10. Introduction to matrix methods in structural mechanics

Contents – 1st revision

10.1 Introduction and terminology

The present chapter contains an introduction to application of matrix methods for structural analysis of linear truss and beam structures. In classical mechanics of materials, the differential equations governing deformations of such structures due to given loads have been derived and considered on explicit form. This has provided a framework where stresses are calculated based on equilibrium between internal and external forces, given constitutive equations (material laws) and (for statically indeterminate structures) equations of compability ensuring that the deformations are physically possible. This, however, leads to a framework where a solution to a large number of differential- and algebraic equations on explicit form is required for larger structures. The explicit solution form, where the differential questions are required fulfilled for all points inside the considered structure (or domain) is often referred to as *strong form* and is the conventional solution form applied for analytical solution of mechanical problems.

A method for structural analysis, which is based on polynomial interpolations of the solutions to the governing differential equations, will now be introduced. Such a solution is often called a *weak form* and has as major advantage, that the equations required to solve a mechanical problem can be derived algebraically and therefore be solved applying linear algebra. While this inevitably also often leads to a large number of equations, these can be shown to be formulated in a modular fashion, which is very convenient for numerical analysis. The introduced methods constitute the basis for linear finite element analysis (FEA) and are implemented in commercially available software for structural analysis (like ANSYS and Abaqus). However, the underlying theoretical framework is very useful in general, both for analytical and numerical calculations.

Our scope will be limited to structures constituted by trusses/bars and beams. The considered systems will be divided into a set of points, which are denoted *nodes*. Each node can move in a number of pre-defined ways (translation/rotation). We refer to this number of ways as *degrees of freedom* (DOF). The part of the structure between two nodes will be denoted an *element*. External loads can only be applied in nodes, so each DOF has a corresponding load component (force/moment). It is our objective to organize our equations in terms of a column vector containing the nodal deformations $(\boldsymbol{X})$ and a column vector containing the corresponding external forces $(\boldsymbol{F})$. These two vectors will be related by a number of stiffness terms organized in a stiffness matrix $[\boldsymbol{K}]$

$$(\boldsymbol{F}) = [\boldsymbol{K}](\boldsymbol{X}) \qquad 1.$$

The deformations along the considered element due to the applied loads are expressed solely in terms of the nodal deformations, which are interpolated by prescribed functions. The functions applied for interpolation of a unit displacement of a DOF will be denoted *shape functions***.** In Figure 1, a two node beam element is shown. Each node can deform by translation in the longitudinal direction, δ, by translation perpendicular to the longitudinal direction (deflection), u and furthermore rotate, θ. One node therefore has tree DOF's yielding a total of 6 DOF's for the entire beam element. These are organized on the following vectorial form for a single node

$$(\boldsymbol{f})_A = \begin{pmatrix} N_A \\ V_A \\ M_A \end{pmatrix} \qquad (\boldsymbol{x})_A = \begin{pmatrix} \delta_A \\ u_A \\ \theta_A \end{pmatrix} \qquad (\boldsymbol{f})_B = \begin{pmatrix} N_B \\ V_B \\ M_B \end{pmatrix} \qquad (\boldsymbol{x})_B = \begin{pmatrix} \delta_B \\ u_B \\ \theta_B \end{pmatrix} \qquad 2.$$

For the entire element, we may now write

$$(\boldsymbol{x})_{AB} = \begin{pmatrix} (\boldsymbol{X})_A \\ (\boldsymbol{X})_B \end{pmatrix} = \begin{pmatrix} \text{DOF1} \\ \text{DOF2} \\ \text{DOF3} \\ \text{DOF4} \\ \text{DOF5} \\ \text{DOF6} \end{pmatrix} = \begin{pmatrix} \delta_A \\ u_A \\ \theta_A \\ \delta_B \\ u_B \\ \theta_B \end{pmatrix} \qquad (\boldsymbol{f})_{AB} = \begin{pmatrix} (\boldsymbol{F})_A \\ (\boldsymbol{F})_B \end{pmatrix} = \begin{pmatrix} N_A \\ F_A \\ M_A \\ N_B \\ F_A \\ M_B \end{pmatrix} \qquad 3.$$

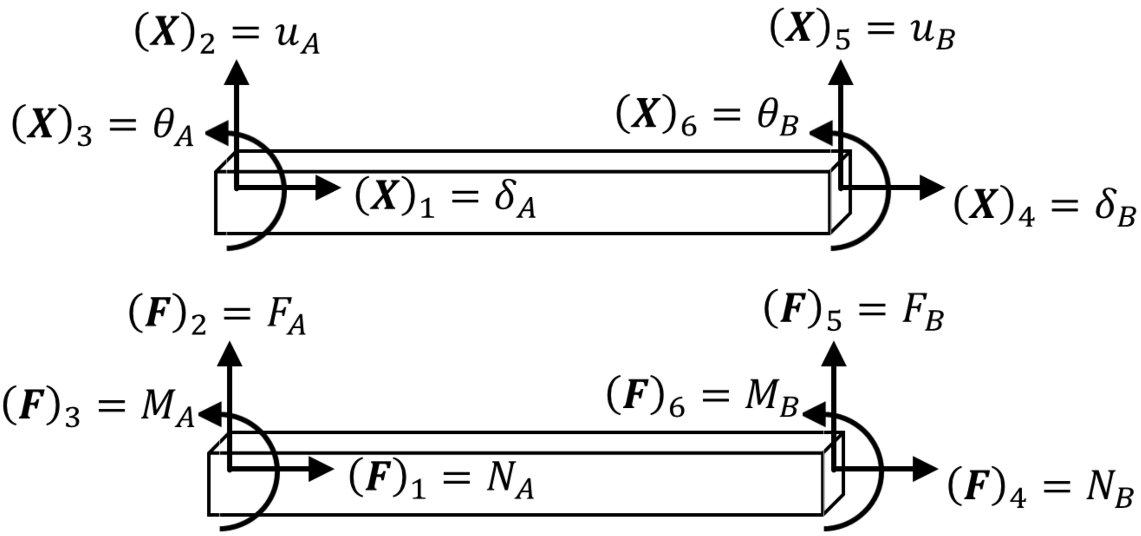

$(\boldsymbol{X})_2 = u_A$
$(\boldsymbol{X})_5 = u_B$
$(\boldsymbol{X})_3 = \theta_A$
$(\boldsymbol{X})_6 = \theta_B$
$(\boldsymbol{X})_1 = \delta_A$
$(\boldsymbol{X})_4 = \delta_B$
$(\boldsymbol{F})_2 = F_A$
$(\boldsymbol{F})_5 = F_B$
$(\boldsymbol{F})_3 = M_A$
$(\boldsymbol{F})_6 = M_B$
$(\boldsymbol{F})_1 = N_A$
$(\boldsymbol{F})_4 = N_B$

Figure 1 DOF's and nodal loads of a two-node 6 DOF beam element.

In the following, it will be demonstrated how to derive the stiffness matrix from chosen interpolation functions and the governing differential equations derived in classical strength of materials. Furthermore, it will be shown how multiple element-wise stiffness matrices can be assembled to form equations for larger structures, and how supports are added as *constrains* to the linear system of equations.

In the current chapter, numerical analysis will be performed using Matlab for the simple reason, that Matlab is available and widely applied at HSRW. The learning objective is, however, that students are able to formulate and solve structural mechanics problems both analytically and numerically with the developed methods. Furthermore, two Matlab scripts are provided with the lecture notes, to demonstrate how the developed formulations are implemented for problems of larger scale:

- `MyFirstFEACode.m` : An ultra simple FEA-code for truss structures similar to the computational engine running the epic computer game *Bridgebuilder*.
- `MrBeam.m` : A FEA-code for plane (2D) beam problems

These can be downloaded here [6].

A note of great importance: Students who hated doing Matlab scripting to solve numerical problems, should stop hating it now. Programming is an extremely useful engineering tool, and the reason we use Matlab is that it has the least steep learning curve. There's an open source Matlab clone called Octave [7] , you can use for most simple Matlab scripts, if you end up in a company without Matlab. Another alternative is one of the Python distributions (fx. Python x-y [7]), which are developed for numerical analysis and datamining.

10.2 The direct stiffness method

Stiffness matrices are often in the literature derived on basis of energy- and variational methods, which qualitatively speaking let work done by inner force (strain energy) balance work done by outer forces (external loads). This approach to matrix methods is elegant, abstract and perfectly suited for master students specializing in structural mechanics. In the present approach, the stiffness matrices will be derived from the well-known governing equations for trusses and beams (derived in chapter 2, 4 and 8). If you are curios and cannot wait to see how to do this in a more general and abstract manner, a brief outline is presented in [1]. For a more detailed description, FEM-guru and professor Carlos Felippa's lectures notes [2] are best-in-class, when it comes to online material.

For now, we will stick to the following framework, known as the *direct stiffness method*, in order to derive our own stiffness matrices:

1. Identify the differential equation governing the deformation of the considered element
2. Guess a general solution, in the considered cases, a polynomial.
3. Apply a unit deformation (or rotation) one by one to each DOF with other DOF's deformations set to zero.

   Note that for a unit deformation, the magnitude of the nodal forces will equal the magnitude of the stiffness constants in linear mechanics
4. Use force- and moment equilibrium to calculate the forces acting on the node due to this particular unit deformation
5. On basis of superposition, to write the total deformation as a sum (linear combination) of the stiffness constants determined in the previous form.

   Note that this when converted to matrix form will provide the stiffness matrix $[\boldsymbol{K}]$.

This method will in section 10.3.110.3 be applied to derive $[\boldsymbol{K}]$ for a truss element and in section 10.4.1 to derive $[\boldsymbol{K}]$ for a beam element.

10.3 The plane truss element

We recall that truss structures are constituted by pinned-pinned bars only loaded in the joints. Each truss is subjected to either tension or compression with no bending stresses. Under such boundary conditions, each truss act like a spring with stiffness k:

$$F = k\delta,\ k = \frac{EA}{L} \qquad 4.$$

The nodal forces are eventually easily obtained from the deformations and can be arranged on to matrix form as follows

$$\left.\begin{aligned} F_A &= k(\delta_A - \delta_B)\\ F_B &= k(\delta_B - \delta_A)\end{aligned}\right\} \rightarrow \begin{pmatrix}F_A\\ F_B\end{pmatrix} = \frac{EA}{L}\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}\begin{pmatrix}\delta_A\\ \delta_B\end{pmatrix} \rightarrow [\boldsymbol{k}] = \frac{EA}{L}\begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix} \qquad 5.$$

The stiffness matrix for a truss element has just been derived. That was easy ! However, we forgot entirely about the direct stiffness method introduced in section 10.2. In order to practice this method, we will re-derive equation 5 in the following.

10.3.1 Governing differential equation and shape functions

Applyinig the direct stiffness method in section 10.2, we proceed step-wise

<u>Step 1: </u>The first thing we have to figure out, is which equation will govern the deformations. Considering the bar in Figure 2, we recall that the strain is given by

$$\varepsilon = \frac{d\delta}{dx} \qquad 6.$$

Recalling from chapter 6 how we used to develop a force quantity over an infinitesimal length segment using a first order (linear) Taylor approximation, we write for a distributed longitudinal load *p(x)*

$$F(dx) = F + dF + p(x)dx \qquad 7.$$

Rearranging this, dividing through with dx and substituting $F = EA\varepsilon$, the following differential equation is obtained

$$\begin{aligned} &\frac{dF}{dx} + p(x) = 0 \\ &\rightarrow \frac{d}{dx}(AE\varepsilon) + p(x) = 0 \\ &\rightarrow AE\frac{d^2\delta}{dx^2} + p(x) = 0 \end{aligned} \qquad 8.$$

Now considering the special case where the distributed load is zero, since we limit ourselves to considering the case where loads solely are applied in the nodes, we have as governing equation

$$AE\frac{d^2\delta}{dx^2} = 0 \qquad 9.$$

<u>Step 2:</u> This equation 10 has a solution on the form

$$\delta(x) = C_1x + C_2 \qquad 10.$$

<u>Step 3/4:</u> We now in general decide, that k_{ij} denotes the stiffness of DOF i due to a unit displacement of DOF j. For a unit longitudinal deformation of the first node (the first DOF) and zero longitudinal deformation of the second node (the second DOF), the following hold:

$$\begin{aligned} &\delta(0) = 1 = C_1 \cdot 0 + C_2 \rightarrow C_2 = 1 \\ &\delta(L) = 0 = C_1 \cdot L + C_2 = C_1 \cdot L + 1 \rightarrow C_1 = -\frac{1}{L} \end{aligned} \qquad 11.$$

This gives us the shape function $N_1 = -\frac{x}{L} + 1$.

We observe from Figure 2, that $k_{11} = \frac{EA}{L}$- Force equilibrium now gives

$$\sum F = k_{11} + k_{41} = 0 \rightarrow k_{11} = -k_{41} = -\frac{EA}{L} \qquad 12.$$

For a unit longitudinal deformation of the second node, and zero deformation of the first node, the following results are obtained:

$$\begin{aligned} &\delta(0) = 0 = C_1 \cdot 0 + C_2 \rightarrow C_2 = 0 \\ &\delta(L) = 1 = C_1 \cdot L + C_2 = C_1 \cdot L \rightarrow C_1 = \frac{1}{L} \end{aligned} \qquad 13.$$

This gives us the shape function $N_2 = \frac{x}{L}$.

Again, we observe from Figure 2, that $k_{44} = \frac{EA}{L}$- Force equilibrium now gives

$$\sum F = k_{14} + k_{44} = 0 \rightarrow k_{14} = -k_{44} = -\frac{EA}{L} \qquad 14.$$

The deformation function throughout the element is now given by

$$\delta(x) = N_1\delta_A + N_2\delta_B \qquad 15.$$

That pretty much gives us an idea of the shape functions as normalized interpolation polynomials.

<u>Step 5:</u> The element stiffness matrix is obtained as

$$[\boldsymbol{k}] = \begin{bmatrix} k_{11} & k_{41} \\ k_{14} & k_{44} \end{bmatrix} = \frac{EA}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad 16.$$

We have now obtained the element stiffness matrix for a single truss element oriented along the global x-axis. It is noted that the stiffness matrix is symmetrical (that’s always the case). Each node currently only has a single DOF and this holds as long as the truss is only loaded along it’s longitudinal direction.

