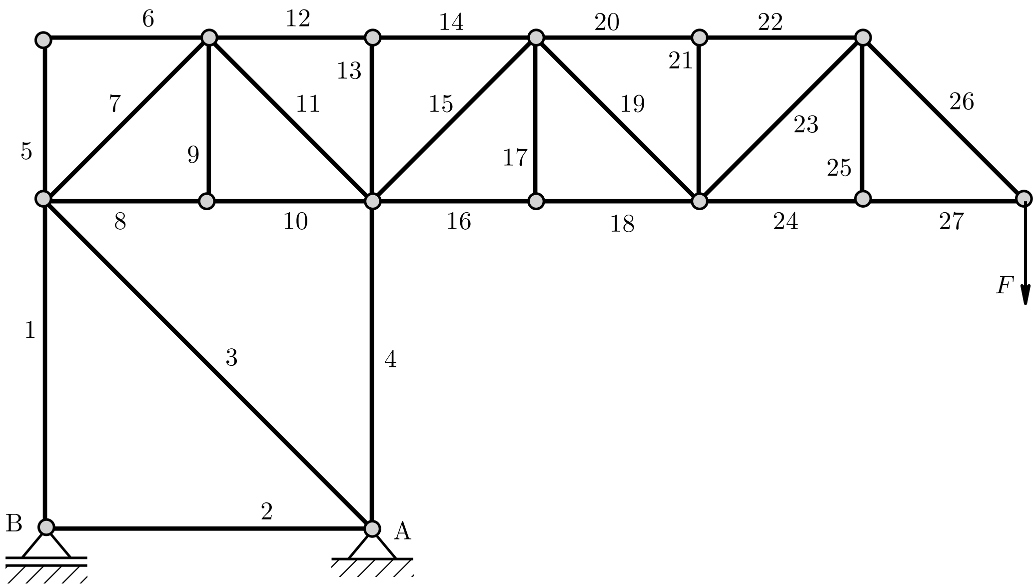
2 Static truss (Experiment LTM)



Contents

The aim of this experiment is to prepare a Matlab script for visualisation of a 2d truss and determination of the reaction and bar forces occurring in it. It is assumed that the truss is an **ideal truss**, which means that the bars in it can only transmit tensile or compressive forces.

2.1 Introduction

A plane truss can be perceived as a system of k initially free-moving nodes that are connected to each other by s bars. With the addition of a support connections by which the truss is borne, the remaining **degree of freedom** f of the bounded system follows as

$$f = 2k - (a + s) \tag{2.1}$$

and the **external** degree of freedom f_a as

$$f_a = 3 - a. \tag{2.2}$$

Unless there is an exceptional case, the following conditions

$$f_a \leq 0 \quad \text{und} \quad f \leq 0 \tag{2.3}$$

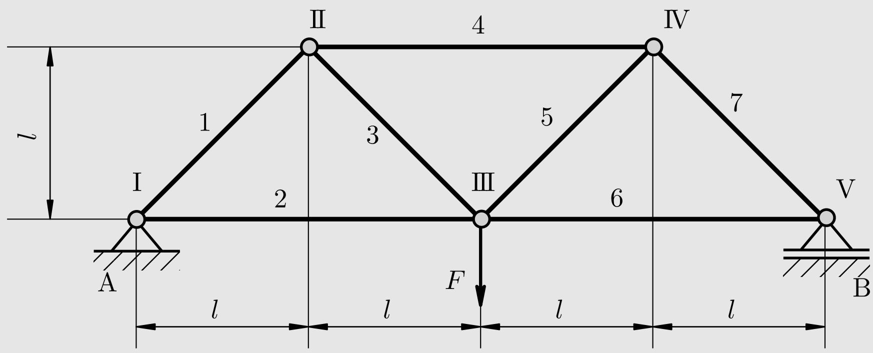
are considered necessary and sufficient for the load-bearing capacity of the truss. For a **statically determined** truss, i.e. $f = 0$, the reaction and bar forces can be calculated using the **method of joints**, in which the horizontal and vertical equilibrium of forces is set up at every node. Based on the total $2k$ equilibrium conditions of the bounded system, a linear, inhomogeneous system of equations of the form

$$\mathbf{A}\mathbf{r} + \mathbf{f} = \mathbf{0} \tag{2.4}$$

can be formulated with the coefficient matrix $\mathbf{A}$, the sought-after a support reactions and s bar forces in the vector $\mathbf{r}$ and the external loads in the vector $\mathbf{f}$.

2.2 Preparation tasks

Task 1
Cut the plane truss shown here free and set up the two equilibrium conditions for every node in the truss.



Then formulate a system of equations in matrix form (2.4) by writing all nodal equilibrium conditions one below the other. Hence, the matrix **A** has the dimension $2k \times (a + s)$.

