Laboratory Excercise 3: Optimization of truss structures

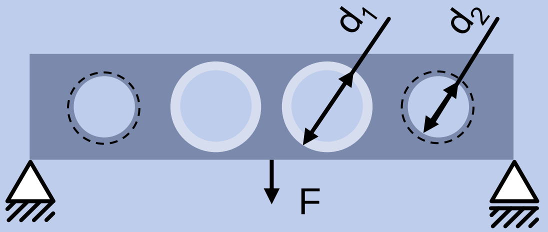
M. Gadinger

- Theoretical basics of optimization
- Optimization with Matlab
- Examples
- Discussion of the task

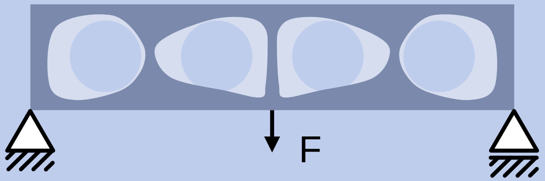
Optimization

Structural optimization

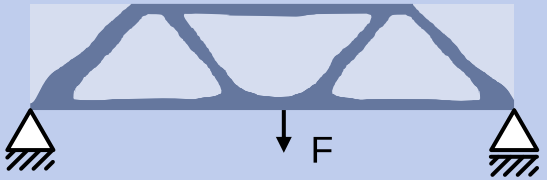
Parameter optimization



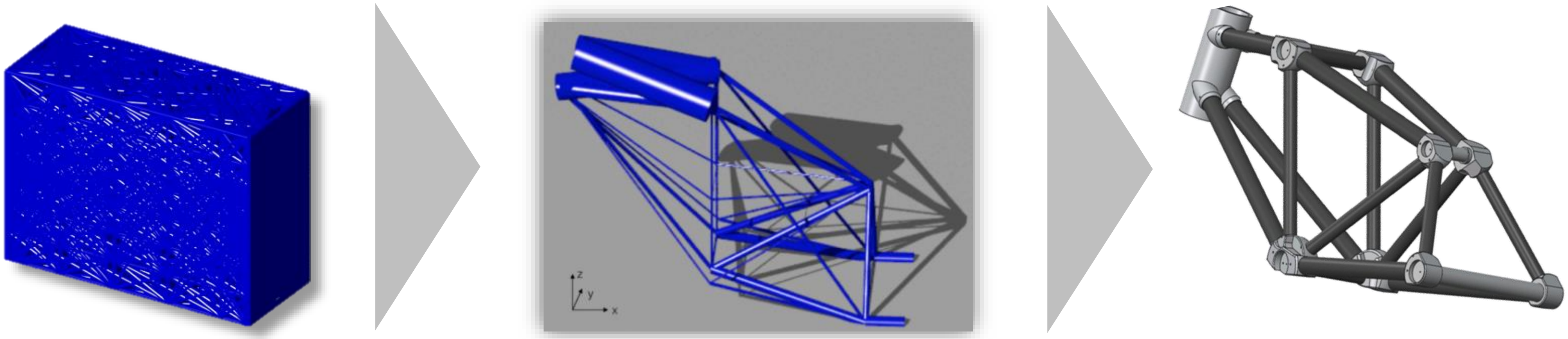
Shape optimization



Topology optimization



- Topology optimization of a motorcycle frame using beams

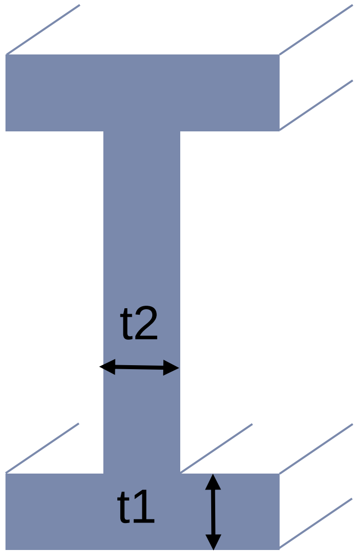


Optimization problem
Components

$\min\ f(\boldsymbol{x})$	**Objective function**	Minimization of mass
$\boldsymbol{x}$	**Design variables**	Thicknesses of the bars t1 and t2

$$s.t.\ \ c(x) \leq 0$$
$$ceq(x) = 0$$
$$Aeq \cdot x = beq$$
$$A \cdot x \leq b$$
$$lb \leq x \leq ub$$

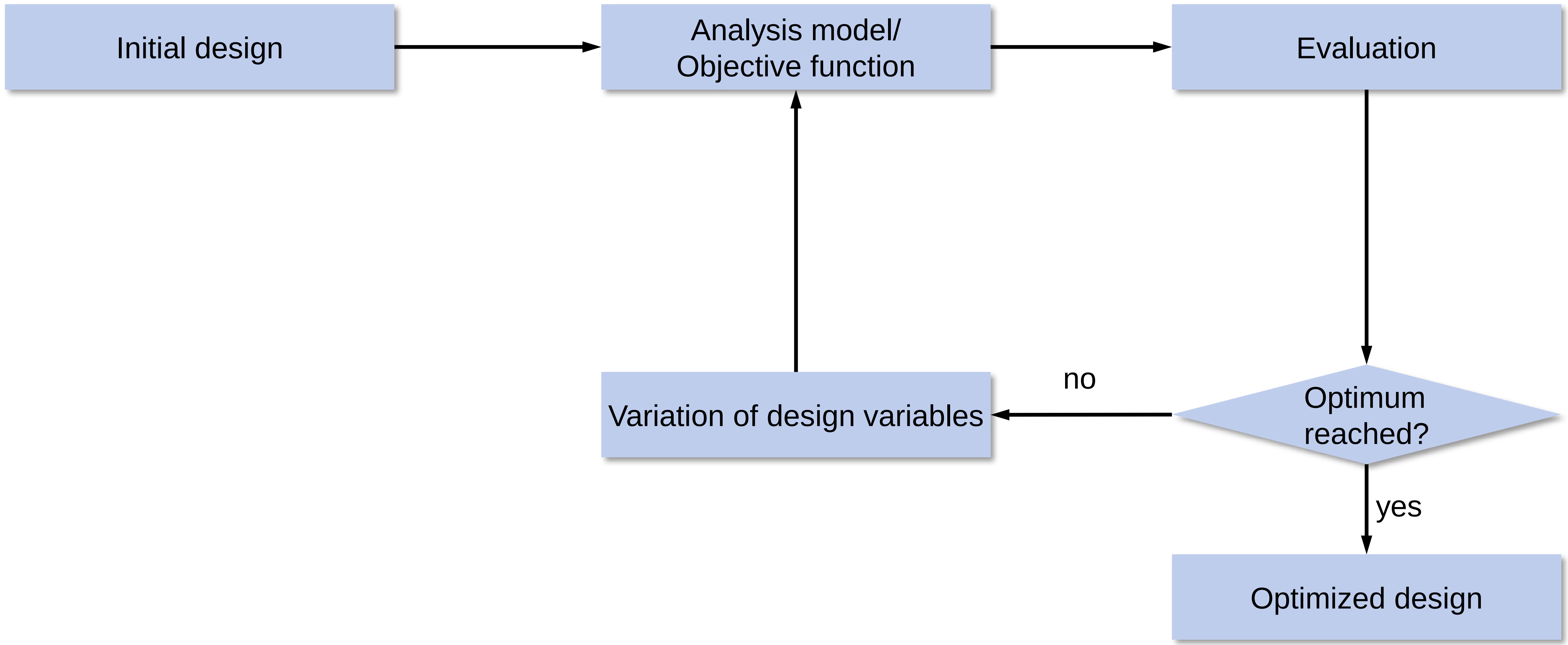
Constraints/
Boundary conditions



- Compliance with a max. permissible deformation
- Compliance with a max. permissible stress
- Compliance with manufacturing restrictions
- Minimum and maximum thickness