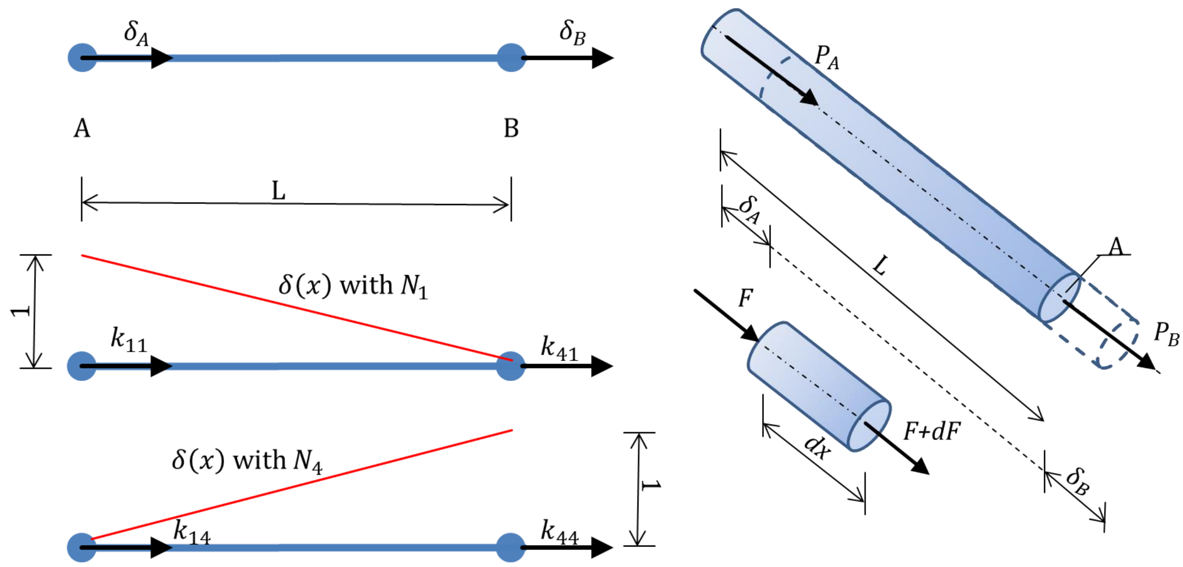


Figure 2, Left: Plane 2-node truss element with shape functions, **Right**: Equilibrium of a truss element

10.3.2 Stiffness matrix assembly

When assembling the stiffness matrix for a system constituted by multiple truss-elements, the unconstrained stiffness matrix $[\boldsymbol{K}]^{(U)}$ is square with dimension equal to the number of nodes times the number of DOF. However, nodes which are included in more than one element, will naturally have stiffness contributions from all of those elements.

There are two ways of implementing supports by constraining the stiffness matrix:

1. A quick and dirty one where a stiffness value, which is sufficiently large to prevent deformations, but not so large that it will turn the stiffness matrix singular, is added to corresponding entry in the stiffness matrix.
2. A theoretically proper one where the constraint DOF is removed from the stiffness matrix and from the corresponding force vector and thereby reducing the dimension of the system of equations when obtaining the constraint stiffness matrix $[\boldsymbol{K}]^{(C)}$.
   In general, the following holds:

$$n_{DOF} = n_{Nodes} \cdot n_{\frac{DOF}{Node}} - n_{constrains}$$

This implies that a constraint is added, everytime a DOF in $(\boldsymbol{X})$ is set to zero. When programming larger systems, it is crucial to remember, that everytime a row and a column are removed from the stiffness matrix to add a constraint, the dimension of the stiffness matrix is altered making it difficult to navigate by using node numbers. This is often overcome by removing the entries for the constrains for nodes with highest node number first and proceeding backwards towards the constrains for the lowest node number.

This might have been a bit abstract and hard to grasp. It's often easier after having seen an example.

10.3.3 Calculated example: Compound cylindrical bar in tension

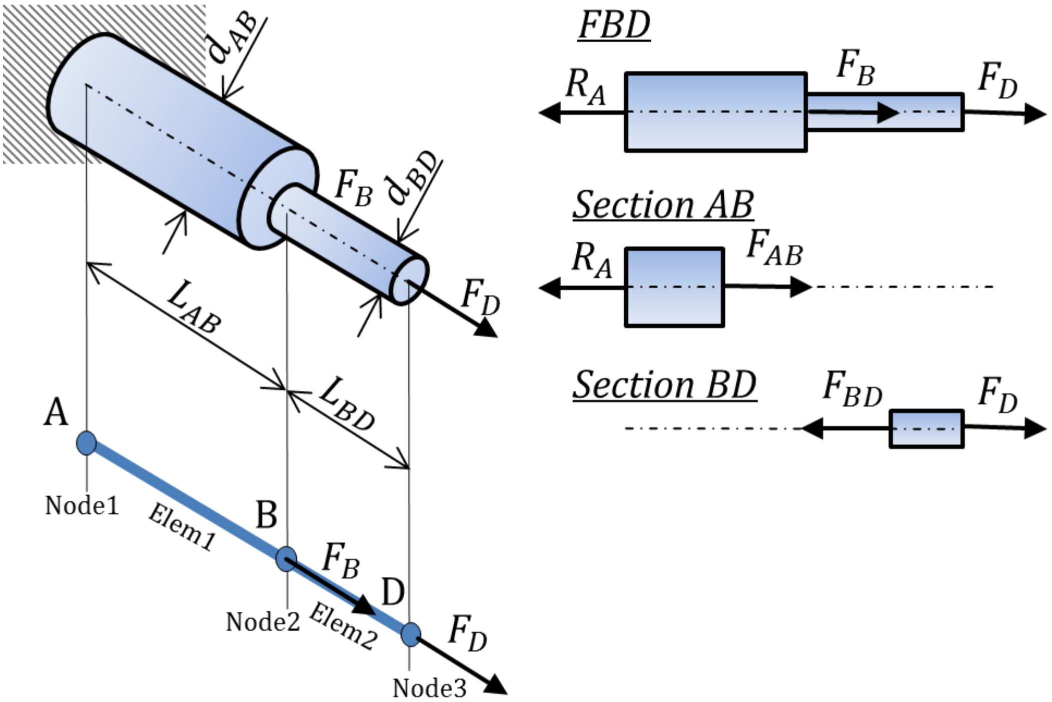


Figure 3 Compound cylinder subjected to multiple longitudinal loads

Problem: The compound cylinder in Figure 3 has dimensions given by
$d_{AB} = 100\ mm \qquad d_{BD} = 75\ mm \qquad L_{AB} = 500\ mm \qquad L_{BD} = 600\ mm$
The cylinder is made of aluminum with elastic modulus $E = 70\ GPa$ and is subjected to loads
$F_B = 100\ kN \qquad F_D = 200\ kN$
Determine: a) the deformation of point D using a) classical strength of materials (see chapter 2), b) matrix methods
Solution: Initially, the cross-sectional areas will be calculated, since these will be needed in both approaches:
$A_{AB} = \frac{\pi}{4}(d_{AB})^2 = 0.0079m^2$
$A_{BD} = \frac{\pi}{4}(d_{BD})^2 = 0.0044m^2$
Approach from classical mechanics: The reaction force at A is calculated based on the FBD in Figure 3. This allows us to lay in sections AB and BD to calculate the internal forces:
$R_A = F_B + F_D = 300kN \qquad F_{AB} = R_A = 100kN \qquad F_{BD} = F_D = 200kN$
The deformation is now calculated by converting everything to SI units and using the expression
$\delta_D = \sum\frac{FL}{EA} = \frac{F_{AB}L_{AB}}{EA_{AB}} + \frac{F_{BD}L_{BD}}{EA_{BD}} = \frac{(F_B+F_D\ )L_{AB}}{EA_{AB}} + \frac{F_D L_{BD}}{EA_{BD}} = 0.66\cdot 10^{-3}mm$
We could get the stresses as internal force by area. Instead, let's try this out using matrices:
Matrix methods approach: As first step, the cylinder is divided into two elements defined by three nodes. Note, that these always should be numbered before proceeding in order for us not to loose overview. The system to solve is given by

$$(\boldsymbol{F}) = [\boldsymbol{K}]^{(C)}(\boldsymbol{X}) \qquad (\boldsymbol{F}) = \begin{pmatrix} N_1 \\ N_2 \\ N_3 \end{pmatrix} = \begin{pmatrix} 0 \\ F_B \\ F_D \end{pmatrix} \qquad (\boldsymbol{d}) = \begin{pmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{pmatrix} = \begin{pmatrix} 0 \\ \delta_2 \\ \delta_3 \end{pmatrix}$$

We just need to get $[\boldsymbol{K}]^{(C)}$. Now the element-wise stiffness matrices are calculated by equation 16:

$$[\boldsymbol{k}]_1^{(AB)} = \begin{bmatrix} k_{11}^{(1)} & k_{41}^{(1)} \\ k_{14}^{(1)} & k_{44}^{(1)} \end{bmatrix} = \frac{EA_{AB}}{L_{AB}}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = 1.0996 \cdot 10^9 \frac{N}{m}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$[\boldsymbol{k}]_2^{(BD)} = \begin{bmatrix} k_{11}^{(2)} & k_{41}^{(2)} \\ k_{14}^{(2)} & k_{44}^{(2)} \end{bmatrix} = \frac{EA_{BD}}{L_{BD}}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = 5.1542 \cdot 10^9 \frac{N}{m}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

In order to assemble the total stiffness matrix $[\boldsymbol{K}]^{(C)}$, we add these two stiffness matrices while accounting for which DOF's the separate entries belong to. Each node only has a single DOF and node 2 is the only node having stiffness contributions from both elements. We obtain

$$[\boldsymbol{K}]^{(U)} = \begin{bmatrix} k_{11}^{(1)} & k_{41}^{(1)} & 0 \\ k_{14}^{(1)} & k_{44}^{(1)} & 0 \\ 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & k_{11}^{(2)} & k_{41}^{(2)} \\ 0 & k_{14}^{(2)} & k_{44}^{(2)} \end{bmatrix} = \begin{bmatrix} \frac{EA_{AB}}{L_{AB}} & -\frac{EA_{AB}}{L_{AB}} & 0 \\ -\frac{EA_{AB}}{L_{AB}} & \frac{EA_{AB}}{L_{AB}} + \frac{EA_{BD}}{L_{BD}} & -\frac{EA_{BD}}{L_{BD}} \\ 0 & -\frac{EA_{BD}}{L_{BD}} & \frac{EA_{BD}}{L_{BD}} \end{bmatrix}$$

This is the unconstrained stiffness matrix. The only problem is that it's singular. The reason is, that it's still unconstrained. The only support acting on the system is the wall support in A yielding $\delta_1 = 0$. This means that DOF1 is constrained. We implement this by removing the first row and the first column from $[\boldsymbol{K}]^{(U)}$ and thereby also the first entry in $(\boldsymbol{F})$. The constrained system is now given by

$$[\boldsymbol{K}]^{(C)} = \begin{bmatrix} \frac{EA_{AB}}{L_{AB}} + \frac{EA_{BD}}{L_{BD}} & -\frac{EA_{BD}}{L_{BD}} \\ -\frac{EA_{BD}}{L_{BD}} & \frac{EA_{BD}}{L_{BD}} \end{bmatrix} \qquad (\boldsymbol{F})^{(C)} = \begin{pmatrix} F_B \\ F_C \end{pmatrix}$$

The deformations are now obtained by

$$(\boldsymbol{X}) = \begin{pmatrix} \delta_2 \\ \delta_3 \end{pmatrix} = \left([\boldsymbol{K}]^{(C)}\right)^{-1}\left(\boldsymbol{F}^{(C)}\right) = \begin{pmatrix} 0.27 \\ 0.66 \end{pmatrix} mm$$

Which corresponds to the results we got using the classical approach. We could also have done this using Matlab:

```
%Calculated example 2.4
clc; close all; clear all;
%Input %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
dAB=0.1;            dBD=0.075;         LAB=0.5;       LBD=0.6;
E=70*10^9;          FB=100*10^3;       FD=200*10^3;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A_AB=pi/4*dAB^2;    A_BD=pi/4*dBD^2;   %Calculate areas
I=[1,-1;-1,1];
kAB=E*A_AB/LAB*I;   kBD=E*A_BD/LBD*I;  %Element stiffness matrices
KU(1:3,1:3)=0;                         %Initialize global stiffness matrix
KU(1:2,1:2)=KU(1:2,1:2)+kAB;           %Add elem1 to global stiffness matrix
KU(2:3,2:3)=KU(2:3,2:3)+kBD;           %Add elem2 to global stiffness matrix
KC=KU(2:3,2:3);                        %Global constrained stiffnes matrix
dlt=inv(KC)*[FB;FD]                    %Solve for deformations
```

While this admittedly was not easier than the classical approach, actually to the contrary, it is crucial to get how the stiffness matrix is constructed as sum of separate elementwise stiffness matrices and constrained by removing DOF's as entries and columns from the stiffness matrix.

10.3.4 Transformation matrix

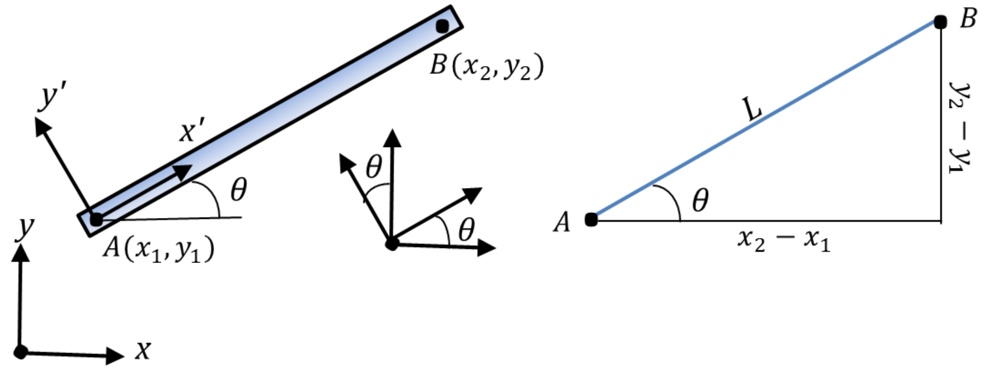


Figure 4 Truss element with spatial orientation

This far, only bars and trusses with longitudinal axis corresponding to the global x-axis have been considered. Furthermore, loads have solely been applied in the longitudinal direction. This is far from always the case, and in order to account for a different spatial orientation of a truss element, displacement, stiffness, and force terms must be transformed from the local element coordinate system (denoted with primes in Figure 4) to the global Cartesian coordinate system. From math, we know that the transformation matrix required to rotate a vector in an angle of θ counter clockwise is given by

$$[\boldsymbol{T}] = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{17.}$$

From the geometry in Figure 4, the directional cosine and sine terms are given by

$$\cos\theta = \frac{y_2 - y_1}{L} \quad \sin\theta = \frac{x_2 - x_1}{L} \quad L = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{18.}$$

It is now important to note, that when loads in any directions are applied to a truss, each node can translate in both the x- and y- directions. Therefore, for spatially oriented trusses, each node has two displacement DOF's and for each element we have

$$\boldsymbol{X} = \begin{pmatrix} \delta_x \\ \delta_y \\ \delta_x \\ \delta_y \end{pmatrix} \qquad \boldsymbol{F} = \begin{pmatrix} F_x \\ F_y \\ F_x \\ F_y \end{pmatrix} \tag{19.}$$

