ME471 PA #7

Hao Yin

Read the attached paper on the course website, download the 99-line Matlab code for topology optimization from http://www.topopt.dtu.dk/files/top.m (or copy and paste from the appendix of the paper), and address the following questions:

(a) Explore the effect of filter size rmin

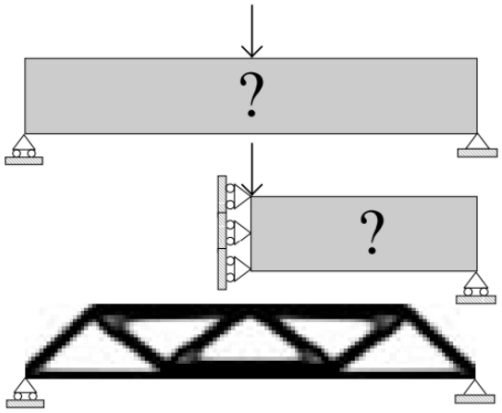


Figure 1. Topology optimization of the MBB-beam (extracted from the paper)

The above figure is the default problem embedded in the Matlab code. As the paper stated, the filtering technique is implemented inside the code to ensure the existence of a solution that is macroscopically feasible and independent of meshes. The parameter of filter size (rmin) is set to implement this technique. Please run the default problem (no need to change the code) using the following command:

$$\text{top}\ (60,20,0.5,3.0,\text{rmin})$$

with $\text{rmin} = 1.0,1.5$ and 5.0, respectively. Show the optimized structures and make comments on the effect of rmin.

(b) Explore the effect of the penalization factor. Use the command top $(60,20,0.5,p,3)$ and $\text{p} = 1,2,3$. What do you observe?

(c) Modify the code to solve the following optimization problem.

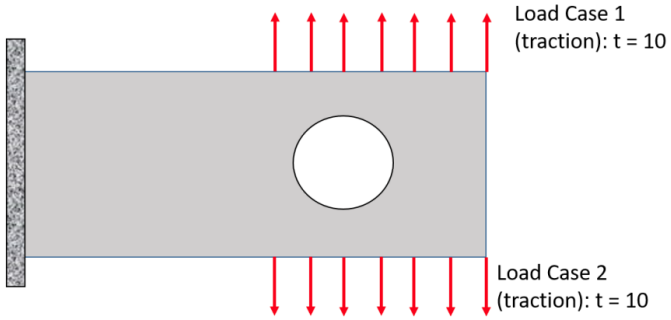


Figure 2. Topology optimization of a cantilever beam with a hole under 2 different load cases

Conditions:
(1) A cantilever beam of length L with left end clamped.
(2) Two different load cases. Load case 1 is a distributed load with the value of 10 exerted at the upper surface, from $x = L/2$ to $x = L$. Load case 2 is a distributed load with the same magnitude of 10 , but opposite direction, exerted at the lower surface, from $x = L/2$ to $x = L$.
(3) There is a hole with the center located at $(2 * \text{nelx}/3, \text{nely}/2)$ and radius of nely/3.
Following the instructions in the paper, make corresponding modifications to the code and run with the command:

top (90,40,0.5,3.0,1.5)

Show the optimized structure and attach your modified code.

a) Effect of filter size ($rmin$)

When $rmin = 1.0, 1.5$ and 5.0 respectively, the optimized structure is shown as in Figure 1(a)(b)(c). It can be observed that the filter size is related with the "geometric resolution" of optimized structure, also the pattern of structure. Larger the filter size, smaller the geometric resolution of optimized structure. This is partially because the filtering technique is used to ensure the existence of solutions of mesh-independent design.

