

BACHELOR

Instationäre Wärmeleitung in dickwandigen Komponenten konventioneller Kraftwerke

Autor:

Daniel Kollmeier

Matrikel-No:

03663836

Betreuer:

Simon van Buren, M. Sc.
Prof. Wolfgang Polifke, Ph. D.
Dipl.-Ing. Franz Binder

7. Mai 2018

Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig verfasst zu haben. Ich habe keine anderen Quellen und Hilfsmittel als die angegebenen verwendet.

Ort, Datum

Daniel Kollmeier

Kurzfassung

In konventionell befeuerten Kraftwerken sind deren Komponenten durch veränderte Betriebszustände vermehrt instationären Vorgängen ausgesetzt. Der Temperaturunterschied zwischen Innen- und Außenoberfläche ruft dabei eine mechanische Schädigung hervor, die bei der Ermüdungsüberwachung berücksichtigt werden muss. Für eine Berechnung der Werkstofferschöpfung ist es notwendig die Innentemperatur präzise zu ermitteln. Die Berechnung des instationären Innentemperaturverlaufs aus aufgezeichneten Außentemperaturen einer Rohrleitung führt zu einer eindimensionalen, transienten Problemstellung der inversen Wärmeleitung. Zu deren Lösung steht sowohl ein analytisches Verfahren mit einer Übertragungsfunktion als auch mehrere numerische Methoden, wie zum Beispiel die Finite-Differenzen-Methode, zur Verfügung.

In dieser Arbeit werden diese Berechnungsarten miteinander verglichen, um ein Verfahren mit einfacher Implementierung und geringer Rechenzeit für die ausgedehnte Überwachung von Rohrleitungsnetzen zu ermitteln. Dazu werden beispielhafte, transiente Innentemperaturverläufe als Testproblem definiert und die resultierenden Außentemperaturverläufe werden mit dem MATLAB `pdepe`-Löser berechnet. Aus diesem Datenmaterial wird mit den betrachteten Lösungsverfahren der Innentemperaturverlauf rekonstruiert, die Ergebnisse werden auf Rechenaufwand, Stabilität und die Anforderungen an einen initialen Filterprozess geprüft.

Es zeigt sich, dass die analytische Lösungsmethode nur in einigen besonderen Fällen anwendbar ist. Unter den Finiten-Differenzen-Verfahren zeigt insbesondere die Methode von Hills & Hensel stabile Ergebnisse hinsichtlich der getesteten Kriterien.

Inhaltsverzeichnis

| | |
|---|-----------|
| Abkürzungsverzeichnis | vi |
| 1 Einleitung | 1 |
| 2 Grundlagen | 3 |
| 2.1 Beanspruchung von Rohrleitungen durch Temperaturtransienten | 3 |
| 2.2 Herleitung der Wärmeleitungsgleichung | 4 |
| 2.3 Inverse Problemstellungen | 8 |
| 2.4 Analytische Lösungsmethode mit einer Übertragungsfunktion | 10 |
| 2.5 Numerische Lösungsmethoden | 13 |
| 2.5.1 Numerische Differentiation | 13 |
| 2.5.2 Finite-Differenzen-Methode (FDM) | 17 |
| 3 Anwendung | 29 |
| 3.1 Modellbildung | 29 |
| 3.2 Eigenschaften der Übertragungsfunktion | 32 |
| 3.3 Definition der Testfälle | 33 |
| 4 Numerische Ergebnisse | 35 |
| 5 Fazit | 43 |
| Anlagen | 44 |
| A Lagrange-Interpolation | 45 |
| B Messtechnik | 47 |
| C MATLAB-Codes | 54 |
| Literaturverzeichnis | 69 |

Abkürzungsverzeichnis

Lateinische Buchstaben

| | | |
|-----------|---|-------------------|
| \dot{Q} | Wärmestrom | [W] |
| \dot{q} | Wärmestromdichte, Wärmefluss | $[\frac{W}{m^2}]$ |
| A | Fläche | $[m^2]$ |
| a | Temperaturleitfähigkeit | $[\frac{m^2}{s}]$ |
| c | spezifische Wärmekapazität | $[\frac{J}{kgK}]$ |
| D | Außendurchmesser | [m] |
| E | Energie | [J] |
| h | Schrittweite der finiten Differenzen | [m] |
| i | zeitlicher Laufindex | [–] |
| I_i | modifizierte Besselfunktion 1.Art i .Ordnung | [–] |
| j | örtlicher Laufindex | [–] |
| k | Lösungsvariable | [–] |
| K_i | modifizierte Besselfunktion 2.Art i .Ordnung | [–] |
| l | Lösungsvariable | [–] |
| M | Anzahl der Zeitschritte | [–] |
| m | Anzahl der gewichteten Zeitschritte nach Hills & Hensel | [–] |
| m | Masse | [kg] |
| N | Anzahl der Ortsschritte | [–] |
| n | Laufindex | [–] |

| | | |
|---------|-----------------------------|---------------|
| Q | Wärme | $[J]$ |
| r | Radius | $[m]$ |
| S | Periodendauer | $[s]$ |
| s | Wandstärke | $[m]$ |
| T | Temperatur | $[^{\circ}C]$ |
| t | Zeit | $[s]$ |
| T_j^i | Temperatur am Knoten i, j | $[^{\circ}C]$ |
| U | innere Energie | $[J]$ |

Griechische Buchstaben

| | | |
|----------------|---|------------------------------|
| Δ | Differenz | $[-]$ |
| $\dot{\omega}$ | Wärmequellendichte | $[\frac{W}{m^3}]$ |
| ϵ | relativer Fehler der Berechnungsverfahren | $[-]$ |
| λ | Wärmeleitfähigkeit | $[\frac{W}{mK}]$ |
| ω | Kreisfrequenz | $[\frac{1}{s}]$ |
| σ | Stabilitätsparameter für Weber-Schema | $[s]$ |
| ρ | Dichte | $[\frac{kg}{m^3}]$ |
| ϑ | Wärmefluss-Temperatur-Vektor | $[\frac{W}{m^2}, ^{\circ}C]$ |
| ξ | Element des Intervalls $[x_0, x_{0+h}]$ | $[m]$ |