In order to derive an expression for the stiffness matrix in the global coordinate system, we utilize that forces and displacements can be transformed from a local to a global coordinate system by

$$(\boldsymbol{F})' = [\boldsymbol{T}](\boldsymbol{F}) \quad (\boldsymbol{X})' = [\boldsymbol{T}](\boldsymbol{X}) \tag{20.}$$

However, since the transformation matrix in equation 17 is valid only for a single vector, and equation 20. contains two vectors organized into a single column, the transformation matrix must in this context be rearranged as

$$[\boldsymbol{T}] = \begin{bmatrix} \begin{matrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{matrix} & 0 \\ 0 & \begin{matrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{matrix} \end{bmatrix} \tag{21.}$$

The following equation can now be written

$$\begin{aligned} & (\boldsymbol{F})' = [\boldsymbol{k}]'(\boldsymbol{X})' \\ \rightarrow\ & [\boldsymbol{T}](\boldsymbol{F}) = [\boldsymbol{k}]\,[\boldsymbol{T}](\boldsymbol{X}) \\ \rightarrow\ & (\boldsymbol{F}) = [\boldsymbol{T}]^{-1}[\boldsymbol{k}]'\,[\boldsymbol{T}](\boldsymbol{X}) \qquad [\boldsymbol{T}]^{-1} = [\boldsymbol{T}]^T \text{ for rotation matrices} \\ \rightarrow\ & (\boldsymbol{F}) = [\boldsymbol{k}](\boldsymbol{X}) \qquad \text{with } [\boldsymbol{k}] = [\boldsymbol{T}]^T[\boldsymbol{k}]'\,[\boldsymbol{T}] \end{aligned} \tag{22.}$$

We now have

$$[k] = [T]^T [k]' \, [T]$$

$$= \begin{bmatrix} \begin{matrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{matrix} & 0 \\ 0 & \begin{matrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{matrix} \end{bmatrix}^T \frac{EA}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \begin{matrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{matrix} & 0 \\ 0 & \begin{matrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{matrix} \end{bmatrix} \qquad 23.$$

$$= \frac{EA}{L} \begin{bmatrix} \cos^2\theta & \cos\theta\sin\theta & -\cos^2\theta & -\cos\theta\sin\theta \\ \cos\theta\sin\theta & \sin^2\theta & -\cos\theta\sin\theta & -\sin^2\theta \\ -\cos^2\theta & -\cos\theta\sin\theta & \cos^2\theta & \cos\theta\sin\theta \\ -\cos\theta\sin\theta & -\sin^2\theta & \cos\theta\sin\theta & \sin^2\theta \end{bmatrix}$$

This gives us the stiffness matrix for a truss element on component form. The internal forces can be calculated by

$$(\boldsymbol{F})^{(i)}_{elem} = \frac{EA}{L} \begin{pmatrix} -\cos\theta & -\sin\theta & \cos\theta & \sin\theta \end{pmatrix} \begin{pmatrix} \delta_x \\ \delta_y \\ \delta_x \\ \delta_y \end{pmatrix} \qquad 24.$$

10.3.5 MyFirstFEMCode.m

As example of a simple code for structural analysis of linear truss structures, a Matlab script was written. However, it's only actual functionality is that it assembles a stiffness matrix for a truss structure on basis of a set of nodal coordinates on matrix form (`nodes`) and a set of elements (`elem`) defined in terms of two node numbers referring back to two rows in the nodal matrix. The moment of inertia (`I`) and the cross sectional area (`Area`) of the elements are constants for all members and must be calculated prior to construction of the total stiffness matrix. The constrains are implemented using the 'quick and dirty' method, where the *k'th* DOF is constrained by replacing the *k'th* times *k'th* entry in the stiffness matrix with a large stiffness term (`BCStiffness`) defined by the user. Furthermore, the external force array must be initiated and defined manually (`F`). An example of the implemented visualization plots from the code are shown in Figure 7.

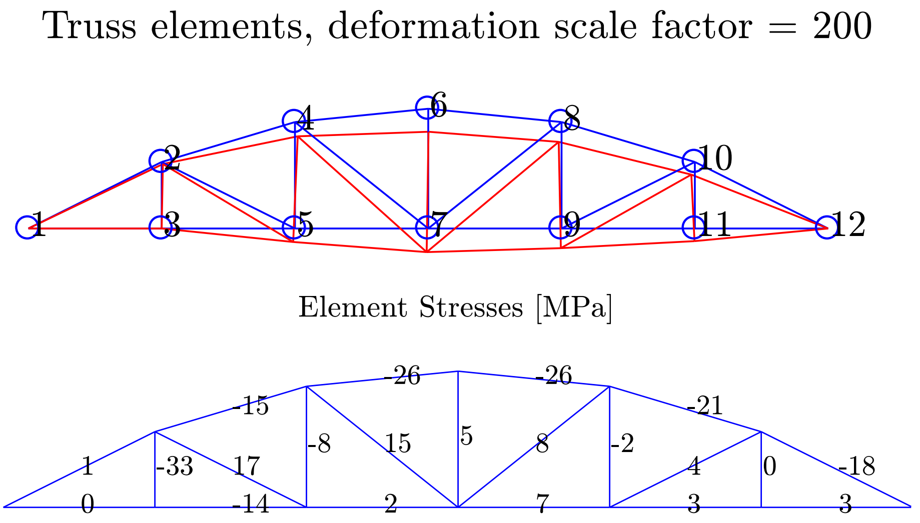


Figure 5 Output from MyFirstFEACode.m for example model, `bridge_geom2.m`

10.3.6 Calculated example: Truss console

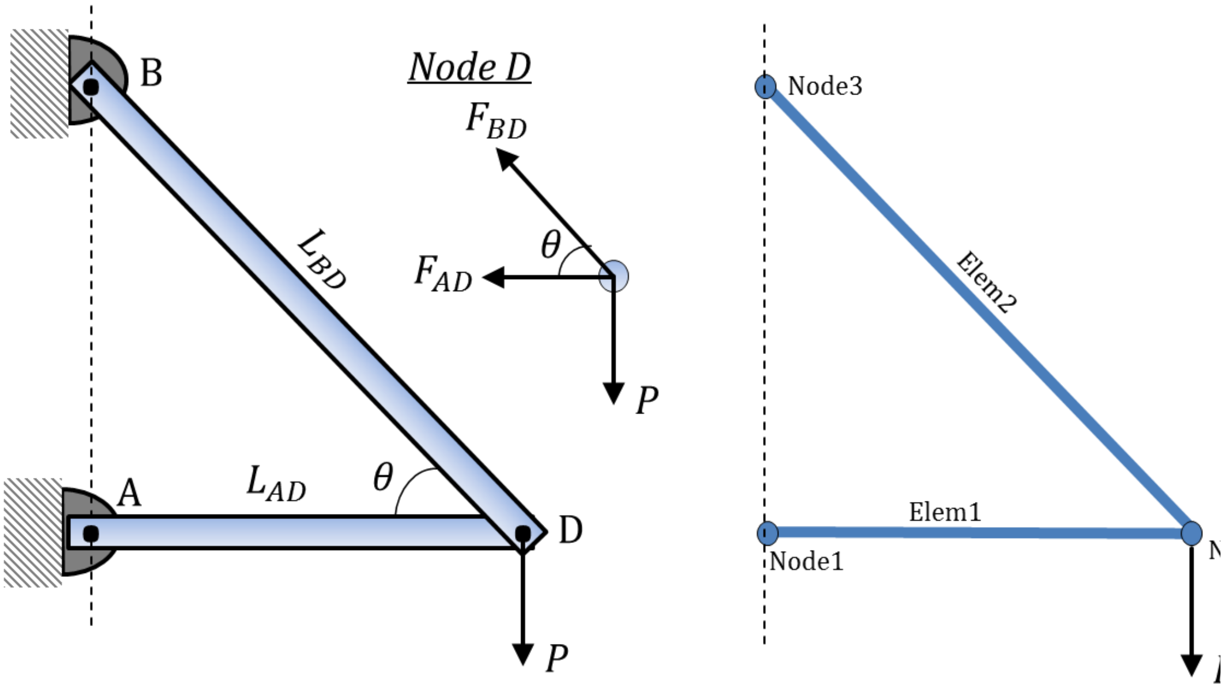


Figure 6 Truss console

Problem: The truss system shown in Figure 6 is constituted by members with rectangular cross-sections with dimensions $h \times b$. The geometry is given by the following set of parameters:

$h = 100\,mm \qquad b = 50\,mm \qquad L_{AD} = 2m \qquad \theta = 60\,deg$

The members are made of steel with elastic modulus $E = 210\,GPa$. The structure is subjected to a downwards load in point D of magnitude P=100 kN.

Determine: the normal stresses in the members using a) classical strength of materials (see chapter 2), b) matrix methods

Solution: initially, the missing geometrical properties are calculated

$A = h \cdot b = (0.100m)(0.050m) = 0.005m^2 \qquad L = \frac{L_{AD}}{\cos\theta} = 4m$

Classical mechanics approach (method of joints): Drawing a FBD of node D and applying static equilibrium, we obtain:

$\sum F_y = 0 = -P + F_{BD}\sin\theta \qquad \sum F_x = 0 = -F_{AD} - F_{BD}\cos\theta$

Solving these equations, the internal forces can be calculated

$F_{AD} = -57.7\,kN$ and $F_{BD} = 115.5\,kN$.

The stresses in the members are

$\sigma_{AD} = \frac{F_{AD}}{A_{AD}} = -12\frac{N}{mm^2} \qquad \sigma_{BD} = \frac{F_{BD}}{A_{BD}} = 23\frac{N}{mm^2}$

Matrix methods approach: The unconstrained system of equations to be solved is given by

$$(\boldsymbol{F}) = [\boldsymbol{K}](\boldsymbol{X}) \qquad (\boldsymbol{F}) = \begin{pmatrix} F_x^{(1)} \\ F_y^{(1)} \\ F_x^{(2)} \\ F_y^{(2)} \\ F_x^{(3)} \\ F_y^{(3)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -P \\ 0 \\ 0 \end{pmatrix} \qquad (\boldsymbol{d}) = \begin{pmatrix} \delta_x^{(1)} \\ \delta_y^{(1)} \\ \delta_x^{(2)} \\ \delta_y^{(2)} \\ \delta_x^{(3)} \\ \delta_y^{(3)} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \delta_x^{(2)} \\ \delta_y^{(2)} \\ 0 \\ 0 \end{pmatrix}$$

The element stiffness matrices are calculated using equation 23. The first element, AD, is horizontal:

$$[k]_1^{AD} = \begin{bmatrix} \frac{EA_{AD}}{L_{AD}} & 0 & -\frac{EA_{AD}}{L_{AD}} & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{EA_{AD}}{L_{AD}} & 0 & \frac{EA_{AD}}{L_{AD}} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Member BD is obtained as rotated 60 deg. clockwise with respect to horizontal, i.e. $\theta = -60\ deg$

$$[k]_2^{BD} = \frac{EA_{BD}}{L_{BD}} \begin{bmatrix} \cos^2\theta & \cos\theta\sin\theta & -\cos^2\theta & -\cos\theta\sin\theta \\ \cos\theta\sin\theta & \sin^2\theta & -\cos\theta\sin\theta & -\sin^2\theta \\ -\cos^2\theta & -\cos\theta\sin\theta & \cos^2\theta & \cos\theta\sin\theta \\ -\cos\theta\sin\theta & -\sin^2\theta & \cos\theta\sin\theta & \sin^2\theta \end{bmatrix}$$

The total unconstrained stiffness matrix is obtained as

$$[\boldsymbol{K}]^{(U)} = \begin{bmatrix} \frac{EA_{AD}}{L_{AD}} & 0 & -\frac{EA_{AD}}{L_{AD}} & 0 & \multicolumn{2}{c}{0} \\ 0 & 0 & 0 & 0 & & \\ -\frac{EA_{AD}}{L_{AD}} & 0 & \cos^2\theta + \frac{EA_{AD}}{L_{AD}} & \cos\theta\sin\theta & -\cos^2\theta & -\cos\theta\sin\theta \\ 0 & 0 & \cos\theta\sin\theta & \sin^2\theta & -\cos\theta\sin\theta & -\sin^2\theta \\ \multicolumn{2}{c}{0} & -\cos^2\theta & -\cos\theta\sin\theta & \cos^2\theta & \cos\theta\sin\theta \\ & & -\cos\theta\sin\theta & -\sin^2\theta & \cos\theta\sin\theta & \sin^2\theta \end{bmatrix}$$

Since node 1 and node 3 are pinned to the world frame and both have two DOF's, we eliminate the 6th,5th,2nd, and 1st rows and columns, i.e. the constrained stiffness matrix is given by

$$[\boldsymbol{K}]^{(C)} = \begin{bmatrix} \cos^2\theta + \frac{EA_{AD}}{L_{AD}} & \cos\theta\sin\theta \\ \cos\theta\sin\theta & -\sin^2\theta \end{bmatrix}$$

Having obtained $[\boldsymbol{K}]^{(C)}$, we may solve for deformations

$$(\boldsymbol{X}) = \begin{pmatrix} \delta_x^{(2)} \\ \delta_y^{(2)} \end{pmatrix} = \left([\boldsymbol{K}]^{(C)}\right)^{-1}\left(\boldsymbol{F}^{(C)}\right) = \left(\begin{bmatrix} \cos^2\theta + \frac{EA_{AD}}{L_{AD}} & \cos\theta\sin\theta \\ \cos\theta\sin\theta & -\sin^2\theta \end{bmatrix}\right)^{-1} \begin{pmatrix} 0 \\ -P \end{pmatrix} = -\begin{pmatrix} 0.11 \\ 0.57 \end{pmatrix} mm$$

The internal forces are finally calculated by equation 24

$$(\boldsymbol{F})_{AD}^{(1)} = \frac{EA}{L_{AD}} \begin{pmatrix} -1 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \delta_x^{(2)} \\ \delta_y^{(2)} \end{pmatrix} = -58kN$$

$$(\boldsymbol{F})_{BD}^{(2)} = \frac{EA}{L_{BD}} \begin{pmatrix} -\cos\theta & -\sin\theta & \cos\theta & \sin\theta \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ \delta_x^{(2)} \\ \delta_y^{(2)} \end{pmatrix} = 115kN$$

Since the internal forces are equal to those obtained by analytical calculations, the stresses will be too. We could also have done this using Matlab:

```
%Calculated example 2.5
%Input %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
h=0.1;              b=0.05;            LAD=2;         theta=60;
E=210*10^9;         P=100*10^3;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A=h*b;              LBD=LAD/cosd(theta);
I=[1,-1;-1,1];
c=1; s=0;
kAD=A*E/LAD*[c^2,c*s,-c^2,-c*s;c*s,s^2,-c*s,-s^2;-c^2,-c*s,c^2,c*s;-c*s,-s^2,c*s,s^2];
c=cosd(-60); s=sind(-60);
kBD=A*E/LBD*[c^2,c*s,-c^2,-c*s;c*s,s^2,-c*s,-s^2;-c^2,-c*s,c^2,c*s;-c*s,-s^2,c*s,s^2];
KU(1:6,1:6)=0; KU(1:4,1:4)=KU(1:4,1:4)+kAD; KU(3:6,3:6)=KU(3:6,3:6)+kBD;
KC=KU(3:4,3:4);
dlt=inv(KC)*[0;-P]                      %Calculate deformations
ForceAD=A*E/LAD*[-1,0,1,0]*[0;0;dlt];   %Internal force, AD
ForceBD=A*E/LBD*[-c,-s,c,s]*[0;
```

