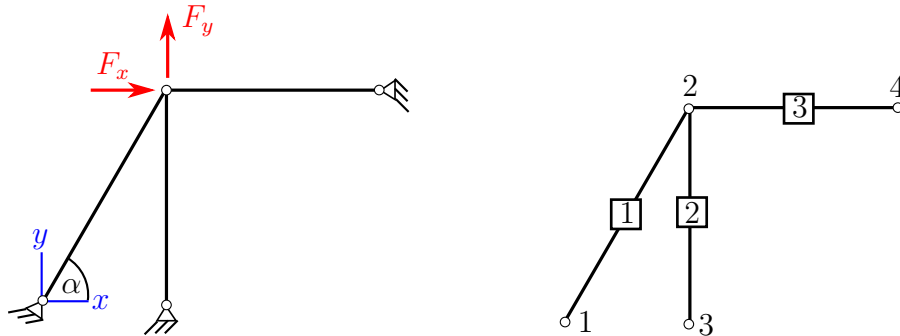


## Übung 5: FEM mit Stabelementen

### Aufgabe 5.1: Elementsteifigkeitsmatrix

Betrachtet wird das Fachwerk aus Aufgabe 4.1.



- Wie lassen sich die Einträge der Elementsteifigkeitsmatrizen  $\underline{\mathbf{k}}^e$  berechnen?  
Werten Sie auch die Spezialfälle  $\alpha \in \{\pi/2, 3\pi/2\}$  und  $\alpha \in \{0, \pi\}$  aus.
- Schreiben Sie die MATLAB-Funktion `PlaneTrussElementStiffness.m` zur Berechnung der Elementsteifigkeitsmatrizen. Verwenden Sie die Parameter aus Tabelle 5.1

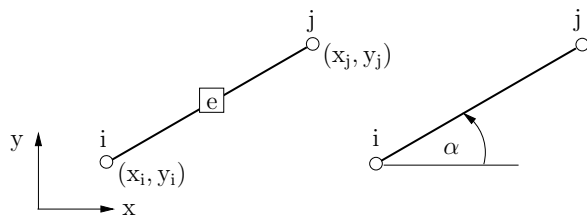
*Hinweis:* `PlaneTrussElementStiffness.m` ist eine Subroutine zur Berechnung der Elementsteifigkeitsmatrix im globalen Koordinatensystem. Die Inputwerte sind E-Modul  $E$ , Stabquerschnitt  $A$  und die Knotenkoordinaten  $i$  und  $j$  (vgl. Abbildung 5.1).

```

node_i(1) = x(1)
node_i(2) = y(1)
node_j(1) = x(2)
node_j(2) = y(2)
  
```

Outputgröße ist die Elementsteifigkeitsmatrix im globalen x,y-Koordinatensystem:

$$\underline{\mathbf{k}}^e = \underline{\mathbf{T}}^T \tilde{\underline{\mathbf{k}}}^e \underline{\mathbf{T}} \quad \text{mit} \quad \tilde{\underline{\mathbf{k}}}^e = \int_{B^e} (\underline{\mathbf{B}}^e)^T E A \underline{\mathbf{B}}^e d\tilde{x}$$



**Abbildung 5.1**

Inputparameter	
E	10000 kN/m <sup>2</sup>
A	0.02 m <sup>2</sup>
node no.1:	[ x y ] [ 0 0 ] m
node no.2:	[ x y ] [ 1 2 ] m
node no.4:	[ x y ] [ 3 2 ] m

**Tabelle 5.1**

- Bestimmen Sie den Lösungsvektor  $\underline{\mathbf{D}}$  mithilfe des in der vorherigen Übung aufgestellten globalen Gleichungssystems. Die Kräfte betragen  $F_x = 0.01$  kN und  $F_y = -0.05$  kN.

## Aufgabe 5.2: Schreiben eines FEM-Programms

Im Folgenden soll ein eigener FE-Code in MATLAB erzeugt werden, mithilfe dessen die Knotenverschiebungen von ebenen Fachwerken berechnet werden können. Der Workflow des Programms kann Abbildung 5.2 entnommen werden. Dabei stellt die Datei `PlaneTruss.m` die Hauptdatei dar. Mit ihr werden zunächst Element- und Knotendaten des Fachwerkes eingelesen. Daraufhin werden mithilfe der Subroutinen `PlaneTrussElementStiffness.m` und `PlaneTrussAssemble.m` die Elementsteifigkeitsmatrizen aufgestellt und im globalen Gleichungssystem assembliert. Schließlich wird der globale Lastvektor aufgestellt und das globale Gleichungssystem gelöst.

