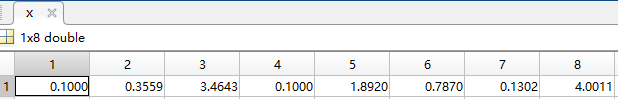


1. Used files: ***get\_cns.m , get\_model\_result.m , get\_obj.m , main.m***

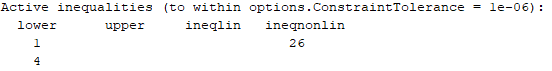
The x result is in this picture:



Compare to Table 1, we can find different groups of truss members have different cross sections.

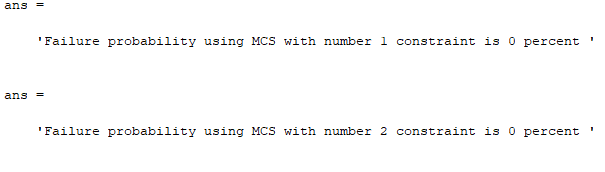


The active constraint is number 26, which is the constraint limits the displacement of node 1 in an allowance value.



1. Used files: ***get\_cns.m , get\_model\_result.m , get\_obj.m , main.m , hw4\_2***

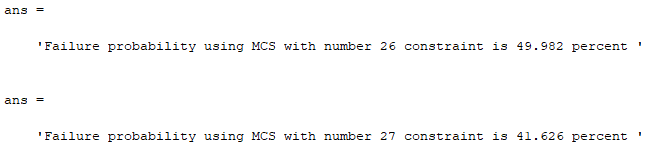
The probability of violating active constraints:



.

.

.





1. Because the displacements of node 1 & node 2 will exceed the allowance value when considering the uncertainties, the new structure of this twenty-five bars tower should become stronger. Thus, when uncertainties are considered, the overall weight will become heavier.

Files: ***get\_cns.m , get\_model\_result.m , get\_obj.m , main.m ,*** ***hw4\_2***

***get\_cns.m***:

function [c, ceq] = get\_cns(x,E)

[Q, stress] = get\_model\_result(x,E);

allow\_stress = 40000; % psi

allow\_disp = 0.35; % inch

c(1:25) = abs(stress)/allow\_stress - 1.0;

disp = zeros(1,2);

for i = 1:2 % node 1, 2

disp(i) = sqrt(Q(3\*i-2)^2 + Q(3\*i-1)^2 + Q(3\*i)^2); % total displacementof node 1,2 =square\_root(displacement in x,y,z directions)

end

c = [c, disp./allow\_disp - 1.0];

ceq = [];

***get\_model\_result.m:***

function [Q, stress] = get\_model\_result(x,E)

%E = 1e7; %E is young's modulus in psi

node\_coord(1,:) = [-37.5, 0, 200];

node\_coord(2,:) = [37.5, 0, 200];

node\_coord(3,:) = [-37.5, 37.5, 100];

node\_coord(4,:) = [37.5, 37.5, 100];

node\_coord(5,:) = [37.5, -37.5, 100];

node\_coord(6,:) = [-37.5,-37.5,100];

node\_coord(7,:) = [-100, 100, 0];

node\_coord(8,:) = [100, 100, 0];

node\_coord(9,:) = [100, -100, 0];

node\_coord(10,:) = [-100, -100, 0]; % node coord.

en\_pair = [ ...

1,2; 1,4; 2,3; 1,5; 2,6; ...

2,4; 2,5; 1,3; 1,6; 3,6; ...

4,5; 3,4; 5,6; 3,10; 6,7; ...

4,9; 5,8; 4,7; 3,8; 5,10; ...

6,9; 6,10; 3,7; 4,8; 5,9]; % elemenet node pair

A = [x(1)\*ones(1,1); ...

x(2)\*ones(4,1); ...

x(3)\*ones(4,1); ...

x(4)\*ones(2,1); ...

x(5)\*ones(2,1); ...

x(6)\*ones(4,1); ...

x(7)\*ones(4,1); ...

x(8)\*ones(4,1)]'; % element section area

F = zeros(18,1);

F(1) = 1;

F(2) = -10;

F(3) = -10;

F(4) = 0;

F(5) = -10;

F(6) = -10;

F(7) = 0.5;

F(8) = 0;

F(9) = 0;

F(16) = 0.6;

F(17) = 0;

F(18) = 0;

F = F\*1e3;

% -- stiffness matrixs

for i = 1:25

ni = en\_pair(i,1);

nj = en\_pair(i,2);