Or it could have been scripted as input for the truss code:

```
%Input for MyFirstFEACode
%Truss console geometry (input section)
%Input nodes
Lb=2.0;   db=Lb*tan(60*pi/180); %Frame geometry
nodes(1,:)=[0.0,0.0];nodes(2,:)=[Lb,0];nodes(3,:)=[0.0,db];

%Input elements [first element node, second element node]
elem(1,:)=[1,2]; elem(2,:)=[2,3];


%Truss console Loads/BCDS (paste after setup of stiffness matrix)
%%%Define boundary conditions
K(1,1)=BCStiffness;
K(2,2)=BCStiffness;
K(5,5)=BCStiffness;
K(6,6)=BCStiffness;
%%%Define load vector
F(6)=0;         %Initialize array
F(4)=-P;        %Define load
```

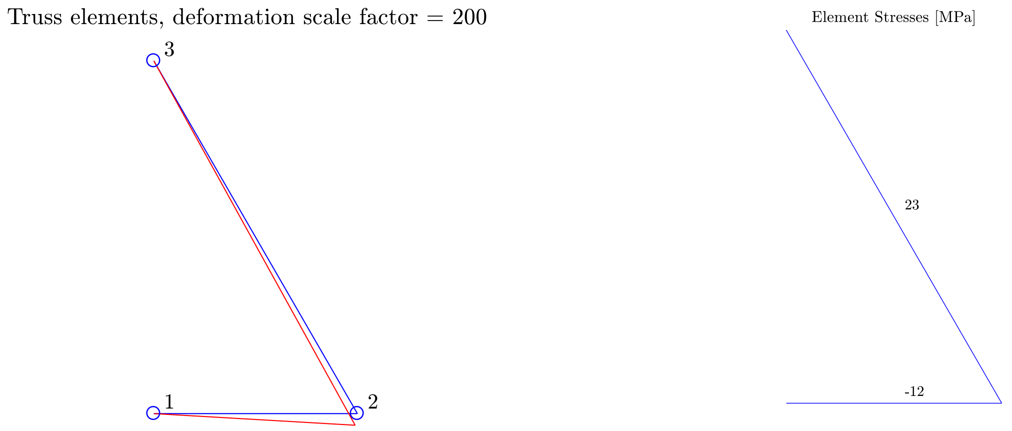


Figure 7 Output from MyFirstFEACode.m, **Left:** Node numbers and scaled deformations, **Right**: element stresses

10.4 The plane truss element

A plane two-node beam oriented along the global x-axis will now be considered, see Figure 8. As long as nodal loads are only added in the transverse direction or as moments, the deformations of the beam can be described in terms of a transverse displacement u and a rotation θ of each node. Therefore, each node has two DOF's yielding a total of four DOF's for the entire element.

10.4.1 Governing differential equation and shape functions

From the Bernoulli-Euler beam theory for long and slender members, we have already obtained a differential equation of fourth order governing the deformations of the beam. Considering the case where no distributed load is applied, the right hand side of the equation is zero, and the differential equations has a solution, which is a polynomial of third order

$$EI\frac{d^4u}{dx^4} = 0 \rightarrow u(x) = Ax^3 + Bx^2 + C_1x + C_2 \tag{25.}$$

Applying the direct stiffness method, our objective is for the beam element to derive a system of equations on the form

$$\begin{pmatrix} F_A \\ M_A \\ F_B \\ M_B \end{pmatrix} = \begin{bmatrix} k_{22} & k_{32} & k_{52} & k_{62} \\ k_{23} & k_{33} & k_{53} & k_{63} \\ k_{25} & k_{35} & k_{55} & k_{65} \\ k_{26} & k_{36} & k_{56} & k_{66} \end{bmatrix} \begin{pmatrix} u_A \\ \theta_A \\ u_B \\ \theta_B \end{pmatrix} \tag{26.}$$

Applying a unit displacement to the first DOF, u_A, while the three remaining DOF's are set to zero, the first shape function and the first line of the stiffness matrix is derived. The internal bending moment in the beam is given by

$$\sum M = M + k_{32} - k_{22}x = 0 \rightarrow M = -k_{32} + k_{22}x \tag{27.}$$

The following expression is now obtained by direct integration

$$\begin{aligned} \theta = \frac{dy}{dx} = \frac{1}{EI}\int M d\,x + C_1 &= \frac{1}{EI}\int -k_{32} + k_{22}x d\,x + C_1 \\ &= \frac{1}{EI}\left(-k_{32}x + \frac{k_{22}}{2}x^2\right) + C_1 \\ y = \int \theta d\,x = \int \frac{1}{EI}\left(-k_{32}x + \frac{k_{22}}{2}x^2\right) &+ C_1 d\,x + C_2 \\ =\frac{1}{EI}\left(-\frac{k_{32}}{2}x^2 + \frac{k_{22}}{6}x^3\right) &+ C_1x + C_2 \end{aligned} \tag{28.}$$

Applying the boundary conditions, the following four algebraic equations are now derived

$$\begin{aligned} &y(0) = 1 = C_2 \\ &\theta(0) = 0 = C_1 \\ &y(L) = 0 = \frac{1}{EI}\left(-\frac{k_{32}}{2}L^2 + \frac{k_{22}}{6}L^3\right) + 1 \\ &\theta(L) = 0 = \frac{1}{EI}\left(-k_{32}L + \frac{k_{22}}{2}L^2\right) \end{aligned} \tag{29.}$$

Solving the above equations, we obtain the values

$$k_{22} = \frac{12EI}{L^3} \qquad k_{32} = \frac{6EI}{L^2} \tag{30.}$$

We can solve for the remaining stiffness terms using force equilibrium in the y-direction and moment equilibrium

$$\sum F_y = 0 \rightarrow k_{52} = -k_{22} = -\frac{12EI}{L^3} \tag{31.}$$

$$\begin{aligned} \sum M_B = 0 \rightarrow k_{62} + k_{32} - k_{22}L = 0 \\ \rightarrow k_{62} = k_{22}L - k_{32} = \frac{12EI}{L^3}L - \frac{6EI}{L^2} = \frac{6EI}{L^2} \end{aligned} \tag{32.}$$

We have now obtained the first row of the stiffness matrix. The shape function for the current case is given by

$$N_1 = \frac{1}{EI}\left(-\frac{1}{2}\left(\frac{6EI}{L^2}\right)x^2 + \frac{1}{6}\left(\frac{12EI}{L^3}\right)x^3\right) + 1 = \frac{2x^3}{L^3} - \frac{3x^2}{L^2} + 1 \tag{33.}$$

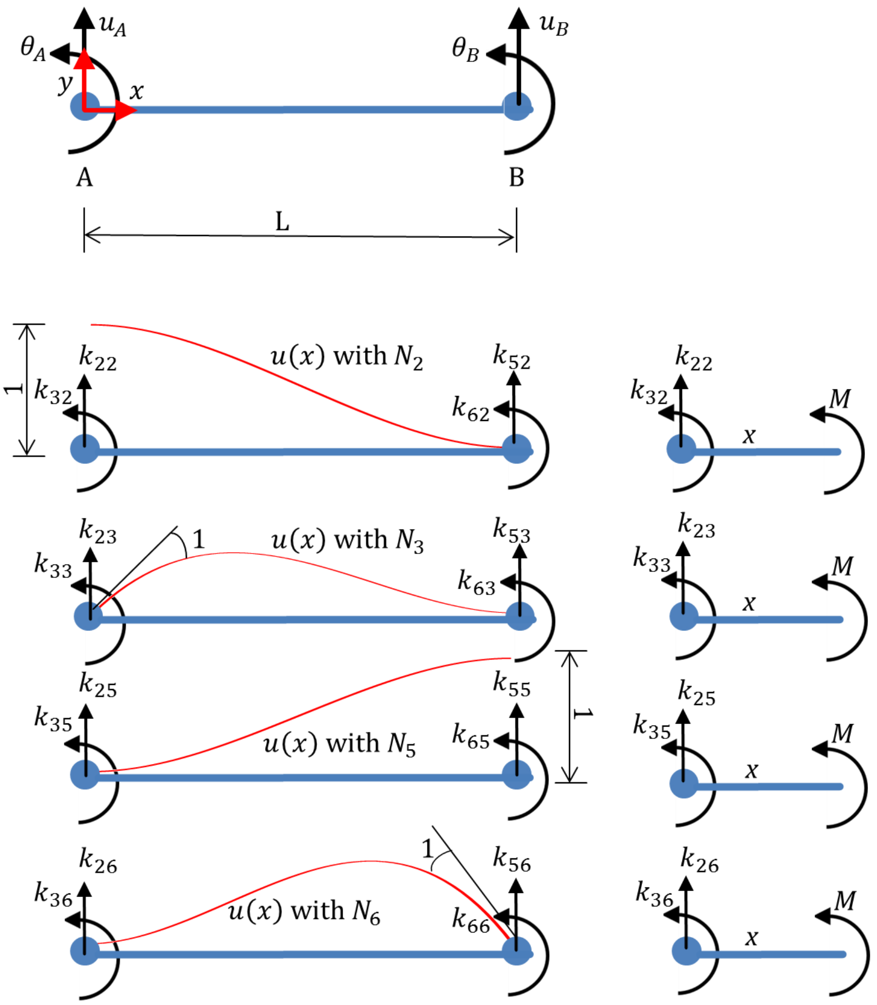


Figure 8 Two node beam element

Now applying a unit rotation to node A with all other displacement DOFs set to zero, the 2nd row of the stiffness matrix can be derived. The internal bending moment is given by

$$M = -k_{33} + k_{23}x \tag{34.}$$

By direct integration, the following relations are obtained

$$\begin{aligned}\theta &= \frac{dy}{dx} = \frac{1}{EI}\int Md\,x + C_1 = \frac{1}{EI}\int -k_{33} + k_{23}xd\,x + C_1 \\ &= \frac{1}{EI}\left(-k_{33}x + \frac{k_{23}}{2}x^2\right) + C_1 \\ y &= \int \theta d\,x = \int \frac{1}{EI}\left(-k_{33}x + \frac{k_{23}}{2}x^2\right) + C_1 d\,x + C_2 \\ &= \frac{1}{EI}\left(-\frac{k_{33}}{2}x^2 + \frac{k_{23}}{6}x^3\right) + C_1 x + C_2\end{aligned} \tag{35.}$$

Applying the boundary conditions, we obtain the following four equations

$$\begin{aligned}y(0) &= 0 = C_2 \\ \theta(0) &= 1 = C_1 \\ y(L) &= 0 = \frac{1}{EI}\left(-\frac{k_{33}}{2}L^2 + \frac{k_{23}}{6}L^3\right) + L \\ \theta(L) &= 0 = \frac{1}{EI}\left(-k_{33}L + \frac{k_{23}}{2}L^2\right) + 1\end{aligned} \tag{36.}$$

Solving the above equations, we obtain the values

$$k_{23} = \frac{6EI}{L^2} \qquad k_{33} = \frac{4EI}{L} \tag{37.}$$

Force equilibrium allows us to solve for the remaining two unknown stiffness terms

$$\sum F_y = 0 \rightarrow k_{53} = -k_{23} = -\frac{6EI}{L^2} \tag{38.}$$

$$\sum M_B = 0 \rightarrow k_{63} + k_{33} - k_{23}L = 0 \tag{39.}$$

$$\rightarrow k_{63} = k_{23}L - k_{33} = \frac{6EI}{L^2}L - \frac{4EI}{L} = \frac{2EI}{L}$$

The corresponding shape function is

$$N_2 = \frac{1}{EI}\left(-\frac{1}{2}\left(\frac{4EI}{L}\right)x^2 + \frac{1}{6}\left(-\frac{6EI}{L^2}\right)x^3\right) + x = x - \frac{2x^2}{L} + \frac{x^3}{L^2} \tag{40.}$$

Solving the two remaining cases for a unit displacement of the right node and finally a unit rotation of the right node, the two remaining rows (3rd and 4th) of the stiffness matrix can be determined, and the two remaining shape functions are given by

$$N_3 = \frac{3x^2}{L^2} - \frac{2x^3}{L^3} \tag{41.}$$

$$N_4 = -\frac{x^2}{L} + \frac{x^3}{L^2} \tag{42.}$$

You are to do those yourself in exercise 10.3.2. You might consider this an act of cruelty, but it is usually necessary for students in order to understand the direct stiffness method.

The element stiffness matrix is now given by

$$[k] = \frac{EI}{L^3}\begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & 6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \tag{43.}$$

As expected, the stiffness matrix can be observed to be symmetrical, and equation 26 can be written

$$\begin{pmatrix} F_A \\ M_A \\ F_B \\ M_B \end{pmatrix} = \frac{EI}{L^3}\begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix}\begin{pmatrix} u_A \\ \theta_A \\ u_B \\ \theta_B \end{pmatrix} \tag{44.}$$

This system of equations is however obtained for a beam with no longitudinal DOF, i.e. it cannot elongate. Adding the DOF's for elongation of a truss from equation 16, the following expression is obtained

$$\begin{pmatrix} N_A \\ F_A \\ M_A \\ N_B \\ F_B \\ M_B \end{pmatrix} = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} & 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix}\begin{pmatrix} \delta_A \\ u_A \\ \theta_A \\ \delta_B \\ u_B \\ \theta_B \end{pmatrix} \tag{45.}$$

This system of equations is on the form required for transformation, when analyzing beams not oriented along the global x-direction.

10.4.2 Transformation matrix and constrains

The transformation matrix derived in section 10.3.4 is still valid for transformation of the beam element system in equation 45. However, it has to be accounted for that rotational DOFs (3rd and 6th) do not require transformation. For a 6DOF beam element, the transformation matrix will therefore be arranged as follows

$$[\boldsymbol{T}] = \begin{bmatrix} \cos\theta & \sin\theta & 0 & & & \\ -\sin\theta & \cos\theta & 0 & & 0 & \\ 0 & 0 & 1 & & & \\ & & & \cos\theta & \sin\theta & 0 \\ & 0 & & -\sin\theta & \cos\theta & 0 \\ & & & 0 & 0 & 1 \end{bmatrix} \qquad 46.$$

The stiffness matrix in the global coordinate system can still be obtained using equation 23 and the principle for assembly of the stiffness matrices known from bar element are generally valid. Constrains for various common support types are shown in Figure 9.

