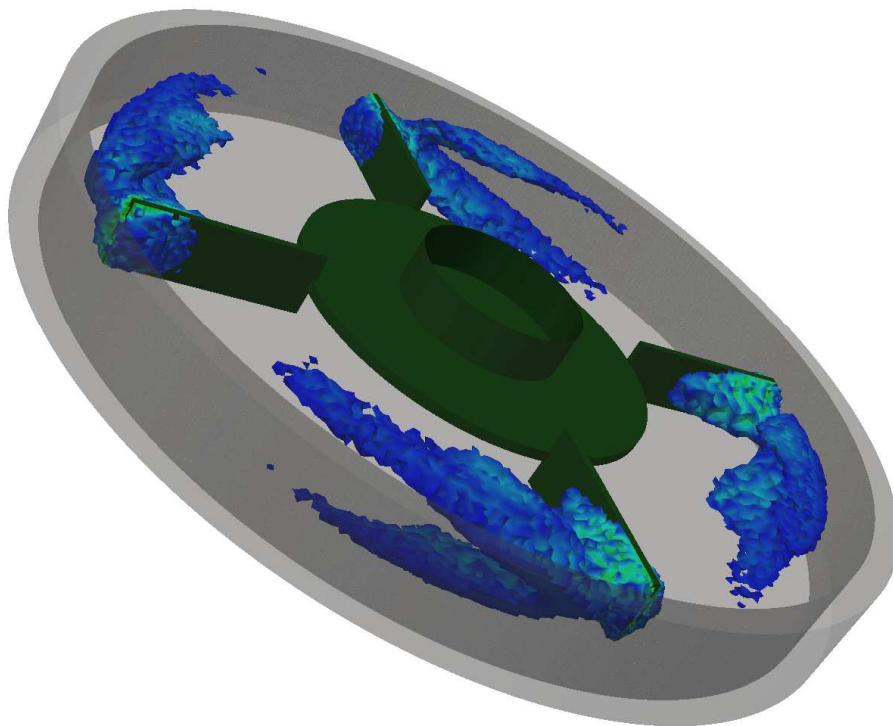


22959
Übungen zu Numerische Strömungssimulation
(CFD für CIW / VT)

Übung 1:

Einführung in die Finite-Differenzen-Methode



1 Motivation

Nach dem erfolgreichen Abschluss Ihres Studiums am KIT sind Sie in einer Firma als Prozess-Ingenieur für den Herstellungsprozess von Metallhalbzeugen verantwortlich. Hierbei werden bei einem Gießprozess lange, quaderförmige Eisenstangen produziert. Bevor die Stangen weiterverarbeitet werden können, muss die mittlere Temperatur der Stange auf unter 600°C abgekühlt sein. Hierzu werden die zunächst 1000°C heißen Halbzeuge in der Fertigungshalle mit siedendem Wasser berieselt.

Im Rahmen einer Prozessoptimierung sind Sie damit beauftragt, die Abkühlzeit zu reduzieren, um so eine Erhöhung des Durchsatzes zu erreichen.

Das Ziel dieser Übung besteht darin, die unter idealen Bedingungen notwendige, minimale Abkühlzeit zu bestimmen.

