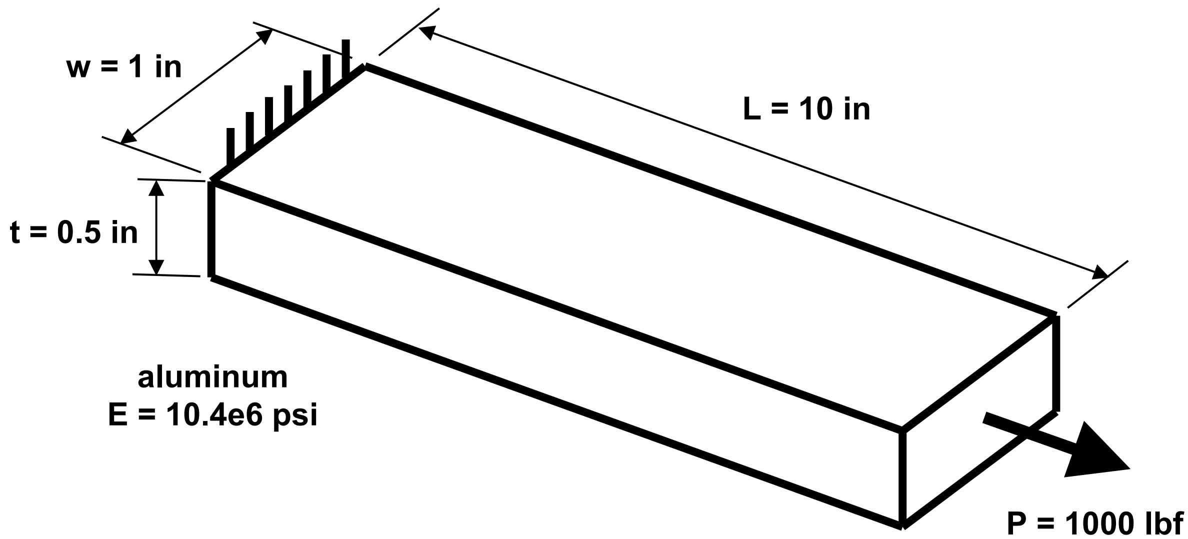
Finite Element Analysis – One Dimensional Example