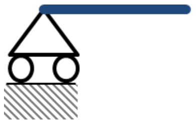
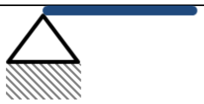
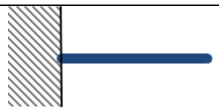
Support type	Constraint equation(s)	Nodal DOF-numbers constrained
	$\delta = 0$	1
	$\delta = 0$ $u = 0$	1,2
	$\delta = 0$ $u = 0$ $\theta = 0$	1,2,3

Figure 9 Nodal constraint equations corresponding common support types

10.4.3 Line loads: Work equivalent nodal forces

This far, only external loads applied in the nodes have been considered. However, structural analysis is often performed of structures with line loads. A constant line load would increase the order of the required polynomial in equation 25 by one. However, using energy methods (which we won't do in the present course), we may replace the distributed load with nodal loads that exerts the same work as the distributed load would. If doing so, loads as shown in Figure 10 can be derived.

You are to try this out on your own when solving problem 10.3.5.

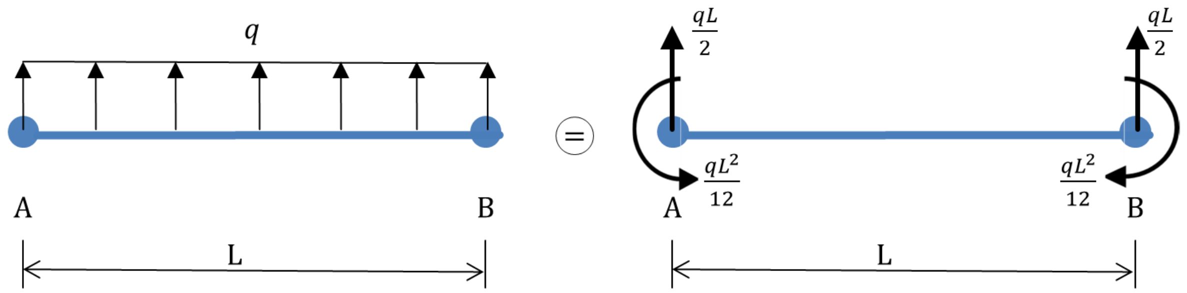


Figure 10 Work equivalent nodal loads replacing a distributed load

10.4.4 Modal analysis: buckling

In chapter 9, Euler buckling was considered by formulating the moment equilibrium of a beam in the deformed shape (assumed to be harmonic), see Figure 11-Left. The following differential equation was derived and we would for given boundary conditions solve this for P yielding the critical load P_{cr}.

$$EI\frac{d^2y}{dx^2}+Py=0 \qquad 47.$$

This type of buckling problem is usually referred as a bifurcation problem, since any compressive load larger than P_{cr}will cause the beam to snap from the initial perfectly straight state to a deformed state. For various boundary conditions, the critical load was obtained in terms of the Euler length L_E, see Figure 11-Right.

$$P_{cr}=\frac{\pi^2 EI}{(L_E)^2} \qquad 48.$$

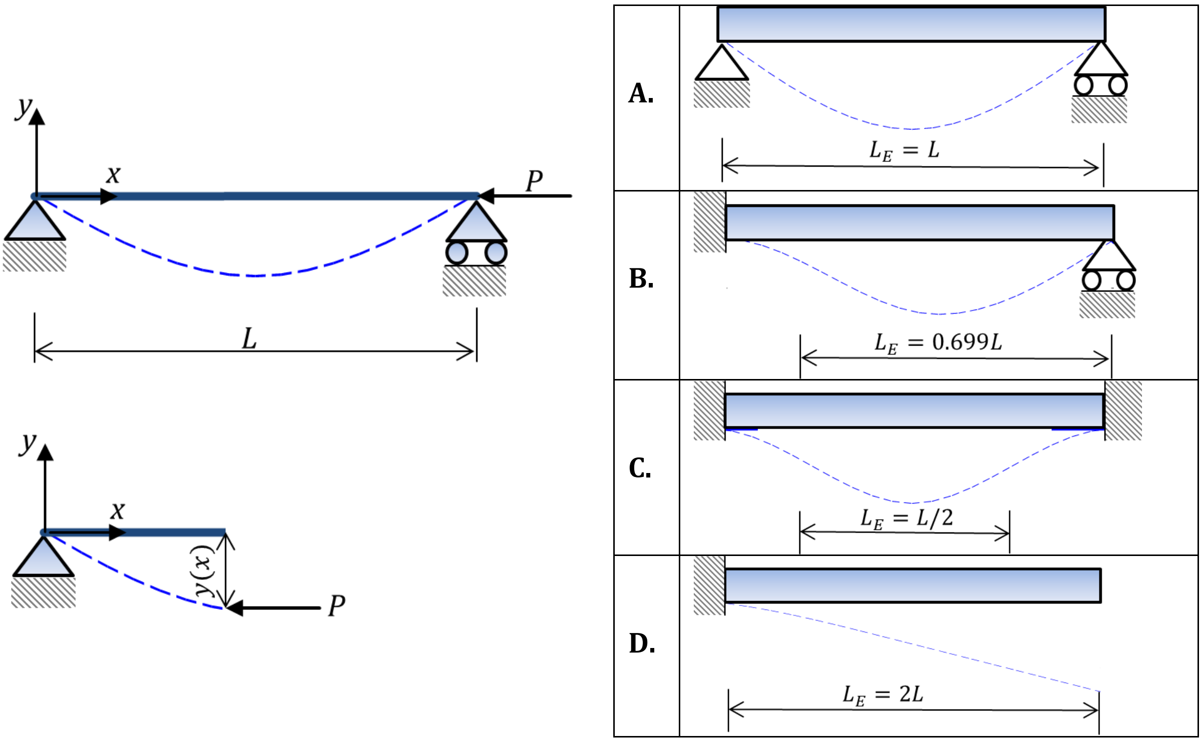


Figure 11, Left: Equilibrium of a long and slender beam in the deformed state, **Right:** Euler lengths for simple cases of supports

If we now apply the direct stiffness method with polynomial shape functions of 3rd order to equation 47, the following equation can be derived

$$(\boldsymbol{F})=[\boldsymbol{k}](\boldsymbol{X})+P[\boldsymbol{k}_G](\boldsymbol{X}) \qquad 49.$$

in which $[\boldsymbol{k}_G]$ is the geometrical (or stress) stiffness matrix, which can be derived to be on the form

$$[\boldsymbol{k}_G]=\frac{1}{30L}\begin{bmatrix} 36 & 3L & -36 & 3L \\ 3L & 4L^2 & -3L & -L^2 \\ 36 & -3L & 36 & -3L \\ 3L & -L^2 & -3L & 4L^2 \end{bmatrix} \qquad 50.$$

It is noticed, that $[\boldsymbol{k}_G]$ depends solely on geometrical parameters and not on materials. Setting $\lambda=-P$ and considering the case where external transverse forces and moments are zero, we obtain the following equation

$$([\boldsymbol{k}]-\lambda[\boldsymbol{k}_G])(\boldsymbol{X})=0 \qquad 51.$$

This is recognized as an eigenvalue problem, which can be solved setting the determinant of the total stiffness matrix equal to zero

$$|([\boldsymbol{k}]-\lambda[\boldsymbol{k}_G])|=0 \qquad 52.$$

The corresponding values for $(\boldsymbol{X})$ are the *mode shapes*. These represent the shape, but not magnitude, of the buckled beam geometry.

This technique for modal analysis is generally valid for buckling analysis of linear elastic structures, and geometrical stiffness matrices may be derived for other structural elements (fx. plates and shells). This turns out to be immensely useful for numerical analysis, since analytical stability analysis often gets quite complicated (the best books on the topic is probably still Timoshenkos epic classic textbook on elastic stability containing a large number of analytical solutions [3]).

10.4.5 Modal analysis: undamped eigenfrequencies

The technique applied for modal analysis to obtain the buckling loads and mode shapes in the previous chapter, can also be applied to determine the eigenfrequencies and mode shapes of beams vibrating without damping. The reader should notice, that this is the first time vibrations of continuous systems have been considered. The differential equation governing the undamped vibrations of a beam can be derived to be on the form

$$EI\frac{\partial^4 y}{\partial x^4}+m\frac{\partial^2 y}{\partial x^2}=q(x,t) \qquad 53.$$

We note, that y is a function of both position x and time t, which is why partial derives are now required. The differential equation has a harmonic solution with eigenfrequency given by

$$\omega_n \approx C\sqrt{\frac{EI}{SL^4}} \qquad 54.$$

in which S denotes mass per unit length (and can be calculated as density times cross-sectional area). The constant C is dependent on the boundary conditions and is given in Figure 12 for various cases of supports.

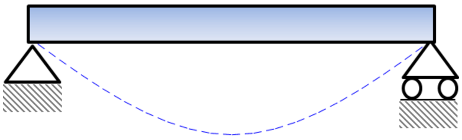
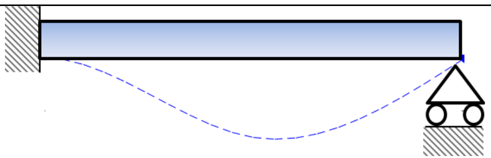
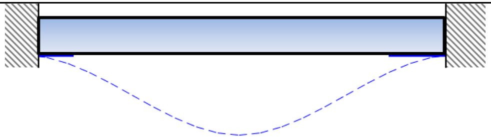
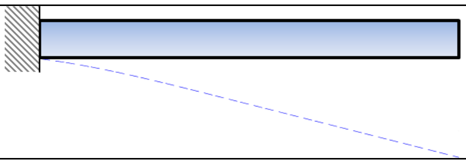
	Boundary conditions and mode shape	Constant C
A.		9.87
B.		15.3
C.		22.4
D.		3.51

Figure 12 Mode shapes and coefficient for eigenfrequency calculation for simple beam cases

Again, applying the direct stiffness method to equation 53. with third order polynomials as shape functions, the following linear system is derived

$$[\mathbf{M}]\{\ddot{\boldsymbol{x}}\}+[\mathbf{K}]\{\boldsymbol{x}\}=0 \qquad 55.$$

In which $[\mathbf{M}]$ is the mass matrix of the system. The mass matrix usually comes on two forms: an inaccurate, but simple form called the lumped mass matrix, in which the mass is assumed concentrated in the nodes, and a more accurate, but complex form called the consistent mass matrix, derived on basis of a distributed mass

$$[\mathbf{M}]_{consistent}=\frac{SL}{420}\begin{bmatrix}156 & 22L & 54 & -13L\\ 22L & 4L^2 & 13L & -3L^2\\ 54 & 13L & 156 & -22L\\ -13L & -3L^2 & -22L & 4L^2\end{bmatrix}\quad [\mathbf{M}]_{lumped}=\frac{SL}{2}\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0\end{bmatrix} \qquad 56.$$

Assuming the solution harmonic for all DOF's, we obtain

$$\{\boldsymbol{x}\}=\{\boldsymbol{x}\}_m\sin(\omega_n t+\varphi)\rightarrow\{\ddot{\boldsymbol{x}}\}=-(\omega_n)^2\{\boldsymbol{x}\}_m\sin(\omega_n t+\varphi) \qquad 57.$$

Substituting this into the dynamics equilibrium, the following equation is derived

$$\begin{aligned}&\{\boldsymbol{x}\}=\{\boldsymbol{x}\}_m\sin(\omega_n t+\varphi)\rightarrow\{\ddot{\boldsymbol{x}}\}=-(\omega_n)^2\{\boldsymbol{x}\}_m\sin(\omega_n t+\varphi)\\ &[\mathbf{M}](-(\omega_n)^2\{\boldsymbol{x}\}_m\sin(\omega_n t+\varphi))+[\mathbf{K}](\{\boldsymbol{x}\}_m\sin(\omega_n t+\varphi))=0\\ &\rightarrow([\mathbf{K}]-(\omega_n)^2[\mathbf{M}])(\{\boldsymbol{x}\}_m\sin(\omega_n t+\varphi))=0\end{aligned} \qquad 58.$$

The derived equation will have a non-trivial solution when

$$([\mathbf{K}]-(\omega_n)^2[\mathbf{M}])\{\boldsymbol{x}\}_m=0 \qquad 59.$$

Which is recognized as an eigenvalue problem, that can be solved using determinants.

$$|([\mathbf{K}]-(\omega_n)^2[\mathbf{M}])|=0 \qquad 60.$$

Again, this technique is generally valid for other linear structures. The best book on structural dynamics is probably Humar's epic classic [4]– that's the best place to look if you need more information about this topic.

10.4.6 Calculated example: simple supported pipe shaped beam (With Matlab – the longest example ever !!!)

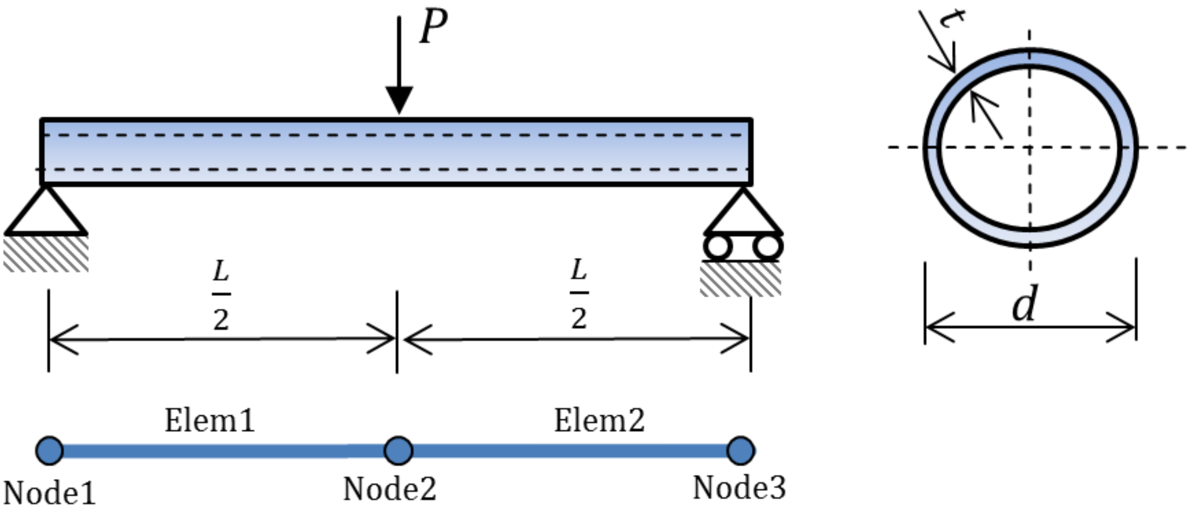


Figure 13 Simply supported pipe with central load

<u>Problem:</u> The system shown in Figure 13 has geometry defined by the following set of parameters

$d = 0.1\,m \qquad t = 0.01\,m \qquad L = 2\,m$

The pipe is made of steel with material parameters

$E = 2.1 \cdot 10^{11}\,\frac{N}{m^2} \qquad \rho = 7800\,\frac{kg}{m^3}$

A load of $P = 10 \cdot 10^3 N$ is applied at mid-span.

