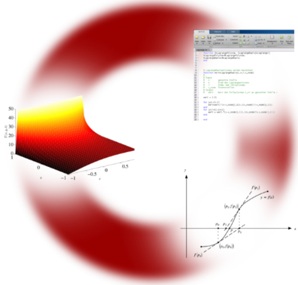
Lehrstuhl für Numerische Mechanik
TUM School of Engineering and Design
Technische Universität München

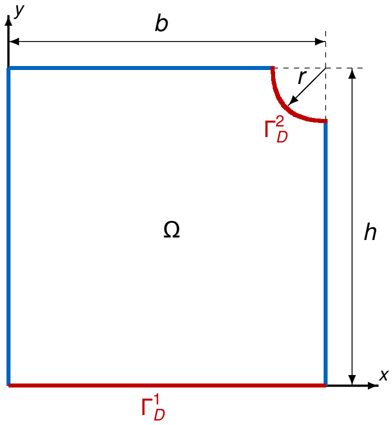
8. Numerische Lösung von instationären pDGL

A Practical Course in Numerical Methods for Engineers

Barbara Wirthl, M.Sc.
Technische Universität München
Lehrstuhl für Numerische Mechanik



2D FE stationär: 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

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

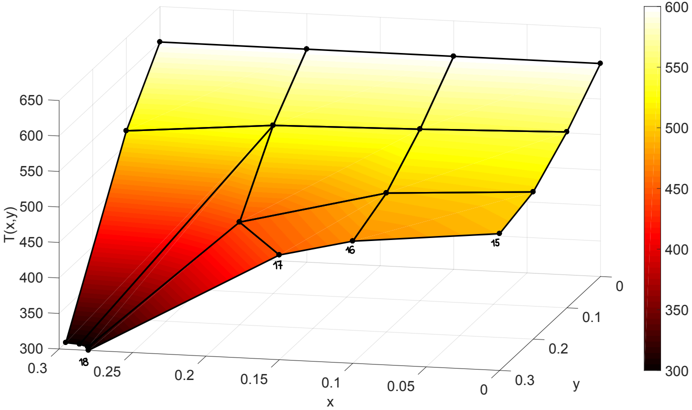
Pingo

Stationäre Wärmeleitung

Wie weit sind Sie bei dieser Aufgabe bekommen?

- Alles korrekt gelöst.
- `evaluate_stat` fehlt/falsch.
- `assemble` fehlt/falsch.
- Etwas anderes fehlt/falsch.

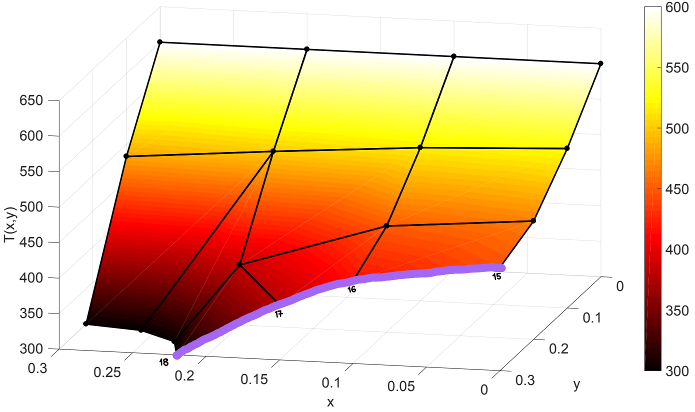
2D FE stationär



$$T_{15} = 492.64,\ \ T_{16} = 472.02,\ \ T_{17} = 447.45,\ \ T_{18} = 300.0$$

2D FE stationär

Maximal zulässige Temperatur $T_k = 450\,\mathrm{K}$



$T_{15} = 492.64,\ T_{16} = 472.02,\ T_{17} = 447.45,\ T_{18} = 300.0,\ r\ = 0.02\,\mathrm{m}$

$T_{15} = 445.48,\ T_{16} = 416.27,\ T_{17} = 377.64,\ T_{18} = 300.0,\ r^* = 0.08\,\mathrm{m}$

Aufgabenblatt 8

FEM – Instationäre Wärmeleitungsgleichung 2D

$T^0 \rightarrow T^1 \rightarrow T^2 .$

$$\rho c \frac{\partial T}{\partial t} - \nabla \cdot (\lambda \nabla T) = \dot{q} \quad \text{in } \Omega \times [0, t^*]$$

