MAE 598 Design Optimization Project 3:

Application of Reduced Gradient in Topology Optimization

Jessica Westerham

Professor Yi Ren

Arizona State University

December 11, 2021

Objective

This project implements an optimization algorithm for minimizing the topology of a beam structure at equilibrium. The code and tutorial are derived from Dr. Sigmund's topology code and paper. The results of this project will provide a mathematical solution to minimize compliance by designing the stiffest structure to withstand applied forces and boundary conditions while limiting the volume.

Problem

The structure is a 2-dimensional beam with dimensions set to 200 x 39. The material is Aluminum Alloy 2024 with a Modulus of Elasticity of 72.4 GPa and a Poissant's ratio of 0.33. Emin is set to 1e-9, which represents the minimum value of E to prevent singularities. The boundary conditions and loading can be seen in Figure 1. There is a fixed pin support at the bottom left corner of the structure that prevents displacement in both the x- and y-directions. There is also a roller support on the lower right section of the structure, one-quarter length away from the corner, that prevents displacement in the y-direction. Loadings are applied in the negative y-direction along the top edge of the structure: one is one-quarter length away from the left corner, and the other is on the right corner.

The objective is to minimize compliance. The constraints are to meet the specific volume requirement by selecting a volume fraction, or physical density, between 0 and 1. Additionally, a penalization factor (typically p = 3) is applied to ensure black-and-white solutions, a filter radius is specified, and sensitivity filtering is selected for this simulation.

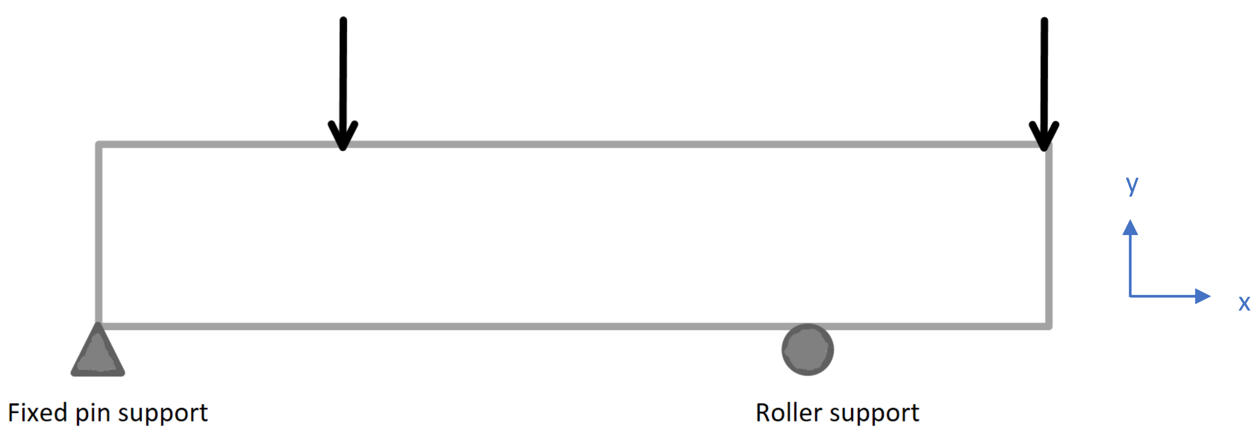

Figure 1. Sketch of 2D Beam with Boundary Conditions and Applied Forces

The code was adjusted to accommodate these new material properties as well as the new loading and boundary conditions (See Appendix). The result is displayed in Figure 2 below.

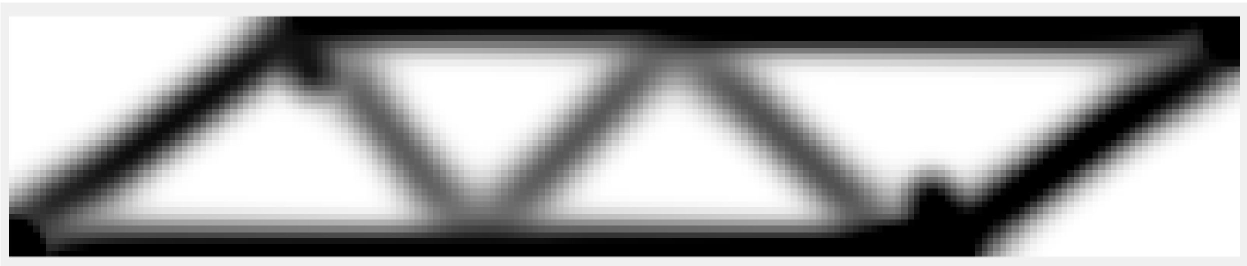

Figure 2. Results of Topology Optimization

The results are reasonable considering that this design is a truss and proves the effectiveness of trusses when it comes to withstanding loadings while minimizing volume of material. Considering the placement of the supports and loadings, this solution makes sense, with the lower left and right corners of the truss being supported by a pin and roller, respectively, and the upper left and right corners of the truss being loaded with a downward force on each, respectively.