straight cantilever beam with constant thickness in pure tension

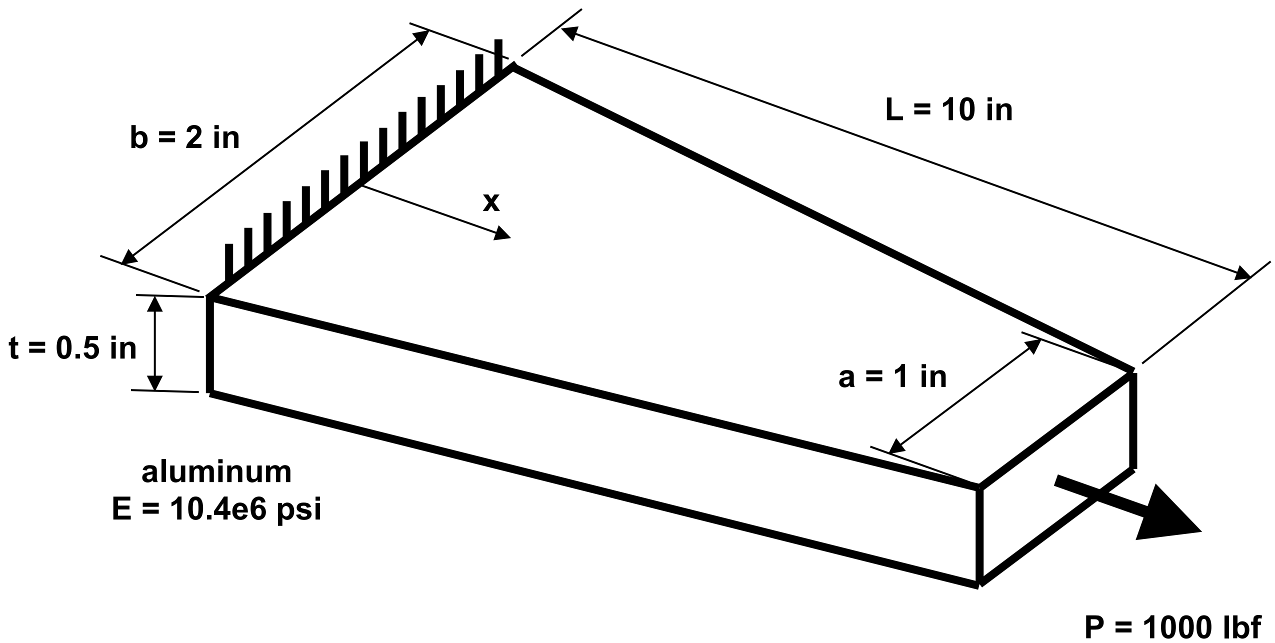


$$\delta = \frac{P\,L}{A\,E} = \frac{P\,L}{t\,w\,E} = \frac{(1000\text{ lbf})(10\text{ in})}{(0.5\text{ in})(1\text{ in})}\left(\frac{\text{in}^2}{10.4\text{x}10^6\text{ lbf}}\right) = 0.001923\text{ in} = 1.923\text{ mil}$$

$$P = k\,\delta \qquad k = \frac{t\,w\,E}{L}$$

$$\sigma = \frac{P}{A} = \frac{P}{t\,w} = \frac{(1000\text{ lbf})}{(0.5\text{ in})(1\text{ in})} = 2.0\text{ ksi}$$

tapered cantilever beam with constant thickness in pure tension



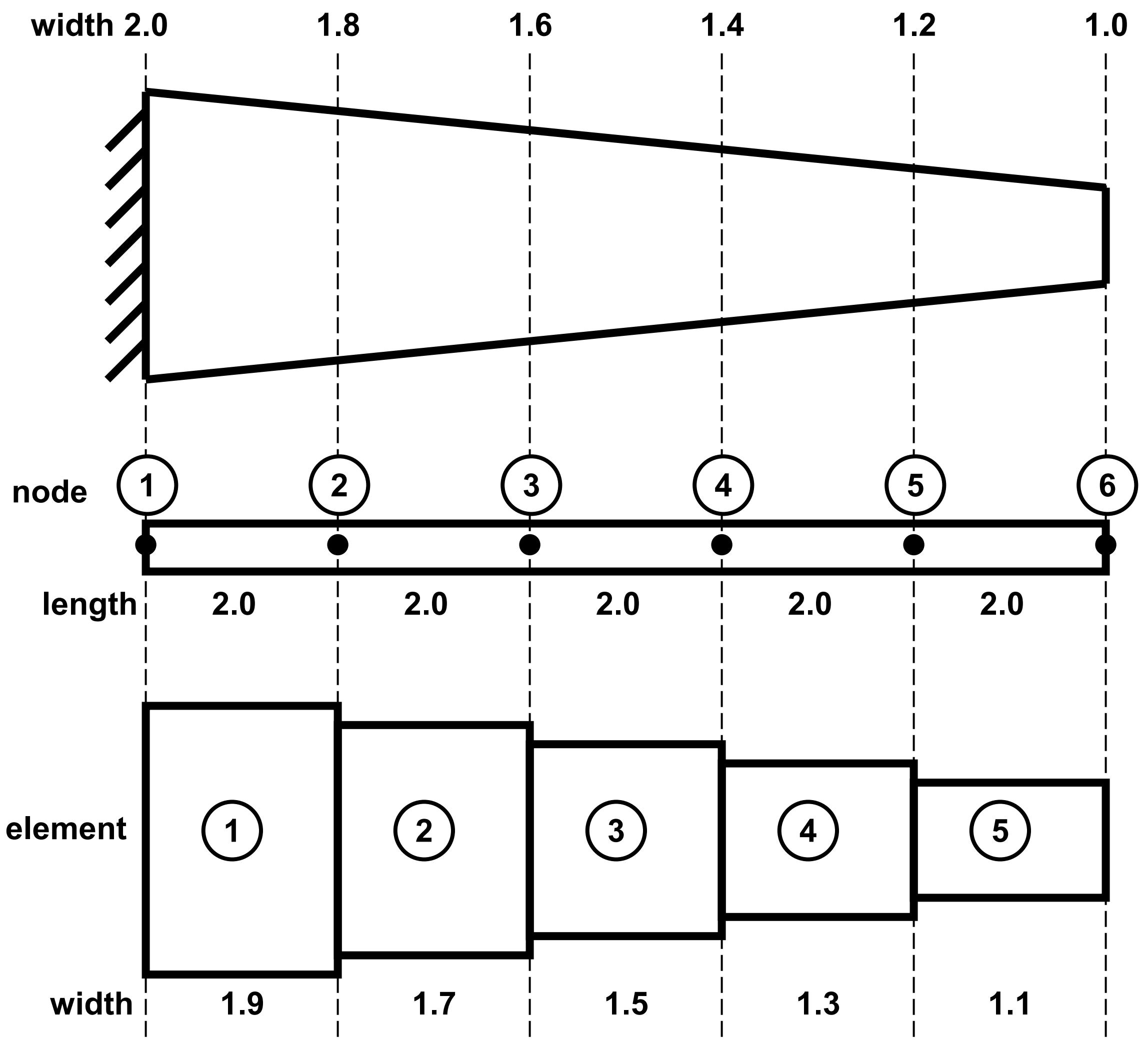
$$w = b - \frac{(b-a)}{L} x = b - m\,x \qquad m = \frac{(b-a)}{L} \qquad 0 \le x \le L$$