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

$$\text{mit} \quad T = T_0 \quad \text{in } \Omega \quad \text{für } t = 0.$$

Die Dichte ρ sowie die spezifische Wärmekapazität c sind vom Material abhängig und gegeben. Unter Beibehaltung aller Einschränkungen und Definitionen aus Arbeitsblatt 7 ($\dot{q} = 0$ und $\lambda \nabla T \cdot \boldsymbol{n} = 0$ an den Neumann-Rändern) soll nun die instationäre Wärmeleitungsgleichung in 2D mittels Matlab gelöst werden. Erweitern Sie das in Arbeitsblatt 7 erstellte Programm für den instationären Fall so, dass die bereits erstellten Funktionen zur Zeitintegration (**Fkt. IX**, **Fkt. X**, **Fkt. XI**, **Fkt. XII**) verwendet werden können.

Schwache Form

Starke Form der instationären Wärmeleitungsgleichung:

$$\rho c \frac{\partial T}{\partial t} - \nabla \cdot (\lambda \nabla T) = \dot{q} \quad \text{in } \Omega \times [0, t^*]$$

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

$$\rho c \int_\Omega v \frac{\partial T}{\partial t} \, d\underline{x} - \int_\Omega v \nabla (\lambda \nabla T) \, d\underline{x} = 0$$

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

$$\rho c \int_\Omega v \frac{\partial T}{\partial t} \, d\underline{x} + \int_\Omega \lambda \nabla v \cdot \nabla T \, d\underline{x} - \int_\Gamma v \lambda \nabla T \cdot \underline{n} \, d\gamma = 0$$

3. Schritt: Ausnutzung der Randbedingungen $\Gamma = \Gamma_N + \Gamma_D$ $\quad \int_\Gamma v \lambda \nabla T \cdot \underline{n} \, d\gamma = 0$

Schwache Form:

$$\rho c \int_\Omega v \frac{\partial T}{\partial t} \, d\underline{x} + \int_\Omega \lambda \nabla v \cdot \nabla T \, d\underline{x} = 0$$

Endlich-dimensionaler Ansatzraum

Ansatzfunktionen für Testfunktion und Temperaturfeld:

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

Einsetzen:

$$\rho c \int_\Omega v^h \frac{\partial T^h}{\partial t}\,\mathrm{d}\boldsymbol{x} + \int_\Omega \lambda \nabla v^h \cdot \nabla T^h\,\mathrm{d}\boldsymbol{x} = 0$$

$$\sum_i \hat{v}_i \sum_j \left[\rho c \int_\Omega N^i(\boldsymbol{x}) N^j(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\,\frac{\partial \hat{T}_j}{\partial t} + \int_\Omega \lambda \nabla N^i(\boldsymbol{x}) \cdot \nabla N^j(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}\,\hat{T}_j\right] = 0$$

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

$$\sum_j \Big[\underbrace{\rho c \int_\Omega N^i(\underline{x}) N^j(\underline{x})\,d\underline{x}}_{M_{ij}} \frac{\partial \hat{T}_j}{\partial t} + \underbrace{\int_\Omega \lambda \nabla N^i(\underline{x}) \cdot \nabla N^j(\underline{x})\,d\underline{x}}_{-B_{ij}}\,\hat{T}_j \Big] = 0 \quad \forall i$$

Vergleichen Sie diese Form mit der DGL auf Blatt 6: $M \frac{\mathrm{d}\phi(t)}{\mathrm{d}t} = B(t)\phi(t) + C(t)$

One–Step–θ-Verfahren zur Zeitintegration

Präsentation 6 (Folie 13) mit $C(t) = 0$:

$$M\phi_{n+1} = M\phi_n + \theta\,\Delta t\, B(t_{n+1})\phi_{n+1} + (1-\theta)\Delta t\, B(t_n)\phi_n$$

$$\sum_j \left[M_{ij} - \theta\,\Delta t\, B_{ij}\right] \underbrace{\hat{T}_j^{n+1}}_{\text{ges.}} = \sum_j \left[M_{ij} + (1-\theta)\Delta t\, B_{ij}\right] \hat{T}_j^{n} \qquad \forall i$$

bekannt

Verwenden Sie die in Arbeitsblatt 6 erstellten Funktionen für jedes einzelne Element:

$$\sum_e \sum_j \underbrace{\left[M_{ij}^{(e)} - \theta\,\Delta t\, B_{ij}^{(e)}\right]}_{LHS_{ij}^{(e)}} \hat{T}_j^{n+1} = \sum_e \underbrace{\sum_j \left[M_{ij}^{(e)} + (1-\theta)\Delta t\, B_{ij}^{(e)}\right] \hat{T}_j^{(e),n}}_{RHS_i^{(e)}} \qquad \forall i$$