Indizes

| | |
|----------|-----------------------------|
| 0 | Bedingung am Anfang |
| ∞ | Eigenschaften der Umgebung |
| a | Bedingung an der Außenseite |
| f | Fluideigenschaften |
| i | Bedingung an der Innenseite |
| iso | Bezug auf Isolationsschicht |
| max | Maximum |

| | |
|-------------|-------------------------------|
| <i>rel</i> | relative Größe |
| <i>Rohr</i> | Bezug auf Rohrleitung |
| <i>S</i> | bezogen auf Souza-Algorithmus |
| <i>W</i> | bezogen auf Weber-Algorithmus |
| Trans | bezogen auf Transiente |

Abkürzungen

| | |
|------|---------------------------------|
| DGL | Differentialgleichung |
| FDM | Finite Differenzen Methode |
| IHCP | Inverse Heat Conduction Problem |

Überlagerung

| | |
|---|-----------------|
| - | komplexe Größe |
| ^ | spektrale Größe |

1 Einleitung

Die Untersuchung von Wärmeleitungsproblemen in dickwandigen Komponenten konventioneller Kraftwerke, z.B. Kesseln, Rohrleitungen oder Ventilen, gehört seit Bestehen der elektrischen Stromversorgung zu den Aufgaben von Ingenieuren in der Energietechnik und im Anlagenbau. Nicht zuletzt durch die Überwachung des Ermüdungszustands sicherheitskritischer Bauteile und die Einhaltung entsprechender Revisionsintervalle kann die Versorgungssicherheit der Gesellschaft mit elektrischem Strom garantiert werden.

In den letzten Jahren ist der durch die „Energiewende“ hervorgerufene Umbruch in der Kraftwerkslandschaft deutlich zu spüren. Das erklärte Ziel, einen Großteil des Energiebedarfs aus regenerativen Energiequellen zu decken, hat einen Einspeisevorrang von sogenanntem „Öko-Strom“ in das Stromnetz zur Folge. Damit wird den bestehenden, konventionell befeuerten Kraftwerken eine erhöhte Flexibilität abverlangt. Deren Betrieb verschiebt sich damit hin zu einem Mindestlastbetrieb mit dynamischen Anteilen an Mittel- und Spitzenlastbetrieb. Bei jedem Wechsel des Betriebszustands stellt sich insbesondere in dickwandigen Komponenten der Kraftwerkstechnik eine heterogene Temperaturverteilung ein, die im Bauteil Wärmespannungen hervorruft.

Mit zunehmendem Wechselbetrieb des Kraftwerks stellt dies eine nicht mehr zu vernachlässigende Belastung der betreffenden Bauteile dar. Bei der ursprünglichen Planung der meisten konventionellen Kraftwerke wurde ein Betrieb, wie er heute häufig erfolgt, vielfach noch nicht berücksichtigt. Somit entsteht ein nachträglicher Nachweisdruck für die Betreiber von konventionellen Kraftwerken.

Für eine exakte Ermüdungsbewertung ist es notwendig die Temperatur an der Rohrinneenseite genau zu kennen. Nur so kann eine zeitliche Temperaturverteilung in der Bauteilwand bestimmt werden, wie sie für die Anwendung technischer Regelwerke, z.B. DIN EN 12952-3 oder TRD 301/303, erforderlich ist.

Die direkte Messung der Temperatur an der Innenseite stellt sich als problematisch heraus. Dazu muss ein Messelement in einer Bohrung bis dicht an die Innenoberfläche geführt werden. Dies bedeutet eine signifikante Schwächung des Bauteilquerschnitts und ist deshalb nur in Ausnahmefällen in Betracht zu ziehen. Ebenso ist das Einbringen einer Temperatur- und Geschwindigkeitsmesseinrichtung in das strömende Fluid, mit deren Hilfe der Wärmeübergangskoeffizient zwischen Fluid und Innenoberfläche berechnet werden kann, nicht für die ausgedehnte Überwachung ganzer Rohrleitungsnetze geeignet.

Für die Messung der Rohraußentemperatur stehen verschiedene Messinstallationen (siehe Anlage B) zur Verfügung, genauso wie entsprechende Erfahrungen zu den zu erwartenden Messergebnissen (siehe [10]). Daher ist es einleuchtend, dass man versucht von der gemessenen Außentemperatur auf die ursächliche Innentemperatur zurückzuschließen.

Dies führt allerdings zu einem inversen Problem der Wärmeleitung. Anders als im Fall des direkten Wärmeleitungsproblems, bei dem die Temperaturverteilung innerhalb eines Körpers aus den gegebenen Werten für Temperatur und Wärmefluss an seinem Rand berechnet wird, ist beim inversen Problem die Temperaturverteilung innerhalb des Bauteils gegeben und gesucht sind die zugrunde liegenden Randbedingungen.

Die ersten inversen Probleme der Wärmeleitung haben bereits Fourier¹, Poisson² und Kelvin³ formuliert. Die von ihnen aufgestellten Probleme hatten aber keine größere Bedeutung, denn solche Probleme galten aus Sicht der reinen Mathematik lange Zeit als nicht lösbar. Erst in den 1950er Jahren mit Beginn der Raumfahrtprogramme gewann das inverse Wärmeleitungsproblem (Inverse Heat Conduction Problem, IHCP) an Wichtigkeit. Eine beispielhafte Anwendung in dieser Zeit war die Auslegung von Hitzeschilden für den Wiedereintritt von Raumkapseln in die Erdatmosphäre.

Der Schwerpunkt dieser Arbeit ist es, aus den verschiedenen Verfahren, die zur Lösung des IHCP zur Verfügung stehen, eines auszuwählen, welches sich mit geringem Aufwand unter Zuhilfenahme von Standard-Software auf einem Computer implementieren lässt. Da Messwerte der Außentemperatur im Allgemeinen über lange Zeiträume zur Verfügung stehen, soll es möglich sein Teilintervalle auszuwerten. Der Algorithmus muss außerdem trotz Messfehlern zuverlässige, stabile Ergebnisse liefern.

Dazu wird im Kapitel 2 zunächst auf die Bauteilbelastung durch instationäre Vorgänge, die Herleitung der Wärmeleitungsgleichung sowie Grundlagen zu inversen Problemen eingegangen. Außerdem werden eine analytische Lösung des IHCP für periodische Eingangsdaten mittels einer Übertragungsfunktion und numerische Lösungen über Finite-Differenzen-Methoden (FDM) vorgestellt. Nachdem einige Finite-Differenzen-Verfahren näher erklärt worden sind, folgt im 3. Kapitel die Anwendung auf das System einer adiabaten Rohrleitung mit der Modellbildung, der Untersuchung der in Kapitel 2 hergeleiteten Übertragungsfunktion und der Vorstellung der exemplarischen Testprobleme. Nachdem im 4. Kapitel die Ergebnisse der numerischen Berechnung diskutiert worden sind, folgt in Kapitel 5 die Empfehlung eines geeigneten Verfahrens.