$$\sigma = \frac{P}{A} = \frac{P}{t\,w} = \varepsilon\,E \qquad \varepsilon = \frac{P}{E\,t\,w}$$

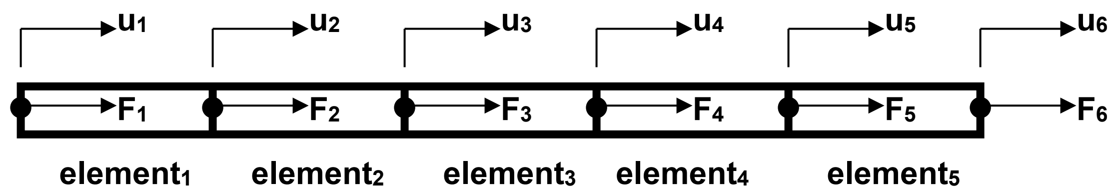
$$\delta = \int_0^L \varepsilon\,dx = \int_0^L \left(\frac{P}{E\,t\,w}\right) dx = \frac{P}{E\,t}\int_0^L \left(\frac{1}{b-m\,x}\right) dx = \frac{P}{E\,t}\left(-\frac{1}{m}\ln(b-m\,x)\right)\Bigg|_0^L = \frac{P\,L}{E\,t\,(b-a)}\ln\left(\frac{b}{a}\right)$$

$$\delta = \frac{(1000\text{ lbf})(10\text{ in})}{(0.5\text{ in})(2\text{ in}-1\text{ in})}\left(\frac{\text{in}^2}{10.4\text{x}10^6\text{ lbf}}\right)\ln\left(\frac{2\text{ in}}{1\text{ in}}\right) = 1.333\text{ mil}$$ (**NOT** 1.923 / 1.5 = 1.282 mil)

