

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.

$x(1,1)$...		$x(1,nelx)$
...	
$x(nely,1)$...		$x(nely,nelx)$

Figure 1: Index de la matrice densité

Nous allons dans cet exercice essayer de voir les différents paramètres de top88, l'effet du maillage, l'effet de la taille du filtre etc...Notez a chaque fois la valeur de la compliance et de la satisfaction des contraintes à convergence.

Le cas test par défaut est le MBB

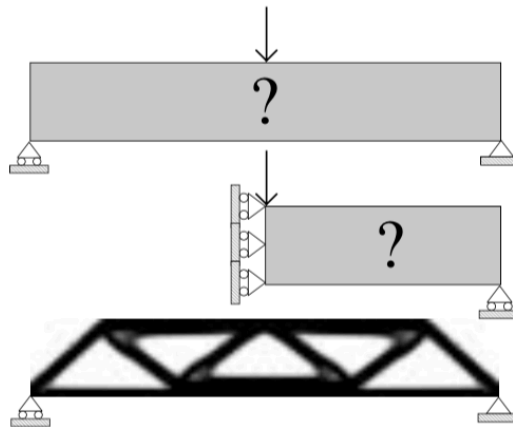


Fig. 1 Topology optimization of the MBB-beam. Top: full design domain, middle: half design domain with symmetry boundary conditions and bottom: resulting topology optimized beam (both halves)

lancer les commandes suivantes
top(40, 20, 0.5, 3, 1.0)
Que constatez vous ?

¹ *d'après le site:
<http://www.topopt.dtu.dk>

top(40, 20, 0.5, 3, 1.5) & top(40, 20, 0.5, 3, 3)
Que constatez vous ?

top(60, 30, 0.5, 3, 1.0)

top(60, 30, 0.5, 3, 2.25)

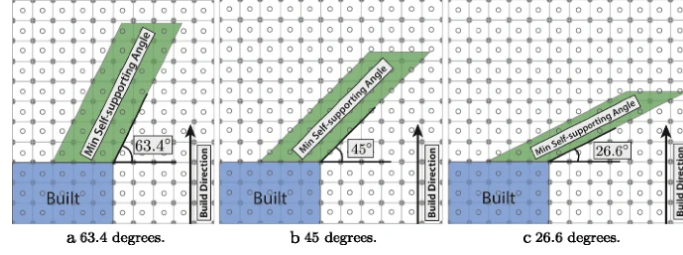
top(40, 20, 0.5, 3, 1.5)

Que constatez vous ?

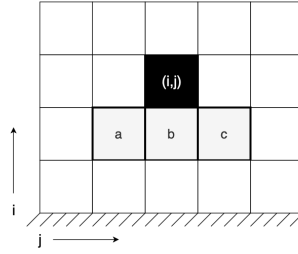
Et la fabrication additive ?

On rajoute la contrainte d'overhang angle but we switch in English...

A limitation of many additive manufacturing methods is the overhang angle: it is impossible to print material at an angle less than about 45° with respect to the building direction. For these methods, standard topology optimization designs are unprintable (Wang et al., 2013).



Examples of different minimum allowable self-supporting angles for satisfying overhang constraints. The blue region is imagined to be built already while the green region indicates the minimum angle at which features may be created without requiring support material. The build direction is assumed upwards and the overhang feature may be leaning to the right (as shown) or to the left



Visualization of overhang: a, b and c support the element above.

Langelaar (2017) departs from the notion that for densities $x_{(i,j)}$ (see Figure 7.1) it should hold that:

$$\begin{aligned} x_{(i,j)} &\leq \max(a, b, c) \\ a &= x_{(i-1,j-1)} \\ b &= x_{(i-1,j)} \\ c &= x_{(i-1,j+1)} \end{aligned} \quad (7.1)$$

The max operator spoils the non-linearity of this constraint that should hold for all elements, except for the first row at the bottom.

A simplistic method is to replace Equation 7.1 by the linear constraint:

$$x_{(i,j)} \leq a + b + c \quad (7.2)$$

If a , b , c and $x_{(i,j)}$ are all binary (0 or 1), then if Equation 7.2 holds then also Equation 7.1 is satisfied. Subsequently, we rely on the penalization in SIMP to push the continuous densities towards zero and one values. The optimization problem becomes

$$\begin{aligned} \text{find} \quad & \mathbf{x} = [x_{11}, x_{12}, x_{21}, x_{22}, \dots, x_{n_y, n_x}]^T \\ \text{minimize} \quad & f_0(\mathbf{x}) = c(\mathbf{x}) \\ \text{subject to} \quad & g_0(\mathbf{x}) = \mathbf{x}^T \mathbf{v} - \bar{v} \leq 0 \\ & g_{(i,j)}(\mathbf{x}) = x_{(i,j)} - x_{(i-1,j-1)} - x_{(i-1,j)} - x_{(i-1,j+1)} \leq 0 \\ & \text{for } (i = 2, \dots, n_y), (j = 1, \dots, n_x) \\ & \mathbf{x} \in \mathcal{X}, \quad \mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{0} \leq \mathbf{x} \leq \mathbf{1}\} \end{aligned} \quad (7.3)$$

Bien sûr on vous demande pas d'implémenter ceci.

Pour deux tailles de maillage 30x20 et 60x30 veuillez lancer le fichier exemple comparison.mlx. Celui-ci compare top88 standard et la version avec prise en compte des overhang angle.

Tracer le front de Pareto Compliance vs volfrac (volfrac= 0.35 :0.1 :0.7) pour le cas Sans et avec contraintes d'overhang angle.

Ajouter des imageries des solutions à convergence x*

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.

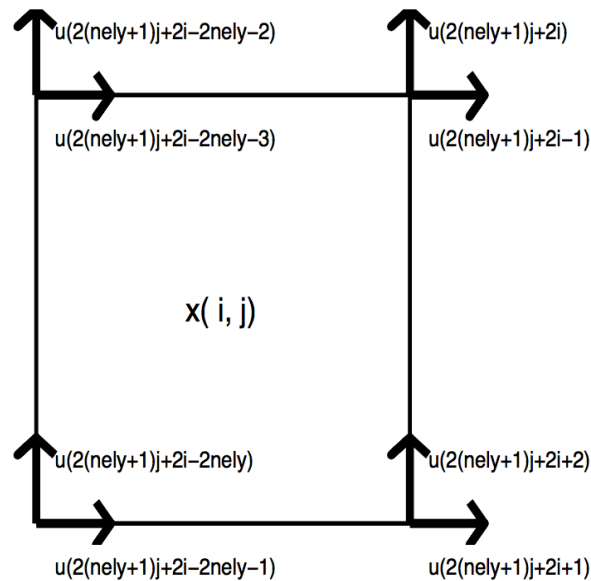


Figure 2: Le déplacement d'un EF quad $x(i, j)$

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+GKS1)*VM1, '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

$$E_{mat} = E/(1-\nu^2) \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix};$$

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 excitation, 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 optimization, 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 ($E \approx 2 \cdot 10^{11} \text{N/m}^2$) ... 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 déplacement cohérent

- 1) choisi un système cohérent de mesure N, mm, MPa.
- 2) $E=210 \times 10^3$;
- 3) multiplie la force fois F fois 54×10^3 ;
- 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 $2 \cdot (nelx+1) \cdot (nely+1)$ par une sur le DOF $(nelx+2) \cdot (nely+1)$

La compliance est en mJ.