Friedrich-Alexander-Universität
Technische Fakultät

Plane truss

Laboratory course Matlab

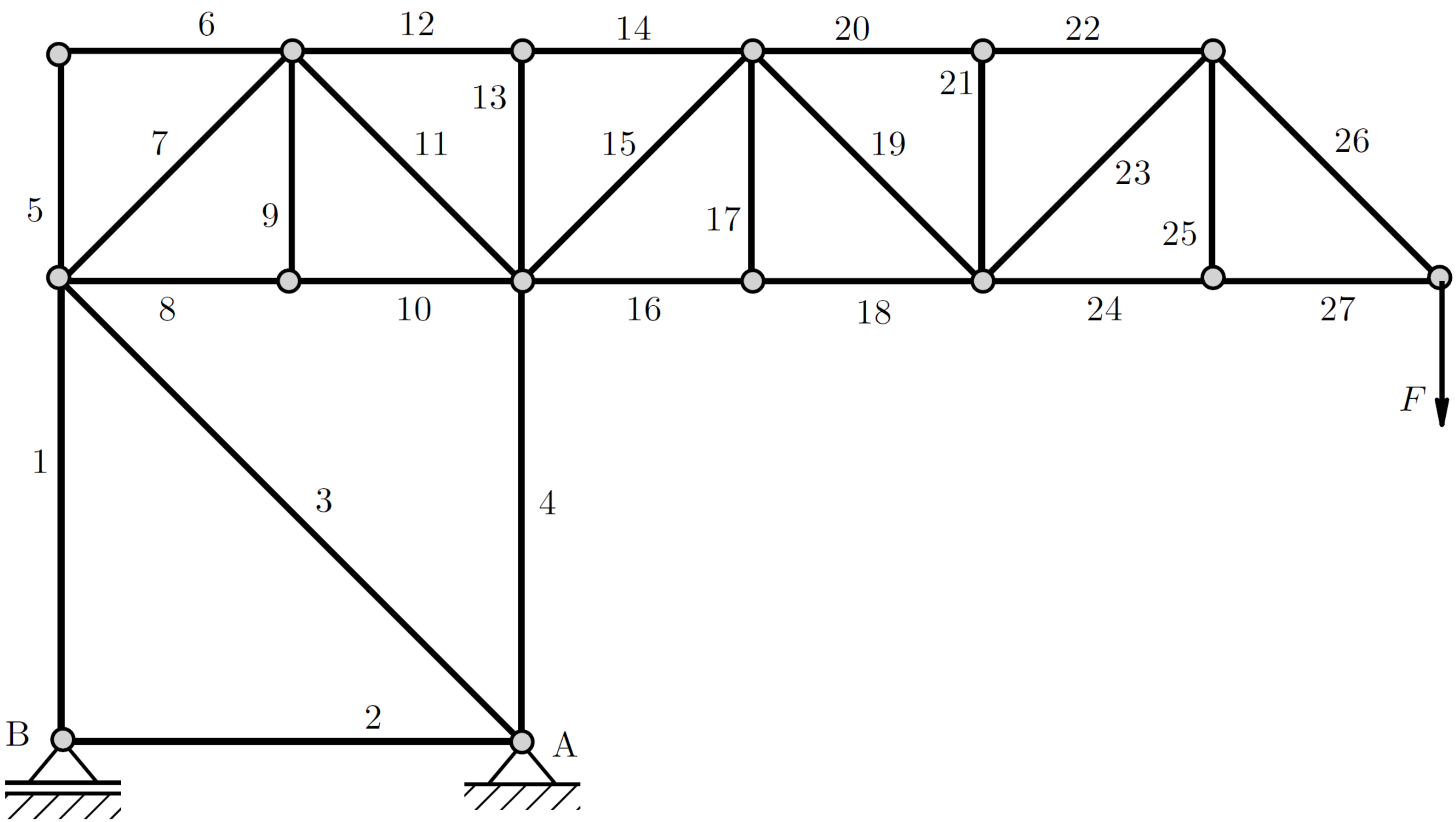
Emely Schaller, M.Sc. (*emely.schaller@fau.de*)
Prof. Dr.-Ing. habil. Kai Willner
October 25th, 2024