Mesh-independency filter works by modifying the element sensitivities as follows:

$$\frac{\partial \tilde{C}}{\partial \rho_i} = \frac{1}{\rho_i \sum_{j=1}^{N} \tilde{H}_j} \sum_{j=1}^{N} \tilde{H}_j \rho_j \frac{\partial C}{\partial \rho_i} \qquad (1)$$

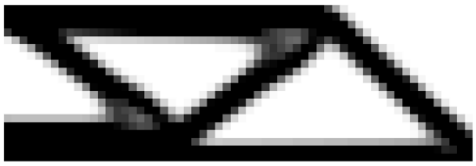
The convolution operator (weight factor) $\widetilde{H}_j$ is written as:

$$\widetilde{H}_j = r_{min} - dist(i,j)$$

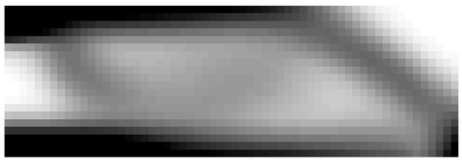
$$\{f \in N \mid dist(i,j) \leq r_{min}\}, \quad i = 1, \ldots, N \tag{2}$$

where the operator $dist(i,j)$ is defined as the distance between center of element i and j. The convolution operator $\widetilde{H}_j$ is zero outside the filter area, and decays linearly with the distance from element i. Thus, smaller filter size $rmin$ means the convolution operator quickly starts decaying when the distance away the center of element i, thus makes the optimized structure more clear.

(a) case of rmin = 1.0



(b) case of rmin = 1.5



(c) case of $\mathrm{rmin} = 5.0$

Figure 1. Topology Optimized Structures of the MBB-beam when varying the filter size ($rmin$)

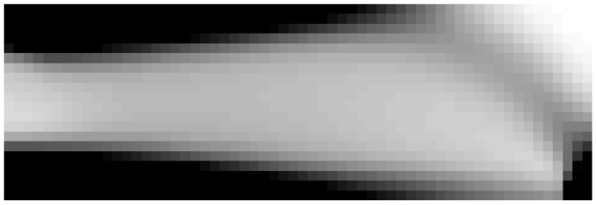
b) Effect of penalization factor (p)

When $p = 1, 2$ and 3 respectively, the optimized structure is shown as in Figure 2(a)(b)(c). It can be observed that the filter size is related with the "geometric resolution" of optimized structure, larger the penalization factor p, higher the geometric resolution of optimized structure. This is because the penalization factor p is used to penalize intermediate densities, resulting distinct solid points and void points, i.e. $\rho = \rho_{min} \approx 0$ for void point and $\rho = 1$ for solid point.

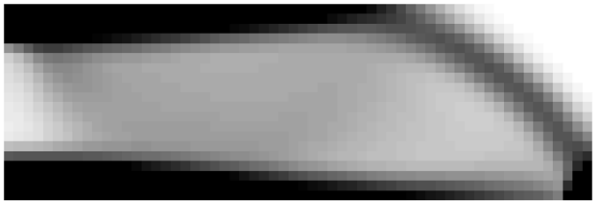
Penalization model works by adding the penalization factor p as follows:

$$\mathbb{C}(x) = \rho^{p}(x)\mathbb{C}^{+} \tag{3}$$

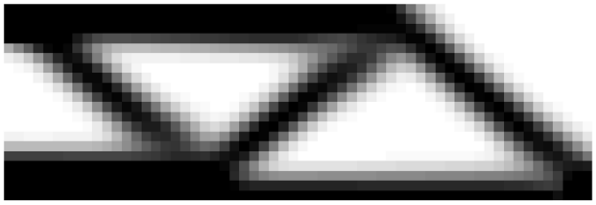
With larger penalization factor p, the $\rho^{\mathrm{p}}(\mathrm{x})$ is closer to 0 or 1, thus results of intermediate densities are reduced, and consequently it makes the optimized design more "black-and-white".



(a) case of $p = 1$



(b) case of $p = 2$



(c) case of $p = 3$

Figure 2. Topology Optimized Structures of the MBB-beam when varying the penalization factor (p)

c) Modified Case

For solving the following optimization problem as shown in Figure 3, topology optimization code was slightly modified. The Structure Modified is shown as in Figure 4. The modified code is followed in Appendix.

Conditions:

(1) A cantilever beam of length L with left end clamped.

(2) Two different load cases. Load case 1 is a distributed load with the value of 10 exerted at the upper surface, from x = L/2 to x = L. Load case 2 is a distributed load with the same magnitude of 10, but opposite direction, exerted at the lower surface, from x = L/2 to x = L.

