Lehrstuhl für Numerische Mechanik
TUM School of Engineering and Design
Technische Universität München

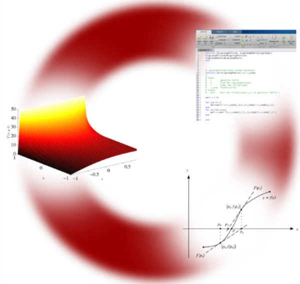
7. Numerische Lösung von partiellen DGL

A Practical Course in Numerical Methods for Engineers

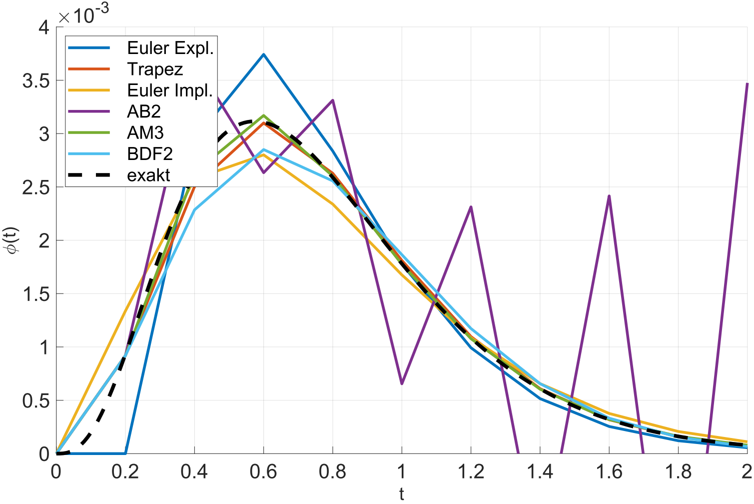
Barbara Wirthl, M.Sc.

Technische Universität München

Lehrstuhl für Numerische Mechanik



6. Aufgabenblatt



- Stabilität OST, AB2, AM3, BDF2

6. Aufgabenblatt: Stabilität

Vergleich Modellgleichung:

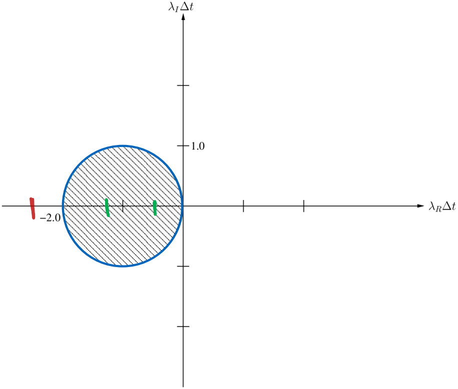
$$\frac{\mathrm{d}\phi}{\mathrm{d}t} = \underline{\lambda\phi} \quad \text{mit} \quad \phi(t=0) = \phi_0.$$

Entscheidendes Maß für Stabilität: $\lambda \Delta t$

Gleichung Aufgabenblatt 6:

$$\frac{\mathrm{d}\phi(t)}{\mathrm{d}t} = t^2 e^{-5t} - 6\phi(t) \quad \text{mit} \quad \phi(t=0) = 0 \quad \Rightarrow \quad \lambda = -6$$

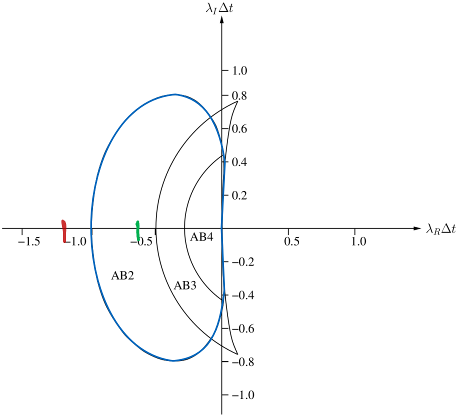
6. Aufgabenblatt



Vorwärts Euler (explizit)

Δt	$\lambda \Delta t$
0.1	-0.6
0.2	-1.2
0.4	-2.4

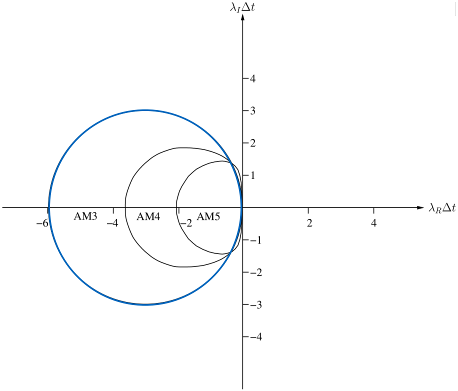
6. Aufgabenblatt



Adams–Bashforth 2. Ordnung (explizit)