¹Jean Baptiste Joseph Fourier, 1768 - 1830; frz. Physiker und Mathematiker

²Siméon Denis Poisson, 1781 - 1840; frz. Mathematiker und Physiker

³William Thomson, 1. Baron Kelvin, 1824 - 1907; brit. Physiker

2 Grundlagen

Im Folgenden werden die grundlegenden Werkzeuge für die Lösung eines inversen Wärmeleitungsproblems zur Verfügung gestellt. Dazu wird zunächst der konkrete Belastungsfall beschrieben. Bevor auf die Besonderheiten inverser Probleme eingegangen wird, ist die Herleitung der Wärmeleitungsgleichung eingeschoben. Im Anschluss werden geeignete Strategien zur Lösung des IHCP vorgestellt.

2.1 Beanspruchung von Rohrleitungen durch Temperaturtransienten

Ändert sich der Lastbetrieb eines konventionellen Kraftwerks, beispielsweise durch Anfahr- oder Abschaltvorgänge, so stellt sich am Innenradius einer Rohrleitung eine Veränderung der Temperatur ein. Dies regt einen thermischen Ausgleichsvorgang in radialer Richtung an, der mit dem Erreichen eines neuen stationären Vorgangs abgeschlossen ist. Das Intervall zwischen zwei stationären Zuständen bezeichnet man im Fall von rotationssymmetrischen Temperaturprofilen auch als „Kolbenströmungstransiente“. Die maßgebliche Schädigung der Rohrleitung entsteht durch die auftretenden, in Abbildung 2.1 dargestellten, radialen Temperaturunterschiede [10].

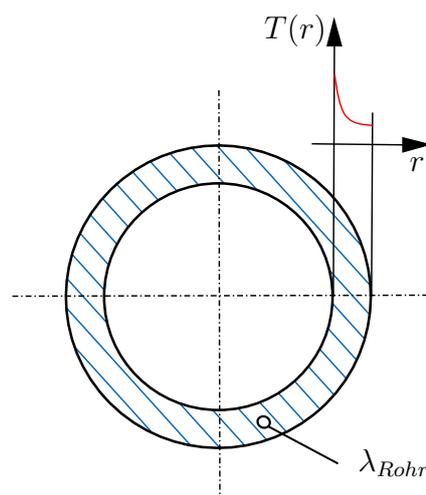


Abbildung 2.1: Eindimensionale, stationäre Temperaturverteilung über den Radius einer Rohrwand bei Kolbenströmungen, nach [10].

Es treten sowohl Kolbenströmungstransienten mit „warmem“ Ausgangszustand und „kaltem“ Endzustand auf als auch umgekehrt (Anfangszustand „kalt“ und Endzustand „warm“). Einer besonderen spannungs- und ermüdungsmäßigen Belastung sind Rohrleitungen ausgesetzt, bei denen transiente Vorgänge beider Art vorkommen [10].

2.2 Herleitung der Wärmeleitungsgleichung

Da jeder Temperaturverlauf eines Wärmeleitungsproblems eine Lösung der Wärmeleitungsgleichung sein muss, wird diese im Folgenden näher betrachtet. Dazu wird zunächst das Fourier'sche Gesetz der Wärmeleitung beleuchtet und danach die Fourier'sche Differentialgleichung über eine Energiebilanz am differentiellen Element hergeleitet.

Fourier'sches Gesetz der Wärmeleitung

Einige Untersuchungen zur Ausbreitung von Wärme hat bereits Fourier vorgenommen. Werden einem Körper unterschiedliche Oberflächentemperaturen $T_i > T_a$ aufgeprägt, so fließt innerhalb eines definierten Zeitintervalls Δt eine Wärmemenge Q . Dabei hängt Q folgendermaßen von den unten aufgeführten Größen ab:

- proportional von der Temperaturdifferenz $\Delta T = T_i - T_a$
- proportional von der isothermen Oberfläche A
- proportional vom Zeitintervall Δt
- invers proportional von der Dicke des Körpers Δx .

Zusätzlich lässt sich eine starke Abhängigkeit von den Materialeigenschaften des Körpers feststellen [16].

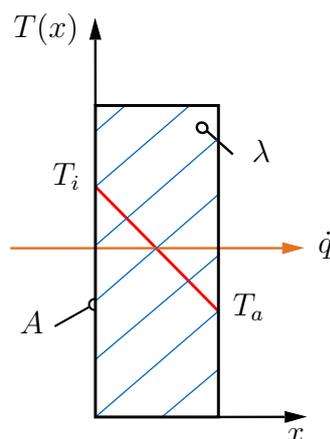


Abbildung 2.2: Stationäre Wärmeleitung in einer umfangsseitig isolierten Platte, nach [11, 16].

Formuliert man diese Zusammenhänge analytisch aus, so erhält man folgende Formel:

$$Q = \lambda \frac{\Delta T}{\Delta x} A \Delta t \quad (2.1)$$

Die Materialabhängigkeit wird mit der spezifischen Wärmeleitfähigkeit λ berücksichtigt. Diese Stoffgröße ist im Allgemeinen eine Funktion der Temperatur, die sich bei Metallen und Gasen im Bereich von $T = 100\text{ K}$ bis $T = 1000\text{ K}$ oft um mehr als eine Größenordnung ändert. Weil derartig große Temperaturunterschiede selten innerhalb eines Bauteils auftreten, reicht es gemäß Incropera und DeWitt [11] in guter Näherung mit einer als konstant angenommenen Wärmeleitfähigkeit zu rechnen [16].