Content:

1. Theory about plane trusses
2. Notes on the task

Task in the laboratory course

Plane trusses



1. Theory about plane trusses

Assumption of an **ideal truss**:

i. The bar axes are straight.

ii. The bar axes intersect in one point, the so-called **nodes**.

iii. At the intersection, the bar axes are connected by frictionless joints.
Thus, **no moments** can be initiated in the bars.

iv. The active forces only act on the nodes.

1. Theory about plane trusses

Condition for static determination (1):

- Degree of freedom of the bounded system:

$$f = 2k - (a + s)$$

k	number of **nodes**
a	number of **support connections**
s	number of **bars**

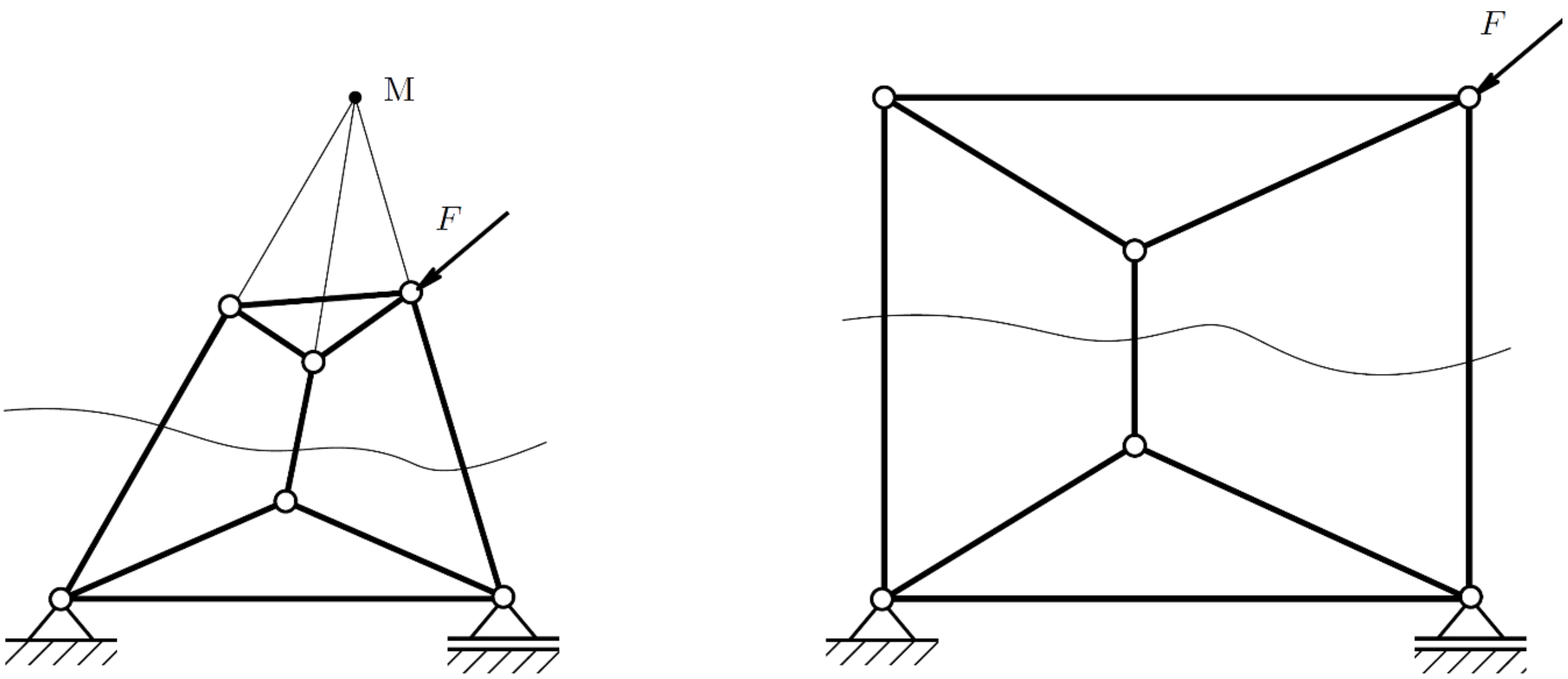
- External degree of freedom: $f_a = 3 - a$