$$2\ mm \leq t1 \leq 20\ mm$$
$$2\ mm \leq t2 \leq 20\ mm$$

Optimization loop



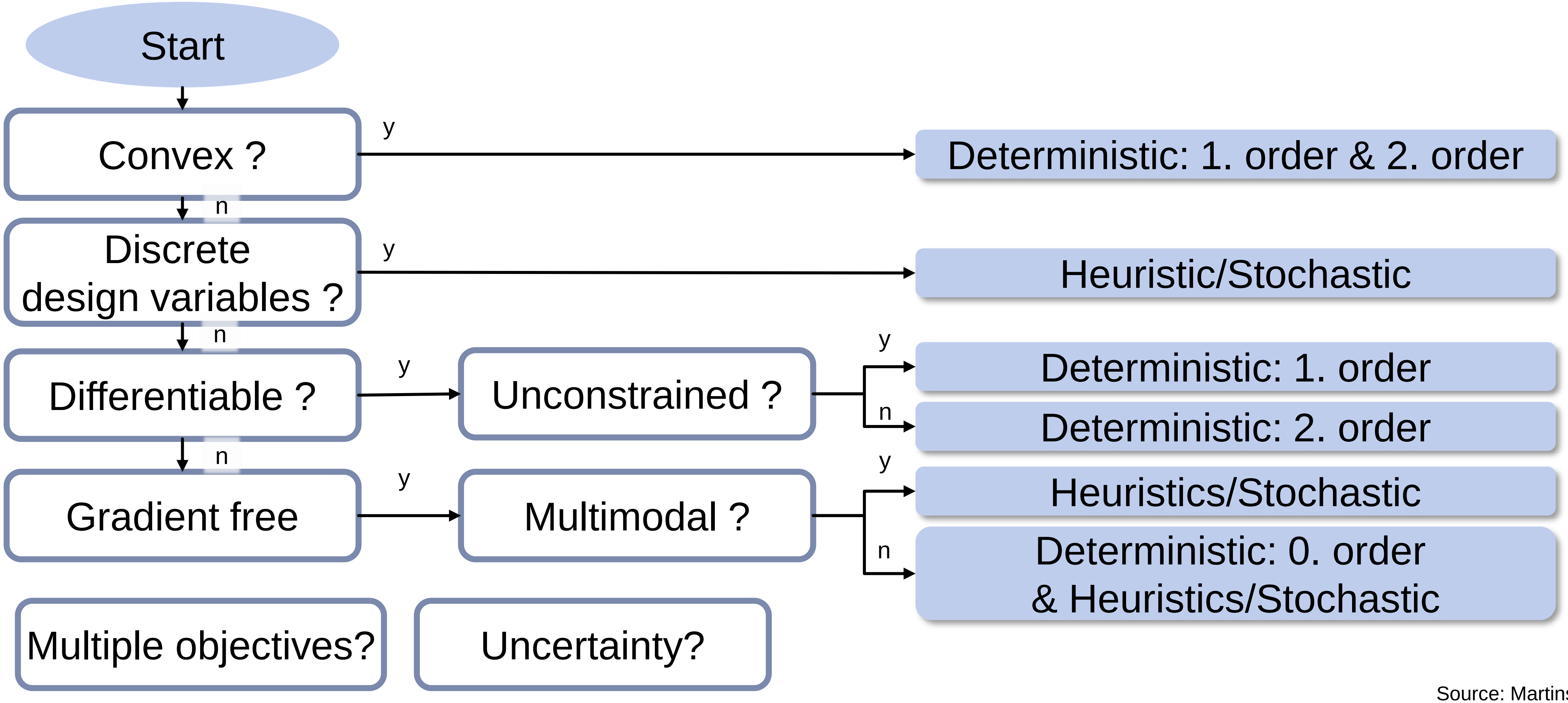
Laboratory 5: Optimization
Classification of optimization algorithms

Optimization algorithms			
Mathematical/Deterministic programming			Stochastic/Heuristic algorithms
0. Order ▪ Objective function value	**1. Order** ▪ Objective function value ▪ Gradient	**2. Order** ▪ Objective function value ▪ Gradient ▪ Hessian matrix	• Objective function value

Source: Müller, S.D.: Bio-inspired optimization algorithm for engineering applications. Dissertation, Swiss Federal Institute of Technology Zurich, Zürich, 2002

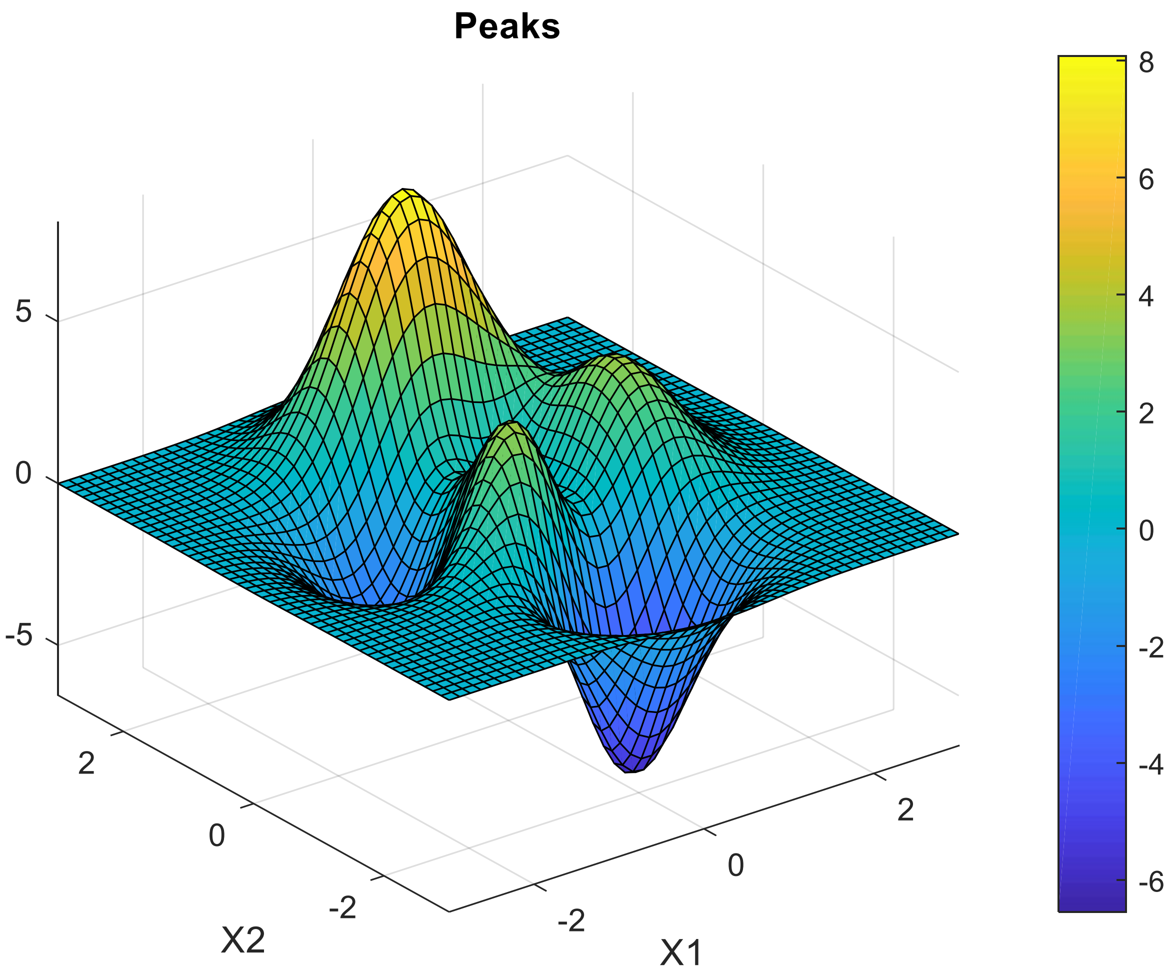
Laboratory 5: Optimization
Choosing an optimization algorithm