(3) There is a hole with the center located at (2*nelx/3, nely/2) and radius of nely/3.

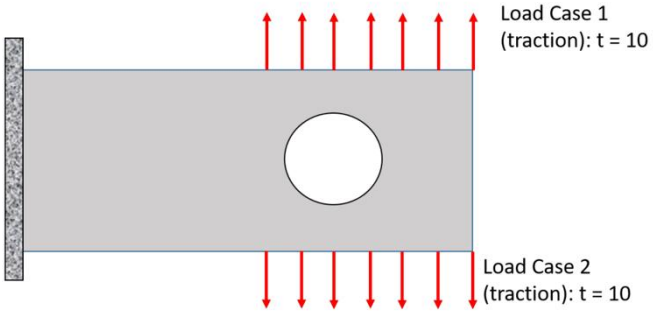


Figure 3. Topology optimization of a cantilever beam with a hole under 2 different load cases

Figure 4. Topology Optimized Structures of a cantilever beam with a hole under 2 different load cases

Appendix: Modified Source Code

top_modified.m

```
%%%% A 99 LINE TOPOLOGY OPTIMIZATION CODE BY OLE SIGMUND, JANUARY 2000 %%%
%%%% CODE MODIFIED FOR INCREASED SPEED, September 2002, BY OLE SIGMUND %%%
function top(nelx,nely,volfrac,penal,rmin);
% INITIALIZE
%passive = zeros(nely,nelx);

x(1:nely,1:nelx) = volfrac;
for ely = 1:nely
      for elx = 1:nelx
            if sqrt((ely-nely/2.)^2+(elx-2*nelx/3.)^2) < nely/3.
                  passive(ely,elx) = 1;
                  x(ely,elx) = 0.001;
            else
                  passive(ely,elx) = 0;
            end
      end
end
loop = 0;
change = 1.;
% START ITERATION
while change > 0.01
   loop = loop + 1;
   xold = x;
% FE-ANALYSIS
   [U]=FE(nelx,nely,x,penal);
% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
   [KE] = lk;
   c = 0.;
   for ely = 1:nely
      for elx = 1:nelx
            dc(ely,elx) = 0.;
            n1 = (nely+1)*(elx-1)+ely;
            n2 = (nely+1)* elx      +ely;
            for i = 1:2
               Ue = U([2*n1-1;2*n1; 2*n2-1;2*n2; 2*n2+1;2*n2+2;2*n1+1;2*n1+2],i);
               c = c + x(ely,elx)^penal*Ue'*KE*Ue;
               dc(ely,elx) = dc(ely,elx) - penal*x(ely,elx)^(penal-1)*Ue'*KE*Ue;
            end
      end
   end
```

```
% FILTERING OF SENSITIVITIES
  [dc]     = check(nelx,nely,rmin,x,dc);
% DESIGN UPDATE BY THE OPTIMALITY CRITERIA METHOD
  [x]      = OC(nelx,nely,x,volfrac,dc,passive);
% PRINT RESULTS
  change = max(max(abs(x-xold)));
  disp([' It.: ' sprintf('%4i',loop) ' Obj.: ' sprintf('%10.4f',c) ...
       ' Vol.: ' sprintf('%6.3f',sum(sum(x))/(nelx*nely)) ...
        ' ch.: ' sprintf('%6.3f',change )])
% PLOT DENSITIES
  colormap(gray); imagesc(-x); axis equal; axis tight; axis off;pause(1e-6);
end
%%%%%%%%%% OPTIMALITY CRITERIA
UPDATE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [xnew]=OC(nelx,nely,x,volfrac,dc,passive)
l1 = 0; l2 = 100000; move = 0.2;
while (l2-l1 > 1e-4)
  lmid = 0.5*(l2+l1);
  xnew = max(0.001,max(x-move,min(1.,min(x+move,x.*sqrt(-dc./lmid)))));
  xnew(find(passive)) = 0.001;
  if sum(sum(xnew)) - volfrac*nelx*nely > 0;
    l1 = lmid;
  else
    l2 = lmid;
  end
end
%%%%%%%%%% MESH-INDEPENDENCY
FILTER %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [dcn]=check(nelx,nely,rmin,x,dc)
dcn=zeros(nely,nelx);
for i = 1:nelx
  for j = 1:nely
    sum=0.0;
    for k = max(i-floor(rmin),1):min(i+floor(rmin),nelx)
      for l = max(j-floor(rmin),1):min(j+floor(rmin),nely)
        fac = rmin-sqrt((i-k)^2+(j-l)^2);
        sum = sum+max(0,fac);
        dcn(j,i) = dcn(j,i) + max(0,fac)*x(l,k)*dc(l,k);
      end
    end
    dcn(j,i) = dcn(j,i)/(x(j,i)*sum);
