Basic FEM for trusses (Day 1)

Ole Sigmund and Kim Lau Nielsen

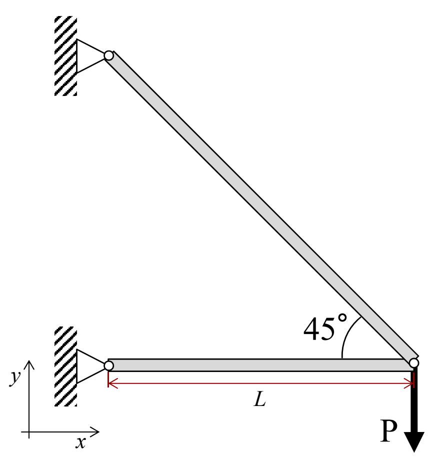
Department of Civil and Mechanical Engineering, Solid Mechanics

Preparatory reading:

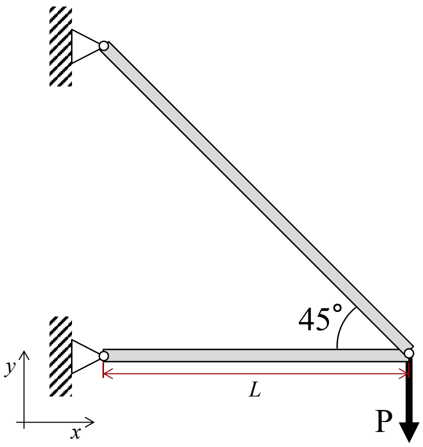
Handout Day 1

Chapter 1 in Cook et al.

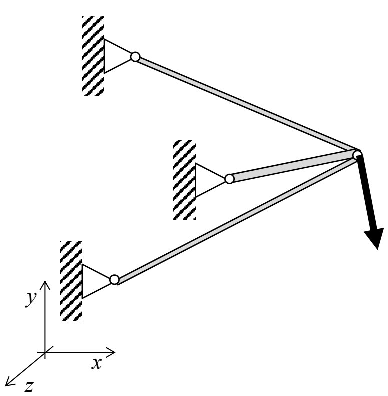
Simple truss structures (brush-up)



Simple truss structures (brush-up)



Simple truss structures (brush-up)



Virtual Work Principle

$$\boxed{\int_V \{\delta\varepsilon\}^T \{\sigma\}\, dV = \int_S \{\delta\mathbf{u}\}^T \{F\}\, dS + \int_V \{\delta\mathbf{u}\}^T \{\Phi\}\, dV + \sum_i \{\delta\mathbf{u}\}_i^T \{p\}_i,}$$

which has to hold for any kinematically admissible displacement variations $\{\delta\mathbf{u}\}$, associated strain variations $\{\delta\varepsilon\}$ and nodal displacement variations $\{\delta D\}$.

Or

$$\boxed{\delta\Omega = \delta W}$$

where

V is structural volume
S is surface area
$\{F\}$ is surface traction
$\{\Phi\}$ is body force
$\{\delta\mathbf{u}\}$ is virtual displacement variation
$\{\sigma\}$ is the stress vector
$\{\delta\varepsilon\}$ is the virtual strain variation vector
$\{p\}_i$ is a concentrated (nodal) load vector

Definition:

$$\boxed{\delta\Omega \equiv \frac{\partial\Omega}{\partial D_i}\delta D_i}$$

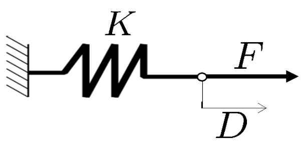
Virtual Work Principle (in words)

Or in words:

Arbitrary virtual displacement variations must lead to identical variations in internal and external work

Basically, it is and alternative way of expressing structural equilibrium (energy balance)

