 1/2 0 -1/2 0
   0 -1/2 0 -1/2 0 1/2 0 1/2
   -1/2 -1/2 -1/2 1/2 1/2 1/2 1/2 -1/2];
```

The constitutive matrix for plane stress

```
Emat = E/(1-nu^2)*[ 1 nu 0
  nu 1 0
  0 0 (1-nu)/2];
```

Short biblio

Bendsøe, M. P. and Sigmund, O.: 2004, Topology Optimization - Theory, Methods and Applications, Springer Verlag, Berlin Heidelberg.

Jensen, J. S.: 2009, A note on sensitivity analysis of linear dynamic systems with harmonic ex- citation, Report, Department of Mechanical Engineering, Technical University of Denmark.

Sigmund, O.: 1997, On the design of compliant mechanisms using topology optimization, Mechanics of Structures and Machines 25(4), 493□524.

Sigmund, O.: 2001, A 99 line topology optimization code written in MATLAB, Structural and Multidisciplinary Optimization 21, 120□127. MATLAB code available online at: www.topopt.dtu.dk.

Sigmund, O.: 2007, Morphology-based black and white filters for topology optimization, Structural and Multidisciplinary Optimization 33(4-5), 401□424.

Svanberg, K.: 1987, The Method of Moving Asymptotes - A new method for structural opti- mization, International Journal for Numerical Methods in Engineering 24, 359□373.

Peut trouvez vous la force et le module d'Young pas réalistes (ainsi que les dimensions, d'où une souplesse pas réaliste…).
Rendez vous par exemple Ligne 87 (E ≈ 2 · 1011N/m2) ... Il est courant de travailler en adimensionnel, d'obtenir le résultat de TopOpt et ensuite de faire l'optimisation paramétrique. Cependant vous auriez pu corriger les unités avec la ligne 39 pour obtenir une solution correcte.
39 while ((l2-l1)/l2 > 1e-4)

Pour obtenir un deplacement cohérent

1) choisi un système cohérent de mesure N, mm,MPa.
2) E=210e3 ;
*3) multiplie la force fois F fois 54*1e3 ;*
4) multiplie la matrice élémentaire fois l'épaisseur en millimètres.
5) le maillage solide 2D est invariant par rapport aux homothéties. Pour respecter les dimensions du problème assures toi que nelx et nely sont en même rapport que les tailles 140 et 50
6) Pour appliquer les conditions limites à 70mm assures toi que nelx soit paire après remplace la condition limite sur le DOF 2(nelx+1)*(nely+1) par une sur le DOF (nelx+2)*(nely+1)*

La compliance est en mJ.