Δt	$\lambda \Delta t$
0.1	-0.6
0.2	-1.2
0.4	-2.4

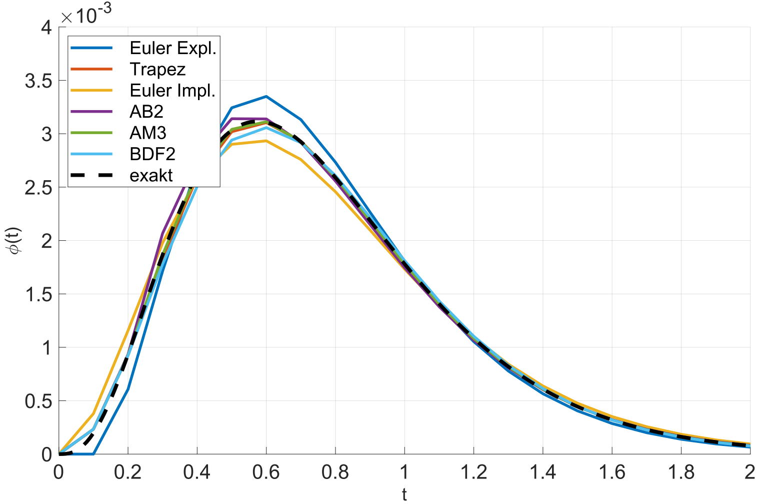
6. Aufgabenblatt



Adams–Moulton 3. Ordnung (implizit)

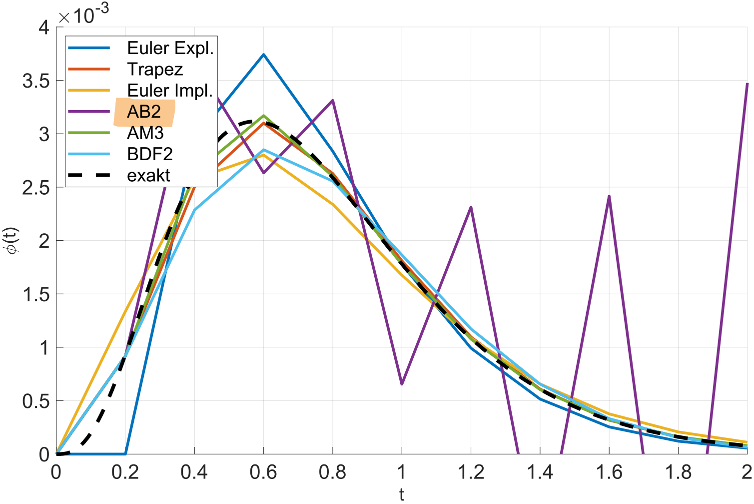
Δt	$\lambda \Delta t$
0.1	−0.6
0.2	−1.2
0.4	−2.4

6. Aufgabenblatt



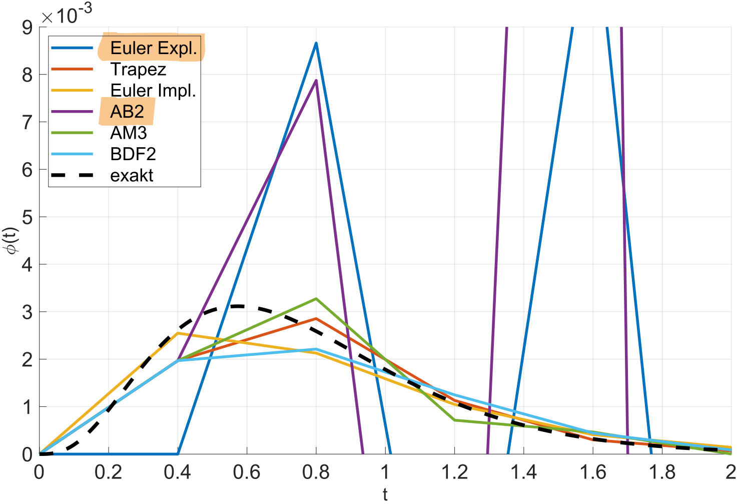
$$\lambda \Delta t = -0.6$$

6. Aufgabenblatt



$$\lambda \Delta t = -1.2$$

6. Aufgabenblatt



$$\lambda \Delta t = -2.4$$

Aufgabenblatt 7

Aufgabe: FEM - Stationäre Wärmeleitungsgleichung 2D

Gegeben ist die stationäre Wärmeleitungsgleichung in 2D:

$$-\nabla \cdot (\lambda \nabla T) = \dot{q} \quad \text{in } \Omega$$

$$\text{mit} \quad T = T_D \quad \text{auf} \quad \Gamma_D, \qquad \lambda \nabla T \cdot \boldsymbol{n} = 0 \quad \text{auf} \quad \Gamma_N.$$