Le(i) = norm( (node\_coord(ni,:) - node\_coord(nj,:)) );

cx(i) = (node\_coord(ni,1) - node\_coord(nj,1)) / Le(i); % x

cy(i) = (node\_coord(ni,2) - node\_coord(nj,2)) / Le(i); % y

cz(i) = (node\_coord(ni,3) - node\_coord(nj,3)) / Le(i); % z

end

K = zeros(30,30); % stiffness matrix

for i = 1:25

ni = en\_pair(i,1);

nj = en\_pair(i,2);

sk = [cx(i), cy(i), cz(i)]'\*[cx(i), cy(i), cz(i)];

tmp = zeros(30,30);

tmp(3\*ni-2:3\*ni, 3\*ni-2:3\*ni) = sk;

tmp(3\*nj-2:3\*nj, 3\*nj-2:3\*nj) = sk;

tmp(3\*ni-2:3\*ni, 3\*nj-2:3\*nj) = -sk;

tmp(3\*nj-2:3\*nj, 3\*ni-2:3\*ni) = -sk;

K = K + E\*A(i)/Le(i)\*tmp;

end

Kr = K(1:18,1:18); % Reduce matrix of K

% -- displacement

Qr = Kr^-1\*F;

Q = [Qr; zeros(12,1)];

% -- stress

stress = zeros(1,25);

for i = 1:25

ni = en\_pair(i,1);

nj = en\_pair(i,2);

stress(1,i) = ...

E/Le(i)\* ...

[-1\*cx(i), -1\*cy(i), -1\*cz(i), cx(i), cy(i), cz(i)]\* ...

[Q(ni\*3-2); Q(ni\*3-1); Q(ni\*3); Q(nj\*3-2); Q(nj\*3-1); Q(nj\*3)];

end

end

***get\_obj.m:***

function weight = get\_obj(x)

node\_coord(1,:) = [-37.5, 0, 200];

node\_coord(2,:) = [37.5, 0, 200];

node\_coord(3,:) = [-37.5, 37.5, 100];

node\_coord(4,:) = [37.5, 37.5, 100];

node\_coord(5,:) = [37.5, -37.5, 100];

node\_coord(6,:) = [-37.5,-37.5,100];

node\_coord(7,:) = [-100, 100, 0];

node\_coord(8,:) = [100, 100, 0];

node\_coord(9,:) = [100, -100, 0];

node\_coord(10,:) = [-100, -100, 0]; % node coord.

en\_pair = [ ...

1,2; 1,4; 2,3; 1,5; 2,6; ...

2,4; 2,5; 1,3; 1,6; 3,6; ...

4,5; 3,4; 5,6; 3,10; 6,7; ...

4,9; 5,8; 4,7; 3,8; 5,10; ...

6,9; 6,10; 3,7; 4,8; 5,9]; % elemenet's node pair

Le = zeros(1,25);

for i = 1:25

ni = en\_pair(i,1);

nj = en\_pair(i,2);

Le(i) = norm( (node\_coord(ni,:) - node\_coord(nj,:)) ); % element length

end

A = [x(1)\*ones(1,1); ...

x(2)\*ones(4,1); ...

x(3)\*ones(4,1); ...

x(4)\*ones(2,1); ...

x(5)\*ones(2,1); ...

x(6)\*ones(4,1); ...

x(7)\*ones(4,1); ...

x(8)\*ones(4,1)]';

D = 0.1; % density lb/in3

weight = sum(D\*A.\*Le);

end

***main.m:***

% -- 25-bar truss optimization

% -- Units: in-lb-s-lbf-psi

clear

close

clc

E=1e7;

x0 = 1.0\*ones(1,8);

lb = 0.1\*ones(1,8);

ub = 5.0\*ones(1,8);

options = optimoptions('fmincon', 'Display', 'iter', 'Algorithm', 'active-set');

[x, fval, exitflag] = fmincon('get\_obj', x0, [], [], [], [], lb, ub, @(x) get\_cns(x,E), options);

***hw4\_2:***

% -- 25-bar truss optimization

% -- Units: in-lb-s-lbf-psi

clear

close

clc

E=1e7;

x0 = 1.0\*ones(1,8);

lb = 0.1\*ones(1,8);

ub = 5.0\*ones(1,8);

options = optimoptions('fmincon', 'Display', 'iter', 'Algorithm', 'active-set');

[x, fval, exitflag] = fmincon('get\_obj', x0, [], [], [], [], lb, ub, @(x) get\_cns(x,E), options);

mux=x; % you should change to the optimal design you obtained

stdx=0.0052\*ones(1,8); % you should change this value according to the homework descriptions

covX=diag(stdx.^2);

std\_E=1e6;

cov\_E=std\_E.^2;

% Basic MCS

N=1e6;

RandX=mvnrnd(mux, covX, N);

RandE=mvnrnd(E,cov\_E,N);

g=zeros(N,27);

for ii=1:N

[c,ceq]=get\_cns(RandX(ii,:),RandE(ii));

g(ii,:)=c;

end

Nf=zeros(1,27);

for ii=1:27

Nf(ii)=sum(g(:,ii)>0);

end

pf=Nf/N;

for ii=1:27

sprintf('Failure probability using MCS with number %d constraint is %0.5g percent ', ii, pf(ii)\*100)

end

bar(pf\*100);

xlabel ConstraintNumber

ylabel ViolatingPercentage

ax=gca;

ax.YLim=[-10,100];