We will calculate:

a) the unconstrained global stiffness matrix
b) the constrained stiffness matrix
c) the equation of the elastic curve
d) the internal moment curve in the beam
e) the maximum deformation of the setup if geometry, loads and BCDs are rotated 90 deg

<u>Solution:</u> First of all, we will define all input parameters

```
%Input section %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Material parameters
E=2.1*10^11;     %Module of elasticity [N/m^2]
rho=7800;        %Density in [kg/m^3]
%Geometry
d=0.1;           %Radius
t=0.01;          %Wall Thickness
L=2;             %Beam length
%Loads
P=-10*10^3;      %End load [N]
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Then we calculate the cross-sectional constants and set the element length

```
%Geometry postprocessing;
A=pi/4*(d^2-(d-2*t)^2);       %cross-sectional area, rectangular cross-section
I=pi/64*(d^4-(d-2*t)^4);        %second order area moment of inertia, rectangular
cross-section
L_elem=L/2;                   %set element length
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

In Figure 13, nodes and elements are defined. Is noticed that the displacement vector is given by

$$(\boldsymbol{X}) = [\delta_1 \quad u_1 \quad \theta_1 \quad \delta_2 \quad u_2 \quad \theta_2 \quad \delta_3 \quad u_3 \quad \theta_3]^T$$

a) Now wanting to determine the global stiffness matrix, the element-wise stiffness matrices are calculated from 43. Since both elements are horizontal, the local and global x-axis coincide and there is no need to transform those to change the spatial orientation

$$[\boldsymbol{k}]_{Elem1} \begin{bmatrix} \begin{bmatrix} \frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix}_{11} & \begin{bmatrix} -\frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} \end{bmatrix}_{12} \\ \begin{bmatrix} -\frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & \frac{6EI}{L^2} & \frac{2EI}{L} \end{bmatrix}_{21} & \begin{bmatrix} \frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix}_{22} \end{bmatrix} = \begin{bmatrix} [\boldsymbol{k}]_{11} & [\boldsymbol{k}]_{12} \\ [\boldsymbol{k}]_{21} & [\boldsymbol{k}]_{22} \end{bmatrix}$$

$$[\boldsymbol{k}]_{Elem2}\begin{bmatrix} \begin{bmatrix}\frac{EA}{L}&0&0\\0&\frac{12EI}{L^3}&\frac{6EI}{L^2}\\0&\frac{6EI}{L^2}&\frac{4EI}{L}\end{bmatrix}_{22} & \begin{bmatrix}-\frac{EA}{L}&0&0\\0&-\frac{12EI}{L^3}&\frac{6EI}{L^2}\\0&-\frac{6EI}{L^2}&\frac{2EI}{L}\end{bmatrix}_{23}\\ \begin{bmatrix}-\frac{EA}{L}&0&0\\0&-\frac{12EI}{L^3}&\frac{6EI}{L^2}\\0&\frac{6EI}{L^2}&\frac{2EI}{L}\end{bmatrix}_{32} & \begin{bmatrix}\frac{EA}{L}&0&0\\0&\frac{12EI}{L^3}&-\frac{6EI}{L^2}\\0&-\frac{6EI}{L^2}&\frac{4EI}{L}\end{bmatrix}_{33}\end{bmatrix} = \begin{bmatrix}[\boldsymbol{k}]_{22}&[\boldsymbol{k}]_{23}\\ [\boldsymbol{k}]_{32}&[\boldsymbol{k}]_{33}\end{bmatrix}$$

The unconstrained global stiffness matrix is now obtained as

$$[\boldsymbol{K}]^{(U)}=\begin{bmatrix}[\boldsymbol{k}]_{11}&[\boldsymbol{k}]_{12}&0\\ [\boldsymbol{k}]_{21}&[\boldsymbol{k}]_{22}+[\boldsymbol{k}]_{22}&[\boldsymbol{k}]_{23}\\ 0&[\boldsymbol{k}]_{32}&[\boldsymbol{k}]_{33}\end{bmatrix}$$

```
%a) Setup unconstrained element stiffness matrix (local)
    k_elem(1:6,1:6)=0;        %Initialize array
    k_elem(1,1)=E*A/L_elem;       k_elem(1,4)=-E*A/L_elem;
    k_elem(2,2)=12*E*I/L_elem^3;  k_elem(2,3)=6*E*I/L_elem^2;
    k_elem(2,5)=-12*E*I/L_elem^3;k_elem(2,6)=6*E*I/L_elem^2;
    k_elem(3,2)=6*E*I/L_elem^2;   k_elem(3,3)=4*E*I/L_elem;
    k_elem(3,5)=-6*E*I/L_elem^2;  k_elem(3,6)=2*E*I/L_elem;
    k_elem(4,1)=-E*A/L_elem;      k_elem(4,4)=E*A/L_elem;
    k_elem(5,2)=-12*E*I/L_elem^3;k_elem(5,3)=-6*E*I/L_elem^2;
    k_elem(5,5)=12*E*I/L_elem^3;  k_elem(5,6)=-6*E*I/L_elem^2;
    k_elem(6,2)=6*E*I/L_elem^2;   k_elem(6,3)=2*E*I/L_elem;
    k_elem(6,5)=-6*E*I/L_elem^2;  k_elem(6,6)=4*E*I/L_elem;

    disp('The element stiffness matrix is'); k_elem

%Setup the global global unconstrained stiffness matrix
    K(1:9,1:9)=0;                        %Initialize array
    K(1:6,1:6)=K(1:6,1:6)+k_elem;        %Add 1st element
    K(4:9,4:9)=K(4:9,4:9)+k_elem;        %Add 2nd element
```

b) The constrained stiffness matrix is obtained by removing the rows and columns corresponding to the constrained DOF's in accordance with the constraint equations given in Figure 9. The fixed support to the left will fix both displacement coordinates, so $\delta_1 = 0$ and $u_1 = 0$. The moving support to the right will fix the vertical displacements only, so $u_3 = 0$. The DOF's to be constraint have numbers 1,2 and 8. We therefore eliminate the corresponding rows and columns in the stiffness matrix (starting with the largest number) to obtain $[\boldsymbol{K}]^{(C)}$

```
  %Determine the constrained stiffness matrix by removing DOFs (eliminate row and
   columns in k_elem starting with the largest)
    K(8,:)=[];               K(:,8)=[];    %Right moving support (fix y)
    K(2,:)=[];               K(:,2)=[];    %Left fixed support (fix y)
    K(1,:)=[];               K(:,1)=[];    %Left fixed support (fix x)
    %Note: nDOF = NumCor - nConstrains = 3*3 - 3 = 6
```

c) Having obtained the constrained stiffness matrix, we may solve for deformation using the nodal force vector

$$(\boldsymbol{F}) = [M_1 \quad N_2 \quad F_2 \quad M_2 \quad N_3 \quad M_3]^T$$
$$= [0 \quad 0 \quad -P \quad 0 \quad 0 \quad 0]^T$$

The deformations are now obtained by

$$(\boldsymbol{X}) = [\theta_1 \quad \delta_2 \quad u_2 \quad \theta_2 \quad \delta_3 \quad \theta_3]^T = \left([\boldsymbol{K}]^{(C)}\right)^{-1}\left(\boldsymbol{F}^{(C)}\right)$$
$$= [\,-0.0041\,rad \qquad 0 \quad -0.0027\,m \qquad 0 \qquad 0 \quad 0.0041\,rad]^T$$

The analytical value for the max. deflection at the midpoint is

$$\delta_2^{(A)} = \frac{PL^3}{48EI} = -0.0027\,m$$

which is equal to the value calculated by numerical analysis.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%c) Determine the maximum deflection
    F=[0;0;P;0;0;0];        %Define load vector,
    X=K\F;                  %Solve for deformations
    yMax=X(3);              %Maximum numerical deformation in y-direction (for x=L)
    yMaxA=P*L^3/(48*E*I);   %Calculate maximum analytical deformation in y-direction
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

In order to calculate the equation of the elastic curve, we will utilize that the deflections over each element are a polynomial of third order. We have

$$y = ax^3 + bx^2 + cx + d$$
$$\frac{dy}{dx} = 3ax^2 + 2bx + c$$

Now considering the first element and inserting $(x_1, y_1) = (0{,}0)$, $(x_2, y_2) = \left(\frac{L}{2}, u_2\right)$, $\left(x, \frac{dy}{dx}\right) = (0, \theta_1)$ and $\left(x, \frac{dy}{dx}\right) = \left(\frac{L}{2}, \theta_2\right)$, the following linear system of equations is obtained and solved for the polynomial coefficients *a*,*b*,*c* and *d*

$$\begin{pmatrix} y_1 \\ y_2 \\ \theta_1 \\ \theta_2 \end{pmatrix} = \begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ 3x_1^2 & 2x_1 & 1 & 0 \\ 3x_2^2 & 2x_2 & 1 & 0 \end{bmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$$

These are in the Matlab implementation simply contained in a vector $Z = [a \quad b \quad c \quad d]^T$
The determined polynomial can be compared with the analytic expression

$$y^{(A)} = \frac{P}{48EI}(4x^3 - 3L^2x)$$

```
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Calculate the equation of the elastic curve by cubic interpolation from x=0 to
    x=L/2
    x1=0;        y1=0;       tht1=X(1);      %BCDS in left end (wall-section)
    x2=L/2;      y2=X(3);    tht2=X(4);      %Extract values from solution (left)

    CoefMat(1:4,1:4)=0;     %Initialize array
    %Setup coefficient matrix and right-hand-side vector
    CoefMat(1,:)=[x1^3,x1^2,x1,1];
    CoefMat(2,:)=[x2^3,x2^2,x2,1];
    CoefMat(3,:)=[3*x1^2,2*x1,1,0];
    CoefMat(4,:)=[3*x2^2,2*x2,1,0];
    B=[y1;y2;tht1;tht2];
    Z=CoefMat\B;
```

```
%Evaluate and plot the obtained solution over the length of the beam on a grid
 of n+1 points
 n=10;
for i=1:n+1
    x(i)=(L/2)/n*(i-1);                              %X-along the beam
    y(i)=Z(1)*x(i)^3+Z(2)*x(i)^2+Z(3)*x(i)+Z(4);     %interpolated deflection
    yA(i)=-P/(48*E*I)*(4*x(i)^3-3*L^2*x(i));
end

figure; plot(x,y,'b.',x,yA,'b-'
xlabel('Length position along beam element [m]')
ylabel('Deflection [m]')
legend('Interpolation from FEM solution','Analytical solution')
grid on
```

d) The internal bending moment in the beam is obtained by differentiation of the rotation response

$$M(x) = EI(6ax + 2b)$$

```
%d) Calculate the internal bending moment curve
    for i=1:n+1
        M(i)=E*I*(6*Z(1)*x(i)+2*Z(2));       %interpolated deflection
        MAna(i)=-P/2*x(i);
    end

    figure; plot(x,M,'b.',x,MAna,'b-');
    xlabel('Length position along beam [m]')
    ylabel('Internal bending moment [Nm]')
    legend('Interpolation from FEM solution','Analytical solution')
    grid on
```

The obtained results are shown in Figure 14 and Figure 15.

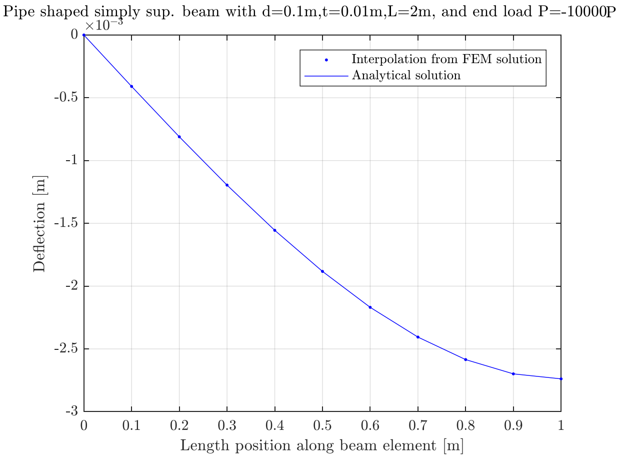


Figure 14 Elastic curve, 1st element

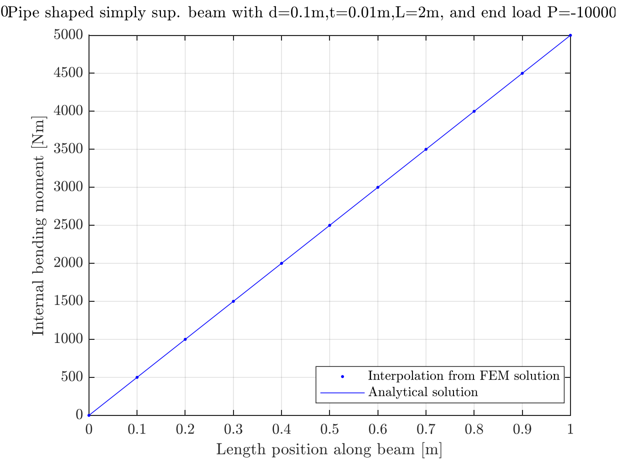


Figure 15 Bending moment, 1st element

e) For a 90 deg. rotation, we assemble a global rotation matrix and apply equation 22 to obtain the global stiffness matrix in the rotated frame

```
alpha=90*pi/180;        %Define angle of rotation
%Setup the transformation matrix for a beam element
T(1:3,1:3)=0;    %Initialize array
T(1,1)=cos(alpha);   T(1,2)=sin(alpha);
T(2,1)=-sin(alpha);  T(2,2)=cos(alpha);
T(3,3)=1;
TG=[T,zeros(3,6);zeros(3,3),T,zeros(3,3);zeros(3,6),T];
TG(8,:)=[];             TG(:,8)=[];     %Right moving support (fix y)
TG(2,:)=[];             TG(:,2)=[];     %Left fixed support (fix y)
TG(1,:)=[];             TG(:,1)=[];     %Left fixed support (fix x)
K_ROT=TG'*K*TG;
F(1:6)=0; F(2)=P;
X=K_ROT\F;                              %Solve for deformations
xMax=X(2);
```

The obtained deflection is as expected equal to the value obtained in b).