VWP – 1 dof spring



$$(KD = F)$$

Elastic (strain) energy:

$$\Omega = \tfrac{1}{2}KD^2$$

Work of external forces:

$$W = FD$$

$$\delta\Omega = \delta W \quad \forall \delta D (\neq 0)$$

$$\boxed{\delta\Omega \equiv \frac{\partial \Omega}{\partial D_i}\delta D_i}$$

VWP – 1 dof spring

Principle of virtual work (VWP):

$$\delta\Omega = \delta W \quad \forall \delta D (\neq 0)$$

variation of strain energy *variation of external work*

Variation of strain energy:

$$\delta\Omega = \delta\left(\tfrac{1}{2}KD^2\right) = KD\ \delta\mathrm{D}$$

Variation of external work:

$$\delta\mathrm{W} = \delta(FD) = F\ \delta\mathrm{D}$$

Insert in VWP

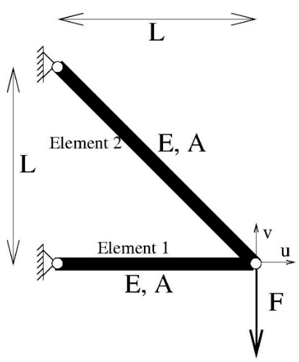
$$KD\delta D = F\delta D\,, \qquad \forall \delta D\ (\neq 0)$$

$$\Downarrow$$

$$KD = F$$

Virtual Work Principle (truss analysis)

$$\boxed{\int_V \{\delta\varepsilon\}^T \{\sigma\}\, dV = \int_S \{\delta\mathbf{u}\}^T \{F\}\, dS + \int_V \{\delta\mathbf{u}\}^T \{\Phi\}\, dV + \sum_i \{\delta\mathbf{u}\}_i^T \{p\}_i,}$$



Simplifications:

LHS: Uniaxial and constant stress/strain in bar elements

$$\{\delta\varepsilon\}^T\{\sigma\} \rightarrow \delta\varepsilon\ \sigma \text{ and } \int_V \rightarrow \sum_e \int_{V_e} \rightarrow \sum_e A^e L_0^e$$

RHS: No distributed forces

$$\{F\} = \{0\}$$

RHS: No volume loads

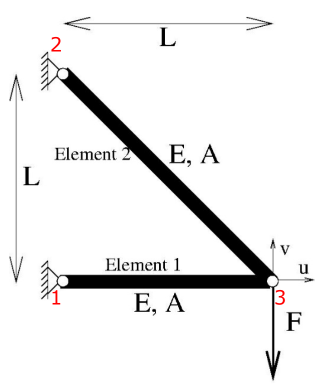
$$\{\Phi\} = \{0\}$$

RHS: Only nodal loads

$$\sum_i \{\delta\mathbf{u}\}_i^T \{p\}_i = \{\delta D\}^T\{P\}$$

Virtual Work Principle (truss analysis)

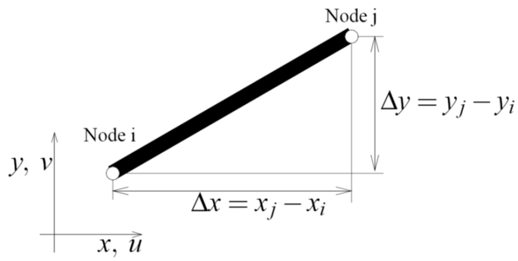
$$\boxed{\sum_e A^e L_0^e \delta\varepsilon\ \sigma = \{\delta D\}^T\{P\}} \quad \forall\{\delta D\} \neq \{0\}$$



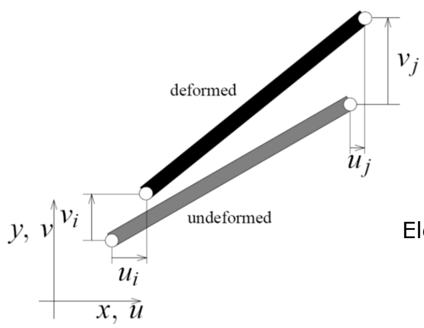
$$\{D\} = \{u_1\ v_1\ u_2\ v_2\ \dots\ u_N\ v_N\}^T$$
$$\{d^1\} = \{u_1\ v_1\ u_3\ v_3\}^T$$
$$\{d^2\} = \{u_2\ v_2\ u_3\ v_3\}^T$$