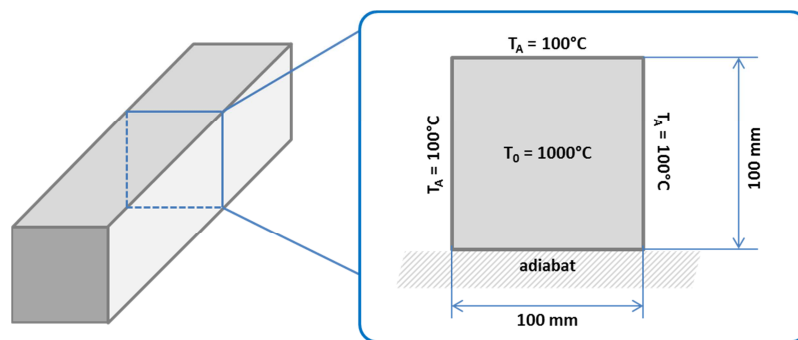


Abbildung 1: Problemstellung

Vereinfachend gehen Sie davon aus, dass der Stab unendlich lang ist, sodass Energieverluste über die Stirnkanten vernachlässigbar sind (vgl. Abbildung 1). Da der Stab auf dem gut isolierten Hallenboden aufliegt, kann die Unterseite des Stabes näherungsweise als adiabat angenommen werden. Ferner nehmen Sie an, dass aufgrund des Siedevorganges der Wärmeübergangskoeffizient zwischen Kühlwasser und Halbzeug groß ist und die Temperatur auf der Oberfläche stets 100°C beträgt.

Tabelle 1: Stoffparameter für Eisen

spezifische Dichte ρ_{Fe}	7874 kg/m ³
spezifische Wärmekapazität $c_{p,\text{Fe}}$	449 J/(kg K)
Wärmeleitfähigkeit λ_{Fe}	80 W/(m K)

2 Theoretische Grundlagen

Für einen Festkörper kann man für die Wärmeleitungsgleichung in drei Dimensionen schreiben

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (\lambda \cdot \nabla T) + \dot{q}. \quad (1)$$

Hierbei ist ρ die spezifische Dichte, c_p die spezifische Wärmekapazität und λ die Wärmeleitfähigkeit des Feststoffes. Die Größe \dot{q} bezeichnet Wärmequellterme innerhalb des Feststoffes.

Vernachlässigt man Wärmequellterme und nimmt ortsunabhängige, isotrope und temperaturunabhängige Stoffgrößen an, vereinfacht sich Gl. (1) zu

$$\rho c_p \frac{\partial T}{\partial t} = \lambda \Delta T. \quad (2)$$

Aufgrund der Annahme eines unendlich langen Stabes (vgl. Abbildung 1) ergibt sich ein Wärmeleitungsproblem in zwei Dimensionen

$$\rho c_p \frac{\partial T}{\partial t} = \lambda \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right). \quad (3)$$

2.1 Numerische Lösung

Sie entschließen sich dazu, Gl. (3) numerisch, mithilfe des finiten Differenzenverfahrens, zu lösen. Dabei werden die Differentialquotienten durch Differenzenquotienten ersetzt, welche auf den an Gitterpunkten definierten Variablen basieren (vgl. Abbildung 2). An jedem dieser Gitterpunkte wird Gl. (3) näherungsweise erfüllt. Entsprechend handelt es sich bei dieser numerischen Lösung um eine Approximation an die exakte Lösung, die umso genauer wird, je mehr Stützstellen vorhanden sind, also je feiner das Rechengitter ist.

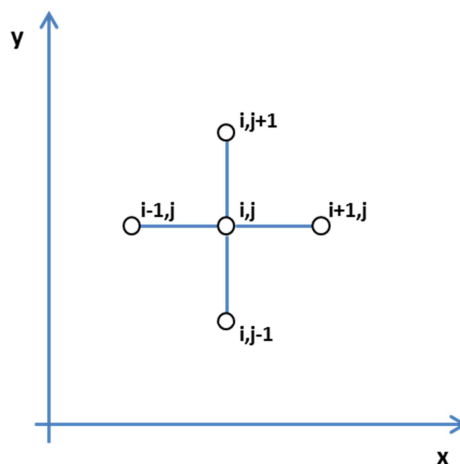


Abbildung 2: Diskretisierung im Raum mithilfe eines kartesischen Netzes

Für die Diskretisierung in der Zeit ergibt sich

$$\frac{\partial T}{\partial t} \approx \frac{T^{n+1} - T^n}{\Delta t}. \quad (4)$$

Hierbei ist $T_{i,j}^{n+1}$ bzw. $T_{i,j}^n$ die Temperatur am Punkt i,j zum Zeitpunkt $n+1$ bzw. n . Weiterhin bezeichnet Δt die im Folgenden als konstant angenommene Zeitschrittweite.