- Decision of an optimization algorithm based on information about the objective function, design variables and constraints



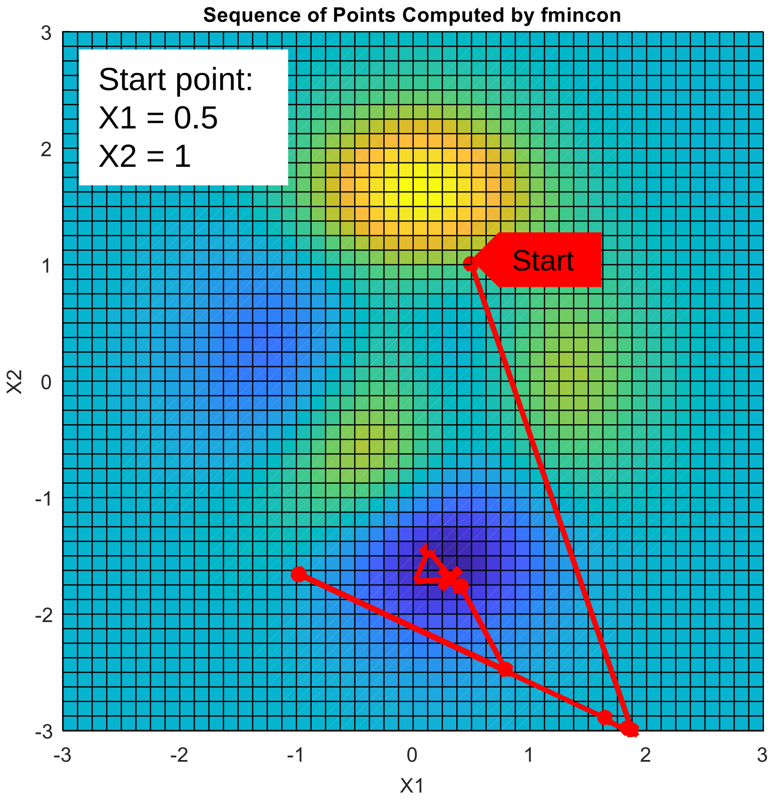
Source: Martins and Ning (2022)