Truss element

Element geometry:



Deformed element:



$$\Delta u = u_j - u_i$$
$$\Delta v = v_j - v_i$$

Element displacement vector

$$\{d^e\} = \{u_i \ v_i \ u_j \ v_j\}^T$$

Truss element

Initial and deformed length:

$$L_0 = \sqrt{\Delta x^2 + \Delta y^2}$$

$$\begin{aligned} L_1 &= \sqrt{(\Delta x + \Delta u)^2 + (\Delta y + \Delta v)^2} \\ &= \sqrt{\Delta x^2 + \Delta y^2 + \Delta u^2 + \Delta v^2 + 2(\Delta x \Delta u + \Delta y \Delta v)} \\ &\approx L_0 + \frac{\Delta x \Delta u + \Delta y \Delta v}{L_0} \end{aligned}$$

From 1st order Taylor-expansion assuming small displacements ($\Delta u \ll \Delta x$)

Elongation:

$$\Delta L = L_1 - L_0 = \frac{\Delta x \Delta u + \Delta y \Delta v}{L_0}$$

Truss element

Relative length change, i.e. strain:

$$\varepsilon = \frac{L_1 - L_0}{L_0} = \frac{\Delta x \Delta u + \Delta y \Delta v}{L_0^2}$$

Engineering/Cauchy strain - not the only choice!

Element displacement vector:

$$\{d^e\} = \{u_i \; v_i \; u_j \; v_j\}^T$$

$$\Rightarrow \qquad \varepsilon = \frac{1}{L_0^2}\{-\Delta x \;\; -\Delta y \;\; \Delta x \;\; \Delta y\}\{d\} = \{B_0\}^T\{d\}$$

Definition of linear strain displacement vector:

$$\{B_0\} = \frac{1}{L_0^2}\{-\Delta x \;\; -\Delta y \;\; \Delta x \;\; \Delta y\}^T$$

Relates only to the initial geometry (NB! Not valid on Day 3!!)

VWP – truss analysis

$$\boxed{\sum_e A^e L_0^e \delta\varepsilon \; \sigma = \{\delta D\}^T\{P\}}$$

Strain-displacement relation

$$\delta\varepsilon = \{B_0\}^T\{\delta d\}$$

Hookes' law (one dimension):

$$\sigma = E\varepsilon = E\{B_0\}^T\{d\}$$

Insert in LHS of VWP:

$$\begin{aligned}\sum_e A^e L_0^e \delta\varepsilon \; \sigma &= \sum_e A^e L_0^e \{B_0\}^T\{\delta d\} E^e \{B_0\}^T\{d\} \\ &= \sum_e \{\delta d\}^T \left[E^e A^e L_0^e \{B_0\}\{B_0\}^T\right]\{d\} \\ &= \sum_e \{\delta d\}^T [k^e]\{d\} = \{\delta D\}^T \left(\sum_e [k^e]\right)\{D\} \\ &= \{\delta D\}^T [K]\{D\}\end{aligned}$$

VWP – truss analysis

$$\boxed{\sum_e A^e L_0^e \delta\varepsilon\ \sigma = \{\delta D\}^T \{P\}}$$

VWP in discretized form:

$$\{\delta D\}^T [K]\{D\} = \{\delta D\}^T \{P\}, \quad \forall \{\delta D\}$$
$$\Downarrow$$
$$\boxed{[K]\{D\} = \{P\}}$$

Local and global stiffness matrices

$$[k^e] = E^e A^e L_0^e \{B_0\}\{B_0\}^T$$

$$[K] = \sum_e [k^e]$$

Element stiffness matrix

Element stiffness matrix for bar element:

$$[k^e] = E^e A^e L_0^e \{B_0\}\{B_0\}^T$$

$$= \frac{E^e A^e}{(L_0^e)^3} \begin{bmatrix} \Delta x^2 & \Delta x \Delta y & -\Delta x^2 & -\Delta x \Delta y \\ \Delta x \Delta y & \Delta y^2 & -\Delta x \Delta y & -\Delta y^2 \\ -\Delta x^2 & -\Delta x \Delta y & \Delta x^2 & \Delta x \Delta y \\ -\Delta x \Delta y & -\Delta y^2 & \Delta x \Delta y & \Delta y^2 \end{bmatrix}$$

In code use short expression for clarity and compactness!