Aus einer Taylor-Reihen-Entwicklung kann die Finite-Differenzen Approximation der zweiten Ableitung in x - und y -Richtung abgeleitet werden, wobei im Folgenden die zentrale Differenz verwendet werden soll.

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{(\Delta x)^2} \quad (5)$$

$$\frac{\partial^2 T}{\partial y^2} \approx \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{(\Delta y)^2} \quad (6)$$

Verwendet man das explizite Euler-Verfahren (Euler-Vorwärts) ergibt sich nach Einsetzen der Gln. (4), (5) und (6) in Gl. (3)

$$\rho c_p \frac{T_{i,j}^{n+1} - T_{i,j}^n}{\Delta t} = \lambda \left(\frac{T_{i+1,j}^n - 2T_{i,j}^n + T_{i-1,j}^n}{(\Delta x)^2} + \frac{T_{i,j+1}^n - 2T_{i,j}^n + T_{i,j-1}^n}{(\Delta y)^2} \right). \quad (7)$$

Die explizite Auflösung nach der einzigen Unbekannten $T_{i,j}^{n+1}$ liefert mit der Tempera-

turleitfähigkeit $a = \frac{\lambda}{\rho c_p}$

$$T_{i,j}^{n+1} = T_{i,j}^n + a \Delta t \left(\frac{T_{i+1,j}^n - 2T_{i,j}^n + T_{i-1,j}^n}{(\Delta x)^2} + \frac{T_{i,j+1}^n - 2T_{i,j}^n + T_{i,j-1}^n}{(\Delta y)^2} \right). \quad (8)$$

2.2 Diskretisierung der Randbedingungen

Im Falle der vorliegenden Übung sind zwei unterschiedliche Randbedingungen an den Rändern des Rechengebietes zu realisieren:

- Vorgegebene Randtemperatur (Dirichlet)
- Vorgegebene Wärmestromdichte am Rand (Neumann)

An denjenigen Rändern, an denen die Temperatur vorgegeben ist, ist es günstig, die Gittereinteilung so zu wählen, dass der Rand mit einer Gitterlinie zusammenfällt. So kann man dort die Temperaturen am Rand direkt in Gl. (8) einsetzen.

Ist an einem Rand die Wärmestromdichte vorgegeben, muss dort folgende Bedingung erfüllt sein

$$\dot{q} \cdot \vec{n} = -\lambda (\nabla T \cdot \vec{n}). \quad (9)$$

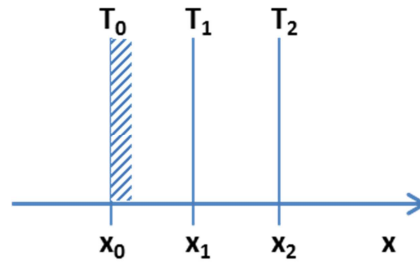


Abbildung 3: Zur Berücksichtigung von Wärmestromdichten am Rand

Nutzt man am linken Rand die einfache Vorwärtsdifferenz, so kann man schreiben

$$-\left(\frac{\partial T}{\partial n}\right)_{x_0}^n = \frac{T_1^n - T_0^n}{\Delta x}. \quad (10)$$

Somit ergibt sich aus den Gln. (9) und (10)

$$T_0^n = T_1^n - \frac{\dot{q} \cdot \Delta x}{\lambda}. \quad (11)$$

In dem vorliegenden Anwendungsfall soll der untere Rand des Eisenquaders als adiabat angenommen werden, sodass $\dot{q} = 0$. Entsprechend ergibt sich aus Gl. (11)

$$T_0^n = T_1^n. \quad (12)$$

3 Implementierung in MATLAB

Laden Sie sich unter Linux die über ILIAS bereitgestellte Übungsdatei herunter und kopieren Sie diese in ein lokales Verzeichnis (z.B. \$HOME/Uebung_1). Navigieren Sie über das Terminal in diesen Ordner und entpacken Sie diese Datei mithilfe des Befehls (vgl. Abbildung 4)