  end
end
```

```
%%%%%%%%%% FE-
ANALYSIS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%
function [U]=FE(nelx,nely,x,penal)
[KE] = lk;
K = sparse(2*(nelx+1)*(nely+1), 2*(nelx+1)*(nely+1));
F = sparse(2*(nely+1)*(nelx+1),2); U = zeros(2*(nely+1)*(nelx+1),2);
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
% DEFINE LOADS AND SUPPORTS (HALF MBB-BEAM)
F(2*(nelx+1)*(nely+1),1) = -5.; F(2*(nelx)*(nely+1)+2,2) = 5.;
for i=1:round((nelx-1)/2)-1
    F(2*(nelx+1-i)*(nely+1),1) = -10.; F(2*(nelx-i)*(nely+1)+2,2) = 10.;
end
F(2*(nelx+1-round((nelx-1)/2))*(nely+1),1) = -5.; F(2*(nelx-round((nelx-1)/2))*(nely+1)+2,2) = 5.;
fixeddofs = [1:2*(nely+1)];
alldofs     = [1:2*(nely+1)*(nelx+1)];
freedofs    = setdiff(alldofs,fixeddofs);
% SOLVING
U(freedofs,:) = K(freedofs,freedofs) \ F(freedofs,:);
U(fixeddofs,:)= 0;
%%%%%%%%%% ELEMENT STIFFNESS
MATRIX %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [KE]=lk
E = 1.;
nu = 0.3;
k=[ 1/2-nu/6   1/8+nu/8 -1/4-nu/12 -1/8+3*nu/8 ...
   -1/4+nu/12 -1/8-nu/8  nu/6       1/8-3*nu/8];
KE = E/(1-nu^2)*[ k(1) k(2) k(3) k(4) k(5) k(6) k(7) k(8)
                  k(2) k(1) k(8) k(7) k(6) k(5) k(4) k(3)
                  k(3) k(8) k(1) k(6) k(7) k(4) k(5) k(2)
                  k(4) k(7) k(6) k(1) k(8) k(3) k(2) k(5)
                  k(5) k(6) k(7) k(8) k(1) k(2) k(3) k(4)
                  k(6) k(5) k(4) k(3) k(2) k(1) k(8) k(7)
                  k(7) k(4) k(5) k(2) k(3) k(8) k(1) k(6)
                  k(8) k(3) k(2) k(5) k(4) k(7) k(6) k(1)];
%
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This Matlab code was written by Ole Sigmund, Department of Solid          %
% Mechanics, Technical University of Denmark, DK-2800 Lyngby, Denmark.       %
% Please sent your comments to the author: sigmund@fam.dtu.dk                %
%                                                                            %
% The code is intended for educational purposes and theoretical details      %
% are discussed in the paper                                                 %
% "A 99 line topology optimization code written in Matlab"                   %
% by Ole Sigmund (2001), Structural and Multidisciplinary Optimization,      %
% Vol 21, pp. 120--127.                                                      %
%                                                                            %
% The code as well as a postscript version of the paper can be               %
% downloaded from the web-site: http://www.topopt.dtu.dk                     %
%                                                                            %
% Disclaimer:                                                                %
% The author reserves all rights but does not guaranty that the code is      %
% free from errors. Furthermore, he shall not be liable in any event         %
% caused by the use of the program.                                          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```