Für den Wärmestrom \dot{Q} , als die in einer Zeiteinheit Δt übertragene Wärmemenge Q definiert, gilt:

$$\dot{Q} \equiv \frac{dQ}{dt} = \lambda \frac{\Delta T}{\Delta x} A \quad (2.2)$$

Wird die Wärmemenge pro Zeit- und Flächeneinheit übertragen, so spricht man auch von der Wärmestromdichte oder Wärmefluss.

$$\dot{q} = \lambda \frac{\Delta T}{\Delta x} \quad (2.3)$$

Damit auch bei komplizierteren Geometrien die Richtungsabhängigkeit der Wärmeleitung eindeutig beschrieben ist, wird der Wärmefluss \dot{q}_x in x -Richtung durch eine (infinitesimal kleine) Fläche dA als

$$\dot{q}_x \equiv \frac{d\dot{Q}_x}{dA} \quad (2.4)$$

definiert. Für die anderen Koordinatenrichtungen gilt diese Beziehung analog. Beim Bilden des Grenzübergangs von $\Delta x \rightarrow 0$ erhält man

$$\dot{q}_x = \lambda \lim_{\Delta x \rightarrow 0} \frac{T(x) - T(x + \Delta x)}{\Delta x} = -\lambda \frac{\partial T}{\partial x}. \quad (2.5)$$

Dies ist bekannt als das Fourier'sche Gesetz, welches den Transport der Wärme in Abhängigkeit des Temperaturgradienten innerhalb des Körpers quantifiziert. Das negative Vorzeichen ist notwendig um zu verdeutlichen, dass die Richtung des Wärmeflusses, wie in Abbildung 2.2 gezeigt, dem Gradienten entgegengesetzt ist [16]. In vektorieller Form lautet das Fourier'sche Gesetz:

$$\underline{\dot{q}}(\underline{x}) = -\lambda \nabla T(\underline{x}) \quad (2.6)$$

Was zunächst als phänomenologischer Ansatz von Fourier formuliert wurde, hat sich als nahezu ausnahmslos geltendes Gesetz erwiesen [16].

Energiebilanz am differentiellen Volumenelement

Mit Hilfe des Prinzips der Energieerhaltung ist es Fourier gelungen eine Differentialgleichung herzuleiten, die die räumlichen und zeitlichen Temperaturveränderungen innerhalb eines Körpers durch Wärmetransport beschreibt [16].

In Abbildung 2.3 wird gezeigt, welche Wärmeflüsse bei der Bilanz der thermischen Energie an einem differentiellen Element dV in Zylinderkoordinaten (r, φ, z) berücksichtigt werden müssen.

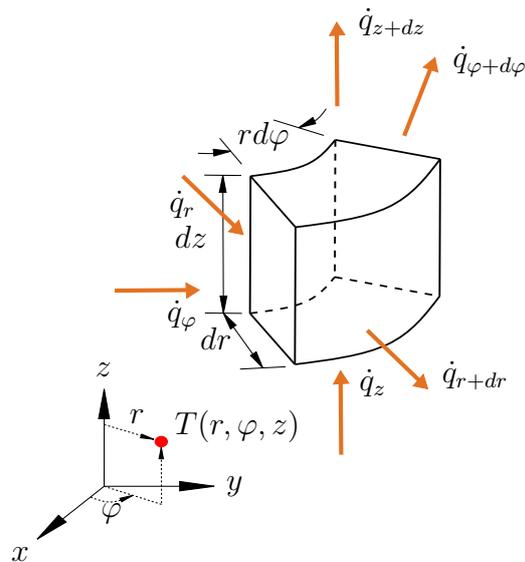


Abbildung 2.3: Wärmebilanz für ein infinitesimales, zylindrisches Volumenelement ohne Zu- und Abfluss von Masse, nach [11].

Aufgrund von Energieerhaltung muss eine Zufuhr von Wärme dQ zu einer Erhöhung der inneren Energie dU führen.

$$\Delta U = \Delta Q \quad (2.7)$$

Wobei die innere Energie für eine Masse dm mit Hilfe des Produkts aus spezifischer Wärmekapazität c , der Masse m und der Temperaturänderung ΔT berechnet werden kann.

$$\Delta U = c\Delta T dm \quad (2.8)$$

Ist die Dichte ρ ebenfalls konstant und temperaturunabhängig über das Volumen, verhält sich die Änderung der inneren Energie gemäß

$$\Delta U = \rho c \Delta T dV \quad (2.9)$$

Dieser Term wird auch als Speicherterm bezeichnet, da er die Einspeicherung innerer Energie im Kontrollvolumen beschreibt [16].

Durch Wärmeleitung kann Energie über den Rand ∂V des betrachteten Systems übertragen werden oder direkt im Kontrollvolumen, z.B. durch elektrische, chemische oder nukleare Effekte, mit einer Wärmequellendichte $\dot{\omega}$ freigesetzt werden. Wird durch den Term $\dot{\omega}dV$ Wärme entzogen, so spricht man von einer Wärmesenke [16].

$$\Delta Q = \left(\dot{Q}_{\partial V} + \dot{\omega}dV \right) \Delta t \quad (2.10)$$

Bilanziert man nun den Wärmeaustausch über den Rand ∂V entlang der Koordinate r , so muss der Wärmefluss \dot{q}_r mit den Seitenflächen des differentiellen Volumens dV multipliziert werden.

$$\dot{Q}_{\partial V,r} = \dot{q}_r|_r r d\varphi dz - \dot{q}_r|_{r+dr} (r + dr) d\varphi dz \quad (2.11)$$

Für die anderen beiden Raumrichtungen können analoge Beziehungen aufgestellt werden. Der Index $\dots|_r$ bezeichnet hier die Seitenfläche des Volumens an der Stelle r . Wie in der Thermodynamik im Allgemeinen üblich bezeichnet ein positiver Wert für $\dot{Q}_{\partial V,r}$ eine Wärmezufuhr [16].

Entwickelt man den Wärmefluss $\dot{q}_r|_{r+dr}$ in eine Taylor-Reihe in r -Richtung, so erhält man

$$\dot{q}_r|_{r+dr} = \dot{q}_r|_r + \frac{\partial \dot{q}_r}{\partial r} \Big|_r dr + \frac{\partial^2 \dot{q}_r}{\partial r^2} \Big|_r \frac{dr^2}{2} + \dots \quad (2.12)$$

Vernachlässigt man alle Terme ab der zweiten Ordnung in dr ergibt sich für die Differenz aus Zu- und Abfluss

$$\dot{Q}_{\partial V,r} = \dot{q}_r|_r r d\varphi dz - \dot{q}_r|_{r+dr} (r + dr) d\varphi dz = - \frac{\partial \dot{q}_r}{\partial r} \Big|_r r d\varphi dz \quad (2.13)$$

Für die φ - und z -Komponente gilt dieser Zusammenhang analog [16].