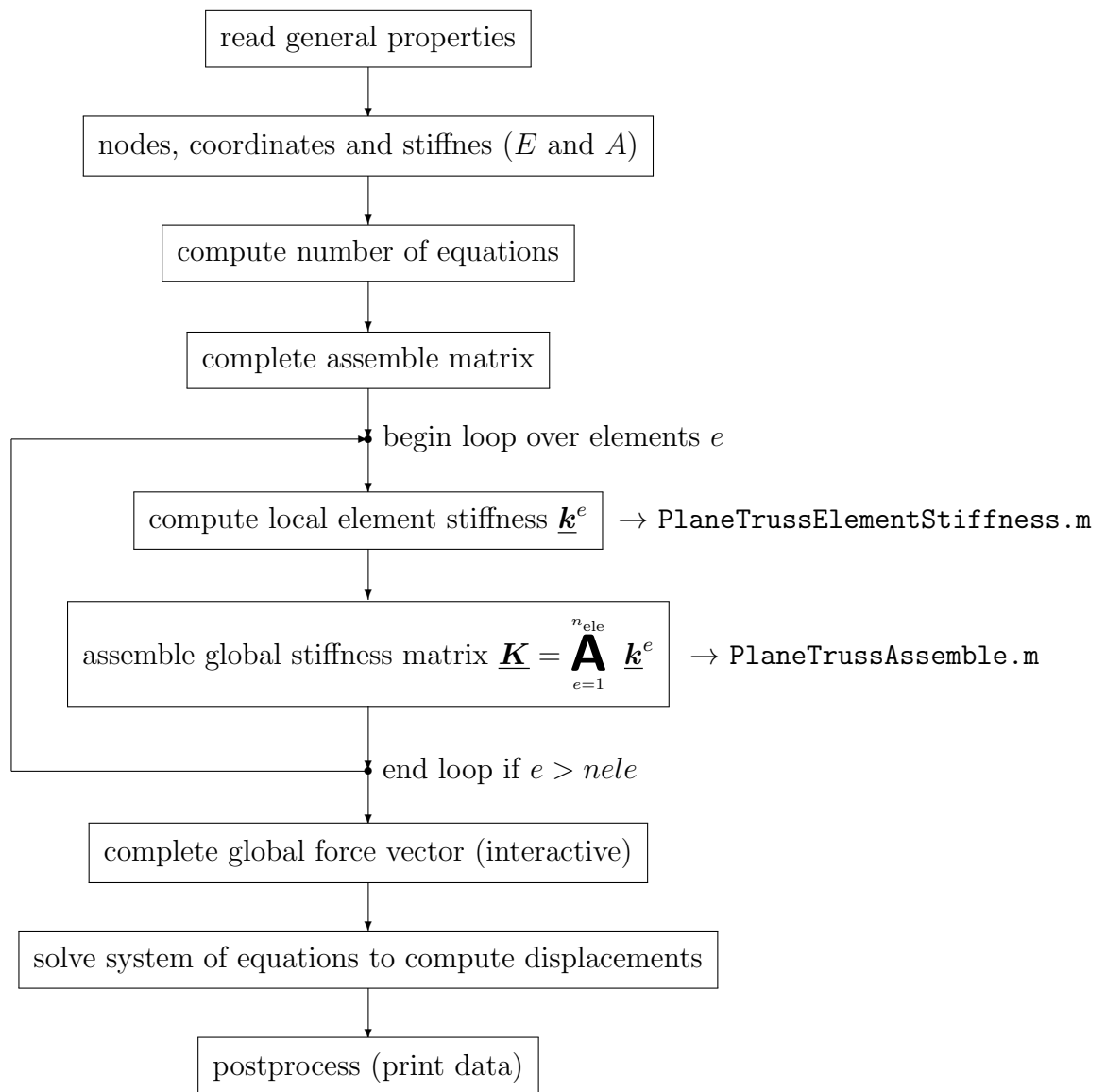
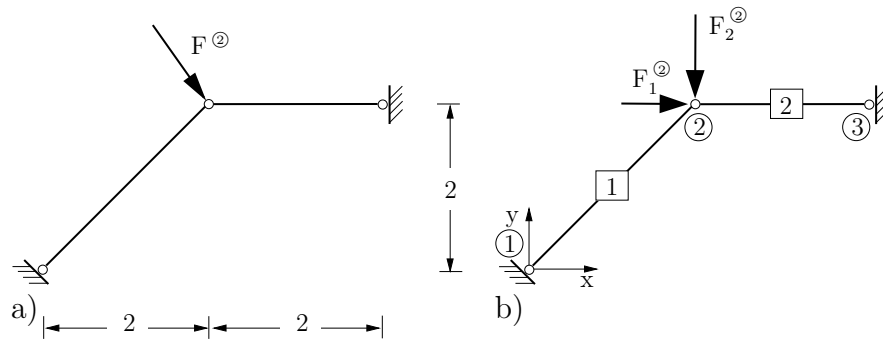