Global stiffness matrix

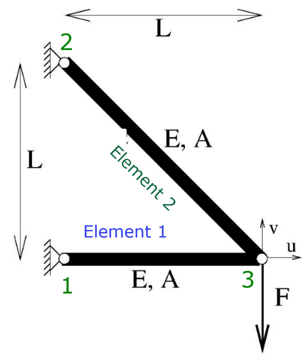
Assembly of element matrices:

$$[k^1] = \begin{bmatrix} k^1_{11} & k^1_{12} & k^1_{13} & k^1_{14} \\ k^1_{21} & k^1_{22} & k^1_{23} & k^1_{24} \\ k^1_{31} & k^1_{32} & k^1_{33} & k^1_{34} \\ k^1_{41} & k^1_{42} & k^1_{43} & k^1_{44} \end{bmatrix}$$

$$[k^2] = \begin{bmatrix} k^2_{11} & k^2_{12} & k^2_{13} & k^2_{14} \\ k^2_{21} & k^2_{22} & k^2_{23} & k^2_{24} \\ k^2_{31} & k^2_{32} & k^2_{33} & k^2_{34} \\ k^2_{41} & k^2_{42} & k^2_{43} & k^2_{44} \end{bmatrix}$$

$$[K] = \sum_e [k^e] = [k^1] + [k^2] =$$

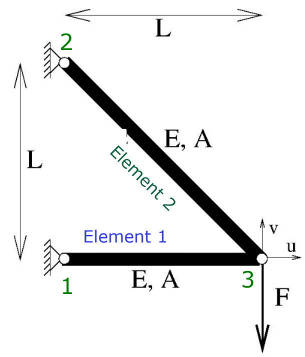
$$\begin{bmatrix} k^1_{11} & k^1_{12} & 0 & 0 & k^1_{13} & k^1_{14} \\ k^1_{21} & k^1_{22} & 0 & 0 & k^1_{23} & k^1_{24} \\ 0 & 0 & k^2_{11} & k^2_{12} & k^2_{13} & k^2_{14} \\ 0 & 0 & k^2_{21} & k^2_{22} & k^2_{23} & k^2_{24} \\ k^1_{31} & k^1_{32} & k^2_{31} & k^2_{32} & k^1_{33} + k^2_{33} & k^1_{34} + k^2_{34} \\ k^1_{41} & k^1_{42} & k^2_{41} & k^2_{42} & k^1_{43} + k^2_{43} & k^1_{44} + k^2_{44} \end{bmatrix}$$



Imposing Boundary conditions

Solve FE problem: $\boxed{[K]\{D\} = \{P\}}$

- Elimination of equations
- **Method of 0's and 1's**
- **Penalty method**
- Lagrangian multipliers



Enforcing BC's with method of 0's and 1's

Enforce $D_2 = 0$

$$\begin{bmatrix} k^1_{11} & 0 & 0 & 0 & k^1_{13} & k^1_{14} \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & k^2_{11} & k^2_{12} & k^2_{13} & k^2_{14} \\ 0 & 0 & k^2_{21} & k^2_{22} & k^2_{23} & k^2_{24} \\ k^1_{31} & 0 & k^2_{31} & k^2_{32} & k^1_{33}+k^2_{33} & k^1_{34}+k^2_{34} \\ k^1_{41} & 0 & k^2_{41} & k^2_{42} & k^1_{43}+k^2_{43} & k^1_{44}+k^2_{44} \end{bmatrix} \begin{Bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \\ D_6 \end{Bmatrix} = \begin{Bmatrix} P_1 \\ 0 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{Bmatrix}$$