„tar -xzf <DATEINAME>“

```
mvm-tutor-0001@scc-pool-c-16:~$ cd Uebung_1/  
mvm-tutor-0001@scc-pool-c-16:~/Uebung_1$ ls  
Übungsdateien_CFD-Uebung_1.gz  
mvm-tutor-0001@scc-pool-c-16:~/Uebung_1$ tar -xzf Übungsdateien_CFD-Uebung_1.gz
```

Abbildung 4: Entpacken der Übungsdateien mithilfe der Konsole

Nach der erfolgreichen Ausführung dieses Befehls müsste der Ordnerinhalt wie in Abbildung 5 dargestellt aussehen.

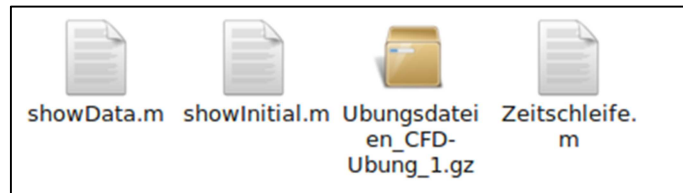


Abbildung 5: Ordnerinhalt nach Entpacken der Übungsdateien

Öffnen Sie nun das Programm MATLAB über das Terminal indem Sie dort den Befehl „matlab“ eingeben und mit „Enter“ bestätigen.

Wählen Sie in MATLAB zunächst mithilfe der Ordner-Navigationsleiste am oberen Fenster-Rand den Pfad aus, unter dem Sie die zur Verfügung gestellten Dateien abgelegt haben.

Nun sehen Sie auf der linken Seite den Inhalt des aktuellen Ordners, rechts den Workspace, in welchem Variablen angezeigt werden. In der Mitte befindet sich die Kommandozeile von MATLAB, mit deren Hilfe man z.B. Funktionen aufrufen kann.

Starten Sie jetzt mit der Implementierung des Programm-Codes zur Lösung des Wärmeleitungsproblems mithilfe des expliziten Finite-Differenzen-Verfahrens.

Alle benötigten Befehle sind im Anhang kurz zusammengefasst.

3.1 Definition der physikalischen Parameter

Öffnen Sie ein neues Skript (New | Skript).

Tragen Sie hier die globalen Parameter ein, die zur Berechnung des Wärmeleitungsproblems benötigt werden (vgl. Abbildung 6). Bei Voranstellen des %-Zeichens erkennt MATLAB das Nachfolgende dieser Zeile als Kommentar, welches durch die grüne Farbe gekennzeichnet ist.

Jede Zeile sollte mit einem Semikolon abgeschlossen werden, um die spätere Ausgabe über die Kommandozeile übersichtlich zu halten.

Speichern Sie das Parameter-Skript unter dem Arbeits-Pfad als „Parameter.m“ ab.

```

% Abmessungen des Querschnitts
xMin = 0; %m
xMax = 0.1; %m
yMin = 0; %m
yMax = 0.1; %m

% Stoffeigenschaften von Eisen
rho = 7874; %kg/m³
cp = 449; %J/(kg*K)
lambda = 80; %W/(m*K)

% Definition des Rechengitters
numX = 101; % Anzahl der Einteilungen in x-Richtung
numY = 101; % Anzahl der Einteilungen in y-Richtung

% Definition der Anfangsbedingung
T0 = 1000; %°C

% Definition der Randbedingung
T_A = 100; %°C

% Definition der Zeitschrittweite
dt = 0.01; %s

% Definition des Startzeitpunktes
t0 = 0; %s

% Definition des Endzeitpunktes
t_end = 30; %s

```

Abbildung 6: Definition der physikalischen Parameter

3.2 Diskretisierung und Initialisierung des Rechengebietes

Für die numerische Lösung muss das Rechengebiet in diskrete Kompartimente unterteilt werden. Dies erfolgt im Folgenden mithilfe von Matrizen.