Die Wärmeleitfähigkeit λ ist eine vom Material abhängige bekannte Größe. Es soll angenommen werden, dass an den Rändern (Γ_D-Dirichlet-Rand) entweder die Temperatur T vorgegeben wird bzw. die Ränder (Γ_N-Neumann-Rand) adiabat sind ($\lambda \nabla T \cdot \boldsymbol{n} = 0$). Des Weiteren sollen keine Quellen oder Senken vorhanden sein: also $\dot{q} = 0$.

Diskretisieren Sie die Gleichung mithilfe der Finite-Elemente-Methode mit bilinearen Vierecks-elementen. Zur numerischen Integration soll die Gauß-Quadratur mit $n = 2$ (Punkte in einer Richtung) verwendet werden.

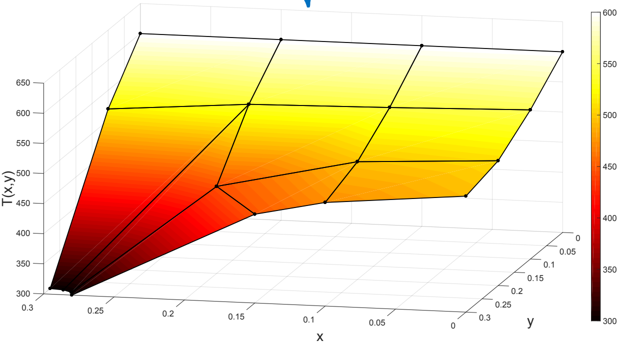
Finite Elemente: Ziel

Matrix $\underline{A}$ berechnen

$$
\begin{bmatrix}
a_{1,1} & a_{1,2} & & & \\
a_{2,1} & a_{2,2} & \ddots & & \\
 & \ddots & \ddots & & \\
 & & & a_{17,17} & a_{17,18} \\
 & & & a_{18,17} & a_{18,18}
\end{bmatrix}
\begin{bmatrix}
\widehat{T}_1 \\ \widehat{T}_2 \\ \dots \\ \widehat{T}_{17} \\ \widehat{T}_{18}
\end{bmatrix}
=
\begin{bmatrix}
f_1 \\ f_2 \\ \dots \\ f_{17} \\ f_{18}
\end{bmatrix}
$$

$$\downarrow$$

$\underline{T}$ berechnen



Schwache Form

Starke Form der stationären Wärmeleitungsgleichung:

$$-\nabla \cdot (\lambda \nabla T) = \dot{q} \quad \text{in } \Omega$$

1. Schritt: Multiplikation mit der Testfunktion v und Integration über das Gebiet Ω

$$-\int_\Omega v \nabla (\lambda \nabla T) \, d\underline{x} = \int_\Omega v \dot{q} \, d\underline{x}$$

2. Schritt: Partielle Integration des Terms auf der linken Seite

> 1. Greensche Identität:
>
> $$\int_\Omega f \, \nabla \cdot (\nabla g) \, \mathrm{d}\boldsymbol{x} = -\int_\Omega \nabla f \cdot \nabla g \, \mathrm{d}\boldsymbol{x} + \int_\Gamma (f \, \nabla g) \cdot \boldsymbol{n} \, \mathrm{d}\gamma$$
>
> $f \mathrel{\hat{=}} v$
>
> $g \mathrel{\hat{=}} \lambda T$

$$\int_\Omega \lambda \nabla v \, \nabla T \, d\underline{x} - \int_\Gamma v \lambda \nabla T \cdot \underline{n} \, d\gamma = \int_\Omega v q \, d\underline{x}$$

Schwache Form

$$\int_\Omega \lambda \nabla v \cdot \nabla T \, d\boldsymbol{x} - \int_\Gamma v \lambda \nabla T \cdot \boldsymbol{n} \, d\gamma = \overbrace{\int_\Omega v \dot{q} \, d\boldsymbol{x}}^{=0}$$

