Application of Reduced Gradient - Topology Optimization

Xiaoyun Fan

1. The compliance minimization problem

Topology optimization is a powerful tool to get the best structure that can reach the design requirement and save the material. The classical objective function for topology optimization problem is to minimum the compliance of the structure. Finite element method is used when simulation. In this case, our optimization function is :

$$\min_x \ c(x) = U^T K U = \sum_{e=1}^{x} E_e(x_e) {u_e}^T k_0 u_e$$

$$s.t. V(x) - v \leq 0$$

$$KU = F$$

$$x \epsilon [0,1]$$

where c is the compliance,
U is the global displacement,
F is the force,
K is the global stiffness matrix,
E_e is the Young's modulus of the element;
u_e is the element displacement vector,
k_0 is the element stiffness matrix for an element with unit Young's modulus,
x is the vector of design variables(density),
N is the number of elements,
$V(x)$ the material volume,
V is max volume for this design,
f is the prescribed volume fraction.

E_e is defined by

$$E(x) = \Delta E * x^p + E_{min}$$

where p (the penalty parameter) is usually set to 3, E_{min} is a small number for numerical stability.

$p = 3$ helps binarize the topologies, i.e., to push the optimal xi to 1 or 0, because of the shape of the cubic function. If the x is small, power 3 will make it more close to 0,

2. Design sensitivity analysis

The reduced gradient (often called design sensitivity) can be calculated as

$$\frac{\partial c}{\partial x} = -p x_e^{p-1} (E_0 - E_{\min}) u_e^T k_0 u$$

The input for this case is nelx,nely,volfrac,penal,rmin,ft= 150,80,0.5,5,3,2
E0 = 300; nu = 0.28;

The result structure is shown in Figure 1.

Figure 1 topology optimization

3. Matlab code

```
%%%% AN 88 LINE TOPOLOGY OPTIMIZATION CODE Nov, 2010 %%%%
%30,12,0.5 ,5,3,2
function top88(nelx,nely,volfrac,penal,rmin,ft)
%% MATERIAL PROPERTIES
E0 = 300;
Emin = 1e-9;
nu = 0.28;
%% 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 (HALF MBB-BEAM)
F = sparse(2,1,-1,2*(nely+1)*(nelx+1),1);
```

```
U = zeros(2*(nely+1)*(nelx+1),1);
fixeddofs = union([1:2:2*(nely+1)],[2*(nelx+1)*(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
  ce = reshape(sum((U(edofMat)*KE).*U(edofMat),2),nely,nelx);
  c = sum(sum((Emin+xPhys.^penal*(E0-Emin)).*ce));
  dc = -penal*(E0-Emin)*xPhys.^(penal-1).*ce;
  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;
  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
    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
%
```