Enforce $D_1 = D_2 = D_3 = D_4 = 0$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & k^1_{33}+k^2_{33} & k^1_{34}+k^2_{34} \\ 0 & 0 & 0 & 0 & k^1_{43}+k^2_{43} & k^1_{44}+k^2_{44} \end{bmatrix} \begin{Bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \\ D_6 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ P_5 \\ P_6 \end{Bmatrix}$$

Can be extended to $D_i \neq 0$ (see Cook 13.2)

Enforcing BC's with penalty method

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} & K_{15} & K_{16} \\ K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & K_{26} \\ K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & K_{36} \\ K_{41} & K_{42} & K_{43} & K_{44} & K_{45} & K_{46} \\ K_{51} & K_{52} & K_{53} & K_{54} & K_{55} & K_{56} \\ K_{61} & K_{62} & K_{63} & K_{64} & K_{65} & K_{66} \end{bmatrix} \begin{Bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \\ D_6 \end{Bmatrix} = \begin{Bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{Bmatrix}$$

Enforce $D_1 = D_4 = 0$

$$\begin{bmatrix} K_{11}+\infty & K_{12} & K_{13} & K_{14} & K_{15} & K_{16} \\ K_{21} & K_{22} & K_{23} & K_{24} & K_{25} & K_{26} \\ K_{31} & K_{32} & K_{33} & K_{34} & K_{35} & K_{36} \\ K_{41} & K_{42} & K_{43} & K_{44}+\infty & K_{45} & K_{46} \\ K_{51} & K_{52} & K_{53} & K_{54} & K_{55} & K_{56} \\ K_{61} & K_{62} & K_{63} & K_{64} & K_{65} & K_{66} \end{bmatrix} \begin{Bmatrix} D_1 \\ D_2 \\ D_3 \\ D_4 \\ D_5 \\ D_6 \end{Bmatrix} = \begin{Bmatrix} P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \\ P_6 \end{Bmatrix}$$

In pratice: ∞ = large number (e.g. 10^8 times max(K_{ii}))

Easy to extend to $D_i \neq 0$

Solving the linear equations

Getting displacements:

$$[K]_{corr}\{D\} = \{P\}_{corr}$$

Solution methods:

- Gauss-elimination using factorization
- Iterative methods

Fast methods:

- Cholesky factorization (Matlab: D=K\P)

Derived quantities

Element strain: $\varepsilon = \{B_0\}^T\{d\}$

Element stress: $\sigma = E\varepsilon$

Element forces: $N = \sigma A$

Reaction forces:

Uncorrected !

$$\{R\}_{reaction} = [K]\{D\} - \{P\} = \sum_e \{B_0\}N^e L_0^e - \{P\}$$

Cheaper !

General FEM procedure

0) Discretize domain into a finite # of elements
   (many ways – the more elements the better)

1) Node and element numbering

2) Build load vector $\{P\}$

3) Calculate $[k^e]$ and assemble $[K]$

4) Enforce BC's

5) Solve for $\{D\}$

6) Compute ε, σ, N …
   (interpolation of solutions between nodes) -> u

7) Plot and evaluate results

Today

- Download matlab template + example files from DTU Learn
- Add basic FEM procedure to fea.m – remember always to keep copies of things that work!
- Work on exercises 1.1 and 1.2 (see them as guidelines for what kind of questions you will get in the first assignment).
- Do competition 1 (obligatory, hand-in Monday midnight!)
- Work on *-exercises 1.3-1.6 (voluntary)

Being able to solve standard problems defined in 1.1-1.2 for complex and realistic structures and report and discuss the results perfectly is sufficient to get top grades

*-exercises are for fun, further insight and may help in grade decision

Exercises

Edit the example1.m file such that you can model the 9-bar truss structure shown in Figure 1.3.