This solution is optimal for Aluminum Alloy 2024 given the generalized boundary conditions and loadings that were simulated in this project.

Appendix

Run the top88 function

```
top88(200,39,0.4,3.0,6,1)
```

Code for top88.m reduced gradient function

Provided: images of formatting, text for simulation

```
%%%% AN 88 LINE TOPOLOGY OPTIMIZATION CODE Nov, 2010 %%%%
function top88(nelx,nely,volfrac,penal,rmin,ft)
%% MATERIAL PROPERTIES
% Modulus of Elasticity and Poissant's Ratio for Aluminum Alloy 2024
E0 = 0.724; % 10^11 Pa
Emin = 1e-9;
nu = 0.33;
%% PREPARE FINITE ELEMENT ANALYSIS
A11 = [12  3 -6 -3;  3 12  3  0; -6  3 12 -3; -3  0 -3 12];
A12 = [-6 -3  0  3; -3 -6 -3 -6;  0 -3 -6  3;  3 -6  3 -6];
B11 = [-4  3 -2  9;  3 -4 -9  4; -2 -9 -4 -3;  9  4 -3 -4];
B12 = [ 2 -3  4 -9; -3  2  9 -2;  4  9  2  3; -9 -2  3  2];
KE = 1/(1-nu^2)/24*([A11 A12;A12' A11]+nu*[B11 B12;B12' B11]);
nodenrs = reshape(1:(1+nelx)*(1+nely),1+nely,1+nelx);
edofVec = reshape(2*nodenrs(1:end-1,1:end-1)+1,nelx*nely,1);
edofMat = repmat(edofVec,1,8)+repmat([0 1 2*nely+[2 3 0 1] -2 -1],nelx*nely,1);
iK = reshape(kron(edofMat,ones(8,1))',64*nelx*nely,1);
jK = reshape(kron(edofMat,ones(1,8))',64*nelx*nely,1);
% DEFINE LOADS AND SUPPORTS
F = sparse([2*(nely+1)*(1/4)*(nelx+1)-(2*nely),2*(nely+1)*(nelx+1)-(2*nely)],[1 2],[-1 -1],2*(nely+1)*(nelx+1),2);
U = zeros(2*(nely+1)*(nelx+1),2);
fixeddofs = union([2*(nely+1)-1,2*(nely+1)],[((nelx+1)*(3/4))*2*(nely+1)]);
alldofs = [1:2*(nely+1)*(nelx+1)];
freedofs = setdiff(alldofs,fixeddofs);
%% PREPARE FILTER
iH = ones(nelx*nely*(2*(ceil(rmin)-1)+1)^2,1);
jH = ones(size(iH));
sH = zeros(size(iH));
k = 0;
for i1 = 1:nelx
  for j1 = 1:nely
    e1 = (i1-1)*nely+j1;
    for i2 = max(i1-(ceil(rmin)-1),1):min(i1+(ceil(rmin)-1),nelx)
      for j2 = max(j1-(ceil(rmin)-1),1):min(j1+(ceil(rmin)-1),nely)
        e2 = (i2-1)*nely+j2;
        k = k+1;
        iH(k) = e1;
        jH(k) = e2;
```

```
        sH(k) = max(0,rmin-sqrt((i1-i2)^2+(j1-j2)^2));
      end
    end
  end
end
H = sparse(iH,jH,sH);
Hs = sum(H,2);
%% INITIALIZE ITERATION
x = repmat(volfrac,nely,nelx);
xPhys = x;
loop = 0;
change = 1;
%% START ITERATION
while change > 0.01
  loop = loop + 1;
  %% FE-ANALYSIS
  sK = reshape(KE(:)*(Emin+xPhys(:)'.^penal*(E0-Emin)),64*nelx*nely,1);
  K = sparse(iK,jK,sK); K = (K+K')/2;
  U(freedofs, :) = K(freedofs,freedofs)\F(freedofs, :);
  %% OBJECTIVE FUNCTION AND SENSITIVITY ANALYSIS
   c = 0;
   dc = 0;
  for i = 1: size(F,2)
      Ui = U(:,i);
      ce = reshape(sum((Ui(edofMat)*KE).*Ui(edofMat),2),nely,nelx);
      c = c + sum(sum((Emin+xPhys.^penal*(E0-Emin)).*ce));
      dc = dc-penal*(E0-Emin)*xPhys.^(penal-1).*ce;
  end
  dv = ones(nely,nelx);
  %% FILTERING/MODIFICATION OF SENSITIVITIES
  if ft == 1
    dc(:) = H*(x(:).*dc(:))./Hs./max(1e-3,x(:));
  elseif ft == 2
    dc(:) = H*(dc(:)./Hs);
    dv(:) = H*(dv(:)./Hs);
  end