five elements – six nodes



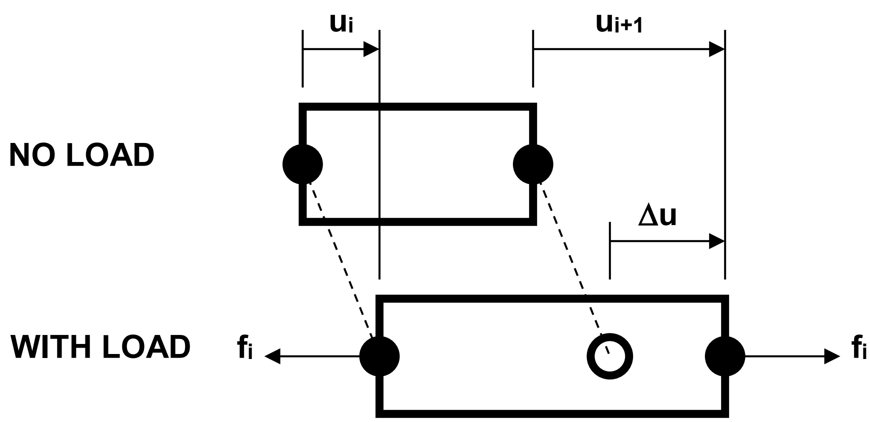
five elements – six nodes



u_i = displacement of node i
F_i = external force applied at node i

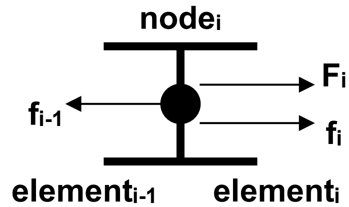
l_i = length of element i
A_i = cross sectional area of element i
t_i = thickness of element i
w_i = width of element i
E_i = modulus for element i

k_i = stiffness of element i
f_i = internal force on element i



$$\Delta u = \frac{f_i\ l_i}{A_i\ E_i} = \frac{f_i\ l_i}{t_i\ w_i\ E_i} \qquad f_i = \frac{t_i\ w_i\ E_i}{l_i}(\Delta u) = k\,(\Delta u) = k_i\,(u_{i+1} - u_i) \qquad k_i = \frac{t_i\ w_i\ E_i}{l_i}$$

$$f_i = k_i\,(u_{i+1} - u_i)$$



ΣF on node 1	$F_1 + f_1 + F_{WALL} = 0$	$F_1 = k_1 u_1 - k_1 u_2 - F_{WALL}$
ΣF on node 2	$F_2 + f_2 - f_1 = 0$	$F_2 = -k_1 u_1 + (k_1 + k_2) u_2 - k_2 u_3$
ΣF on node 3	$F_3 + f_3 - f_2 = 0$	$F_3 = -k_2 u_2 + (k_2 + k_3) u_3 - k_3 u_4$
ΣF on node 4	$F_4 + f_4 - f_3 = 0$	$F_4 = -k_3 u_3 + (k_3 + k_4) u_4 - k_4 u_5$
ΣF on node 5	$F_5 + f_5 - f_4 = 0$	$F_5 = -k_4 u_4 + (k_4 + k_5) u_5 - k_5 u_6$
ΣF on node 6	$F_6 - f_5 = 0$	$F_6 = -k_5 u_5 + k_5 u_6$

$$\begin{bmatrix} +k_1 & -k_1 & 0 & 0 & 0 & 0 \\ -k_1 & k_1+k_2 & -k_2 & 0 & 0 & 0 \\ 0 & -k_2 & k_2+k_3 & -k_3 & 0 & 0 \\ 0 & 0 & -k_3 & k_3+k_4 & -k_4 & 0 \\ 0 & 0 & 0 & -k_4 & k_4+k_5 & -k_5 \\ 0 & 0 & 0 & 0 & -k_5 & +k_5 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{Bmatrix} = \begin{Bmatrix} F_1 + F_{WALL} \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{Bmatrix}$$

impose restraints $u_1 = 0$ and $F_1 = 0$ in first row of equations

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -k_1 & k_1+k_2 & -k_2 & 0 & 0 & 0 \\ 0 & -k_2 & k_2+k_3 & -k_3 & 0 & 0 \\ 0 & 0 & -k_3 & k_3+k_4 & -k_4 & 0 \\ 0 & 0 & 0 & -k_4 & k_4+k_5 & -k_5 \\ 0 & 0 & 0 & 0 & -k_5 & +k_5 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{Bmatrix} = \begin{Bmatrix} 0 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{Bmatrix} \qquad [K]\{u\} = \{F\}$$