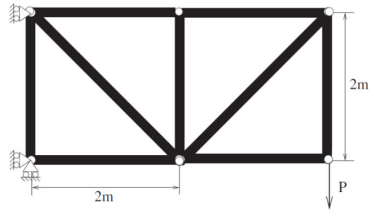


Figure 1.3: A 9-bar truss structure.

Modify and expand the subroutines in "fea" to *find displacements, strains and stresses for the truss structures in Fig. 1.2 and Fig. 1.3.*
Steps in the program development:

▪ ▪ ▪

Check the results obtained by running the example from Fig. 1.3 with a simple hand-calculation for support reactions and element forces (calculate analytical element forces in at least 3 elements). **Hints:** In the FE-program, the support reactions may be found by calculating the residual (1.10) without inserting boundary conditions.

Exercises

Exercise 1.2

Use the interactive preprocessor program "FlExtract" (download it from the course web-page) to generate a large, more complex truss structure. Draw elements in tension with blue, elements in compression with red and non-loaded elements with green.

Handout Code

```
function fea()

close all
clc

%--- Input file ----------------------------------------------------------%
example1                 % Input file
%test1                   % Input file

neqn = size(X,1)*size(X,2);          % Number of equations
ne = size(IX,1);                     % Number of elements
disp(['Number of DOF ' sprintf('%d',neqn) ...
    ' Number of elements ' sprintf('%d',ne)]);

%--- Initialize arrays ---------------------------------------------------%
Kmatr=zeros(neqn,neqn);                      % Stiffness matrix
P=zeros(neqn,1);                         % Force vector
D=zeros(neqn,1);                         % Displacement vector
R=zeros(neqn,1);                         % Residual vector
strain=zeros(ne,1);                      % Element strain vector
stress=zeros(ne,1);                      % Element stress vector

%--- Calculate displacements ---------------------------------------------%
[P]=buildload(X,IX,ne,P,loads,mprop);       % Build global load vector

[Kmatr]=buildstiff(X,IX,ne,mprop,Kmatr);    % Build global stiffness matrix

[Kmatr,P]=enforce(Kmatr,P,bound);           % Enforce boundary conditions

                                            % Solve system of equations

[strain,stress]=recover(mprop,X,IX,D,ne,strain,stress); % Calculate element
                                                        % stress and strain

%--- Plot results --------------------------------------------------------%
PlotStructure(X,IX,ne,neqn,bound,loads,D,stress)        % Plot structure

return
```

0) Discretize domain into a finite # of elements
   (many ways – the more elements the better)

1) Node and element numbering

2) Build load vector {*P*}

3) Calculate [k^e] and assemble [*K*]

4) Enforce BC's

5) Solve for {*D*}

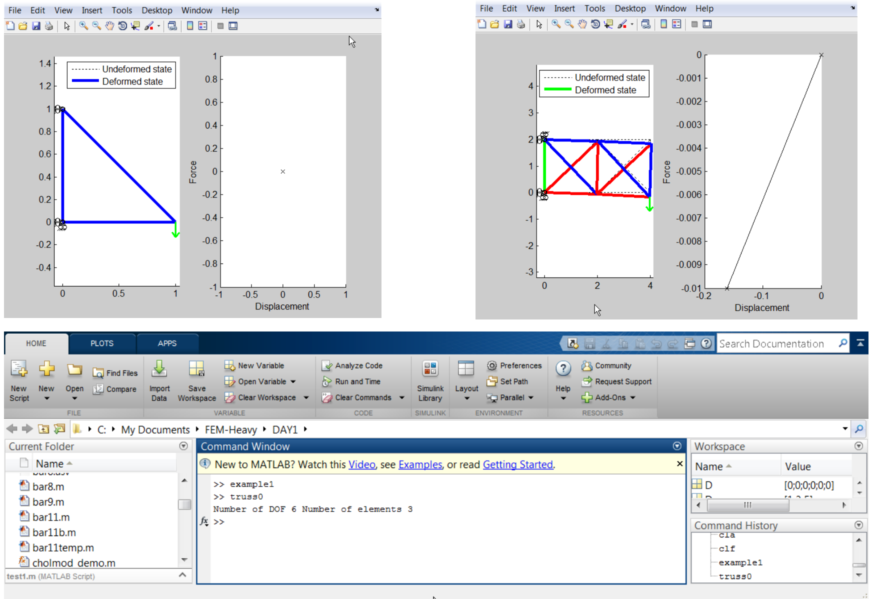
6) Compute ε, σ, *N* …
   (interpolation of solutions between nodes) -> *u*