Abbildung 5.2: Workflow von `PlaneTruss.m`

- a) Erzeugen sie zunächst in der Hauptdatei `PlaneTruss.m` ein Interface, mit welchem die Element- und Knotendaten sowie die Dirichlet-Randbedingungen eingelesen werden.

*Hinweis:* Anhand des einfachen Fachwerks in Abbildung 5.3 soll veranschaulicht werden wie die entsprechenden Daten eingelesen und abgespeichert werden.



**Abbildung 5.3:** Fachwerk: a) Dimension (in Metern) und Randbedingungen, b) Diskretisierung.

Nach dem Programmstart soll zunächst folgender Input erfolgen:

```
>> PlaneTruss
give the total number of elements. num_ele = 2
give the total number of nodes.  num_nodes = 3
```

Im nächsten Schritt wird die Matrix `connectivity` erstellt. In ihr wird abgespeichert, welche globale Knotennummer zu welchem element gehört. Sie hat die Dimension `[nodes_ele × num_ele]`, wobei `nodes_ele = 2` die Anzahl der Knoten je Element darstellt. In diesem Beispiel besitzt Element 1 die globalen Knoten 1 und 2; Element 2 hat die Knoten 2 und 3. Die `connectivity`-Matrix sieht wie folgt aus:

$$\text{connectivity} = \begin{bmatrix} 1 & 2 \\ 2 & 3 \end{bmatrix}$$

Der E-Modul sei mit  $E = 10000 \text{ kN/m}^2$  festgelegt und die Querschnittsfläche der Elemente 1 und 2 sei

$$0.02 \text{ m}^2 \quad \text{und} \quad 0.015 \text{ m}^2.$$

Die Elementdaten werden nun wie folgt eingegeben

```
element no.1
  global node for 1st local node = 1
  global node for 2nd local node = 2
  element data      [ E A ] = [10000 0.02]

element no.2
  global node for 1st local node = 2
  global node for 2nd local node = 3
  element data      [ E A ] = [10000 0.015],
```

wobei die eingegebenen Knotennummern in der `connectivity`-Matrix gespeichert werden.

Nun werden für alle Knoten die  $(x,y)$ -Koordinaten eingegeben. Die entsprechende `X`-Matrix der Knotenkoordinaten der Dimension  $[\text{dof\_nodes} \times \text{num\_nodes}]$  lautet:

$$\mathbf{X} = \begin{bmatrix} 0 & 2 & 4 \\ 0 & 2 & 2 \end{bmatrix}$$

Der Input sieht wie folgt aus:

```
node no.1
  global node coordinates [ x y ] = [0 0]
```

```
node no.2
  global node coordinates [ x y ] = [2 2]
```

```
node no.3
  global node coordinates [ x y ] = [4 2]
```

Die Anzahl an Dirichlet-Randbedingungen `num_bc` beträgt 4. Nach der Eingabe

```
give the total number of displacement boundary conditions
num_bc = 4
```

kann nun die Anzahl an verbleibenden Freiheitsgraden mit

$$\text{num\_eq} = \text{num\_nodes} * \text{dof\_nodes} - \text{num\_bc} \quad (5.1)$$

bestimmt werden. Die Horizontal- und Vertikalverschiebungen der Knoten 1 und 3 werden auf Null gesetzt:

$$d_1^{(1)} = d_2^{(1)} = d_1^{(3)} = d_2^{(3)} = 0,$$

bzw.