3. Schritt: Ausnutzung der Randbedingungen $\Gamma = \Gamma_N + \Gamma_D$

$$\int_\Gamma v \lambda \nabla T \cdot \boldsymbol{n} \, d\gamma = \underbrace{\int_{\Gamma_D} v \lambda \nabla T \cdot \boldsymbol{n} \, d\gamma}_{\substack{\textbf{Dirichlet-Rand} \\ v = 0}} + \underbrace{\int_{\Gamma_N} v \lambda \nabla T \cdot \boldsymbol{n} \, d\gamma}_{\substack{\textbf{Neumann-Rand} \\ \lambda \nabla T \cdot \underline{n} = 0}} = 0$$

Schwache Form:

$$\int_\Omega \lambda \nabla v \cdot \nabla T \, d\underline{x} = 0$$

Endlich-dimensionaler Ansatzraum

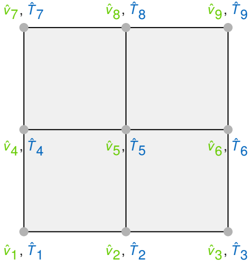
Ansatzfunktionen für Testfunktion und Temperaturfeld:

$$v^h(\boldsymbol{x}) = \sum_i N^i(\boldsymbol{x})\widehat{v}_i \qquad T^h(\boldsymbol{x}) = \sum_j N^j(\boldsymbol{x})\widehat{T}_j$$

Einsetzen:

$$\int_\Omega \lambda \nabla v^h \cdot \nabla T^h \,\mathrm{d}\boldsymbol{x} = 0$$

Beispiel:



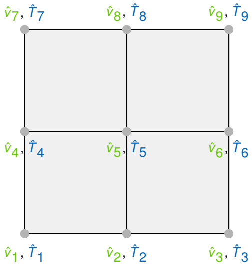
$$\sum_i \hat{v}_i \sum_j \int_\Omega \lambda \nabla N^i(\underline{x}) \; \nabla N^j(\underline{x}) \; d\underline{x} \; \hat{T}_j = 0$$

Muss für beliebige Werte $\widehat{v}_i$ gelten:

$$\sum_j \underbrace{\int_\Omega \lambda \nabla N^i(\underline{x}) \; \nabla N^j(\underline{x}) \; d\underline{x}}_{A_{ij}} \; \hat{T}_j = 0 \quad \forall i$$

Elementweise Betrachtung und numerische Integration

$$\sum_j \underbrace{\int_{\Omega} \lambda \nabla N^i(\boldsymbol{x}) \cdot \nabla N^j(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}}_{A_{ij}}\, \widehat{T}_j = 0 \qquad \forall\, i$$



Summe über alle Elemente e:

$$\sum_e \sum_j \underbrace{\int_{\Omega^{(e)}} \lambda \nabla N^i(\boldsymbol{x}) \cdot \nabla N^j(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}}_{A_{ij}^{(e)}}\, \widehat{T}_j = 0 \qquad \forall\, i$$

Transformation auf Referenzelement:

$$\sum_e \sum_j \int_{\Omega^{(e)}_{ref}} \lambda \nabla N^i(\xi) \cdot \nabla N^j(\xi) \det(J(\xi))\, \mathrm{d}\xi\, \widehat{T}_j = 0 \qquad \forall\, i$$

Gauß-Quadratur:

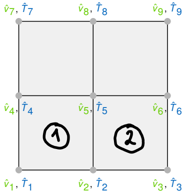
$$\sum_e \sum_j \underbrace{\sum_k \lambda \nabla N^i(\xi_k) \cdot \nabla N^j(\xi_k) \det(J(\boldsymbol{\xi_k}))w_k}_{A_{ij}^{(e)}}\, \widehat{T}_j = 0 \qquad \forall\, i$$

Umsetzung im Programm: Problembeschreibung

Beschreibung durch Knoten, Elemente und Dirichlet-Randbedingungen:

- Matrix `nodes`:
  Zeile ... Knoten, Spalte ... x, y-Koordinate

$$\begin{bmatrix} x_1 & y_1 \\ \cdot & \cdot \\ x_9 & y_9 \end{bmatrix}$$



- Matrix `elements`:
  Zeile ... Element, Spalte ... Knotenidx. [$n1$, $n2$, $n3$, $n4$]

$$\begin{bmatrix} 1 & 2 & 5 & 4 \\ 2 & 3 & 6 & 5 \\ & \cdot\cdot & & \end{bmatrix}$$