[K] = system stiffness matrix

{u} = vector of nodal displacements

{F} = vector of external forces on nodes

displacement analysis - solve for $\{u\} = [K]^{-1}\{F\}$

stress analysis

ε_i = strain in element i
σ_i = stress in element i

$$\varepsilon_i = \frac{u_{i+1} - u_i}{l_i} \qquad \sigma_i = E_i \ \varepsilon_i = \frac{E_i}{l_i}\left(u_{i+1} - u_i\right)$$

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \end{Bmatrix} = \begin{bmatrix} E_1/l_1 & 0 & 0 & 0 & 0 \\ 0 & E_2/l_2 & 0 & 0 & 0 \\ 0 & 0 & E_3/l_3 & 0 & 0 \\ 0 & 0 & 0 & E_4/l_4 & 0 \\ 0 & 0 & 0 & 0 & E_5/l_5 \end{bmatrix} \begin{Bmatrix} u_2 - u_1 \\ u_3 - u_2 \\ u_4 - u_3 \\ u_5 - u_4 \\ u_6 - u_5 \end{Bmatrix} \qquad \{\sigma\} = [C]\{\Delta u\}$$

$\{\sigma\}$ = vector of stresses on each element

[C] = elasticity matrix

$\{\Delta u\}$ = vector of differences in nodal displacements

SUMMARY FOR THREE-DIMENSIONAL ANALYSIS

1) Know material properties – E, G, ν (Poisson's ratio)

2) Generate mesh – create elements and nodes
   reduce mesh size by factor of 2, increases number of nodes by factor of 8

3) Compute stiffness matrix – order n = number of degrees of freedom (DOF)
   n = 3 * number of nodes

4) Define restraints – modify stiffness matrix to force specific nodal displacements to zero

5) Define loading – form right-hand-side (RHS) vector

6) Invert stiffness matrix and pre-multiply loading RHS vector to solve for nodal displacements
   computation time proportional to n^2

7) Compute nodal deformations RHS vector (differences of nodal displacements)

8) Compute elasticity matrix

9) Pre-multiply elasticity matrix times deformation RHS vector to solve for stresses at nodes
   find von Mises stress, maximum shear stress, normal stress, Dowling stress

10) Know strength of material – S_Y, S_{UT}, S_{UC}

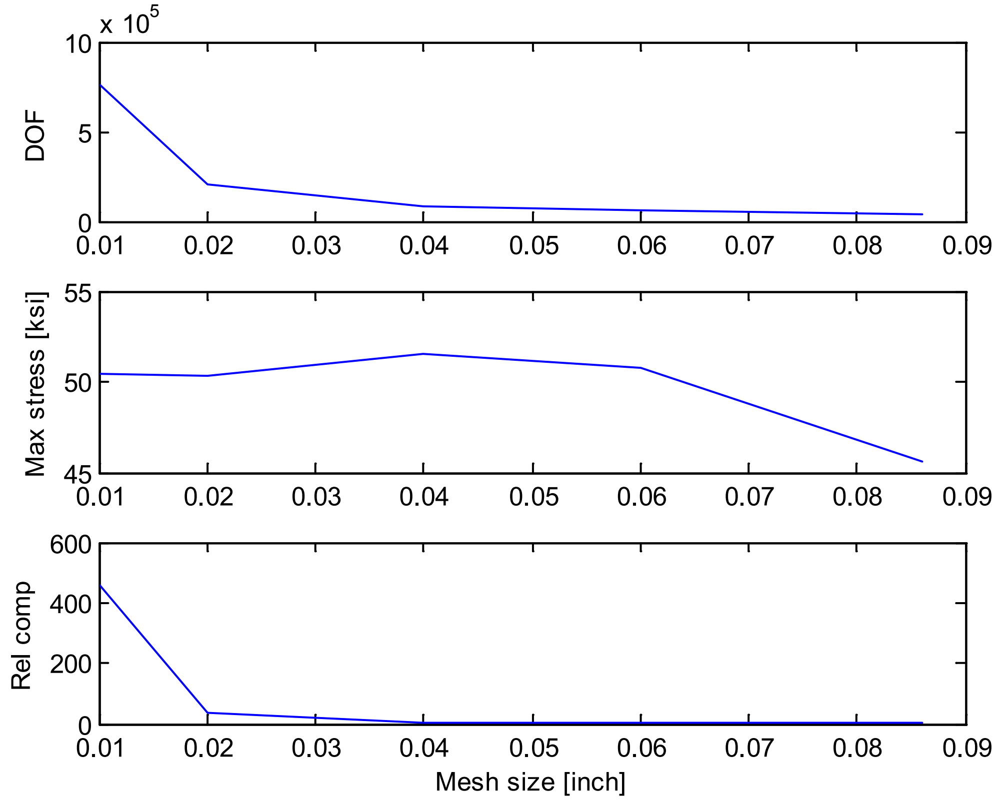
11) Compute factor of safety at each node

