MAE 598 Design Optimization

Topology optimization (Application of Reduced Gradient)

Onkar Chavan

ASU id: 1223314248

Topology Optimization:

Topology Optimization is a useful method in reducing the material used and strain energy of a structure. Topology optimization is a mathematical method which spatially optimizes the distribution of material within a defined domain, by fulfilling given constraints previously established and minimizing a predefined cost function.

Problem Definition:

The objective is to minimize the following function:

$$\min_{\mathbf{x}} \quad \mathbf{f} := \mathbf{d}^T\mathbf{K}(\mathbf{x})\mathbf{d}$$

$$\text{subject to:} \quad \mathbf{h} := \mathbf{K}(\mathbf{x})\mathbf{d} = \mathbf{u},$$

$$\mathbf{g} := V(\mathbf{x}) \leq v,$$

$$\mathbf{x} \in [0, 1].$$

Where,

x = set of densities = $\{x_i\}$ for I = 1,2,3…. n

d = displacement of structure under load ‘u’,

V(x) = Total Volume,

v = upper bound of volume,

K(x) = global stiffness matrix

Where,

$K_i = K_e * E(x_i)$ and

$K(x) = G[K_1, K_2, K_3, \ldots K_n]$

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

here p is the penalty parameter, which is usually set to 3, E_{min} is provided for numerical stability.

The penalty parameter helps reduce the topologies to binary values, as x_i is closer to 0, the cubic function reduces it 0.

Design Sensitivity:

The design sensitivity (reduced gradient) can be calculated as follows:

$$\frac{df}{d\mathbf{x}} = \frac{\partial f}{\partial \mathbf{x}} - \frac{\partial f}{\partial \mathbf{u}}\left(\frac{\partial \mathbf{h}}{\partial \mathbf{u}}\right)^{-1}\frac{\partial \mathbf{h}}{\partial \mathbf{x}},$$

$$\frac{df}{d\mathbf{x}} = -\mathbf{u}^T\frac{\partial \mathbf{K}}{\partial \mathbf{x}}\mathbf{u}.$$

$$\mathbf{u}^T\mathbf{K}\mathbf{u} = \sum_{i=1}^{n}\mathbf{u}_i^T\mathbf{K}_i\mathbf{u}_i,$$

$$-\mathbf{u}^T\frac{\partial \mathbf{K}}{\partial \mathbf{x}}\mathbf{u} = -\frac{\partial \mathbf{u}^T\mathbf{K}\mathbf{u}}{\partial \mathbf{x}}$$

$$= -\frac{\partial \sum_{i=1}^{n}\mathbf{u}_i^T\mathbf{K}_i\mathbf{u}_i}{\partial \mathbf{x}}$$

$$= [\ldots, -\frac{\partial \mathbf{u}_i^T\mathbf{K}_i\mathbf{u}_i}{\partial x_i}, \ldots]$$

$$= [\ldots, -\mathbf{u}_i^T\frac{\partial \mathbf{K}_i}{\partial x_i}\mathbf{u}_i, \ldots]$$

$$= [\ldots, -\mathbf{u}_i^T\frac{\partial \bar{\mathbf{K}}_e \Delta E x_i^3}{\partial x_i}\mathbf{u}_i, \ldots]$$

$$= [\ldots, -3\Delta E x_i^2\mathbf{u}_i^T\bar{\mathbf{K}}_e\mathbf{u}_i, \ldots]$$

Onkar Chavan

ASU ID : 1223314248

MAE 598 Project 3 Report

```
%%%% AN 88 LINE TOPOLOGY OPTIMIZATION CODE Nov, 2010 %%%%

nelx = 120;
nely = 60;
volfrac = 0.5;
penal = 3;
rmin = 3;
ft = 2;

%% MATERIAL PROPERTIES
E = 200;
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]);
fprintf("nelx = %d \n",nelx);
```

```
nelx = 120
```

```
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);
```

```matlab
% 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 GAUSSIAN 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*(E-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);
    % element-wise strain energy
    c = sum(sum((Emin + xPhys.^penal*(E-Emin)).*ce));
    % total strain energy
    dc = -penal*(E-Emin)*xPhys.^(penal-1).*ce;
    % design sensitivity
    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 DmoveENSITIES
    s_1 = 0;
    s_2 = 1e9;
    s = 0.2;
    while (s_2 - s_1)/(s_1 + s_2) > 1e-3
        s_m = 0.5*(s_2 + s_1);
        x_n = max(0, max(0, max(x-s, min(1, min(x + s, x.*sqrt(-dc./dv/s_m))))));
        if ft==1
            xPhys = x_n;
        elseif ft == 2
            xPhys(:) = (H*x_n(:))./Hs;
        end
        if sum(xPhys(:)) > volfrac*nelx*nely
            s_1 = s_m;
        else
            s_2 = s_m;
        end
    end
    change = max(abs(x_n(:)-x(:)));
    x = x_n;
%% PLOT DENSITIES
    colormap(gray);
    imagesc(1-xPhys);
    caxis([0 1]);
    axis equal;
    axis off;
    drawnow;
end
```

```
%% PRINT RESULTS
fprintf(' It.:%5i Obj.:%11.4f Vol.:%7.3f ch.:%7.3f\n',loop,c, ...
        mean(xPhys(:)),change);
```

```
 It.:  514 Obj.:     0.4372 Vol.:  0.500 ch.:  0.009
```