Trägt man alle Ergebnisse der Wärmeströme in Gleichung (2.7) zusammen und dividiert durch das differentielle Volumen dV , so erhält man für ein infinitesimales Zeitintervall ∂t

$$\rho c \frac{\partial T}{\partial t} = - \frac{\partial \dot{q}_r}{\partial r} - \frac{\partial \dot{q}_\varphi}{\partial \varphi} - \frac{\partial \dot{q}_z}{\partial z} + \dot{\omega}. \quad (2.14)$$

Insgesamt bilanziert Gleichung (2.14) den Wärmetransport über die Grenzen des Kontrollvolumens zusammen mit der Einspeicherung und der Freisetzung von Energie durch die Quellendichte $\dot{\omega}$ [16].

Drückt man nun die Wärmeströme innerhalb von Gleichung (2.14) durch das Wärmetransportgesetz von Fourier (2.6) aus,

$$\dot{q}_r = -\lambda \frac{\partial T}{\partial r} \quad (2.15)$$

$$\dot{q}_\varphi = \frac{-\lambda}{r} \frac{\partial T}{\partial \varphi} \quad (2.16)$$

$$\dot{q}_z = -\lambda \frac{\partial T}{\partial z} \quad (2.17)$$

erhält man folgende partielle Differentialgleichung für das Temperaturfeld $T(r, \varphi, z, t)$:

$$\rho c \frac{\partial T}{\partial t} = \frac{1}{r} \frac{\partial}{\partial r} \left(\lambda r \frac{\partial T}{\partial r} \right) + \frac{1}{r^2} \frac{\partial}{\partial \varphi} \left(\lambda \frac{\partial T}{\partial \varphi} \right) + \frac{\partial}{\partial z} \left(\lambda \frac{\partial T}{\partial z} \right) + \dot{w}. \quad (2.18)$$

Dies ist die allgemeine Form der Fourier'schen Differentialgleichung (DGL) für isotrope Medien mit zylindrischer Geometrie, konstanter Dichte ρ und konstanter Wärmekapazität c [16]. Kann man außerdem auch für die Wärmeleitfähigkeit λ Orts- und Temperaturunabhängigkeit voraussetzen, vereinfacht sich Gleichung (2.18) folgendermaßen:

$$\frac{1}{a} \frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} + \frac{1}{r^2} \frac{\partial^2 T}{\partial \varphi^2} + \frac{\partial^2 T}{\partial z^2} + \frac{\dot{w}}{\lambda}. \quad (2.19)$$

Wobei die Temperaturleitfähigkeit a eine zusammengesetzte Stoffgröße ist, die die Wärmeleitfähigkeit λ mit dem Speichervermögen ρc ins Verhältnis setzt.

$$a \equiv \frac{\lambda}{\rho c} \quad (2.20)$$

Sie gibt Auskunft darüber wie schnell Inhomogenitäten des Temperaturfeldes im Material übertragen werden [11, 16].

2.3 Inverse Problemstellungen

In den Natur- und Ingenieurwissenschaften spricht man von einem **inversen Problem** wenn

- eine Abbildung $F : \mathbb{U} \rightarrow \mathbb{W}$ gegeben ist, die eine Kausalität zwischen einer Ursache $u \in \mathbb{U}$ und einer Wirkung $F(u) \in \mathbb{W}$ beschreibt und
- die Aufgabe besteht, von der Wirkung $w \in \mathbb{W}$ auf die Ursache $u \in \mathbb{U}$ mit $F(u) = w$ zurückzuschließen.

Umgekehrt bezeichnet man die Berechnung von $w = F(u)$ bei gegebenem $u \in \mathbb{U}$ als **direktes Problem** [17].

Die im Folgenden behandelte Berechnung der Innentemperatur eines Rohrs aus der bekannten Außentemperatur stellt somit ein inverses Problem dar.

Eigenschaften inverser Probleme

Inverse Probleme, wie beispielsweise das Inverse Wärmeleitungsproblem (IHCP), sind im Allgemeinen schwierig zu berechnen, weil die Lösung sehr stark von den Eingangsdaten abhängig ist. So können geringe Änderungen der Wirkung bereits starke Schwankungen der Ursache hervorrufen. Besonders bei fehlerbehafteten Messwerten, wie sie in der Praxis generell vorkommen, stellt es eine besondere Herausforderung dar, aus den vorhandenen Daten ausreichende Informationen zu gewinnen [17].

Als Beispiel für die Sensitivität inverser Problemstellungen den jeweiligen Eingangsdaten sei an dieser Stelle folgender Zusammenhang genannt: Während bei einem direkten Wärmeleitungsproblem die numerische Lösung der Wärmeleitungsgleichung mit kleineren Zeitschritten zu wesentlich stabileren Ergebnissen führt, kann es beim IHCP bei kleineren Zeitschritten bereits zu Stabilitätsproblemen kommen. Eine Verkleinerung des Zeitschritts hat demnach häufig den gegenteiligen Effekt auf die Genauigkeit der Lösung [2].

Lösungsmethoden für inverse Probleme

Zur Lösung inverser Berechnungsaufgaben gibt es in der Literatur verschiedene Ansätze - ebenso für das IHCP. Einige davon basieren auf einer analytischen Lösung. Es gibt zum Beispiel ein exaktes Verfahren von Burggraf [3], diverse Polynomansätze zur Lösung des IHCP, z.B. Mulholland [15] und Frank [8], oder Integraltransformationen nach Stolz [18] und Beck [1]. Diese Verfahren sind auf die Lösung linearer, eindimensionaler Probleme mit speziellen Anfangs- und Randbedingungen beschränkt. Daher wird als einzige analytische Berechnungsmethode ein Lösungsverfahren über eine Fourier-Transformation mit komplexer Übertragungsfunktion nach Carslaw und Jaeger [4] vorgestellt. Da numerische Lösungsverfahren, wie Finite-Differenzen-Methoden (FDM), praktisch auf jede Problemstellung anwendbar sind, werden in Kapitel 2.5.2 die Methoden nach Souza [6], Weber [22] und Hills & Hensel [9] näher betrachtet. Auf Verfahren, in denen das Problem mittels Regularisierung stabilisiert wird, wird nicht eingegangen, da vertiefte Kenntnisse der Funktionalanalysis für das Verständnis notwendig sind. Beispiele hierfür sind Miller [14] oder Tikhonov und Arsenin [20].

Weil die Berechnungen zur inversen Wärmeleitung mit dem Computer durchgeführt werden sollen, weshalb ausschließlich endlich dimensionale und diskrete Probleme behandelt werden können, wird die allgemeine Theorie zur Lösung inverser Probleme an dieser Stelle nicht behandelt. Vielmehr werden exemplarische Verfahren, die mit wenig Aufwand in einem Computersystem zu implementieren sind, herangezogen.