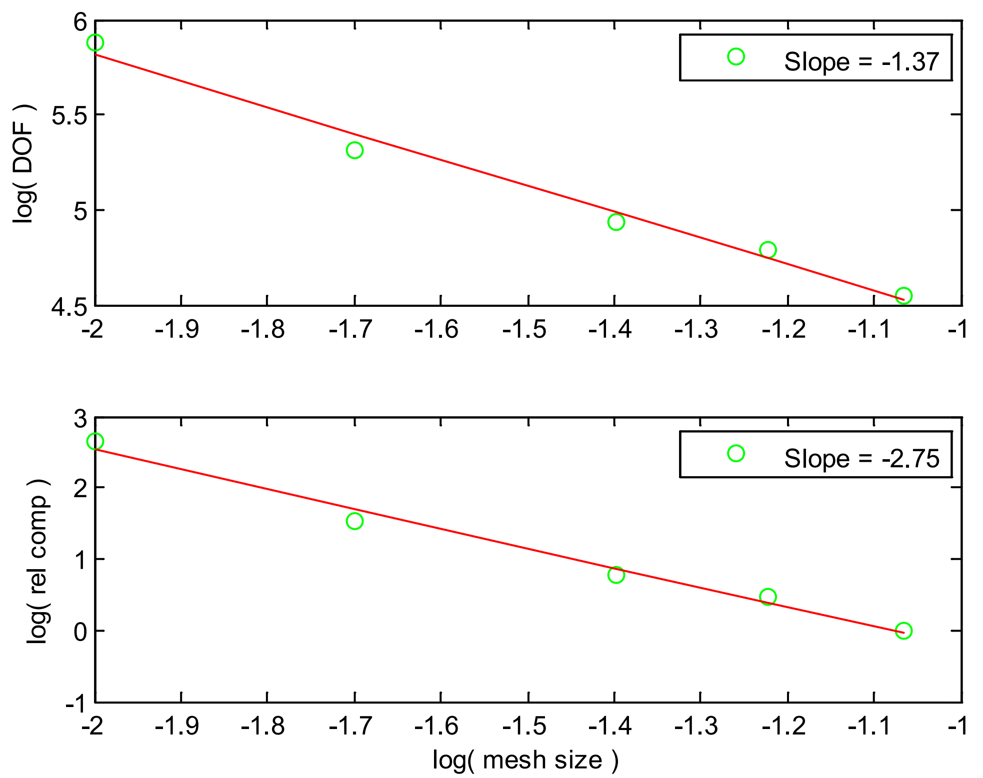
```
% fea_1d.m - one dimensional finite element analysis
% HJSIII, 12.09.05

% axial load, geometry, material
P = 1000;      % axial load [lbf]
L = 10.0;      % length [inch]
t = 0.5;       % thickness [inch]
E = 10.4e6;    % Young's modulus for aluminum [psi]

% analytical solution - constant width
w = 1.0;       % width [inch]
delta_const_w_mil = P *L /w /t /E *1000;    % [mils]

% analytical solution - tapered
a = 1.0;       % width at tip [inch]
b = 2.0;       % width at root [inch]
delta_taper_mil = P *L /t /E /(b-a) *log(b/a) *1000;    % [mils]

% element stiffness - tapered
n = 5;                                 % number of finite elements
li = L/n * ones(n,1);                  % length [inch]
ti = t * ones(n,1);                    % thickness [inch]
Ei = E * ones(n,1);                    % Young's modulus for aluminum [psi]
wi = [ 1.9  1.7  1.5  1.3  1.1 ]';     % width [inch]

k = wi .*ti .*Ei ./li;

% global stiffness
K = [ +1      0           0           0           0           0    ;
      -k(1)  +k(1)+k(2)  -k(2)        0           0           0    ;
       0     -k(2)       +k(2)+k(3)  -k(3)        0           0    ;
       0      0          -k(3)       +k(3)+k(4)  -k(4)        0    ;
       0      0           0          -k(4)       +k(4)+k(5)  -k(5) ;
       0      0           0           0          -k(5)       +k(5) ];

% external forces
F = [ 0  0  0  0  0  P ]';

% displacements
u = inv(K) * F;
delta_fea_mil = u(6) *1000;  % [mils]

delta_const_w_mil
delta_taper_mil
delta_fea_mil

% stress
C = diag( Ei ./li );
strain = diff( u );
stress = C * strain;
stress_ksi = stress /1000
```

```
>> fea_1d

delta_const_w_mil =

    1.9231


delta_taper_mil =

    1.3330


delta_fea_mil =

    1.3306


stress_ksi =

    1.0526
    1.1765
    1.3333
    1.5385
    1.8182
```

