Numerische Lösung von partiellen Differentialgleichungen

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 Viereckselementen. Zur numerischen Integration soll die Gauß-Quadratur mit $n = 2$ (Punkte in einer Richtung) verwendet werden. Erstellen Sie ein Programm in Matlab, welches die Lösung der stationären Wärmeleitungsgleichung in 2D unter den gegebenen Einschränkungen für eine beliebige Anordnung von Viereckselementen und vorgegeben Dirichlet-Randbedingung berechnet.

Hinweis zum Programmaufbau:

Schleife über alle Elemente

    `Funktion Evaluieren()`

        Schleife über alle Gaußpunkte im Element

            Schleife über alle Zeilen der Elementmatrix

                Schleife über alle Spalten der Elementmatrix

    `Funktion Assemblieren()`

`Funktion Dirichlet()`

`Funktion Lösen()`

Vorgehen: Die ersten 2 Punkte sind von Hand durchzuführen

1. Leiten Sie ausgehend von der gegebenen Gleichung die schwache Form der Gleichung unter Berücksichtigung der gegebenen Einschränkungen her.

2. Ersetzen Sie den unendlich-dimensionalen Funktionenraum der Testfunktionen sowie der Temperatur $T(\boldsymbol{x})$ durch einen endlich-dimensionalen Funktionenraum. Stellen Sie diesen mithilfe einer global nodalen Basis durch elementweise Lagrange'sche bilineare Ansatzfunktionen dar.

3. Erstellen Sie die Funktion **Fkt. XVII** zur Berechnung der Elementmatrizen $\boldsymbol{A}^{(e)}$. Verwenden Sie dazu die bereits vorbereiteten Funktionen aus den letzten Arbeitsblättern.

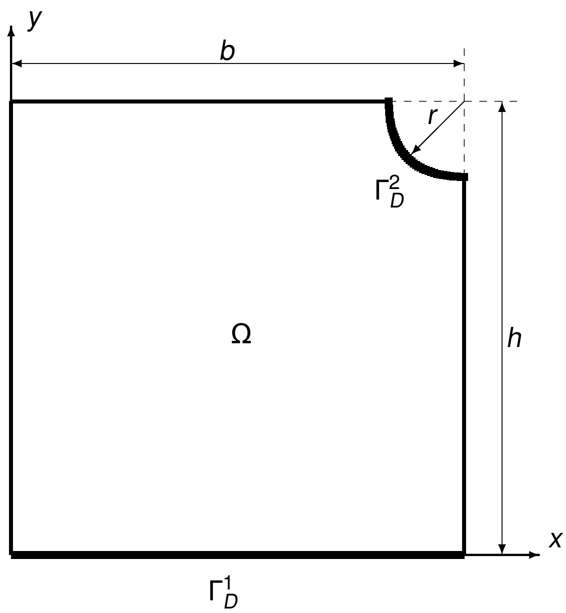
4. Erstellen Sie die Funktion **Fkt. XVIII** zur Assemblierung der Elementmatrizen $\boldsymbol{A}^{(e)}$ in die globale Matrix $\boldsymbol{A}$.

5. Erstellen Sie die Funktion **Fkt. XIX** zur Aufbringung der Dirichlet-Randbedingungen in die globale Matrix $\boldsymbol{A}$ sowie in den globalen Lastvektor $\boldsymbol{f}$.

6. Lösen Sie das Gleichungssystem $\boldsymbol{A}\boldsymbol{T} = \boldsymbol{f}$ mithilfe von Matlab (`Lösen`).

Wenden Sie das erstellte Programm auf folgende Problemstellung an:

Ein Stahlträger liegt an der Fläche $y = 0$ (Γ_D^1) auf einem Bauteil mit einer Temperatur von 600 K auf. Zur Kühlung des Trägers soll eine Aussparung (Γ_D^2), die von Wasser durchflossen wird, angebracht werden. Dadurch wird die Oberfläche dieser Aussparung auf eine Temperatur von 300 K gekühlt. Alle anderen Ränder werden perfekt isoliert. Die Temperaturverteilung im Bauteil und im Speziellen an der Fläche $y = h$ soll analysiert werden. Diese Analyse soll durch ein 2D-Modell am Querschnitt erfolgen.