- Matrix `dbc`:
  Zeile ... Dirichlet Boundary Condition, Spalte ... [*Knotenindex*, *Wert*]

$$\begin{bmatrix} 7 & 100 \\ 8 & 10.8 \\ \multicolumn{2}{c}{\ldots} \end{bmatrix}$$

Umsetzung im Programm: Programmaufbau

$$A_{ij}^{(e)} = \sum_k \lambda \nabla N^i(\xi_k) \cdot \nabla N^j(\xi_k) \det(J(\xi_{\boldsymbol{k}})) w_k$$

Schleife über alle Elemente

`Funktion Evaluieren()` $\longrightarrow \underline{A}^{(e)}$

Schleife über alle Gaußpunkte im Element $\sum_k$

Schleife über alle Zeilen der Elementmatrix i

Schleife über alle Spalten der Elementmatrix j

...

`Funktion Assemblieren()` $\underline{A}^{(e)} \longrightarrow \underline{A}$

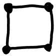
`Funktion Dirichlet()`

`Funktion Lösen()`

Umsetzung im Programm: `Evaluieren()`

$$A_{ij}^{(e)} = \sum_k \lambda \nabla N^i(\xi_k) \cdot \nabla N^j(\xi_k) \det(J(\boldsymbol{\xi_k})) w_k$$

```
[elemat,elevec] = evaluate_stat(elenodes,gpx,gpw)
```



Berechne die Elementsteifigkeitsmatrix $\boldsymbol{A}^{(e)}$

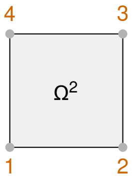
$\rightarrow$ (4×4)-Matrix (i, j) für lineares Viereckselement

$\underline{A}^{(e)}$ 4×4

- Größen: $\xi_k, w_k, \det(J(\boldsymbol{\xi_k})), \frac{\partial N^i}{\partial \xi}, J(\boldsymbol{\xi_k})^{-1} = \frac{\partial \xi}{\partial \boldsymbol{x}}$ aus bereits erstellten Funktionen
- $\nabla N^i(\xi_k) = \frac{\partial N^i}{\partial \boldsymbol{x}} = \frac{\partial N^i}{\partial \xi} \frac{\partial \xi}{\partial \boldsymbol{x}}$
- Enthält somit: Schleife über Zeilen und Spalten der Elementsteifigkeitsmatrix sowie Schleife über Gaußpunkte
- Für den allgemeinen Fall (instationär) wird hier auch die rechte Seite der Gleichung berechnet (Elementlastvektor)

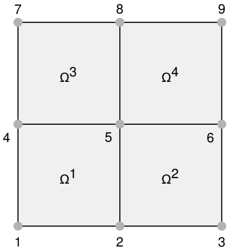
Umsetzung im Programm: `Assemblieren()`

Lokal:



1	2
2	3
3	6
4	5

Global:



elemat =

	1	2	3	4
1	A	B	C	D
2	E	F	G	H
3	...			
4				

sysmat =

			1	2		4	3			
		1	2	3	4	5	6	7	8	9
	1									
1	2		A	B		D	C			
2	3		E	F		H	G			
	4									
4	5									
3	6		..							
	7									
	8									
	9									

Umsetzung im Programm: `Assemblieren()`

```
[sysmat, rhs] = assemble(elemat,elerhs,sysmat,rhs,ele)
```

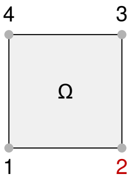
- Funktion Assemblieren: Assembliere die Elementsteifigkeitsmatrix $\boldsymbol{A}^{(e)}$ (und den Elementlastvektor $\boldsymbol{f}^{(e)}$) in die globale Systemmatrix $\boldsymbol{A}$ (sowie den globalen Lastvektor $\boldsymbol{f}$)
- Zusammenhänge zwischen den lokalen Indizes und globalen Indizes liegen in der Matrix `elements`
- Schleife über Spalten, Zeilen der Elementsteifigkeitsmatrix - Addieren in der globalen Systemmatrix

Umsetzung im Programm: `Dirichlet()`

```
[dbcsysmat,dbcrhs] = assignDBC(sysmat,rhs,dbc)
```

- Funktion Dirichlet: Berücksichtigung der Dirichlet Ränder in der Systemmatrix $\boldsymbol{A}$ bzw. im Systemvektor $\boldsymbol{f}$
- Wenden Sie folgendes Vorgehen auf alle Zeilen mit einer Dirichlet-Randbedingung an:
  1. Ersetzen Sie die Zeile der Systemmatrix durch 0-Einträge und setzen Sie den Wert auf der Hauptdiagonalen zu 1.
  2. Ersetzen Sie den entsprechenden Eintrag im Systemvektor durch den Dirichletwert.