Ergibt die Elementsteifigkeitsmatrix und den Elementlastvektor:

$LHS_{ij}^{(e)} \longrightarrow$ elemat 4x4

$RHS_i^{(e)} \longrightarrow$ elevec 4x1

Umsetzung im Programm: Programmaufbau

Zusätzliche Zeitschleife um den gesamten restlichen Code:

Schleife über alle Zeitschritte

- Schleife über alle Elemente
  - `Funktion Evaluieren()`
    - Schleife über alle Gaußpunkte im Element
      - Schleife über alle Zeilen der Elementmatrix
        - Schleife über alle Spalten der Elementmatrix
          - Anwenden der Zeitintegrationsmethoden
          - z.B. `Funktion OST()`, ... hier nicht LHS\RHS ↯
  - `Funktion Assemblieren()` elemat → sysmat elevec → rhs
- `Funktion Dirichlet()`
- `Funktion Lösen()` erst hier sysmat \ rhs !

Umsetzung im Programm: Schematischer Workflow

$$M_{ij}^{(e)} = \rho c \int_{\Omega^{(e)}} N^i(\boldsymbol{x}) N^j(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}$$

$$B_{ij}^{(e)} = -\int_{\Omega^{(e)}} \lambda \nabla N^i(\boldsymbol{x}) \cdot \nabla N^j(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}$$

`elesol` $= \boldsymbol{T}^{(e),n}$

`eleosol` $= \boldsymbol{T}^{(e),n-1}$ (AB2, AM3, BDF2)

↓

$\boxed{\text{OST, AB2, AM3, BDF2}}$

↓

Beiträge: $LHS_{ij}^{(e)}$, $RHS_{i}^{(e)}$

↓

alle Zeilen i, Spalten j sowie Summe über alle Gaußpunkte: elemat, elevec

↓

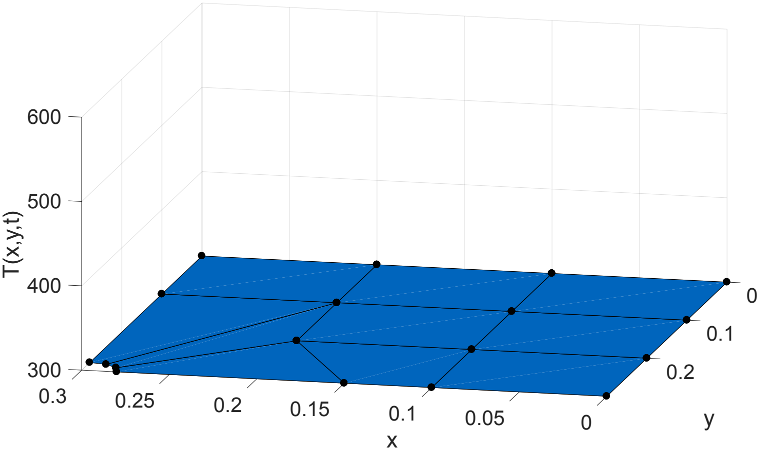
Elemente assemblieren + Randbedingungen aufbringen

↓

$$\underline{\underline{A}}\, \underline{T}^{n+1} = \underline{f}$$

Umsetzung im Programm: Initialbedingung

Zur Zeitintegration wird die Lösung T^0 zum Zeitpunkt $t = 0$ benötigt: $T^0 = 300\,\mathrm{K}$



Wärmeleitung 2D: Ausblick

Einige Punkte, die zur Vereinfachung weggelassen wurden:

- $\dot{q} \neq 0$
- $\lambda = \lambda(x, y)$
- $T_D = g(t)$ auf Γ_D
- $\lambda \nabla T \cdot \boldsymbol{n} \neq 0.0$ auf Γ_N
- andere Elementtypen (z.B. Dreieckselemente)
- Funktion, die das Netz (Knoten, Elemente, DBC) erstellt

(nicht überprüfungsrelevant)

Und los...

Nächste Tutorsprechstunden:

Montag	19.12.	10:00 – 12:15 Uhr	MW1264
Mittwoch	21.12.	15:30 – 17:45 Uhr	MW1264
Montag	09.01.	10:00 – 12:15 Uhr	MW1264
Mittwoch	11.01.	15:30 – 17:45 Uhr	MW1264

Nächstes Aufgabenblatt:

Donnerstag 12.01. 17:00 – 17:45 Uhr MW2050 + Zoom