10.4.7 Calculated example: MrBeam

The example in the previous section can also be solved using the plane beam FEA code MrBeam[1] on basis of the following input commands ran after the input block from a) in the previous example. The deformed shape in Figure 18 and the moment curves in Figure 16 and Figure 17 look as expected. The model is contained in the example input file `ExModel_SimplySuppertedPipe.m`

```
%Horisontal simply supperted pipe
Nodes=[0,0;L/2,0;L,0];
BeamElem=[1,2,A,I,E;2,3,A,I,E];
PointLoads=[2,2,P];
%Constraint simply support
Constrains=[1,2;3,2];
```

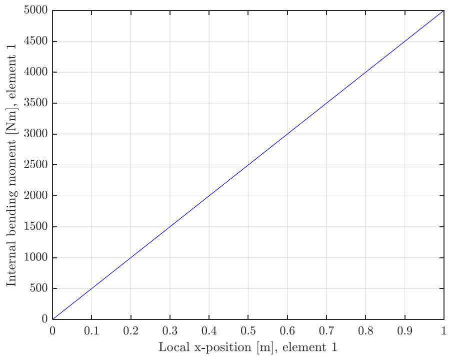


Figure 16 Moment curve, 1st element

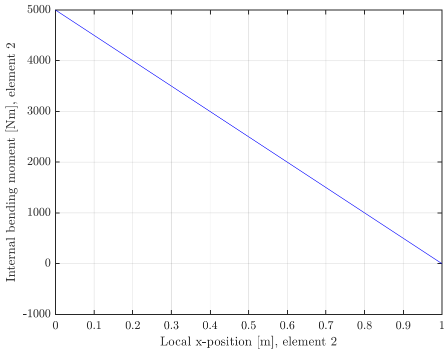


Figure 17 Moment curve, 2nd element

[1] Name invented by former mech. Eng. Student Daniel Neubauer. This is highly appreciated by the author, since the code inevitably would have been after a weird fantasy character with it's own web-comic, should I have made the decision

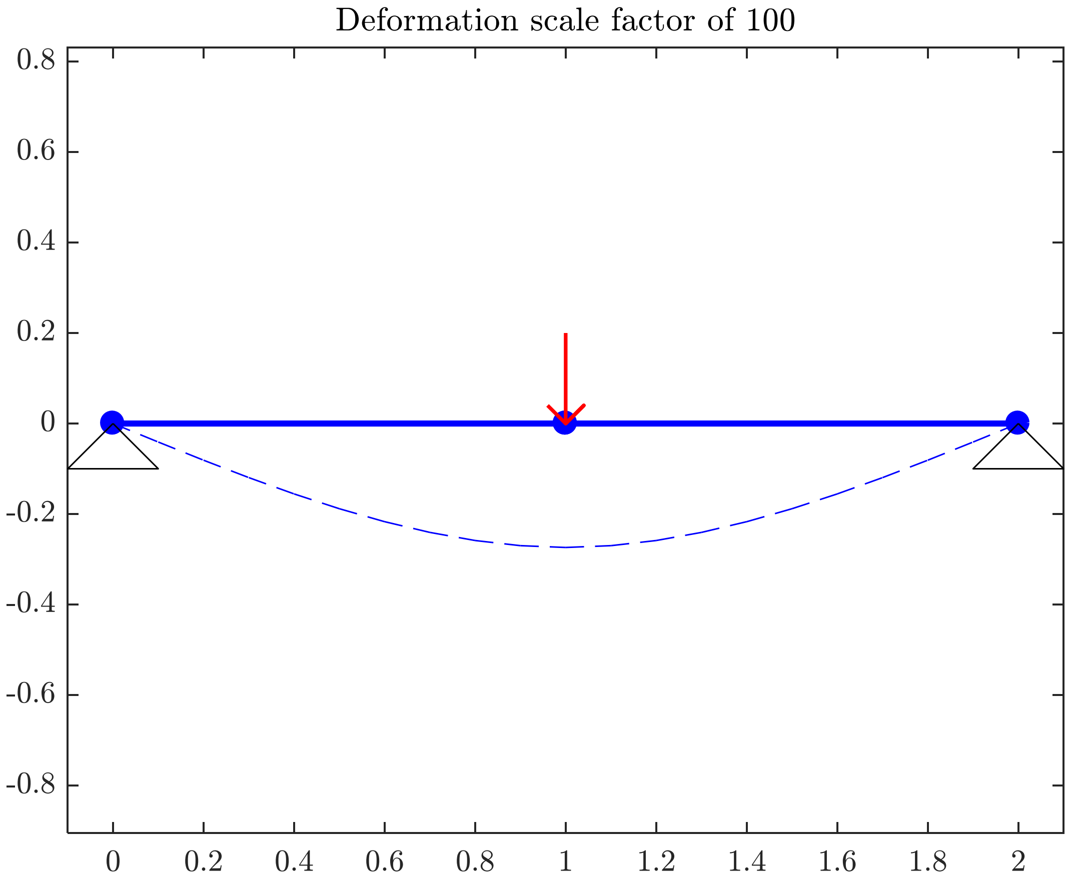


Figure 18 Beam model and deflected shape

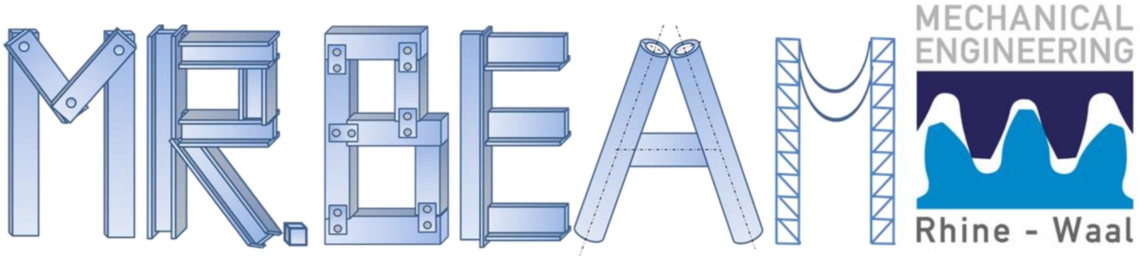


Figure 19 The MrBeam logo

10.4.8 Calculated example: modal analyses

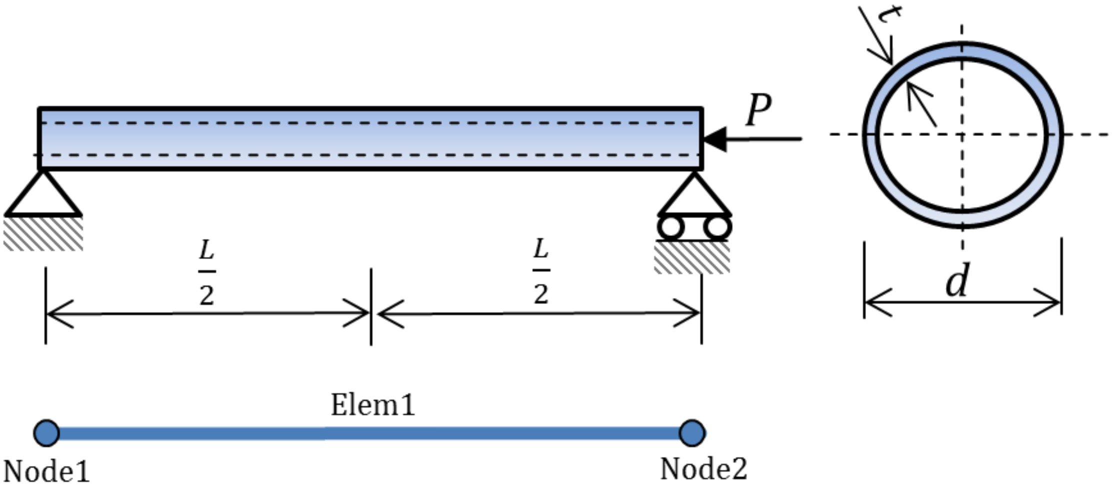


Figure 20 Critical load of simply supported pipe shaped beam

Problem: for geometry and material properties like those given in section 10.4.6, determine the critical load of the beam.

Solution: Since the beam is oriented horizontally, the local and global x-axes coincide and we do not need the DOF-corresponding to extension. We may therefore divide the structure into a single element, and obtain the stiffness matrix along with the geometric stiffness matrix using equations 43 and 50 respectively. After constraining both matrices, the standard eigenvalue solver in Matlab, `eigs`, will be applied to determine the eigenvalue, which can be compared to the analytically calculated critical load.

```
clear k_elem; L_elem=L;
k_elem(1:4,1:4)=0;           %Initialize array
k_elem(1,1)=12*E*I/L_elem^3; k_elem(1,2)=6*E*I/L_elem^2;
k_elem(1,3)=-12*E*I/L_elem^3;k_elem(1,4)=6*E*I/L_elem^2;
k_elem(2,1)=6*E*I/L_elem^2;  k_elem(2,2)=4*E*I/L_elem;
k_elem(2,3)=-6*E*I/L_elem^2; k_elem(2,4)=2*E*I/L_elem;
k_elem(3,1)=-12*E*I/L_elem^3;k_elem(3,2)=-6*E*I/L_elem^2;
k_elem(3,3)=12*E*I/L_elem^3; k_elem(3,4)=-6*E*I/L_elem^2;
k_elem(4,1)=6*E*I/L_elem^2;  k_elem(4,2)=2*E*I/L_elem;
k_elem(4,3)=-6*E*I/L_elem^2; k_elem(4,4)=4*E*I/L_elem;

kG_elem(1:4,1:4)=0;          %Initialize array
kG_elem(1,1)=6/(5*L_elem);   kG_elem(1,2)=1/10;
kG_elem(1,3)=-kG_elem(1,1);  kG_elem(1,4)=kG_elem(1,2);
kG_elem(2,1)=kG_elem(1,2);   kG_elem(2,2)=2*L_elem/15;
kG_elem(2,3)=-kG_elem(1,4);  kG_elem(2,4)=-L_elem/30;
kG_elem(3,1)=-kG_elem(1,1);  kG_elem(3,2)=-kG_elem(1,2);
kG_elem(3,3)=kG_elem(1,1);   kG_elem(3,4)=-kG_elem(1,2);
kG_elem(4,1)=kG_elem(1,4);   kG_elem(4,2)=kG_elem(2,4);
kG_elem(4,3)=kG_elem(3,4);   kG_elem(4,4)=kG_elem(2,2);

K=k_elem;   KG=kG_elem;
K(3,:)=[];  K(:,3)=[];       KG(3,:)=[];  KG(:,3)=[];
K(1,:)=[];  K(:,1)=[];       KG(1,:)=[];  KG(:,1)=[];

[Modes,EigVals]=eigs(K,KG)  %Solve eigenvalue problem
Pcr=EigVals(1,1)            %Extract critical load
PcrA=pi^2*E*I/L^2           %Compare to analytical value
```

10.4.9 Frame structures

One of the major advantages of the developed methods is, that these enable analysis of frame structures by solving systems of equations using linear algebra. The techniques demonstrated in section 10.4.6 can easily be applied to develop larger structures. This section should contain a calculated example, but the author decided to turn this into exercise problems (as an act of cruelty). However, an example of a frame structure is shown below in Figure 21.

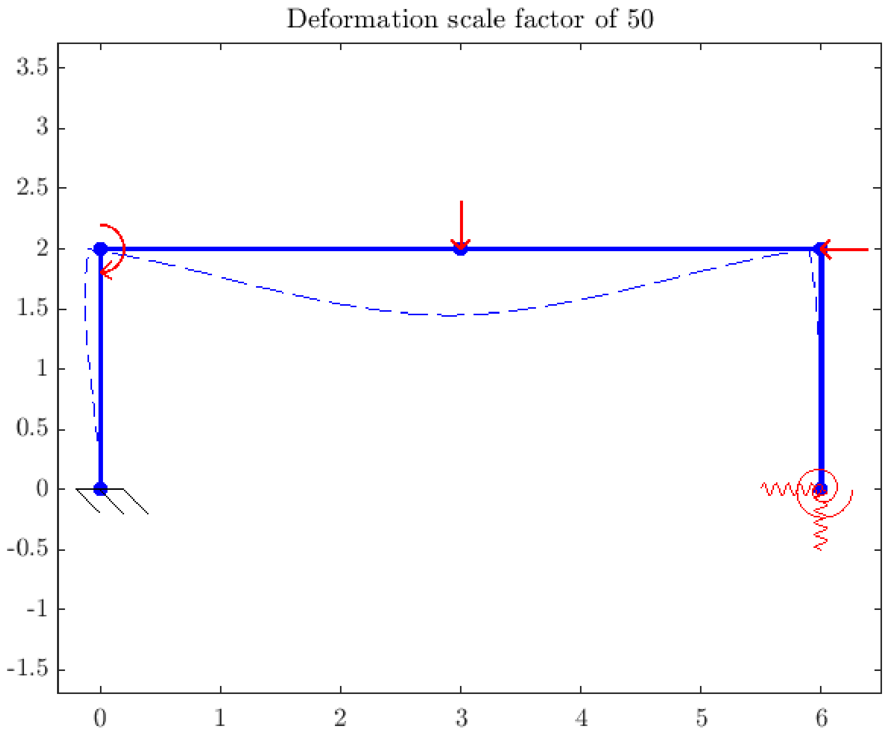
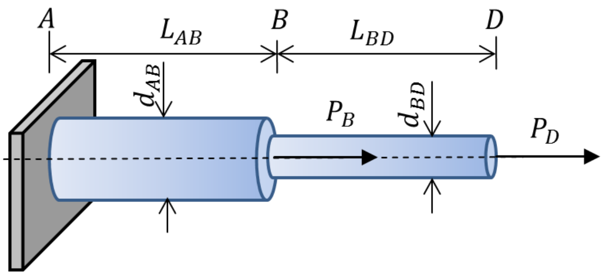


Figure 21 Example of frame structure modeled using MrBeam

Problems

Bars and trusses



Problem 10.2.1

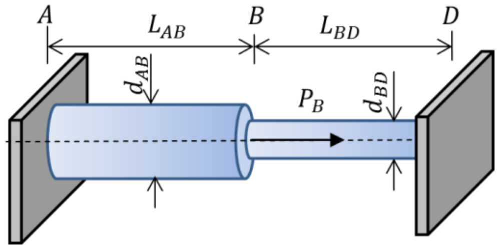
The compound cylinder ABD is fixed against deformations in section A and subjected to axial loads $P_B = 300kN$ and $P_D = 200kN$ as shown in the figure. The geometry is given by the following set of parameters:

$d_{AB} = 120\ mm \qquad d_{BD} = 100\ mm$

$L_{AB} = L_{BD} = 520\ mm$

The cylinder is made of structural steel with elastic modulus of 210 GPa. Determine the deformation of the points B and D using a) classical mechanics, b) matrix methods

Ans: $\delta_B = 0.105\ mm, \delta_D = 0.166\ mm$

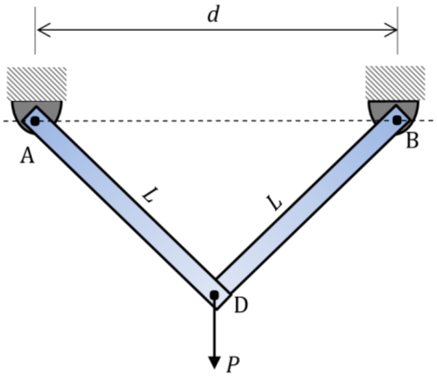


Problem 10.2.2

For the compound cylinder in the previous problem, the load in D is replaced with a wall, so both ends A and D are fixed against deformations. Determine the deformation of point B and the reactions in A and D using a) classical mechanics, b) matrix methods