- Internal degree of freedom: $f_i = f - f_a = 2k - (3 + s)$

- Necessary condition for the rigidity: $f \leq 0$

1. Theory about plane trusses

Condition for static determination(2): Exceptional cases



1. Theory about plane trusses

Condition for static determination (3):

- Necessary and sufficient condition for the rigidity and load-bearing capacity of the truss (no exceptional case)

$$f \leq 0 \qquad \text{and} \qquad f_a \leq 0$$

- $2k$ equilibrium conditions for the determination of the $(a + s)$ support connections and bar forces

$$f \begin{cases} > 0 & \text{movable (not stable)} \\ = 0 & \text{statically determined} \\ < 0 & \text{statically undetermined} \end{cases}$$

1. Theory about plane trusses

Method of joints (1):

1. Cutting all nodes free
2. Setting up the equilibrium of forces at all nodes:
   - equilibrium of forces in x-direction
   - equilibrium of forces in y-direction
   - equilibrium of moments is not necessary, since no moments are initiated in the bars

$$\boxed{\begin{aligned}\sum F_x &\doteq 0\\ \sum F_y &\doteq 0\end{aligned}}$$

$$\sum M_n = 0$$

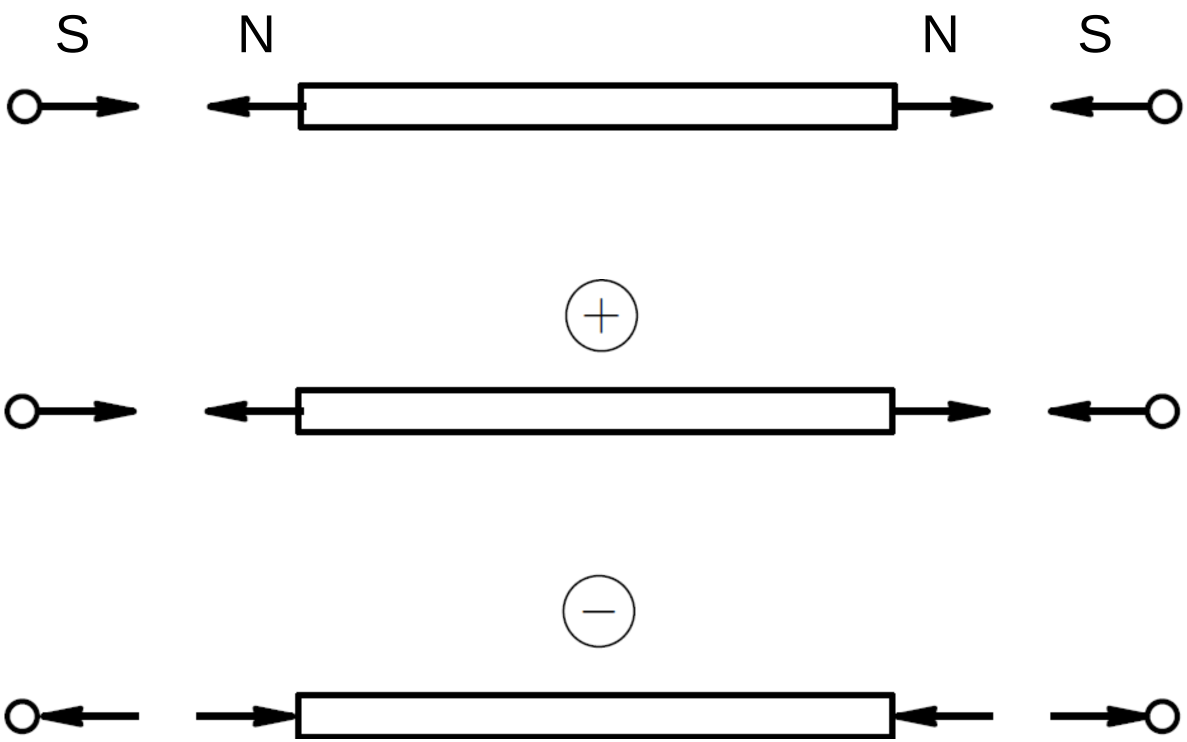
3. Summarising the equilibrium conditions in a linear and inhomogeneous system of equations with the dimension:

$$\boxed{2k \text{ x } (a + s)}$$

Alternative: Method of sections

1. Theory about plane trusses

Sign convention:

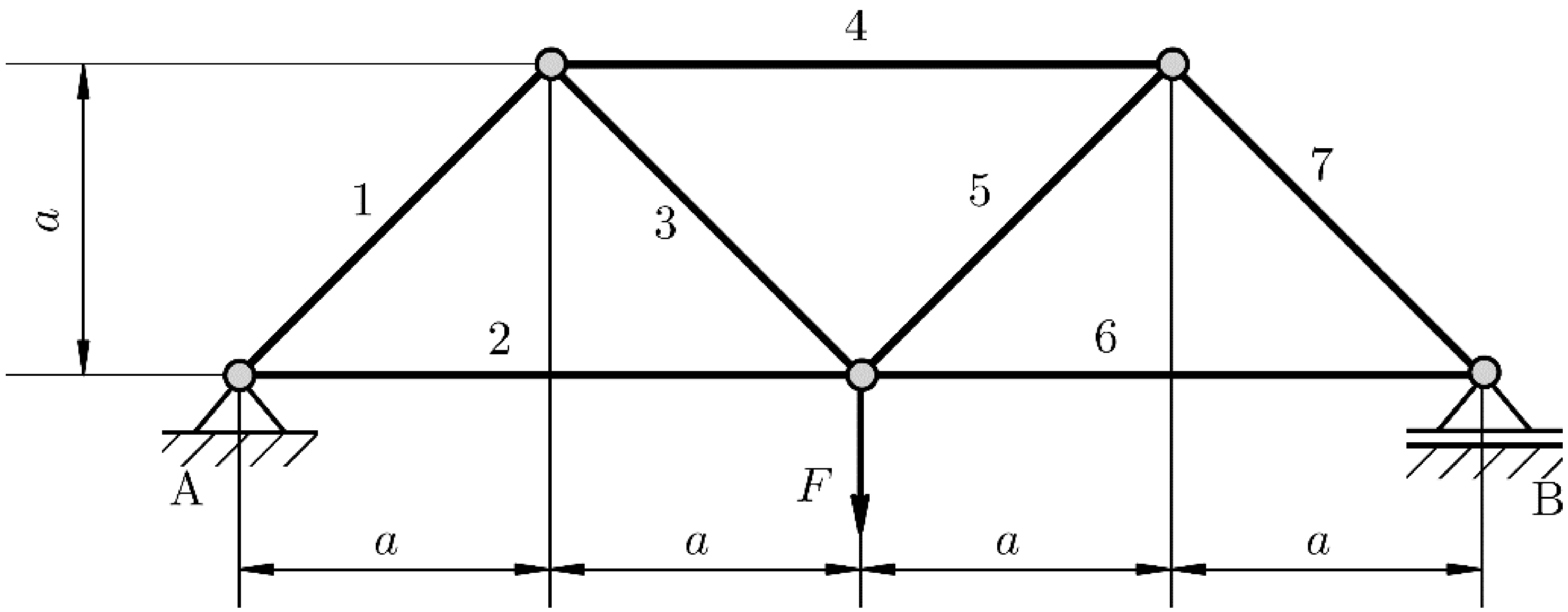


Tension bar
(S > 0 N)