2.3 Experimental procedure

To determine the reaction and bar forces of a plane truss, a script is to be prepared in this experiment that reads in the required parameters for any given truss, visualises it, sets up and solves the corresponding linear system of equations as well as displays and saves the results. The following trusses (`mat` files) are available as input parameters:

`truss_1.mat` `truss_2.mat` `truss_3.mat` `truss_4.mat`

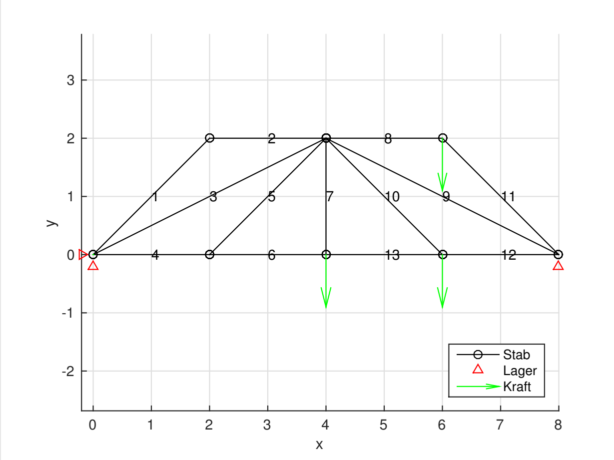
In the respective **structure arrays** all relevant information about the truss is summarised as follows:

- `coord`: Each row of this matrix contains the (x,y)-coordinates of a node.
- `conn`: This variable contains the connectivity matrix, i.e. each row describes a bar by indicating the two nodes between which the bar lies.
- `bearing`: The bearing of the truss is summarised here in a matrix row by row. The first column contains the node index at which the movement in a certain direction is constrained. The second column defines the direction in which a displacement is inhibited, where $1 \mathrel{\hat{=}} x$ and $2 \mathrel{\hat{=}} y$.
- `F`: Each external load that should act on the truss is in a separate row of this matrix. The first column defines the node on which the force acts and the second and third column the x- and y-components of the force vector.

The following tasks serve as support and outline the essential steps required to develop the script.

Task 2
Load the variables of one of the given `mat` files into the workspace and extract the coordinates of the nodes, the connectivity matrix, the bearings as well as the external forces as separate variables. To enable an interactive selection of the input file, a matlab dialogue box can be utilised by exploiting the Matlab function `uigetfile` .

Task 3
Visualise the given truss with its bars (and numbers), nodes, bearings and forces. The labelling of the axes and the legend should also be given. An exemplary representation is given below.

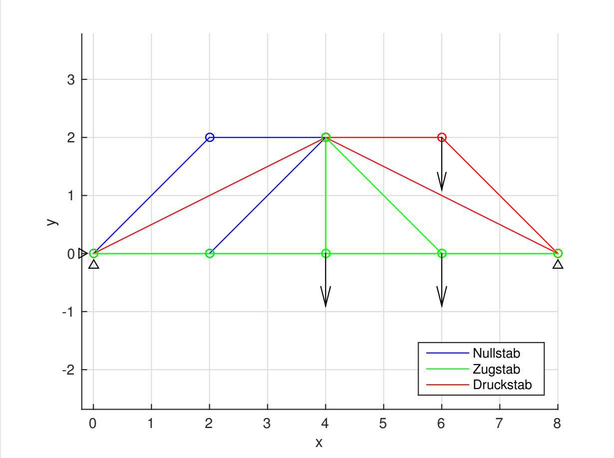


Task 4
Check whether the necessary condition $f = 0$ of static determinateness as well as (2.3) are fulfilled and write out an appropriate and informative message in the `Command Window`. If necessary, **terminate** the script with an **error** message.

Task 5
Calculate the angle α that every bar enclose with the x-axis. Use either the Matlab function `atan` or `atan2` for your implementation. Consider here the difference of these two functions and the alignment of the bar in the Cartesian coordinate system. Additionally, note that the direction of the bar force at its two nodes is not identical.

Task 6
Formulate the linear system of equations in matrix form (2.4) by using the given input variables and already generated variables from the previous tasks. Note the dimensions of the bounded system (see task 1). Then solve this linear system of equations for the unknown vector $\mathbf{r}$.

Task 7
Visualise the solution of the calculation by colouring the bars according to their load condition, i.e. zero, tension or compression bar. An exemplary representation is shown here.



Note that in numerical calculations the number 0 is rarely determined exactly because of floating point arithmetic and rounding errors. Matlab uses a numerical procedure to solve the linear system of equations (2.4) for the unknown vector **r**. Thus, a **very small error bound** is recommended to check for zero bars.
Check whether the resulting bar forces (zero, tension or compression bar) of all four trusses are plausible according to the respective load case.

Task 8
Save the results of the calculation (support and bar forces) as well as the input parameters in a new `mat` file, e.g. labeled `results_truss1.mat`.