$$D_1 = D_2 = D_5 = D_6 = 0.$$

Im Zusammenhang mit diesem Schritt wird die Matrix `assembleid` erstellt. Sie hat die Dimension  $[\text{dof\_nodes} \times \text{num\_nodes}]$  und beinhaltet alle verbleibenden Freiheitsgrade. An den Positionen, an denen Dirichlet-Randbedingungen vorliegen, befinden sich Nulleinträge. Für das Beispiel lautet sie also:

$$\text{assembleid} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 2 & 0 \end{bmatrix}$$

Die Inputsequenz dazu lautet:

```
input of global dofs with zero boundary displacements
  for i from 1 to num_bc
    global node of constrained degree of freedom = 1
    degree of freedom to constrained (1=x,2=y) = 1
```

```

global node of constrained degree of freedom = 1
degree of freedom to constrained (1=x,2=y) = 2
global node of constrained degree of freedom = 3
degree of freedom to constrained (1=x,2=y) = 1
global node of constrained degree of freedom = 3
degree of freedom to constrained (1=x,2=y) = 2

```

- b) Setzen Sie die Assemblierung der Elementsteifigkeitsmatrizen zur globalen Steifigkeitsmatrix algorithmisch in form von MATLAB Code in der Hauptdatei `PlaneTruss.m` und mithilfe der Subroutinen `PlaneTrussElementStiffness.m` und `PlaneTrussAssemble.m` um.

*Hinweis:* Die Assemblierung findet in einer Schleife über die Elemente statt, in der für jedes Element zunächst die lokale Steifigkeitsmatrix  $\underline{k}^e$  (mithilfe von `PlaneTrussElementStiffness.m`) erzeugt wird. Mithilfe der Subroutine `PlaneTrussAssemble.m` soll die globale Steifigkeitsmatrix nun an den passenden Einträgen mit den jeweiligen Einträgen der lokalen Steifigkeitsmatrix besetzt werden. Um diesen Prozess zu veranschaulichen wird das Fachwerk in Abbildung 5.4 betrachtet

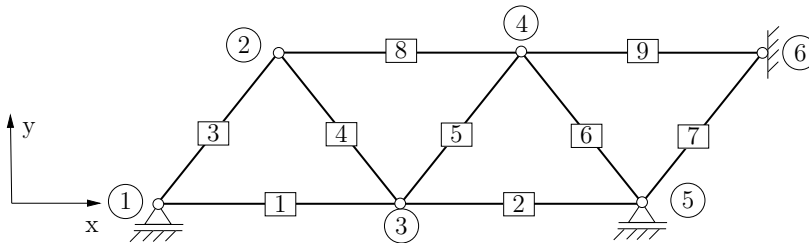


Abbildung 5.4

Die `connectivity`-Matrix lautet:

$$\text{connectivity} = \begin{bmatrix} 1 & 3 & 1 & 2 & 3 & 4 & 5 & 2 & 4 \\ 3 & 5 & 2 & 3 & 4 & 5 & 6 & 4 & 6 \end{bmatrix}$$

Das Randwertproblem besteht aus neun Stabelementen, sechs Knoten und vier essentiellen Randbedingungen, d.h.

$$\text{num\_ele} = 9, \quad \text{num\_nodes} = 6, \quad \text{num\_bc} = 4.$$

Die Elementsteifigkeitsmatrix  $\underline{k}^e$  hat die Dimension  $[\underline{k}_{\text{dim}}^e \times \underline{k}_{\text{dim}}^e]$  mit

$$\underline{k}_{\text{dim}}^e = \text{nodes\_ele} * \text{dof\_nodes} = 4.$$

Die Dimension der globalen Steifigkeitsmatrix  $\underline{K}$  unter Berücksichtigung der Dirichlet Randbedingungen ist  $[\text{num\_eq} \times \text{num\_eq}]$  mit  $\text{num\_eq} = 8$  (5.1). Die `assembleid`-Matrix lautet

$$\text{assembleid} = \begin{bmatrix} 1 & 2 & 4 & 6 & 8 & 0 \\ 0 & 3 & 5 & 7 & 0 & 0 \end{bmatrix}.$$