Gut und schlecht gestellte Probleme

Nach Hadamard¹ ist die Wohlgestelltheit eines Problems folgendermaßen definiert:

¹Jaques Salomon Hadamard, 1865 - 1963; frz. Mathematiker

\mathbf{X} und \mathbf{Y} seien normierte \mathbb{K} -Vektorräume und

$$F : \mathbb{U} \in \mathbf{X} \rightarrow \mathbb{W} \in \mathbf{Y} \quad (2.21)$$

eine Abbildung. Es wird das inverse Problem betrachtet, die Gleichung

$$F(u) = w, u \in \mathbb{U}, w \in \mathbb{W} \quad (2.22)$$

zu gegebenem w nach u aufzulösen. Dieses Problem heißt **wohlgestellt** (*properly posed*), wenn:

1. für jedes $w \in \mathbb{W}$ eine Lösung $u \in \mathbb{U}$ existiert (**Existenzbedingung**)
2. diese Lösung eindeutig ist (**Eindeutigkeitsbedingung**) und
3. die Umkehrfunktion $F^{-1} : \mathbb{W} \rightarrow \mathbb{U}$ stetig ist (**Stabilitätsbedingung**).

Andernfalls gilt das inverse Problem als **schlecht gestellt** (*ill posed*) [17].

Wie von Beck, Blackwell und St. Clair [2] auf verschiedene Weise gezeigt, können Existenz und Eindeutigkeit der Lösung leicht bewiesen werden. Allerdings ist keine Stabilität gegeben, d.h. kleine Änderungen der Eingangsdaten können große Änderungen der Ausgangsdaten zur Folge haben. Da nicht alle drei Kriterien für ein gut gestelltes Problem erfüllt sind, ist das IHCP ein schlecht gestelltes Problem.

2.4 Analytische Lösungsmethode mit einer Übertragungsfunktion

Zur Analyse von elektrischen Schaltkreisen werden diese in der Vierpoltheorie [19] zunächst als eine „Black Box“ betrachtet. Aus den Informationen, die über die Ein- und Ausgangsdaten vorliegen, kann im nächsten Schritt eine Übertragungsfunktion modelliert werden. Mit Hilfe dieser kann dann auf den Aufbau der Schaltkreise geschlossen werden. Ist allerdings der Schaltkreis bekannt, so können bei Kenntnis der Eingangsdaten die Ausgangsdaten direkt angegeben werden. Dieses Vorgehen aus der Elektrotechnik lässt sich analog für die Wärmeleitung übernehmen unter der Voraussetzung, dass es sich bei den Signalen um periodische Schwingungen handelt. Dies stellt keine Einschränkung dar, da jedes nichtperiodische Signal mittels Fourier-Transformation in eine Linearkombination harmonischer Schwingungen mit kontinuierlichem Spektrum zerlegt werden kann [12]. In Abbildung 2.4 sind die Lösungsschritte im Zeit- und Frequenzbereich dargestellt.

Ist der wärmeleitende Körper bekannt, so kann eine geeignete Übertragungsfunktion in Form einer Matrix $\underline{\underline{A}} \in \mathbb{M}^{2 \times 2}$ bestimmt werden. Mit Hilfe dieser lässt sich demnach aus bekannter Temperatur und Wärmestrom an der Außenseite unmittelbar durch Multiplikation Temperatur und Wärmestrom an der Innenseite der Geometrie berechnen:

$$\hat{\underline{\underline{T}}}_i = \underline{\underline{A}} \hat{\underline{\underline{T}}}_a \quad (2.23)$$

Wobei $\underline{\hat{T}}$ ein Vektor mit Einträgen für Temperatur und Wärmefluss am jeweiligen Ort ist.

$$\underline{\hat{T}} = \begin{bmatrix} \hat{T} \\ \hat{q} \end{bmatrix} \quad (2.24)$$

Die Darstellung der Übertragungsfunktion in Matrixschreibweise geht auf Strecker und Feldtkeller zurück [19].

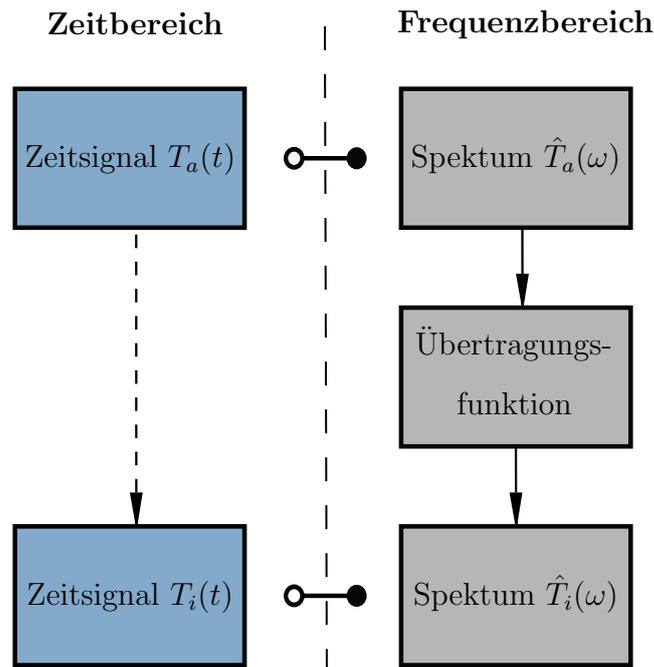


Abbildung 2.4: Lösungsweg für periodische Signale mit einer Übertragungsfunktion.

Herleitung einer Übertragungsfunktion

Unter der Annahme, dass sich Temperaturänderungen in Form einer Welle mit Amplitude \hat{T} und Periode S in einem Voll- oder Hohlzylinder ausbreiten („Temperaturwellen“), führt dies mit Hilfe des Produktansatzes $T(r, t) = \hat{T}(r)e^{i\omega t}$ mit $\omega = \frac{2\pi}{S}$ die Gleichung (2.19) im eindimensionalen Fall auf

$$\frac{\partial^2 \hat{T}}{\partial r^2} + \frac{1}{r} \frac{\partial \hat{T}}{\partial r} - \frac{i\omega}{a} \hat{T} = 0. \quad (2.25)$$

Der Zeitfaktor $e^{i\omega t}$ mit dem jede Größe stets multipliziert werden muss, wird im Folgenden zur Verbesserung der Übersichtlichkeit weggelassen. Am Ende der Berechnung wird er wieder hinzugefügt und es wird entschieden, ob der reelle oder imaginäre Anteil der Schwingung maßgeblich für das Ergebnis ist [4].