FEA sensitivity study for tee_01.sldprt

mesh control size parameter [inch]	max von Mises stress [ksi]	DOF	relative computations
0.086	45.59	35496	1
0.06	50.80	61335	2.99
0.04	51.55	87162	6.03
0.02	50.38	205329	33.46
0.01	50.42	762018	460.86


x 10^5
DOF
Max stress [ksi]
Rel comp
Mesh size [inch]


Slope = -1.37
log(DOF)
Slope = -2.75
log(rel comp)
log(mesh size)

```
% tee_01.m - plot results for SW FEA sensitivity study
% HJSIII, 13.09.16

clear

% data
%        mesh size   DOF  max von Mises
%        [inch]           [ksi]
sen = [  0.086     35496  45.59 ;
         0.06      61335  50.80 ;
         0.04      87162  51.55 ;
         0.02     205329  50.38 ;
         0.01     762018  50.42 ];
ms  = sen(:,1);
dof = sen(:,2);
vm  = sen(:,3);
flops = dof / dof(1);
flops = flops .* flops;

% plot raw data
figure( 1 )
clf
  subplot( 3, 1, 1 )
  plot( ms, dof )
  ylabel( 'DOF' )

  subplot( 3, 1, 2 )
  plot( ms, vm )
  ylabel( 'max stress [ksi]' )

  subplot( 3, 1, 3 )
  plot( ms, flops )
  xlabel( 'Mesh size [inch]' )
  ylabel( 'Rel comp' )

% plot log-log
lms = log10( ms );
ldof = log10( dof );
lflops = log10( flops );

pdof = polyfit( lms, ldof, 1 );
pflops = polyfit( lms, lflops, 1 );

ldof_fit = polyval( pdof, lms );
lflops_fit = polyval( pflops, lms );

figure( 2 )
clf
  subplot( 2, 1, 1 )
  plot( lms,ldof,'go', lms,ldof_fit,'r' )
  ylabel( 'log( dof )' )
  legend( [ 'Slope = ' num2str(pdof(1),3) ] )

  subplot( 2, 1, 2 )
  plot( lms,lflops,'go', lms,lflops_fit,'r' )
  xlabel( 'log( mesh size )' )
  ylabel( 'log( rel comp )' )
  legend( [ 'Slope = ' num2str(pflops(1),3) ] )
```