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

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

$T = 600\,\mathrm{K}$ auf Γ_D^1

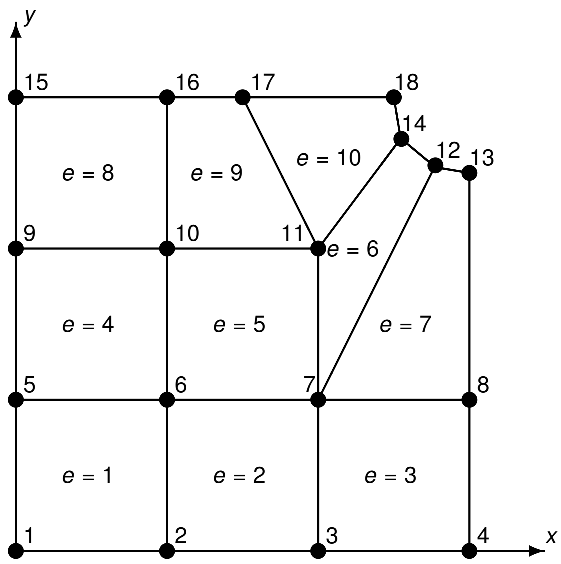
$T = 300\,\mathrm{K}$ auf Γ_D^2

$r = 0.02\,\mathrm{m}$

$b = 0.3\,\mathrm{m}$

$h = 0.3\,\mathrm{m}$

Vernetzen Sie die Problemstellung folgendermaßen:



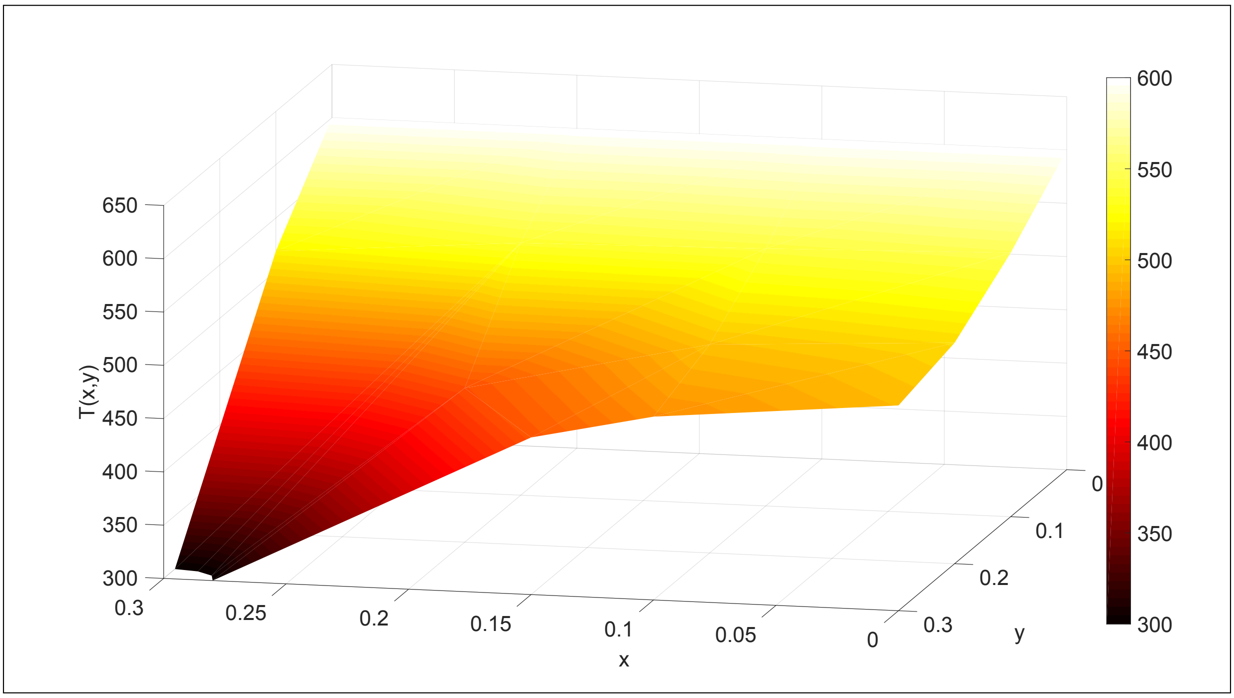
Spezielle Knotenkoordinaten:

$$\boldsymbol{x}_{12} = \begin{bmatrix} b - r\sin(\frac{\pi}{6}) \\ h - r\cos(\frac{\pi}{6}) \end{bmatrix}, \quad \boldsymbol{x}_{13} = \begin{bmatrix} b \\ h - r \end{bmatrix}, \quad \boldsymbol{x}_{14} = \begin{bmatrix} b - r\cos(\frac{\pi}{6}) \\ h - r\sin(\frac{\pi}{6}) \end{bmatrix}, \quad \boldsymbol{x}_{17} = \begin{bmatrix} \frac{b}{2} \\ h \end{bmatrix}, \quad \boldsymbol{x}_{18} = \begin{bmatrix} b - r \\ h \end{bmatrix}$$

Alle weiteren Knoten liegen auf dem strukturierten Gitter mit Weite $\frac{h}{3}$, bzw. $\frac{b}{3}$.

- Werten Sie die Temperatur in den Knoten $(15, 16, 17, 18)$ aus und erstellen Sie einen 3D-Plot der gesamten Temperaturverteilung (mithilfe von **Fkt. 0**).