Die Gleichung (2.25) stellt eine modifizierte Bessel'sche Differentialgleichung 2. Ordnung dar. Die Lösung dieser DGL erfordert die Bestimmung von zwei Integrationskonstanten, hierzu genügen bereits die bekannten Randbedingungen an einer Außenseite der Rohrleitung. Wie von McLachlan [13] vertieft ausgeführt, ist

$$\hat{T} = kI_0(\bar{r}) + lK_0(\bar{r}) \quad (2.26)$$

eine Lösung von Gleichung (2.25), wobei $\bar{r} = r\sqrt{\frac{i\omega}{a}}$. Zusammen mit dem Wärmefluss

$$\hat{q} = -\lambda \frac{\partial \hat{T}}{\partial r} = k - \lambda \sqrt{\frac{i\omega}{a}} I_1(\bar{r}) + l\lambda \sqrt{\frac{i\omega}{a}} K_1(\bar{r}) \quad (2.27)$$

gibt es zwei Gleichungen zur Bestimmung der Koeffizienten k und l in Abhängigkeit der gegebenen Randbedingungen. Die modifizierten Besselfunktionen mit komplexen Argumenten in Gleichung (2.26) und Gleichung (2.27) können dabei nach Carslaw und Jaeger [4] als numerisch bekannt angenommen werden.

Nimmt man nun \hat{T}_i und \hat{q}_i für Temperatur und Wärmestrom an der Rohrrinnenwand an und \hat{T}_a und \hat{q}_a für Temperatur und Wärmestrom an der Rohraußenwand, so ergibt sich durch Auflösen nach k und l und Einsetzen der beiden Variablen folgende Übertragungsfunktion

$$\begin{bmatrix} \hat{T}_a \\ \hat{q}_a \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \hat{T}_i \\ \hat{q}_i \end{bmatrix}, \quad (2.28)$$

mit folgenden Matrixeinträgen

$$A = \bar{r}_i [I_0(\bar{r}_a) K_1(\bar{r}_i) + K_0(\bar{r}_a) I_1(\bar{r}_i)] \quad (2.29)$$

$$B = \frac{\bar{r}_a}{\lambda} [I_0(\bar{r}_i) K_0(\bar{r}_a) - K_0(\bar{r}_i) I_0(\bar{r}_a)] \quad (2.30)$$

$$C = \lambda \frac{\omega}{a} \bar{r}_i [I_1(\bar{r}_a) K_1(\bar{r}_i) - K_1(\bar{r}_a) I_1(\bar{r}_i)] \quad (2.31)$$

$$D = \bar{r}_i [I_0(\bar{r}_a) K_1(\bar{r}_i) + K_0(\bar{r}_a) I_1(\bar{r}_i)]. \quad (2.32)$$

Für die Berechnung der Randbedingungen an der Innenseite bei gegebenen Randbedingungen an der Außenseite muss die Matrix der Übertragungsfunktion lediglich invertiert werden

$$\begin{bmatrix} \hat{T}_i \\ \hat{q}_i \end{bmatrix} = \frac{r_a}{r_i} \begin{bmatrix} D & -B \\ -C & A \end{bmatrix} \begin{bmatrix} \hat{T}_a \\ \hat{q}_a \end{bmatrix}. \quad (2.33)$$

Schließlich führt ein Einfügen des Zeitfaktors $e^{i\omega t}$ und das Berechnen des Realteils zur gewünschten Lösung [4].

$$T_i(t) = \text{Re} \left(\hat{T}_i e^{i\omega t} \right) \quad (2.34)$$

2.5 Numerische Lösungsmethoden

Da es für partielle Differentialgleichungen, wie z.B. die Fourier'sche DGL (2.18), im Allgemeinen kein exaktes Lösungsverfahren gibt, ist man in der Praxis oft auf numerische Lösungsmethoden angewiesen. Eine Klasse dieser Verfahren ist die im Folgenden betrachtete *Finite-Differenzen-Methode*. Durch die Approximation der Differentialquotienten der DGL durch Differenzenquotienten kann auf einem diskretisierten Berechnungsgebiet eine Lösung der ursprünglichen partiellen Differentialgleichung angegeben werden. Hierfür werden zunächst die benötigten *Finiten Differenzen* hergeleitet, bevor verschiedene FDM-Verfahren erläutert werden.

2.5.1 Numerische Differentiation

Ist eine analytische Differentiation nicht möglich beziehungsweise zu aufwendig, wird es notwendig auf numerische Verfahren der Differentiation zurückzugreifen. Da die Ableitung einer Funktion f an einer Stelle x_0 für ein kleines $h \neq 0$ folgendermaßen definiert ist

$$\frac{\partial f}{\partial x} = \lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h} \quad (2.35)$$

liegt es nahe die Ableitung durch die **Zweipunkte-Formel**

$$\frac{\partial f}{\partial x} \approx \frac{f(x_0 + h) - f(x_0)}{h} \quad (2.36)$$

zu approximieren. Der Fehler dieser Approximation, die in Abbildung 2.5 dargestellt ist, wird dabei noch nicht näher spezifiziert [7].

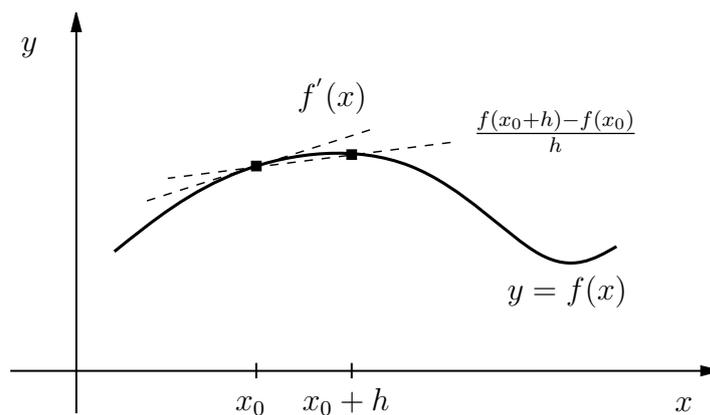


Abbildung 2.5: Approximation der Ableitung mit Hilfe der Zweipunkte-Formel, nach [7, 21].