Laboratory 5: Optimization
Problems in finding the optimum

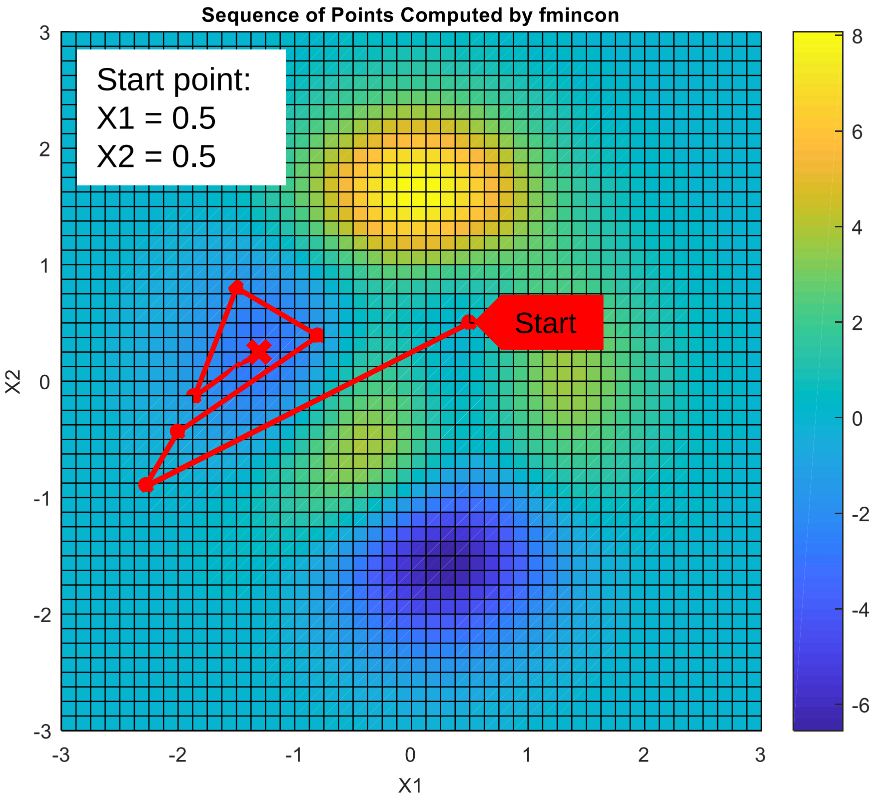


Laboratory 5: Optimization
Problems in finding the optimum

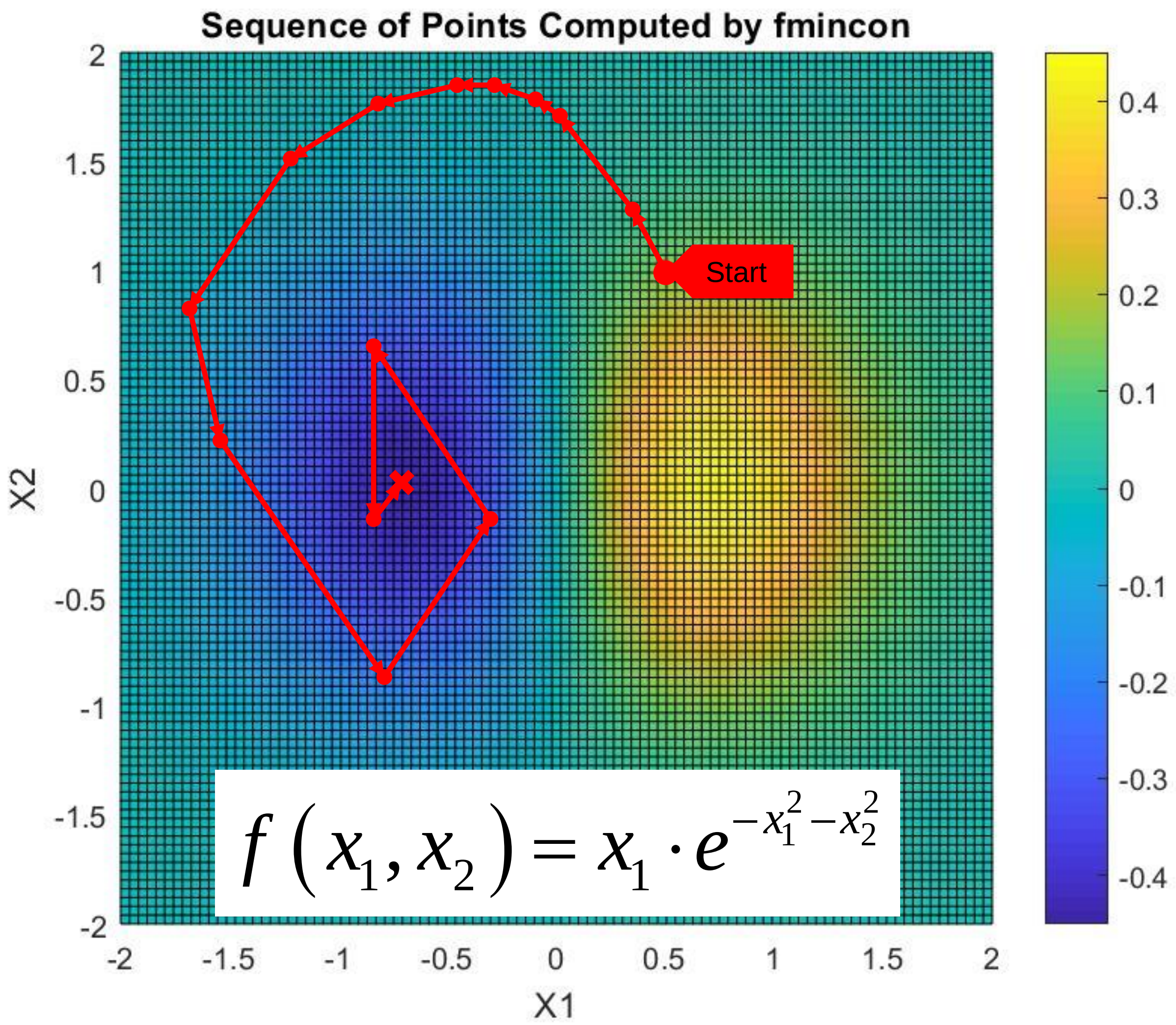
Global minimum



Local minimum



Laboratory 5: Optimization
Optimization history



Laboratory 5: Optimization

Syntax: Calling optimization in Matlab

Mathematical optimization formulation

$$\min \; f(\boldsymbol{x}) \qquad \text{(Objective function)}$$

$$s.t.\;\; c(x) \leq 0 \qquad \text{(Constraints)}$$

$$ceq(x) = 0$$

$$Aeq \cdot x = beq$$

$$A \cdot x \leq b$$

$$lb \leq x \leq ub$$

Matlab optimization functions:

- `fminbnd` (Scalar minimization)
- `fminsearch` (Unconstrained minimization)
- `fmincon` (Constrained minimization)
- `fminimax` (Multiobjective – Minimax)
- `fsolve` (Equation Solving – Nonlinear equations)
- `ga` (Genetic algorithm)
- `particleswarm` (Particle swarm optimization)
- Overview of various algorithms:
  - mathematic
  - stochastic

Matlab function: `fmincon`

```
[x, fval] = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
```

Example:

```
[t_opt,m]=...
    fmincon(@(t)objective(t,B,H,L,rho),t_0,[],[],[],[],lb,ub,...
    @(t)constraints(t,F,L,E,B,H),options)
```

Laboratory 5: Optimization

Syntax: function handle

- A function handle is a reference to a function
- It can be used like a variable
- Functions can be called from anywhere with function handles, because the storage location of the function is known to the function handle

Matlab-Syntax for a function handle:

```
function [mass]=calculateMass(length,width,height,density)
volume = length*width*height;
mass = volume * density;
End
```

Function Handle:

```
massfun = @calculateMass
L = 100;
W = 5;
H = 5;
Mass = massfun(L,W,H,7.85)
```

Laboratory 5: Optimization

Example 1

General example:

Find values for x that minimize the following objective function *f(**x**)*:

$$f(\boldsymbol{x}) = -x_1 x_2 x_3$$

Use as starting point:

$$\boldsymbol{x} = [10; 10; 10]$$

The following constraints should be considered:

$$0 \leq x_1 + 2x_2 + 2x_3 \leq 72$$

Laboratory 5: Optimization

General example: objective function and constraints

- Write the objective function in a new m-file:

```
function f = myfun(x)
f = -x(1) * x(2) * x(3);
```

- Reformulate constraints:
  - $-x_1 - 2x_2 - 2x_3 \leq 0$
  - $x_1 + 2x_2 + 2x_3 \leq 72$

- Since both constraints are linear, they can be formulated as matrix inequality $A \cdot x \leq b$, with

$$A = \begin{bmatrix} -1 & -2 & -2 \\ 1 & 2 & 2 \end{bmatrix}$$

$$b = \begin{bmatrix} 0 \\ 72 \end{bmatrix}$$

Laboratory 5: Optimization

General example: objective function and constraints

- Start point and call of `fmincon`:

```
x0 = [10; 10; 10];
[x,fval] = fmincon(@myfun,x0,A,b);
```

- After `fmincon` stops, the solution for **x** is:

```
x=
      24.000
      12.000
      12.000
```

- The function value `fval` results in:

```
fval=
      -3.4560e+03.
```

- The linear inequality constraints result in:

```
A*x-b=
      -72.000
      -0.000
```

Optimization example

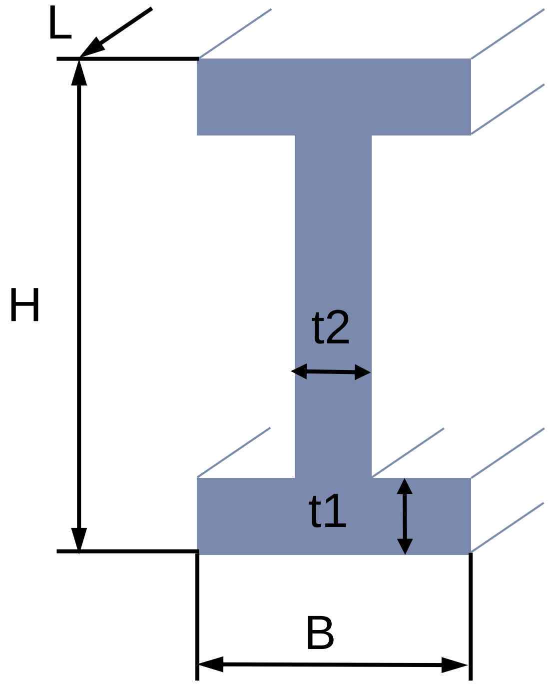
Minimization of the mass of an I-beam

Objective:
Minimization of mass m

Constraints:
Deformation $f \leq$ 5mm

Given:

E = 200.000 MPa	B = 100 mm
ρ = 7,85 g/cm^3	H = 200 mm
F = 10 kN	t_{min} = 2 mm
L = 2000 mm	t_{max} = 20 mm



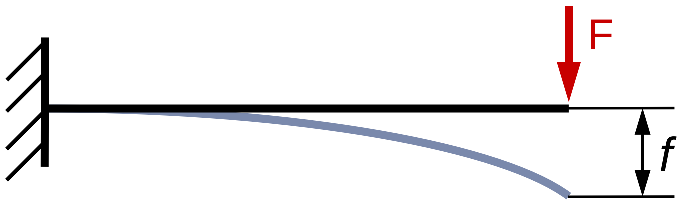
According to ME table book TH3.1:

$$I = \frac{BH^3 - bh^3}{12}$$

mit: $b = B - t2$
$h = H - 2 \cdot t1$

According to ME table book TH3.2:

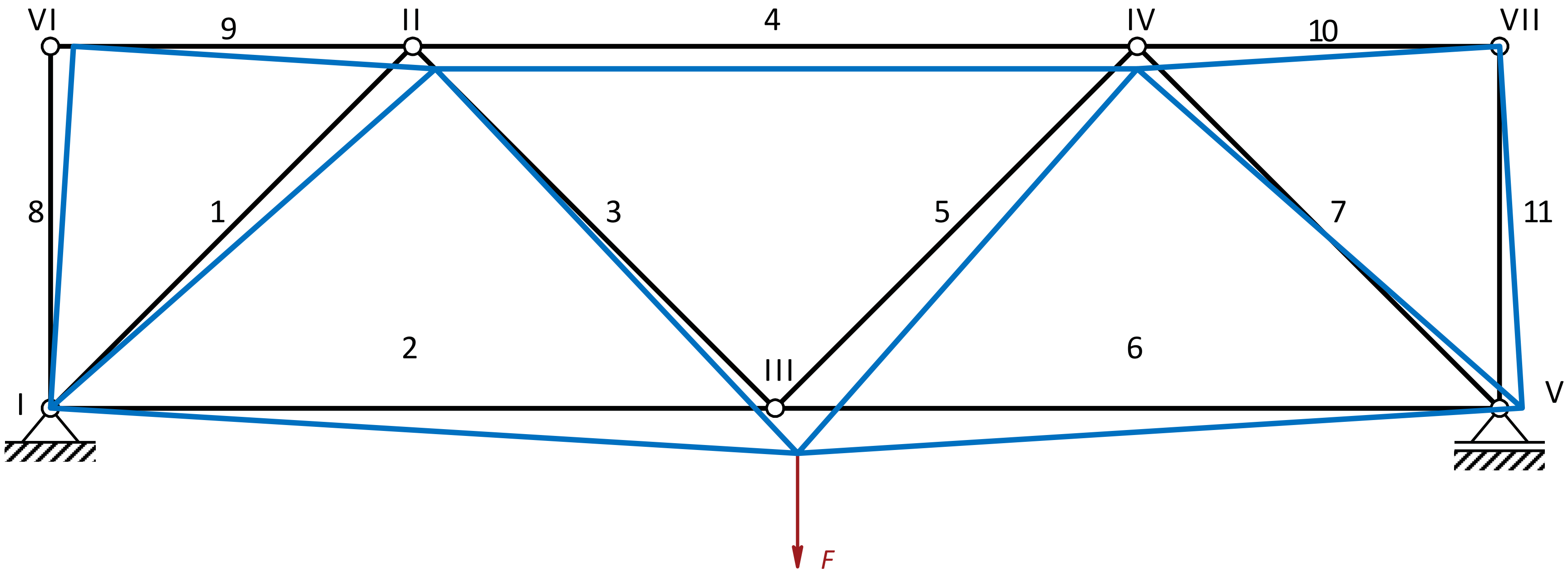
$$f = \frac{FL^3}{3EI}$$



Laboratory 5: Optimization

<u>Objective:</u>

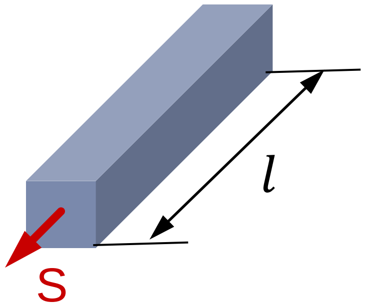
The objective is to optimize the cross-sections of a known 2D truss structure in such a way that the highest possible stiffness is realized while taking into account a weight constraint.



$$\min u(x)$$

Repetition truss structure

Truss element



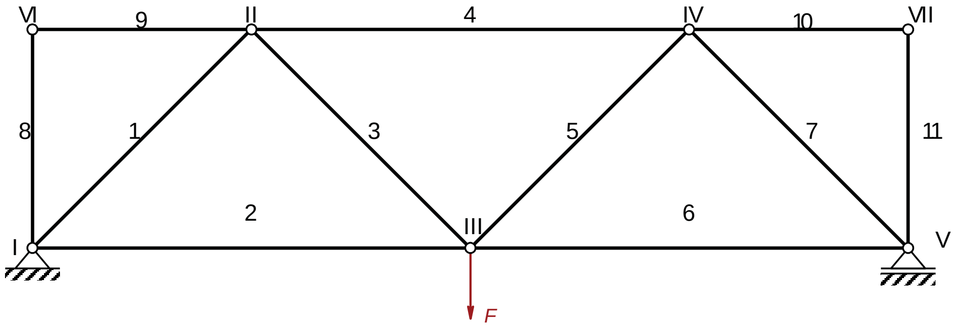
- Only forces in normal direction (truss element force S)
- Truss element has tensile stiffness EA
  - Elastic Modulus E (material property)
  - Cross section area A
- Displacement of a truss element in axial direction

$$\Delta l = \frac{Sl}{EA}$$

- Normal stress of truss element

$$\sigma = \frac{S}{A}$$

Truss structure



- Truss structure consists of several truss elements
- Articulated joint of the truss elements
- Static determinancy (Laboratory LTM)
- Displacements of the entire truss structure by basic equation of the finite element method (function Stabtragwerk)

$$u = K^{-1}F$$

Input files

Input data truss structure

- coord – Matrix of node coordinates x and y

	x	y
1	0	0
2	2	2
n_K	…	…

- conn – Connectivity matrix; node numbers of one truss element

	K1	K2
1	1	2
2	1	3
n_S	…	…

- boundaryCond – support; limited degrees of freedom

	K	Dim
1	1	1
2	1	2
n_L	…	…

- force – force vector

	K	Fx	Fy
1	3	0	-1.5

Function truss structure

```
function[u,S]=calcTrussStructure...
(EA,nNode,nTruss,coord,conn,boundaryC
ond,force)
```

Output:

- Function for calculating the nodal displacements u and the truss element forces S

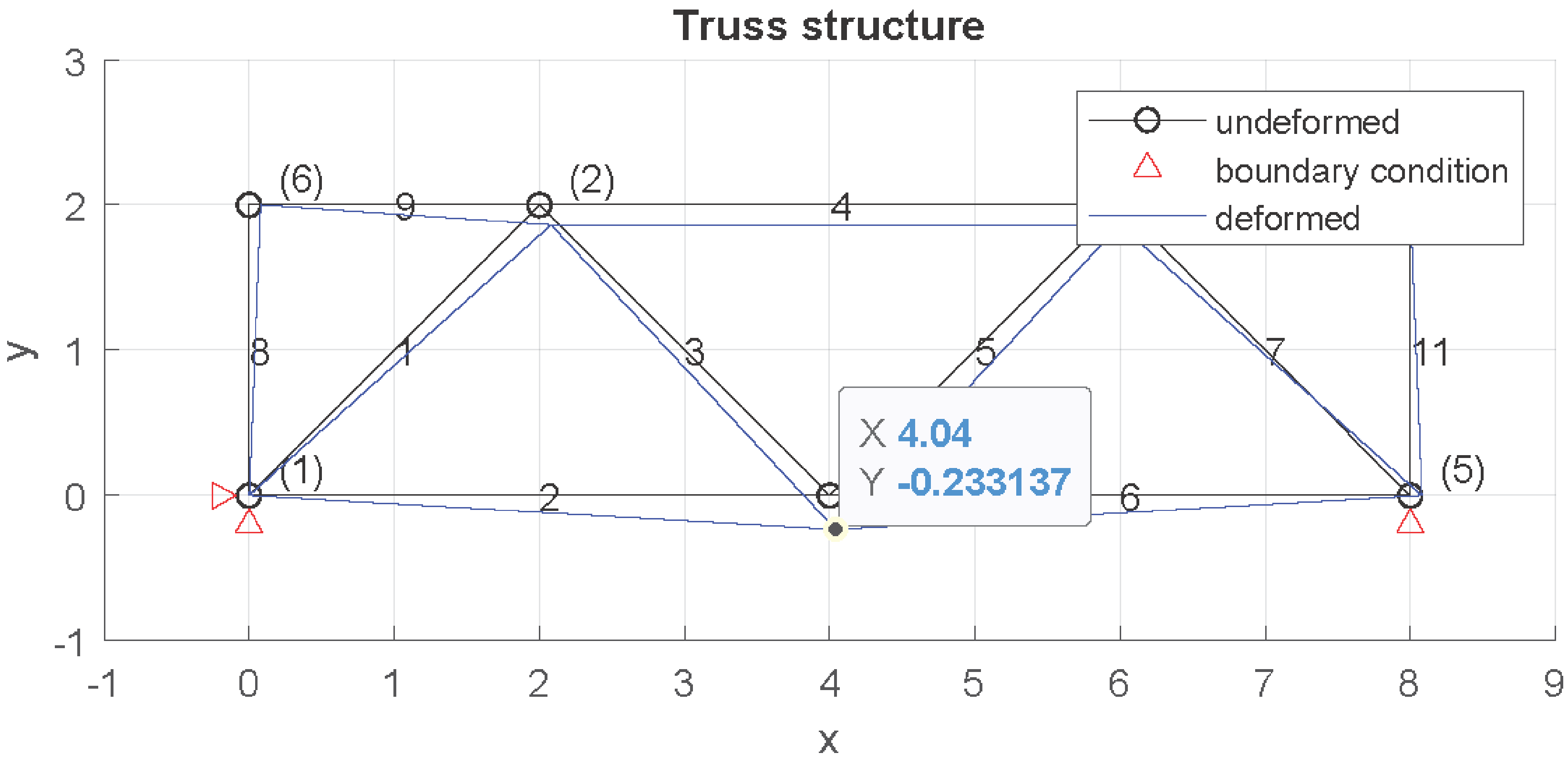
Input:

- EA – Vector of tensile stiffness of truss elements (length nTruss)
- nNode – Number of nodes
- nTruss – Number of truss elements
- coord – Matrix of node coordinates x and y
- conn – Connectivity matrix; node numbers of one truss element
- boundaryCond – support; limited degrees of freedom
- force – force vector

Sub task 1: Goal and task description

Goal:

- Plot of the undeformed truss structure
- Plot of the deformed truss structure



Sub task 1: Displacement of the truss structure
Approach

Step 1: Plot of the undeformed truss structure. (Use code from LTM excercise)

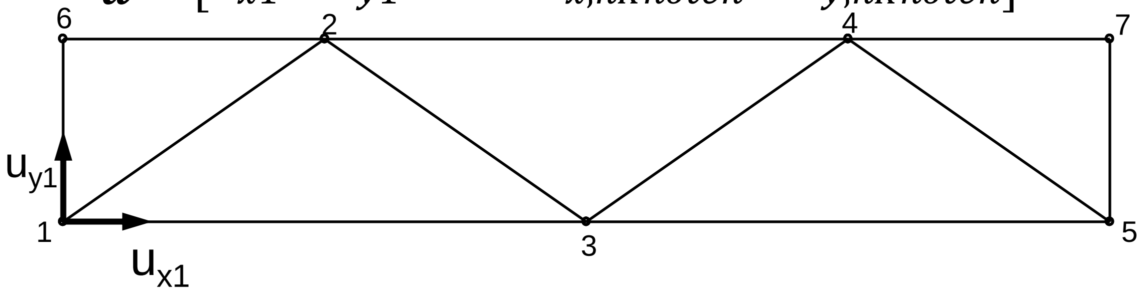
Step 2: Build vector of tensile stiffness

$$\boldsymbol{EA} = \begin{bmatrix} EA_1 \\ \vdots \\ EA_{nStab} \end{bmatrix}$$

Step 3: Calculate nodal displacements using the **calcTrussStructure** function:

```
[u,S]=calcTrussStructure(EA,nNode,nTruss,coord,conn,boundaryCond,force)
```

$$\boldsymbol{u} = [u_{x1} \quad u_{y1} \quad \cdots \quad u_{x,nKnoten} \quad u_{y,nKnoten}]$$



Step 4: Plot of the deformed truss structure

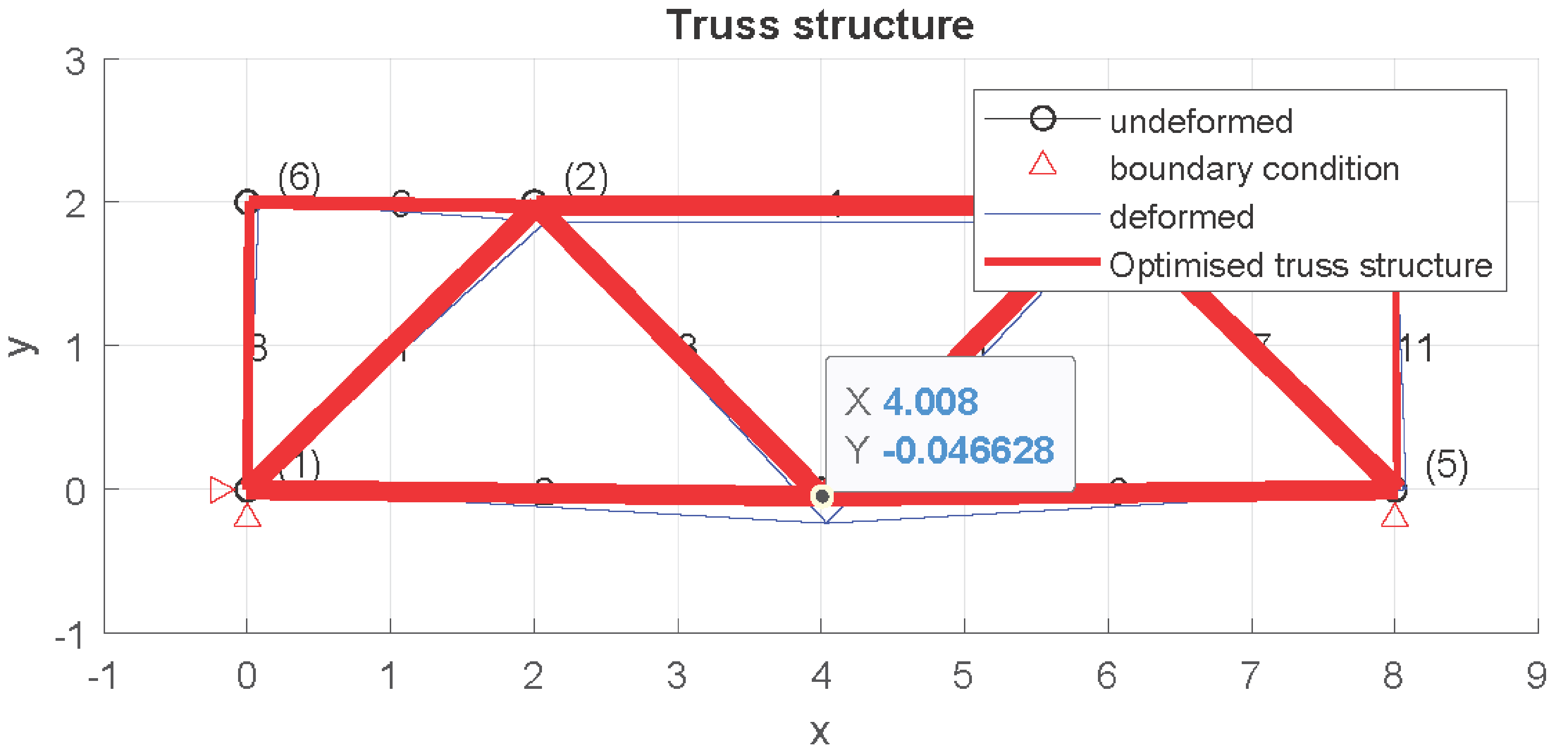
```
plotx=[coord(conn(i,1),1)+u(2*conn(i,1)-1), ...
p3=plot(plotx,ploty,'Color','b');
```


Friedrich-Alexander-Universität Erlangen-Nürnberg
Prof. Dr.-Ing. Sandro Wartzack

Sub task 2: Goal and task description

Goal:

- Minimization of the y-displacement at the force application point



Sub task 2: Stiffness optimization
Objective

Objective:

```
function Y =
objectiveFunction(x_opt,EA,nNode,nTruss,coord,conn,boundaryCond,force)
```

- Scaling of the tensile stiffnesses of the individual truss elements:

```
EA_scaled=x_opt.*EA;
```

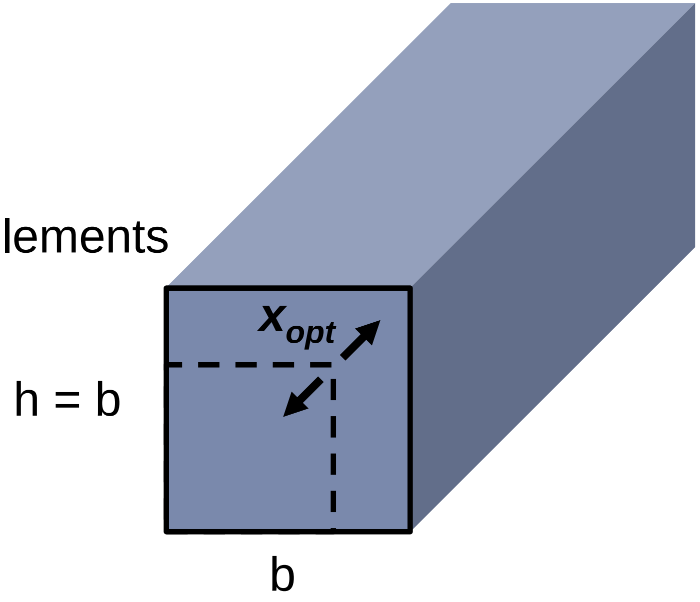
with:

$E = const.$

$A = h \cdot b$

→ The design parameter vector $\boldsymbol{x}_{opt}$ scales the cross section of the truss elements

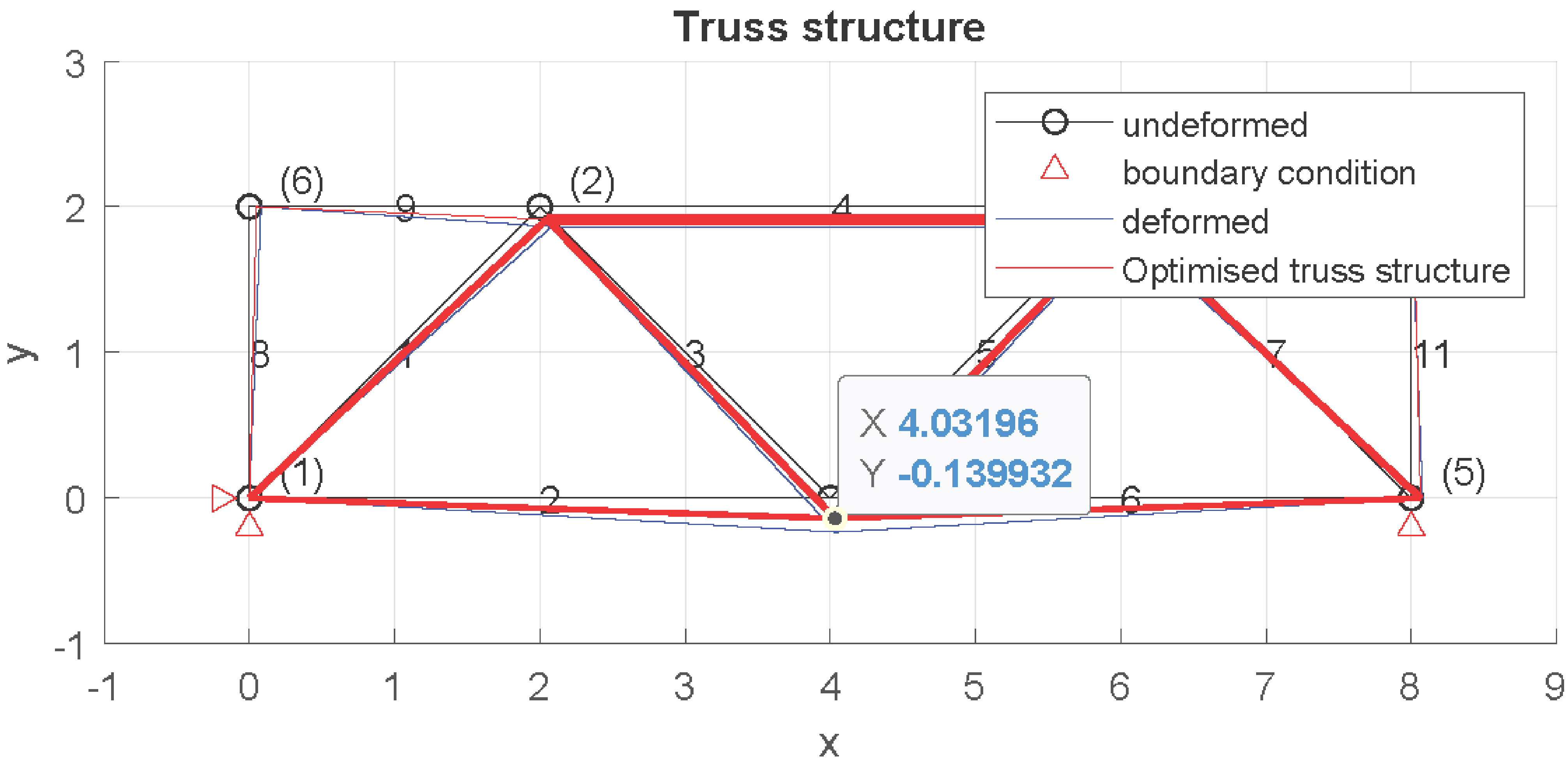
- Calculation of displacement u using the **calcTrussStructure** function
- Pick out the function value to be minimized from ***u***



Sub task 3: Goal and task description

Goal:

- Minimization of the y-displacement at the force application point with equality (mass) constraints



Sub task 3: Stiffness optimization with equality constraints

Linear constraints:

- Setting up the equality constraints $A_{eq} \cdot x = b_{eq}$
- Determine `Aeq` and `beq`
- Hint:

$$[EA1 \quad EA2 \quad EA3] \cdot \begin{bmatrix} x1 \\ x2 \\ x3 \end{bmatrix} = EA1 \cdot x1 + EA2 \cdot x2 + EA3 \cdot x3 = \sum_{i=1}^{N} EA_i x_i$$

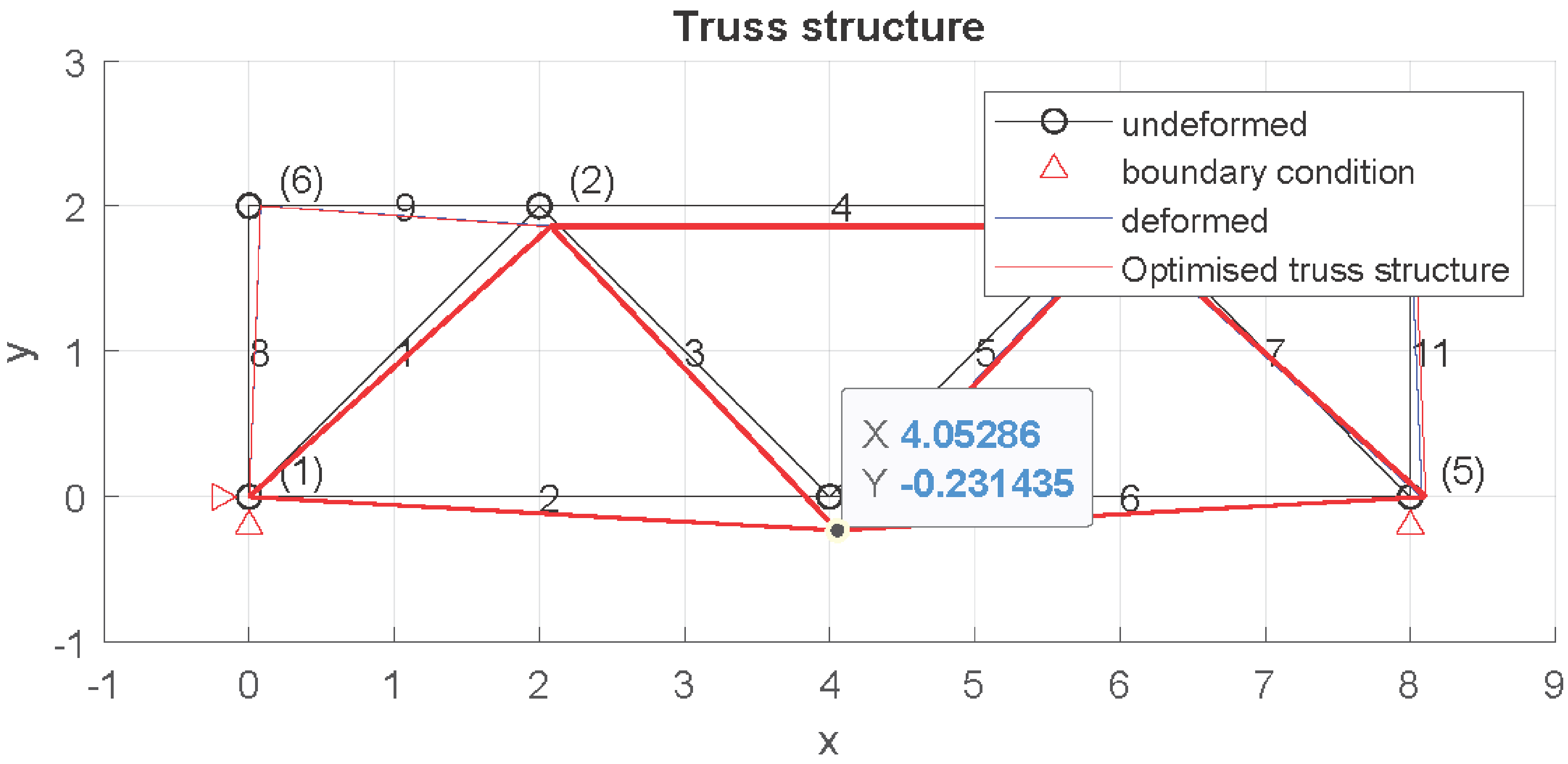
- Adjusting the optimization function `fmincon`

```
[x, fval] = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
```