Wie aus Gl. (8) hervorgeht, benötigen Sie zur Berechnung der neuen Temperatur die Informationen über das lokale alte Temperaturfeld. Aus diesem Grund bietet es sich für die spätere Rechnung an, mit zwei Rechengebieten identischer Größe zu rechnen. Für eine bessere Übersichtlichkeit wird zunächst das Anfangs-Temperaturfeld entwickelt, welches dann auf die beiden Arbeits-Temperaturfelder übertragen wird (vgl. Abbildung 7).

Öffnen Sie, wie in Kap. 3.1 beschrieben, ein neues Matlab-Skript.

```

% Initialisierung des Rechengebietes

% Definieren der Anfangsbedingung
T_t0 = T0*ones(numX,numY);

% Inkludieren der Randbedingung
T_t0(:,1) = T_A;
T_t0(:,end) = T_A;
T_t0(1,:) = T_A;
T_t0(end,:) = T0;

% Initialisieren des Rechengebietes für die Zeitschritte n und n+1
T_n = T_t0;
T_np1 = T_t0;

```

Abbildung 7: Diskretisierung und Initialisierung des Rechengebietes

Zu Beginn hat der quadratische Stabquerschnitt, der in diskreter Form aus $\text{numX} \cdot \text{numY}$ Elementen besteht, im Inneren die konstante Temperatur von T_0 (vgl.

Abbildung 7). Die jeweils erste und letzte Zeile bzw. erste und letzte Spalte enthalten die Größen für die Randbedingung des Problems und werden entsprechend gesetzt. Hierbei ist zu beachten, dass eine Matrix in MATLAB jeweils in der oberen linken Ecke mit dem Index 1 beginnt. Schließlich wird das Anfangs-Temperaturfeld auf die beiden Arbeits-Temperaturfelder T_n und T_{np1} übertragen.

Speichern Sie das erstellte Skript unter dem Arbeits-Pfad als „Initialisierung.m“ ab. Zur besseren Übersichtlichkeit öffnen Sie nun ein weiteres, neues Matlab-Skript, in welchem die zur numerischen Lösung benötigten Parameter bestimmt werden (vgl. Abbildung 8). Speichern Sie dieses Skript unter dem Namen „num_Parameter.m“ ab.

```
% Bestimmen der numerischen Parameter

% Gitterweite in x-Richtung
dx = (xMax-xMin)/(numX-1);
dy = (yMax-yMin)/(numY-1);

% Temperaturleitfähigkeit
a = lambda/(rho*cp);

% Berechnung der Anzahl der Zeitschritte
num_Zeitschritte = (t_end - t0)/dt;

% Darstellen der Anfangsbedingung
showInitial
```

Abbildung 8: Berechnung der für die Numerik benötigten Parameter

Hier wurde ein Skript mit dem Namen „showInitial“ eingebunden, welches das Temperaturfeld zu Beginn visualisiert. Diese Datei befindet sich unter den bereits heruntergeladenen Übungsdateien.

3.3 Anzeigen der Anfangsbedingung

Lassen Sie sich vor der eigentlichen Lösung des Wärmeleitungsproblems die Anfangsbedingung anzeigen.