7) Plot and evaluate results

Initial input file

```
% File example1.m, DAY1, modified 3/9 by OS
%
% Example: 3-bar truss
% No. of Nodes: 3
% No. of Elements : 3
clear all

% Coordinates of 3 nodes,
X = [  0.00  0.00
       0.00  1.00
       1.00  0.00 ];

% Topology matrix IX(node1,node2,propno),
IX = [ 1  3  1
       1  2  1
       2  3  1 ];

% Element property matrix mprop = [ E A ],
mprop = [ 1.0 1.0 ];

% Prescribed loads mat(node,dof,force)
loads = [ 3   2 -0.01 ];

% Boundary conditions mat(node,dof,disp)
bound = [ 1  1  0.0
          1  2  0.0
          2  1  0.0 ];

% Control Parameters
plotdof = 6;
```

Todays goal!

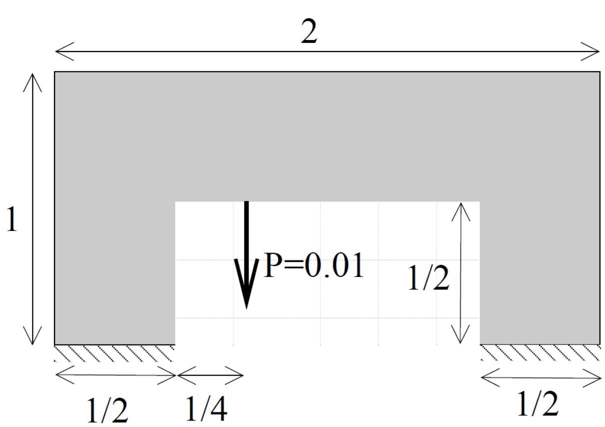


Competition 1

Here is a competition: Who can make the stiffest truss structure? This corresponds to the truss structure with the smallest displacement (in the direction of the load) where the load is applied! [1]

Rules: The truss structure should fit into the grey domain shown below. Supports may be used anywhere on the lower edge, whereas the position and size of the load are fixed. The bar areas and Young's moduli are fixed to $A = 1$ and $E = 1$, respectively. The maximum total length of bar elements is 6 and you may use as many hinges and elements as you want.

Create an input file called "group#.m" (where # is your group number) and mail it to Kim (knia@dtu.dk) by Monday, September 5th (at midnight). The teachers will evaluate the proposals and announce the prize-winner in the next lecture!



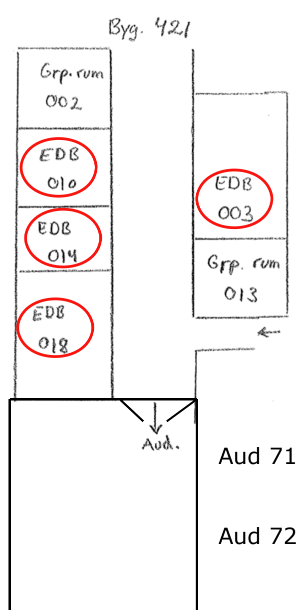
[1]The submission (not the ranking!) is part of the evaluation of the first report in the course!

Groups

- Form groups of two:
  - Team up with partner with similar time availability and ambitions
  - If no programming experience: find partner who has !
  - If no solid mechanics experience: find partner who has !

Computer exercises

Building 421



- Distribute in computer rooms
- Organize groups
- Read notes/exercises
- Use notes, book, brain, other students – then teachers …

Good luck and have fun!!