Umsetzung im Programm: `Dirichlet()`



Gleichungssystem: $\boldsymbol{A}\widehat{\boldsymbol{T}} = \boldsymbol{0}$

Beispiel Dirichlet-Randbedingung: $\widehat{T}_2 = 0.5$

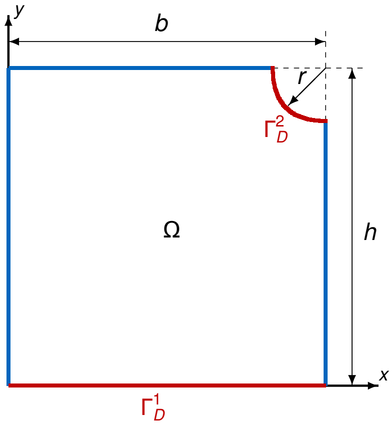
$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} \widehat{T}_1 \\ \widehat{T}_2 \\ \widehat{T}_3 \\ \widehat{T}_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & 1 & 0 & 0 \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} \widehat{T}_1 \\ \widehat{T}_2 \\ \widehat{T}_3 \\ \widehat{T}_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0.5 \\ 0 \\ 0 \end{pmatrix}$$

Umsetzung im Programm: `Lösen()`

- Lösen Sie das Gleichungssystem in Matlab mit `mldivide`, $\backslash$
- Lineares Gleichungssystem: $\boldsymbol{A}\widehat{\boldsymbol{T}} = \boldsymbol{f}$

Problemstellung



$\lambda(x, y) = 48.0 \, \frac{\mathrm{W}}{\mathrm{mK}}$ in Ω

Neumann-Rand:

$\lambda \nabla T \cdot \boldsymbol{n} = 0.0$ auf $\{\Gamma \setminus (\Gamma_D^1 \cup \Gamma_D^2)\}$

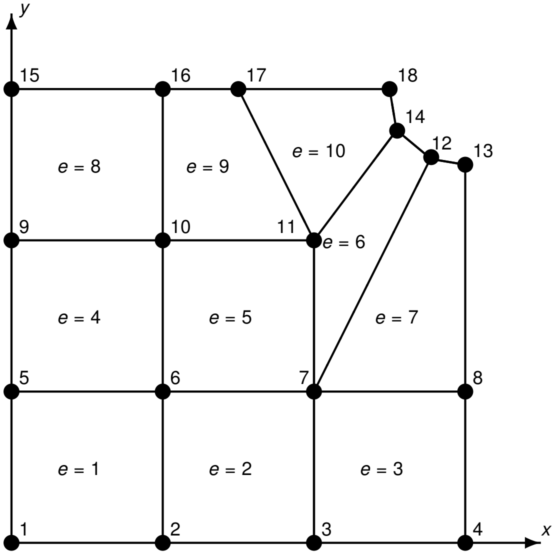
Dirichlet-Rand:

$T = 600\,\mathrm{K}$ auf Γ_D^1

$T = 300\,\mathrm{K}$ auf Γ_D^2

$r = 0.02\,\mathrm{m}$, $b = 0.3\,\mathrm{m}$, $h = 0.3\,\mathrm{m}$

Problemstellung



$$\boldsymbol{x}_{12} = \begin{bmatrix} b - r\sin(\frac{\pi}{6}) \\ h - r\cos(\frac{\pi}{6}) \end{bmatrix}$$

$$\boldsymbol{x}_{13} = \begin{bmatrix} b \\ h - r \end{bmatrix}$$

$$\boldsymbol{x}_{14} = \begin{bmatrix} b - r\cos(\frac{\pi}{6}) \\ h - r\sin(\frac{\pi}{6}) \end{bmatrix}$$

$$\boldsymbol{x}_{17} = \begin{bmatrix} \frac{b}{2} \\ h \end{bmatrix}$$

$$\boldsymbol{x}_{18} = \begin{bmatrix} b - r \\ h \end{bmatrix}$$

Und los...

Nächste Tutorsprechstunden:

Montag	12.12. 10:00 – 12:15 Uhr	MW1264
Mittwoch	14.12. 15:30 – 17:45 Uhr	MW1264

Nächstes Aufgabenblatt:

Donnerstag	15.12. 17:00 – 17:45 Uhr	MW2050 + Zoom