  Lsg.: $T_{15} = 492.6433557959438$, $T_{16} = 472.0167230894156$, $T_{17} = 447.4502045054674$, $T_{18} = 300.0$



- An die Fläche $y = h$ soll ein temperaturkritisches Bauteil montiert werden. Die Effekte durch das angrenzende Bauteil werden nicht modelliert. Die maximal zulässige Temperatur $T_k = 450\,\mathrm{K}$ darf auf keinen Fall überschritten werden. Überprüfen Sie, ob dies bereits erfüllt ist. Falls nicht, erhöhen Sie den Radius der Ausnehmung schrittweise mit $r_{neu} = r + 0.01\,\mathrm{m}$, um die Kühlungsfläche zu erhöhen. Finden Sie somit den kleinst möglichen Radius r^*, um das angrenzende Bauteil nicht zu zerstören. Verschieben Sie nur die Punkte 12, 13, 14, 18 der Vernetzung. Erstellen Sie einen 3D-Plot der gesamten Temperaturverteilung in dieser Konfiguration (mithilfe von **Fkt. 0**).

Lsg.: $T_{15} = 445.47787636935$, $T_{16} = 416.2656515525$, $T_{17} = 377.6438051321$, $T_{18} = 300.0$, $r^* = 0.08\,\mathrm{m}$

Matlab-Funktionen

Folgende Funktionen sollen bei der Bearbeitung dieses Aufgabenblattes erstellt werden, da diese für spätere Aufgabenblätter wiederverwendet werden sollen. Erstellen Sie die Funktionen in Matlab und speichern Sie diese in eigenen *.m-files ab.

- **Fkt. XVII**: `function [elemat,elevec] = evaluate_stat(elenodes,gpx,gpw)`

  [elenodes ... [Knotenpositionen(Zeile: Knoten i, Spalte: x,y)], gpx ... [Positionen ξ_i Gauss-Integration], gpw ... [Gewichte w_i Gauss-Integration]]

  Rückgabewert: Elementmatrix $\boldsymbol{A}^{(e)}$ und Elementvektor $\boldsymbol{f}^{(e)}$

  Testen Sie die Funktion mit: ([0, 0; 1, 0; 1, 2; 0, 2], `gx2dref(3)`, `gw2dref(3)`) $\rightarrow$
  [[40.000000000000000,-28.000000000000004,-20.000000000000000,7.999999999999998;
  -28.000000000000004,40.000000000000007,8.000000000000000,-20.000000000000000;
  -20.000000000000000,8.000000000000000,39.999999999999993,-28.000000000000000;
  7.999999999999998,-20.000000000000000,-28.000000000000000,40.000000000000000],[0;0;0;0]]

- **Fkt. XVIII**: `function [sysmat,rhs] = assemble(elemat,elevec,sysmat,rhs,ele)`

  [elemat ... Elementmatrix $\boldsymbol{A}^{(e)}$, elevec ... Elementvektor $\boldsymbol{f}^{(e)}$, sysmat ... Systemmatrix $\boldsymbol{A}$, rhs ... Systemvektor $\boldsymbol{f}$, ele ... globale Knotenindex (als Zeilenvektor)]

  Rückgabewert: Systemmatrix $\boldsymbol{A}$, Systemvektor $\boldsymbol{f}$

  Testen Sie die Funktion mit:

  ([1, 2, 3, 4; 5, 6, 7, 8; 9, 10, 11, 12; 13, 14, 15, 16], [17; 18; 19; 20], `eye(5,5)`, `ones(5,1)`, [5, 3, 1, 2]) $\rightarrow$
  [[12,12,10,0,9;15,17,14,0,13;7,8,7,0,5;0,0,0,1,0;3,4,2,0,2],[20;21;19;1;18]]

- **Fkt. XIX**: `function [sysmat,rhs] = assignDBC(sysmat,rhs,dbc)`

  [sysmat ... Systemmatrix $\boldsymbol{A}$, rhs ... Systemvektor $\boldsymbol{f}$,
  dbc ... Dirichlet Randbedingung-Matrix (Zeile: (dbc), Spalte (1: Knotenindex, 2: Wert)]

  Rückgabewert: Systemmatrix $\boldsymbol{A}$, Systemvektor $\boldsymbol{f}$

  Testen Sie die Funktion mit:

  ([12, 12, 10, 0, 9; 15, 17, 14, 0, 13; 7, 8, 7, 0, 5; 0, 0, 0, 1, 0; 3, 4, 2, 0, 2], [20; 21; 19; 1; 18], [2, −7; 5, −2]) $\rightarrow$
  [[12,12,10,0,9;0,1,0,0,0;7,8,7,0,5;0,0,0,1,0;0,0,0,0,1],[20;-7;19;1;-2]]