  %% OPTIMALITY CRITERIA UPDATE OF DESIGN VARIABLES AND PHYSICAL DENSITIES
  l1 = 0; l2 = 1e9; move = 0.2;

% create a hole in the center of the beam for a potential cross-beam
%   passive = zeros(nely,nelx);
%   for i = 1:nelx
%       for j = 1:nely
%           if sqrt((j-nely/2)^2+(i-nelx/3)^2) < nely/3
%               passive(j,i) = 1;
%           end
%       end
%   end
  while (l2-l1)/(l1+l2) > 1e-3
    lmid = 0.5*(l2+l1);
    xnew = max(0,max(x-move,min(1,min(x+move,x.*sqrt(-dc./dv/lmid)))));
    if ft == 1
      xPhys = xnew;
    elseif ft == 2
      xPhys(:) = (H*xnew(:))./Hs;
    end
    % for hole in center of beam
    % xPhys(passive == 1) = 0;
    % xPhys(passive == 2) = 1;
    if sum(xPhys(:)) > volfrac*nelx*nely, l1 = lmid; else l2 = lmid; end
  end
  change = max(abs(xnew(:)-x(:)));
  x = xnew;
  %% PRINT RESULTS
  fprintf(' It.:%5i Obj.:%11.4f Vol.:%7.3f ch.:%7.3f\n',loop,c, ...
    mean(xPhys(:)),change);
  %% PLOT DENSITIES
  colormap(gray); imagesc(1-xPhys); caxis([0 1]); axis equal; axis off; drawnow;
end
```

```
%%%% AN 88 LINE TOPOLOGY OPTIMIZATION
CODE Nov, 2010 %%%%
function
top88(nelx,nely,volfrac,penal,rmin,ft)
%% MATERIAL PROPERTIES
% Modulus of Elasticity and Poissant's
Ratio for Aluminum Alloy 2024
E0 = 0.724; % 10^11 Pa
Emin = 1e-9;
nu = 0.33;
%% PREPARE FINITE ELEMENT ANALYSIS
A11 = [12  3 -6 -3;  3 12  3  0; -6  3
12 -3; -3  0 -3 12];
A12 = [-6 -3  0  3; -3 -6 -3 -6;  0 -3
-6  3;  3 -6  3 -6];
B11 = [-4  3 -2  9;  3 -4 -9  4; -2 -9
-4 -3;  9  4 -3 -4];
B12 = [ 2 -3  4 -9; -3  2  9 -2;  4  9
2  3; -9 -2  3  2];
KE = 1/(1-nu^2)/24*([A11 A12;A12'
A11]+nu*[B11 B12;B12' B11]);
nodenrs =
reshape(1:(1+nelx)*(1+nely),1+nely,1+n
elx);
edofVec = reshape(2*nodenrs(1:end-
1,1:end-1)+1,nelx*nely,1);
```

```
edofMat =
repmat(edofVec,1,8)+repmat([0 1
2*nely+[2 3 0 1] -2 -1],nelx*nely,1);
iK =
reshape(kron(edofMat,ones(8,1))',64*ne
lx*nely,1);
jK =
reshape(kron(edofMat,ones(1,8))',64*ne
lx*nely,1);
% DEFINE LOADS AND SUPPORTS
F = sparse([2*(nely+1)*(1/4)*(nelx+1)-
(2*nely),2*(nely+1)*(nelx+1)-
(2*nely)],[1 2],[-1 -
1],2*(nely+1)*(nelx+1),2);
U = zeros(2*(nely+1)*(nelx+1),2);
fixeddofs = union([2*(nely+1)-
1,2*(nely+1)],[((nelx+1)*(3/4))*2*(nel
y+1)]);
alldofs = [1:2*(nely+1)*(nelx+1)];
freedofs = setdiff(alldofs,fixeddofs);
%% PREPARE FILTER
iH = ones(nelx*nely*(2*(ceil(rmin)-
1)+1)^2,1);
jH = ones(size(iH));
sH = zeros(size(iH));
k = 0;
for i1 = 1:nelx
  for j1 = 1:nely
```

```
    e1 = (i1-1)*nely+j1;
    for i2 = max(i1-(ceil(rmin)-
1),1):min(i1+(ceil(rmin)-1),nelx)
      for j2 = max(j1-(ceil(rmin)-
1),1):min(j1+(ceil(rmin)-1),nely)
        e2 = (i2-1)*nely+j2;
        k = k+1;
        iH(k) = e1;
        jH(k) = e2;
        sH(k) = max(0,rmin-sqrt((i1-
i2)^2+(j1-j2)^2));
      end
    end
  end
end
H = sparse(iH,jH,sH);
Hs = sum(H,2);
%% INITIALIZE ITERATION
x = repmat(volfrac,nely,nelx);
xPhys = x;
loop = 0;
change = 1;