Schreiben Sie dazu ein neues Skript mit dem Namen „Global.m“, in welchem Sie die bisher angefertigten Skripte einbinden (vgl. Abbildung 9).

```
% Einlesen der Parameter
Parameter

% Initialisierung des Rechengebietes
Initialisierung

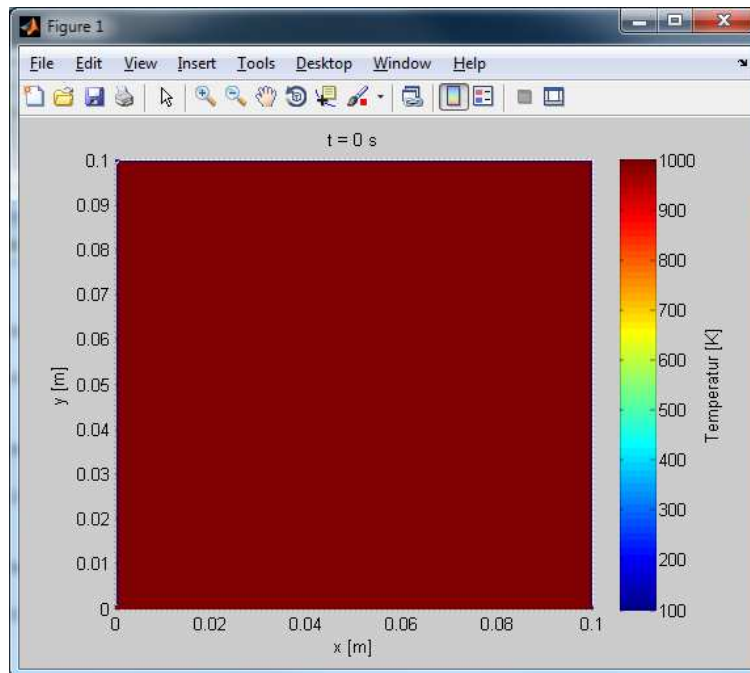
% Berechnung der numerischen Parameter
num_Parameter
```

Abbildung 9: Skript für die Anzeige der Anfangsbedingung

Führen Sie dieses Skript aus, indem Sie entweder den „Run“-Button an der Oberseite des Fensters klicken, oder in der Kommandozeile „Global“ eintippen und mit Enter bestätigen (vgl. Abbildung 10).

**Abbildung 10: Ausführen des Skripts**

Es sollte nun ein Fenster erscheinen, in welchem die Anfangsbedingung visualisiert ist (vgl. Abbildung 11).

**Abbildung 11: Darstellung der Anfangsbedingung**

3.4 Programmieren der Zeitschleife

Beginnen Sie nun mit dem Herzstück des Programmes, welches die numerische Lösung des Wärmeleitungsproblems vollführt.

Hierbei bietet es sich an mit sogenannten for-Schleifen zu arbeiten. Wie in Abbildung 12 dargestellt, wird dabei eine Laufvariable definiert (hier i), mit einem Startwert initialisiert (hier 1) und bei jedem Schleifendurchlauf um eins inkrementiert. Dies geschieht so lange, bis das Abbruchkriterium (hier 10) erfüllt ist. Jede for-Schleife wird mit dem end-Befehl beendet.

```
1      % Beispiel für eine for-Schleife
2
3      a=0;
4
5      for i=1:10
6
7          a = a+i;
8
9      end
```

Abbildung 12: Beispiel für eine for-Schleife in MATLAB

Öffnen Sie nun das von den Übungsleitern zur Verfügung gestellte Skript mit dem Namen „Zeitschleife.m“.

Versuchen Sie den Quellcode für die Zeitschleife zur Lösung des Wärmeleitungsproblems selbst nachzuvollziehen (vgl. Abbildung 13). Bei Fragen wenden Sie sich an die Übungsleiter.

Finden Sie heraus welcher Teil hier vergessen wurde und programmieren Sie diesen fehlenden Teil selbständig hinzu.

```
% Zeitschleife
for t=1:num_Zeitschritte

    % Durchlaufen aller Punkte in Y-Richtung
    for j=2:(numY-1)

        % gleichzeitig Durchlaufen aller Punkte in X-Richtung
        for i=2:(numX-1)

            end

        end

    end

    % Kopieren der Lösung für den Zeitschritt n+1 als Startbedingung für
    % den neuen Zeitschritt

    T_n = T_np1;

    % Gesonderte Behandlung der Neumann-Randbedingung am unteren Rand
    T_n(end,:) = T_np1(end-1,:);

    % Darstellen der neuen Lösung
    showData;

    mittelTemperatur = mean(mean(T_np1));
    disp(['Die mittlere Temperatur zum Zeitpunkt t = ' ...
        num2str(t*dt) ' s beträgt ' num2str(mittelTemperatur) ' °C']);

end
```