Ans: $\delta_B = 3.73 \cdot 10^{-5}\ m,$

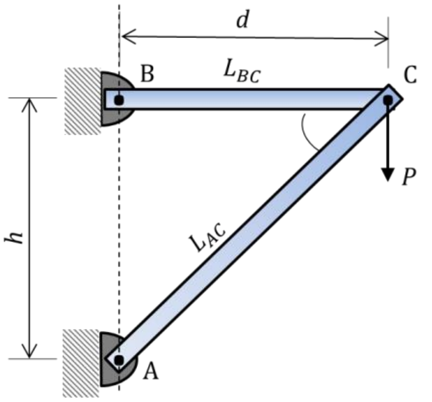
$R_A = -1.77 \cdot 10^5 N, R_D = 1.23 \cdot 10^5 N$



Problem 10.2.3

The frame ADB is constituted by truss elements with cross-sectional areas $A = 9.0 \cdot 10^{-4} m^2$ and length $L = 1.5\ m$ pinned in A and B at a distance $d = 2.0\ m$. Point D is subjected to a vertical load $P = 20\ kN$. The trusses are made of structural steel with elastic modulus of 210 GPa. Determine the internal forces in both members using a) classical mechanics, b) matrix methods, c) draft the input file for the Matlab truss script `MyFirstFEACode.m`

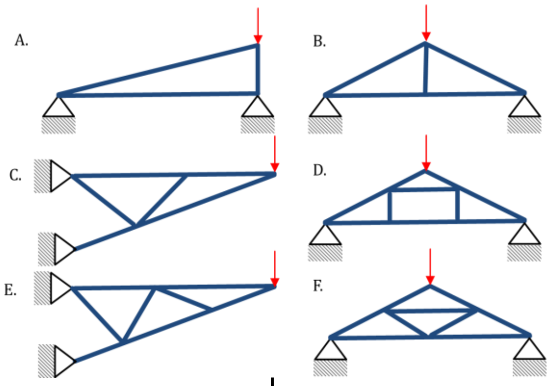
Ans: $F = 1.34 \cdot 10^4 N$



Problem 10.2.4

The truss console structure ACB is made up of elements with cross-sectional areas $A = 1.2 \cdot 10^{-3} m^2$ and is subjected to a vertical load $P = 40\ kN$ in point C. The dimensions of the trusses are given by the parameters $h = 2m$ and $b = 1m$. Determine the internal forces in both members using a) classical mechanics, b) matrix methods, c) draft the input file for the Matlab truss script `MyFirstFEACode.m`

Ans: $F_{AC} = -4.47 \cdot 10^4 N, F_{BC} = 2.00 \cdot 10^4 N$

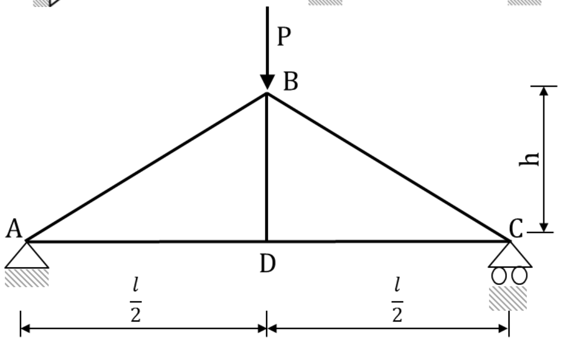


Problem 10.2.5

For the truss structures to the left determine the number of

a) Nodes
b) Elements
c) DOF's for the unconstrained system
d) DOF's for the constrained system

Furthermore, determine the size of the constrained stiffness matrix



Problem 10.2.6

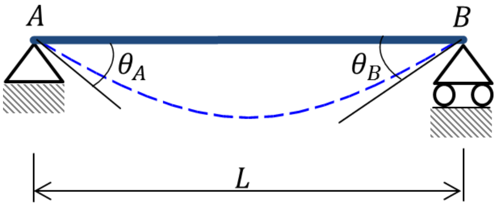
The frame shown is loaded by a vertical force P=245 kN in point B. The cross sectional areas are 2500 mm^2 and 1800 mm^2 for trusses AB and AD. All trusses are made of structural steel with elastic modulus E=210 GPa. The frame is of height h=3 m and length l=10 m.

For all members in the frame, determine a) the internal forces using a) classical mechanics, b) matrix methods, c) draft the input file for the Matlab truss script `MyFirstFEACode.m`

HINT: you'll need Matlab from question b)

a) F_{AB}=-238.1 kN, F_{AD}=204.2 kN

Beams

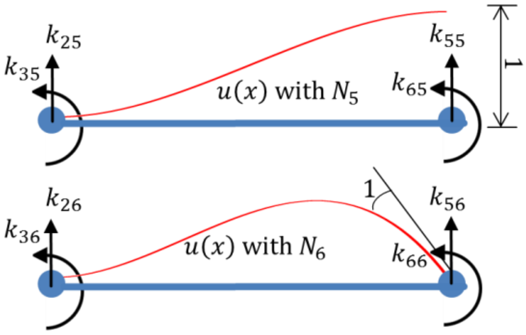


Problem 10.3.1

A simply supported beam of length L=$2m$ is subjected to some loads. These are unfortunately unknown. However, we have measured the rotations in each end to be of magnitude $\theta = 2.86deg$. If the beam is modeled using a single element a) Determine the equation of the elastic curve, b) Estimate the average curvature, c) how would you estimate the maximum curvature of the beam?

Ans.: $y = 0 \cdot x^3 + 0.025m^{-1}x^2 - 0.05x$

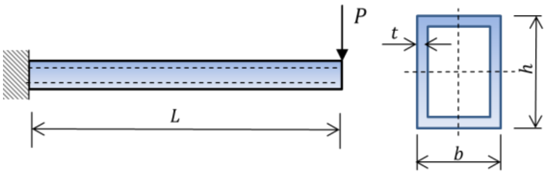
$\kappa_{av} = 0.05m^{-1}$



Problem 10.3.2

Carefully review the derivation of the two first rows of the stiffness matrix in section 10.4.1, before you (entirely on your own) derive the two last rows (3 and 4).

Ans: see equation 44



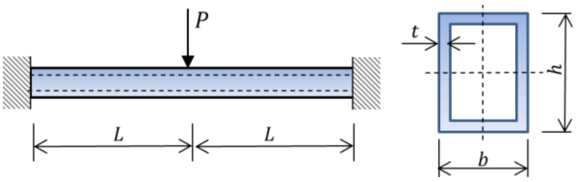
Problem 10.3.3

A cantilevered rectangular hollow beam with height $h = 0.1m$, width $b = 0.05m$, wall thickness $t = 0.01m$, and length $L = 2.0m$ is subjected to a concentrated end load of magnitude *P=10kN*. The beam is made of structural steel with elastic modulus of 210 GPa. Determine the maximum deflection, the equation of the elastic curve and the equation of the internal bending moment curve using a) classical mechanics, b) matrix methods

Ans: $u_{max} = -0.044m,$

$y = 0.0027x^3 - 0.0165x^2$

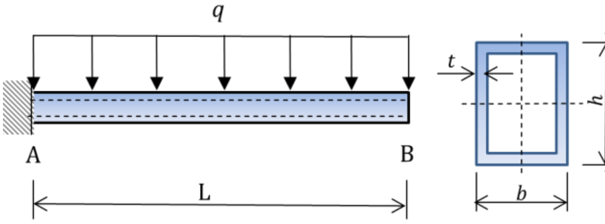
$M = 9.82 \cdot 10^3 x - 2.0 \cdot 10^4$



Problem 10.3.4

A beam defined by the same set of parameters as in the previous problem (double length) is subjected to a central load P as double cantilever. Determine the support moments and reactions using a) classical mechanics, b) matrix methods

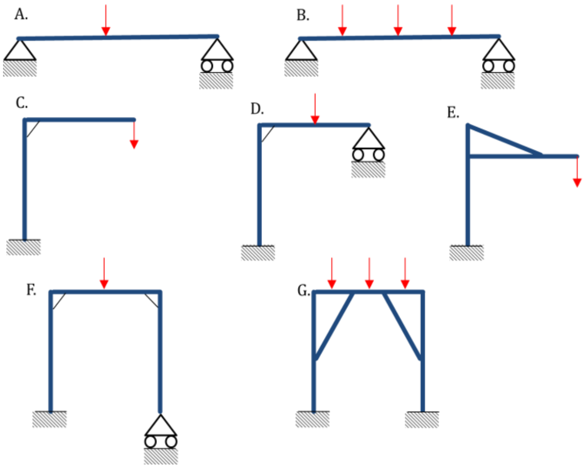
Ans: *M=-5000Nm, V=-5000N*



Problem 10.3.5

A beam with geometry like in problem 10.3.3 is subjected to a distributed load $q = 1000N/m$. Determine the end deflection and rotation using a) classical mechanics, b) matrix methods

Ans: $u_B = -0.033\ m, \theta_B = -0.022rad$



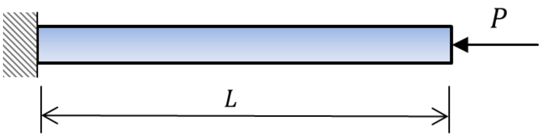
Problem 10.3.6

For the plane beam structures to the left determine the number of

a) Nodes
b) Elements
c) DOF's for the unconstrained system
d) DOF's for the constrained system

Furthermore, determine the size of the constrained stiffness matrix

Modal analysis

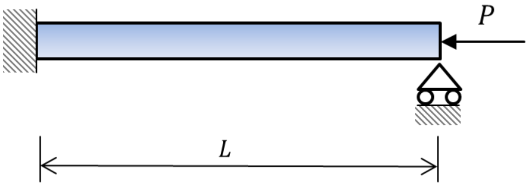


Problem 10.3.7

Determine the critical compressive load of a cantilever beam using a single beam element in terms of *E*,*I*, and *L*. Calculate the error with respect to the analytical critical Euler load

Hint: yes, that's right – you have to solve an eigenvalue problem

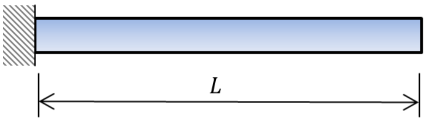
Ans: 0.8%



Problem 10.3.8

Determine the critical compressive load of an end supported cantilever beam using a single beam element in terms of *E*,*I*, and *L*. Calculate the error with respect to the analytical critical Euler load

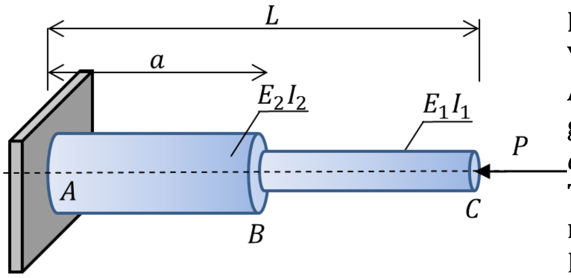
Ans: 49%



Problem 10.3.9

Determine the lowest eigenfrequency of a cantilever beam using a single beam element in terms of *E*,*I*,*L*, and *S*. Calculalte the error with respect to the analytical eigenfrequency

Ans: 0.6%



Problem 10.3.10

You'll need Matlab for this problem !

A compound cylinder is defined by the following geometrical set of parameters:

$d_{AB} = 0.1\ m,\ d_{BC} = 0.084\ m, L = 2\ m, a = 1.33\ m$

The beam is made of structural steel with elastic modulus of 210 GPa.

Determine the critical compressive load of the cylinder. Check your results against the solution tables in Roark, pp. 718 [5]

Hint for Problem 10.3.10: define a function of *E*,*I* and *L* which returns the element wise stiffness matrix. In a similar fashion, define a function of *L* which returns the geometrical stiffness matrix. This makes life so much easier

Problem 10.3.11 – solve with Matlab

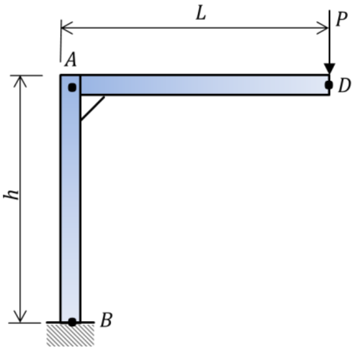
How would you, for all modal problems in the current section, check the validity of the obtained eigenvalues

Frames

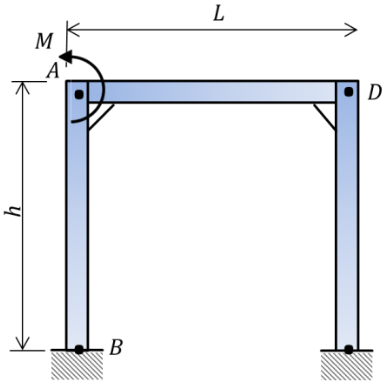
In the following problems, the cross-sectional dimensions are the same as in problem 10.3.3.
The load (if any) is $P = 10kN$ downwards and the moment (if present) is $M = 1000Nm$.
The length dimension is *L=2 m* and the height dimension is *h=3 m*.
Determine the nodal deflections and the internal bending moments along the lengths of all members using:

a) Your own Matlab scripts (Hint: remember that functions are handy)
b) The `MrBeam.m` Matlab script

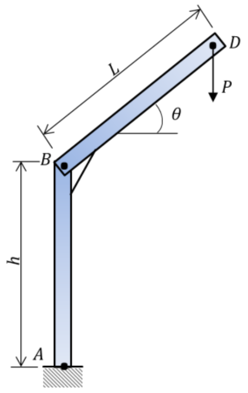
Problem 10.3.12



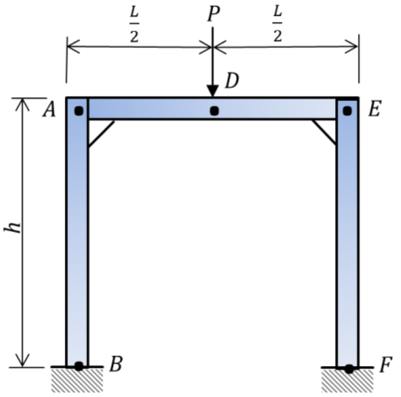
Problem 10.3.13



Problem 10.3.14



Problem 10.3.15



References

[1] Lecture notes on finite element analysis (Introduction to FEA theory for dummies), NH Østergaard, HSRW, 2015

[2] Introduction to finite element methods (download here), Online lecture notes by Carlos Felippa, University of Colorado at Boulder, 2005

[3] Theory of elastic stability, SP Timoshenko & JM Gere, Courier Corporation, 2012

[4] JL Humar, Dynamics of structures, CRC-Press, 2012

[5] Roark's formulaes for stress and strain, WC Young & RG Budynas, McGraw-Hill, 2002

[6] Course FEA scripts for download: https://drive.google.com/drive/folders/1fpAf_pt-i1-WJtKz4mJIpMAJtwddw-VG

[7] GNU Octave project page

[8] Python x-y project page