%% START ITERATION
while change > 0.01
  loop = loop + 1;
  %% FE-ANALYSIS
```

```
  sK =
reshape(KE(:)*(Emin+xPhys(:)'.^penal*(
E0-Emin)),64*nelx*nely,1);
  K = sparse(iK,jK,sK); K = (K+K')/2;
  U(freedofs, :) =
K(freedofs,freedofs)\F(freedofs, :);
  %% OBJECTIVE FUNCTION AND
SENSITIVITY ANALYSIS
   c = 0;
   dc = 0;
  for i = 1: size(F,2)
      Ui = U(:,i);
      ce =
reshape(sum((Ui(edofMat)*KE).*Ui(edofM
at),2),nely,nelx);
      c = c +
sum(sum((Emin+xPhys.^penal*(E0-
Emin)).*ce));
      dc = dc-penal*(E0-
Emin)*xPhys.^(penal-1).*ce;
  end
  dv = ones(nely,nelx);
  %% FILTERING/MODIFICATION OF
SENSITIVITIES
  if ft == 1
    dc(:) =
H*(x(:).*dc(:))./Hs./max(1e-3,x(:));
  elseif ft == 2
```

```
    dc(:) = H*(dc(:)./Hs);
    dv(:) = H*(dv(:)./Hs);
  end
  %% OPTIMALITY CRITERIA UPDATE OF
DESIGN VARIABLES AND PHYSICAL
DENSITIES
  l1 = 0; l2 = 1e9; move = 0.2;
% create a hole in the center of the
beam for a potential cross-beam
%   passive = zeros(nely,nelx);
%   for i = 1:nelx
%       for j = 1:nely
%           if sqrt((j-nely/2)^2+(i-
nelx/3)^2) < nely/3
%               passive(j,i) = 1;
%           end
%       end
%   end
  while (l2-l1)/(l1+l2) > 1e-3
    lmid = 0.5*(l2+l1);
    xnew = max(0,max(x-
move,min(1,min(x+move,x.*sqrt(-
dc./dv/lmid)))));
    if ft == 1
      xPhys = xnew;
    elseif ft == 2
      xPhys(:) = (H*xnew(:))./Hs;
    end
```

```
    % for hole in center of beam
    % xPhys(passive == 1) = 0;
    % xPhys(passive == 2) = 1;
    if sum(xPhys(:)) >
volfrac*nelx*nely, l1 = lmid; else l2
= lmid; end
  end
  change = max(abs(xnew(:)-x(:)));
  x = xnew;
  %% PRINT RESULTS
  fprintf(' It.:%5i Obj.:%11.4f
Vol.:%7.3f ch.:%7.3f\n',loop,c, ...
    mean(xPhys(:)),change);
  %% PLOT DENSITIES
  colormap(gray); imagesc(1-xPhys);
caxis([0 1]); axis equal; axis off;
drawnow;
end
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This Matlab code was written by E.
Andreassen, A. Clausen, M.
Schevenels,%
% B. S. Lazarov and O. Sigmund,
Department of Solid
Mechanics,               %
```

```
%  Technical University of
Denmark,
 %
%  DK-2800 Lyngby,
Denmark.
        %
% Please sent your comments to:
sigmund@fam.dtu.dk
%
%
                            %
% The code is intended for educational
purposes and theoretical details    %
% are discussed in the
paper
    %
% "Efficient topology optimization in
MATLAB using 88 lines of code,       %
% E. Andreassen, A. Clausen, M.
Schevenels,
%
% B. S. Lazarov and O. Sigmund, Struct
Multidisc Optim, 2010               %
% This version is based on earlier 99-
line code                            %
% 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 authors reserves all rights but
do not guaranty that the code is      %
% free from errors. Furthermore, we
shall not be liable in any
event         %
% caused by the use of the
program.
 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```