Compression bar
(S < 0 N)

2. Notes on the task

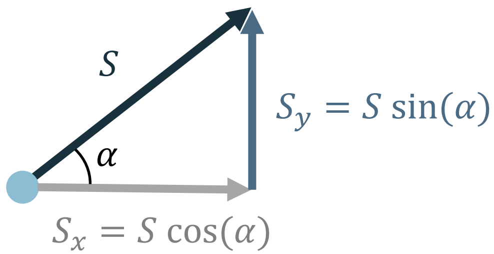
Method of joints (2) - **Task 1 in the script as preparation for the programming task**:

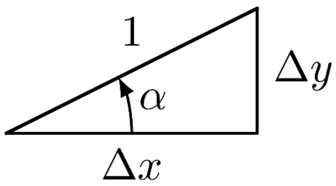


➡ Cutting free and setting up the linear system of equations

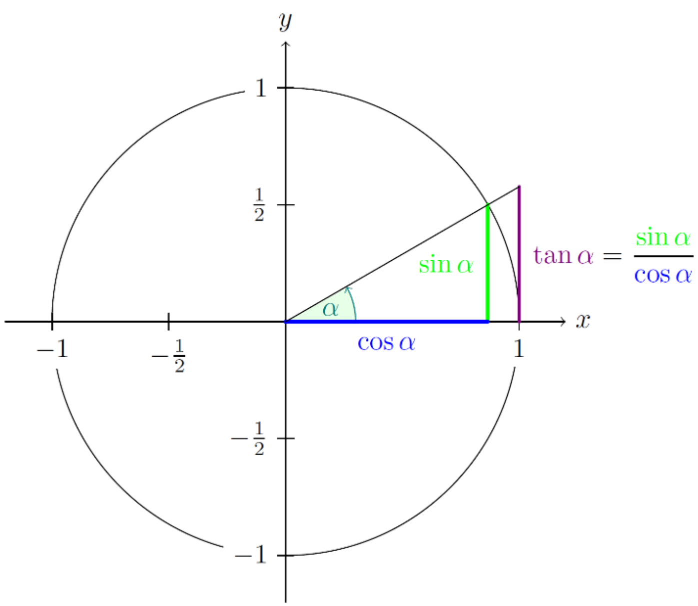
2. Notes on the task

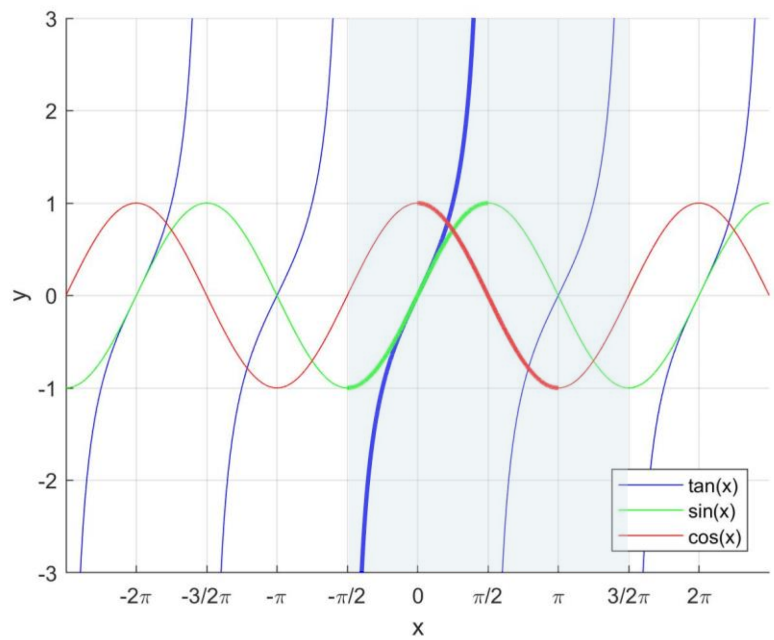
Arctangent:





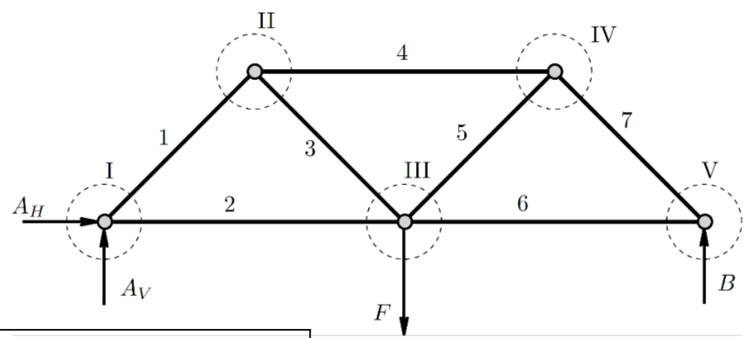
$$\alpha = \arctan\left({}^{\Delta y}/_{\Delta x}\right)$$





2. Notes on the task

Method of joints (3) - Task 1 in the script:



Result $\quad \mathbf{Ar} = -\mathbf{f} \longrightarrow \mathbf{r} = -A^{-1}\mathbf{f}$

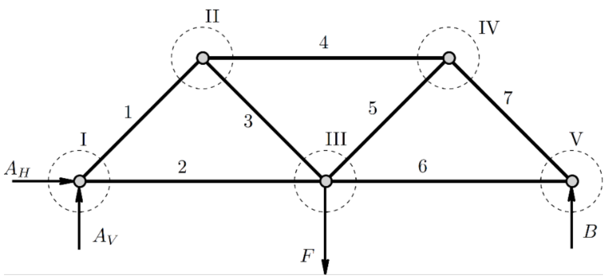
Kn.	A_H	A_V	B	S_1	S_2	S_3	S_4	S_5	S_6	S_7
I	1	0	0	$\sqrt{2}/2$	1	0	0	0	0	0
	0	1	0	$\sqrt{2}/2$	0	0	0	0	0	0
	…	…	…	…	…	…	…	…	…	…
II	0	0	0	$-\sqrt{2}/2$	0	$\sqrt{2}/2$	1	0	0	0
	0	0	0	$-\sqrt{2}/2$	0	$-\sqrt{2}/2$	0	0	0	0
	…	…	…	…	…	…	…	…	…	…
III	0	0	0	0	-1	$-\sqrt{2}/2$	0	$\sqrt{2}/2$	1	0
	0	0	0	0	0	$\sqrt{2}/2$	0	$\sqrt{2}/2$	0	0
	…	…	…	…	…	…	…	…	…	…
IV	0	0	0	0	0	0	-1	$-\sqrt{2}/2$	0	$\sqrt{2}/2$
	0	0	0	0	0	0	0	$-\sqrt{2}/2$	0	$-\sqrt{2}/2$
	…	…	…	…	…	…	…	…	…	…
V	0	0	0	0	0	0	0	0	-1	$-\sqrt{2}/2$
	0	0	1	0	0	0	0	0	0	$\sqrt{2}/2$

$$\mathbf{A} \begin{bmatrix} A_H \\ A_V \\ \cdots \\ B \\ S_1 \\ \cdots \\ S_2 \\ S_3 \\ \cdots \\ S_4 \\ S_5 \\ \cdots \\ S_6 \\ S_7 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \\ 0 \\ 0 \\ \\ 0 \\ F \\ \\ 0 \\ 0 \\ \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{A} \qquad \mathbf{r} = -\mathbf{f}$$

2. Notes on the task

Data structure (e.g. `truss_3.mat`)