Mit ihrer Hilfe werden die einzelnen Einträge  $[\mathbf{k}^e]_{ij}$  der Elementsteifigkeitsmatrix den Einträgen  $[\mathbf{K}]_{IJ}$  der globalen Steifigkeitsmatrix zugeordnet. Bevor dies geschieht wird jedoch zunächst die leere globale Steifigkeitsmatrix aufgestellt mit

$$K_{IJ} = 0.0 \quad \text{für} \quad I, J = 1, \dots, 8.$$

Nun wird der Element-Schleifendurchlauf gestartet. In jedem Iterationsschritt wird zunächst die lokale Elementsteifigkeitsmatrix  $\mathbf{k}^e$  mit `PlaneTrussElementStiffness.m` erzeugt. Daraufhin wird mithilfe der Subroutine `PlaneTrussAssemble.m` die globale Steifigkeitsmatrix  $\mathbf{K}$  geupdatet. Um diesen Vorgang zu verdeutlichen wird der erste Iterationsschritt (Element  $e = 1$ ) betrachtet. Die Positionen I und J in der globalen Matrix können den Einträgen der `connectivity`- und `assembleid`- Matrix entnommen werden:

$$\begin{aligned} I &= \text{assembleid}(:, \text{connectivity}(1, 1)) = [1 \ 0] \\ J &= \text{assembleid}(:, \text{connectivity}(1, 2)) = [4 \ 5] \end{aligned}$$

mit der Elementsteifigkeitsmatrix

$$\mathbf{k}^1 = \begin{array}{c|cccc} & 1 & 0 & 4 & 5 & \text{global dof} \\ \hline \left[ \begin{array}{cccc} k_{11}^1 & k_{12}^1 & k_{13}^1 & k_{14}^1 \\ k_{21}^1 & k_{22}^1 & k_{23}^1 & k_{24}^1 \\ k_{31}^1 & k_{32}^1 & k_{33}^1 & k_{34}^1 \\ k_{41}^1 & k_{42}^1 & k_{43}^1 & k_{44}^1 \end{array} \right] & 1 & 0 & 4 & 5 & . \end{array}$$

Die ersten zwei Zeilen und Spalten von  $\mathbf{k}^1$  sind dem vertikalen und horizontalen Freiheitsgrad des globalen Knotens 1 zugeordnet; die dritte und vierte Zeile und Spalte sind dem globalen Knoten 3 zugeordnet. Mit der dieser Information folgt nun der Update-Vorgang

$$\begin{aligned} K_{11} &\leftarrow K_{11} + k_{11}^1, & K_{14} &\leftarrow K_{14} + k_{13}^1, & K_{15} &\leftarrow K_{15} + k_{14}^1, \\ K_{41} &\leftarrow K_{41} + k_{31}^1, & K_{44} &\leftarrow K_{44} + k_{33}^1, & K_{45} &\leftarrow K_{45} + k_{34}^1, \\ K_{51} &\leftarrow K_{51} + k_{41}^1, & K_{54} &\leftarrow K_{54} + k_{43}^1, & K_{55} &\leftarrow K_{55} + k_{44}^1. \end{aligned} \quad (5.2)$$

Es wird deutlich, dass die Einträge der zweiten Zeile und Spalte nicht in das Update der globalen Matrix eingehen, da die zugehörigen globalen Freiheitsgrade aufgrund der Dirichlet-Randbedingungen in der `assembleid`-Matrix mit Null gekennzeichnet sind. Für den zweiten Iterationsschritt  $e = 2$ , bei dem das zweite Element betrachtet wird, folgt:

$$\underline{\mathbf{k}}^{[2]} = \begin{array}{c|cccc} & 4 & 5 & 8 & 0 \\ \hline \begin{bmatrix} k_{11}^{[2]} & k_{12}^{[2]} & k_{13}^{[2]} & k_{14}^{[2]} \\ k_{21}^{[1]} & k_{22}^{[2]} & k_{23}^{[2]} & k_{24}^{[2]} \\ k_{31}^{[2]} & k_{32}^{[2]} & k_{33}^{[2]} & k_{34}^{[1]} \\ k_{41}^{[2]} & k_{42}^{[2]} & k_{43}^{[2]} & k_{44}^{[1]} \end{bmatrix} & \text{global dof} \\ \hline & 4 & 5 & 8 & 0 \end{array} .$$