Sub task 4: Goal and task description

Goal:

- Minimization of the y-displacement at the force application point with equality (mass) and inequality (stress) constraints



Sub task 4: Stiffness optimization with equality and inequality constraints

<u>Constraint:</u>

```
function [c,ceq] = constraintFunction(x_opt,A,EA,S,nTruss)
```

- Scaling of the extensional stiffness of the individual truss elements:

  ```
  EA_scaled=x_opt.*EA;
  ```

- Setting up the equality constraints `ceq` (see formula in the task description)
- Setting up the inequality constraint

  Hints:

  - The stress must be calculated for each truss element → vector
  - The internal force in the truss element can be positive or negative depending on the direction. The maximum stress should be maintained in both cases (tension or compression).

Sub task 5: Goal and task description

Goal:

- Evaluation of the efficiency of different optimization algorithms

Task description:

Now the efficiency of different optimization algorithms will be investigated. Optimize the problem defined in **sub task 4** with the function `fmincon`, but with the algorithm of the Sequential-Quadratic-Programming. For this purpose, set the required iterations as output using the keyword `Display` and set `sqp` as the algorithm with the additional command `optimoptions`. Check the required iterations when changing the start value. Then limit the maximum number of iterations that the algorithm may perform. Compare the results.

Sub task 6: Goal and task description

Goal:

- Saving the results

Task description:

Save the displacements u of the optimized truss structure in a new .mat file.

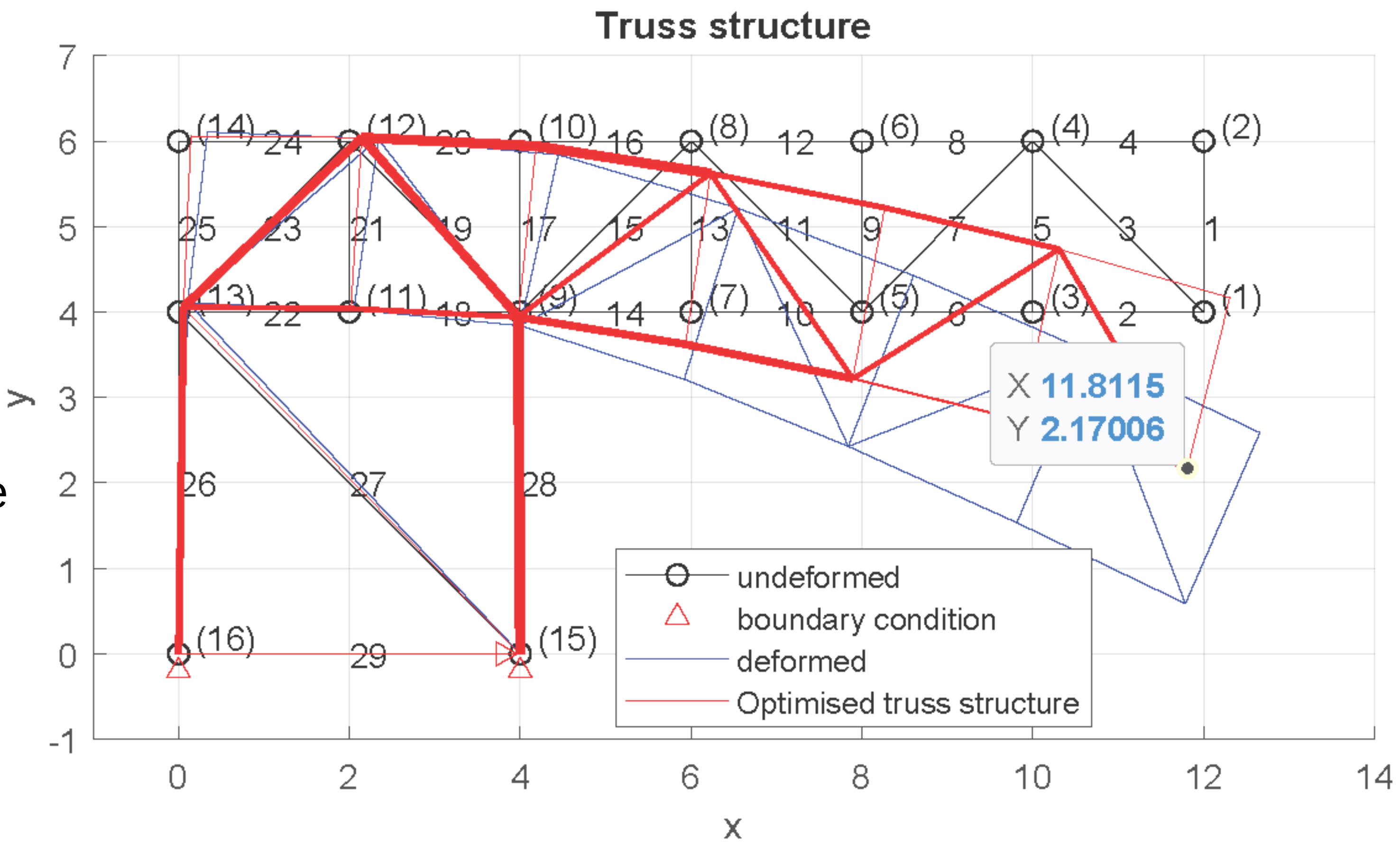
Sub task 7: Goal and task description

<u>Goal:</u>

- Use your code to optimize a crane's truss structure

<u>Task description:</u>

- Load the input file: V4_Input_Crane.mat
- Restrict the total stiffness to 2175 kN
- Optimize the truss structure and save the results



Organizational matters

- Questions can be asked in the StudOn forum or during the laboratory. The supervisor and the student assistents will answer them in a timely manner.
- For further questions, comments and feedback please use the contact below

Kontakt:
Marc Gadinger
gadinger@mfk.fau.de
Tel.: 09131/85-23215

Further literature

- Bendsøe, M.P. and Sigmund, O. (2004), Topology optimization: Theory, methods, and applications, Second edition, corrected printing, Springer, Berlin, New York.
- Harzheim, L. (2019), Strukturoptimierung: Grundlagen und Anwendungen, 3. Auflage, Europa-Lehrmittel, Haan.
- Schumacher, A. (2013), Optimierung mechanischer Strukturen, Springer Berlin Heidelberg, Berlin, Heidelberg.
- Martins, J.R.R.A. and Ning, A. (2022), Engineering Design Optimization. Cambridge University Press