Connectivity (`conn`)

local node 1	local node 2
1	2
1	3
2	3
2	4
3	4
3	5
4	5

Coordinates (`coord`)

x	y
0	0
2	2
4	0
6	2
8	0

Bearing (`bearing`)

node	dof
1	1
1	2
5	2

External loading (`F`)

node	F_x	F_y
3	0	-1.5

2. Notes on the task

Reading and saving variables

- `[filename , pathname] = uigetfile`
  opens a graphic dialog box to select a file that, for example, can subsequently be loaded with the `load` function. The `uigetfile` function gives back two output-variables: the `filename` and the `pathname` to the folder that contains the file

- concatenation: `filepath = [pathname, filename]`

- `input = load(filepath)`
  loads variables of a .`mat`-file in the Matlab workspace

- `` save(`results.mat`) ``
  saves the entire workspace in a . `mat`-file called `results`

2. Notes on the task

Access to data in structure arrays

- `s = struct(field1, value1 , field2, value2, … )` summarises different field arrays, denoted by `field1` and `field2`, with the values `value1` and `value2` in one structure `s`. The same can be obtained by using

```
s.field1 = value1;
s.field2 = value2;
```

- Access to the values of a structure field via a dot `.`

```
input = load(filename);
forces = input.F;
```

2. Notes on the task

Access to data in matrices

- Saving data in principle by a matrix or multidimensional array, e.g.

  `A = [1 , 2 ; 3 , 4]` $\longrightarrow$ $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

- Access and assignment via indices `A(i,j)` (row `i` und column `j`)

  `A(1,2) = 5` $\longrightarrow$ $A = \begin{bmatrix} 1 & 5 \\ 3 & 4 \end{bmatrix}$

- `l = length(v)` or `[m , n] = size(A)`
  query the length `l` of a vector `v` or the number of rows `m` and columns `n` of a `[m x n]` matrix `A` .
  Using `length(A)` returns the maximum value of `m` and `n`.

2. Notes on the task

Generating graphs (1)

- `figure`
  generates a new window to plot graphs;
  plotting multiple graphs is only possible with the command
  `hold on`

- `p1 = plot( [x1 , x2] , [y1 , y2])`
  draws a 2D line plot from point `(x1,y1)` to point `(x2,y2)`

- ``legend([p1 , p2] , `bar`, `force`)``
  creates legend of the line plots of the current window

2. Notes on the task

Generating graphs (2)

- `text(x1, y1, `node 1`)`
  adds the text `node 1` to the data point (`x1,y1`)

- `quiver(x2, y2, force(1), force(2))`
  draws the vector `force` at point (`x2,y2`)

- Uniform arrow size with the additional option
  `quiver(x2, y2, force(1), force(2), …`
  `` `MaxHeadSize`, 1/norm(force)) ``

2. Notes on the task

Angle calculation and solving the linear system of equation

- `atan(dy/dx)`

  Arctangent in rad for the value range $[-\frac{\pi}{2}, +\frac{\pi}{2}]$

  → Function with **only one** input parameter

- `atan2(dy , dx)`

  Arctangent in rad for the value range $[-\pi, +\pi]$

  → Function with **two** input parameters

- `x = A/b`

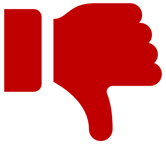
  solves the linear system of equations $\mathbf{Ax} = \mathbf{b}$ for the vector $\mathbf{x}$

2. Notes on the task

Hard vs. generic coding for plane truss code

- **Hard coding**:
  The script is explicitly coded for the trusses given in the input files, e.g. for truss 3:

```
Nnodes = 5;
...
```



- **Generic coding**:
  The script is coded in a general way, such that any input file defining a truss can be evaluated

```
Nnodes = size(coords,1);
...
```

Thank you for your attention!

If you have any questions on the lecture, please use the forum „Forum: experiment LTM“ on StudOn.

You can also contact me via email *emely.schaller@fau.de*.

LTM-task

on November 8th (group A)
on November 15th (group B)
from 10 am to 4 pm
in the MB & CBI CIP-Pool