$$\begin{aligned} K_{44} &\Leftarrow K_{44} + k_{11}^{[2]}, & K_{45} &\Leftarrow K_{45} + k_{12}^{[2]}, & K_{48} &\Leftarrow K_{48} + k_{13}^{[2]}, \\ K_{54} &\Leftarrow K_{54} + k_{21}^{[2]}, & K_{55} &\Leftarrow K_{55} + k_{22}^{[2]}, & K_{58} &\Leftarrow K_{58} + k_{23}^{[2]}, \\ K_{84} &\Leftarrow K_{84} + k_{31}^{[2]}, & K_{85} &\Leftarrow K_{85} + k_{32}^{[2]}, & K_{88} &\Leftarrow K_{88} + k_{33}^{[2]}. \end{aligned} \quad (5.3)$$

In gleicher Weise wird für die weiteren Elemente verfahren bis der Schleifendurchlauf über alle Elemente abgeschlossen ist. Schließlich bleibt noch zu berücksichtigen, dass ein Element je nach vorliegenden Dirichlet Randbedingungen eine unterschiedliche Anzahl von Freiheitsgraden aufweisen kann. Um diese Tatsache algorithmisch zu handhaben kann die Subroutine `PlaneTrussAssemble.m` als Kontrollroutine betrachtet werden welche mithilfe von Fallunterscheidung prüft wieviele aktive Freiheitsgrade im jeweiligen Element vorliegen. Besitzt ein Element nur einen aktiven Freiheitsgrad so wird zum Update der globalen Steifigkeitsmatrix die Unterfunktion `Assemble1.m` aufgerufen. Liegen zwei aktive Freiheitsgrade vor wird die Unterfunktion `Assemble2.m` aufgerufen, und so weiter. In den Unterfunktionen selbst wird dann das Update, wie in (5.2) und (5.3) veranschaulicht, durchgeführt

- c) Versetzen Sie die Hauptdatei `PlaneTruss.m` nun mit einem Interface, welches den globalen Lastvektor (Neumann-Randbedingungen) einliest.
- d) Lassen Sie das Programm schließlich das globale Gleichungssystem lösen und die `connectivity`-Matrix,  $\underline{\mathbf{K}}$ ,  $\underline{\mathbf{R}}$  und  $\underline{\mathbf{D}}$  ausgeben.

*Hinweis:* Zum Verdeutlichen des Einlesevorgangs des Lastvektors  $\mathbf{R}$  ( $[\text{num\_eq} \times 1]$ ) wird wieder das Beispiel aus Abbildung 5.3 betrachtet. Wie Abbildung 5.3 zu entnehmen ist, besteht dieser aus dem lokalen Vektor  $\mathbf{F}^{(2)}$ , wofür folgende Werte gegeben seien:

$$\mathbf{F}^{(2)} = [\mathbf{F}_1^{(2)}, \mathbf{F}_2^{(2)}]^T = [0.01 \text{ kN}, -0.05 \text{ kN}]^T.$$

Das Interface sieht folgendermaßen aus:

```
give the total number of nodes with non-zero load vectors
                                num_loads = 1
                                input of load vectors for i from 1 to num_loads
no. of global node with non-zero load vector = 2
                                force in global x-direction = 0.01
                                force in global y-direction = -0.05
```

Der erste Output ist die zur **connectivity**-Matrix gehörige Tabelle

=====					
	e		Node i		Node j
=====					
	1		1		2
	2		2		3
=====					

Da **K**, **R** und **D** in dem Programm als Variablen auftreten kann der Output dieser Größen nach der Lösung direkt erfolgen.

- e) Verifizieren, ob ihr Programm richtig funktioniert, indem Sie Ihr FE-Programm für das Fachwerk aus Aufgabe 5.1: auswerten und mit der Lösung von Aufgabe 5.1:c) vergleichen.
- f) Was passiert wenn Sie bei der Eingabe keine Lagerungsrandbedingungen angeben? Wie lässt sich das beobachtete Verhalten begründen?