Abbildung 13: Zeitschleife

Erweitern Sie nun die Datei „Global.m“, indem Sie in dieser Datei das Skript „Zeitschleife.m“ einbinden (vgl. Abbildung 14) und starten Sie die Simulation.

```
% Einlesen der Parameter
Parameter

% Initialisierung des Rechengebietes
Initialisierung

% Berechnung der numerischen Parameter
num_Parameter

% Starten der Zeitschleife
Zeitschleife
```

Abbildung 14: Vervollständigung der Global-Datei

Nach einer Simulationszeit von 2 Sekunden sollte das Temperaturfeld der Darstellung in Abbildung 15 entsprechen.

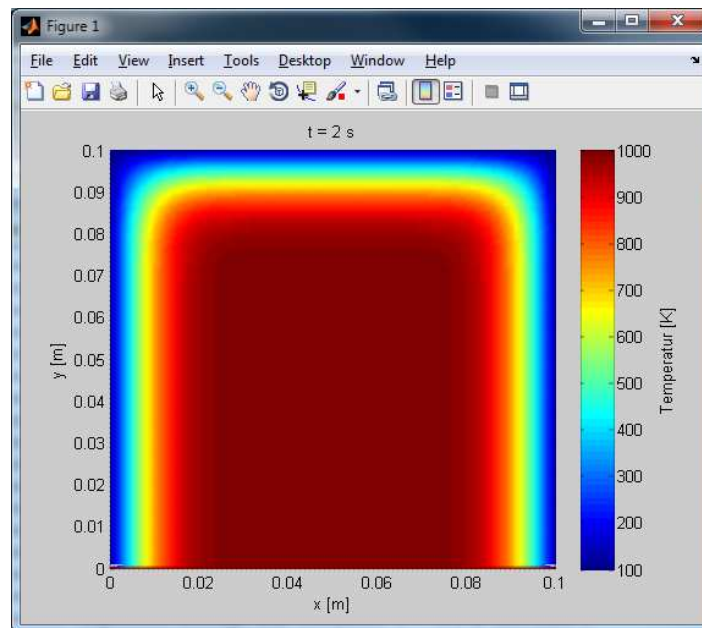


Abbildung 15: Visualisierung des Temperaturfeldes nach 2 Sekunden

4 Aufgabenstellung

Nachdem Sie nun das Wärmeleitungsproblem numerisch gelöst haben, möchte Ihr Vorgesetzter von Ihnen wissen, wie groß unter den gegebenen Bedingungen die minimal notwendige Abkühlzeit der Eisenstangen ist.

Wie lautet Ihre Antwort?

Ihr Vorgesetzter ist von Ihrem Ergebnis sehr beeindruckt und möchte die minimale Abkühlzeit für in einer anderen Abteilung gefertigte Aluminiumstangen identischer Form, Größe und Herstellungsbedingungen wissen. Die Stoffparameter für Aluminium sind in Tabelle 2 zusammengefasst.

Tabelle 2: Stoffparameter für Aluminium

spezifische Dichte ρ_{Al}	2700 kg/m ³
spezifische Wärmekapazität $c_{p,\text{Al}}$	897 J/(kg K)
Wärmeleitfähigkeit λ_{Al}	235 W/(m K)

Was sagen Sie ihm?

5 Anhang: einige MATLAB-Befehle

Befehl	Bedeutung
ones(x,y)	Erzeugung einer Matrix mit x Zeilen und y Spalten, wobei alle Matrixelemente eine 1 enthalten.
clear	Alle Variablen löschen
Tastenkombination (STRG+C)	Abbrechen der Simulation
Matrixoperationen:	
a(1,1)	Gibt den Wert der ersten Zeile und der ersten Spalte in der Matrix a aus.
a(1,:)	Gibt alle Werte der ersten Zeile der Matrix a aus.
a(end,1)	Gibt den Wert der Matrix a aus, der in der letzten Zeile und der ersten Spalte gespeichert ist.