Herleitung der finiten Differenzen

Formeln für sogenannte **finite Differenzen** können im Allgemeinen mit Lagrange-Polynomen (vgl. Anlage A) hergeleitet werden. Nimmt man $(n + 1)$ Stützstellen x_0, \dots, x_n in einem Intervall an, so kann $f(x)$ durch das n -te Lagrange-Polynom approximiert werden:

$$f(x) = \sum_{i=0}^n f(x_i) L_{ni}(x) + \frac{f^{(n+1)}(\xi(x))}{(n+1)!} (x-x_0) \dots (x-x_n). \quad (2.37)$$

Die erste Ableitung dieses Polynoms lautet inklusive Fehlerterm:

$$\begin{aligned} \frac{\partial f(x_0)}{\partial x} = \sum_{i=0}^n f(x_i) \frac{\partial L_{ni}(x)}{\partial x} + \frac{f^{(n+1)}}{(n+1)!} \frac{d}{dx} [(x-x_0) \dots (x-x_n)] \\ + \frac{d}{dx} \left[\frac{f^{(n+1)}(\xi(x))}{(n+1)!} \right] \underbrace{(x-x_0) \dots (x-x_n)}_{=0, \text{ wenn } x=x_j} \end{aligned} \quad (2.38)$$

Durch Einsetzen von $x = x_j$ erhält man:

$$\frac{\partial f(x_j)}{\partial x} = \underbrace{\sum_{i=0}^n f(x_i) \frac{\partial L_{ni}(x_j)}{\partial x}}_{\text{numerische Approximation}} + \underbrace{\frac{f^{(n+1)}(\xi(x_j))}{(n+1)!} \prod_{i=0, j \neq i}^n (x_j - x_i)}_{\text{Fehlerterm}} \quad (2.39)$$

Für zwei gegebene Punkte, $x_0, x_0 + h$, lässt sich hieraus die endgültige **Zweipunkte-Formel** herleiten

$$\frac{\partial f(x_0)}{\partial x} = \frac{f(x_0 + h) - f(x_0)}{h} - \underbrace{\frac{h}{2} \frac{\partial^2 f(\xi)}{\partial x^2}}_{\text{Fehlerterm}} \quad (2.40)$$

wobei $\xi \in [x_0, x_0 + h]$. Ist $h > 0$ handelt es sich um eine aufsteigende Differenz, wenn $h < 0$ um eine absteigenden Differenz [7].

Für die Approximation der Ableitung an inneren Stützstellen ist die **Dreipunkte-Mittelpunkt-Formel** (siehe Abbildung 2.6) besser geeignet. Mit Hilfe des zweiten Lagrange-Polynoms lässt sich diese für drei gegebene Punkte $x_0 - h, x_0$ und $x_0 + h$ herleiten,

$$\frac{\partial f(x_0)}{\partial x} = \frac{1}{2h} [f(x_0 + h) - f(x_0 - h)] - \underbrace{\frac{h^2}{6} f^{(3)}}_{\text{Fehlerterm}} \quad (2.41)$$

wobei hier $\xi \in [x_0 - h, x_0 + h]$.

Ableitungen höherer Ordnung lassen sich in analoger Weise herleiten [7]. Die zweite Ableitung kann beispielsweise mit der Dreipunkte-Mittelpunkt-Formel für gegebene $x_0 - h, x_0, x_0 + h$ approximiert werden zu

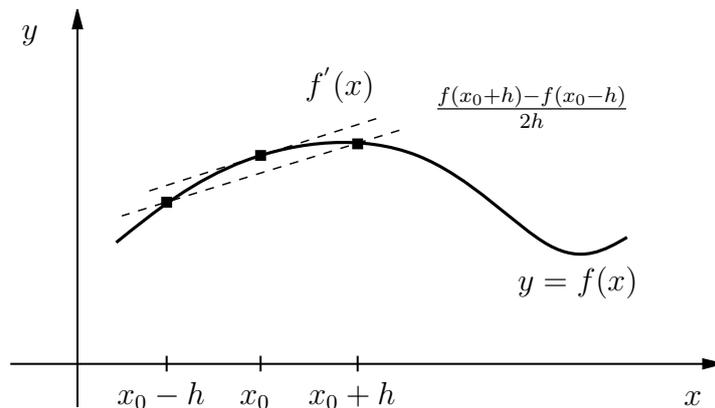


Abbildung 2.6: Approximation der Ableitung mit Hilfe der Dreipunkte-Mittelpunkt-Formel, nach [7, 21].

$$\frac{\partial^2 f(x_0)}{\partial x^2} = \frac{1}{h^2} [f(x_0+h) - 2f(x_0) + f(x_0-h)] - \underbrace{\frac{h^2}{12} f^{(4)}}_{\text{Fehlerterm}}. \quad (2.42)$$

Wobei auch hier wieder $\xi \in [x_0 - h, x_0 + h]$.

Konvergenz der Verfahren

Der Effekt des numerischen Rundungsfehlers darf bei der numerischen Differentiation nicht außer Acht gelassen werden. Durch Reduzieren der Schrittweite h lässt sich im Allgemeinen der Verfahrensfehler der finiten Differenzen weiter senken, aber der Rundungsfehler steigt stark an. Da in jedem Rechenschritt Zahlen voneinander subtrahiert werden, die sich nur um $h \frac{\partial f(x_0)}{\partial x}$ unterscheiden, gehen pro Rechenoperation $\log_{10}(\frac{1}{h})$ Stellen durch Auslöschung verloren. Eine zu kleine Schrittweite ist deshalb unvorteilhaft, da in diesem Fall der Rundungsfehler die Berechnung dominiert [21]. Dies wird in Abbildung 2.7 besonders deutlich. Der Betrag der Differenz des analytisch berechneten Differentialquotienten der gebrochen rationalen Funktion $f(x) = \frac{5x^4}{(x+1)^6}$ ausgewertet an der Stelle $x = 0.6$ und den numerischen Approximationen ist als relativer Fehler über der Schrittweite h aufgetragen.

Demzufolge wird die numerische Differentiation auch als **instabiles Verfahren** bezeichnet. Je höher der Grad der Ableitung, desto größer wird auch der Einfluss von Ungenauigkeiten in den Eingangsdaten und der Einfluss der Rundungsfehler. Dies ist damit zu begründen, dass durch eine Potenz von h dividiert werden muss und bei der numerischen Division durch kleine Zahlen hohe Rundungsfehler auftreten [21].

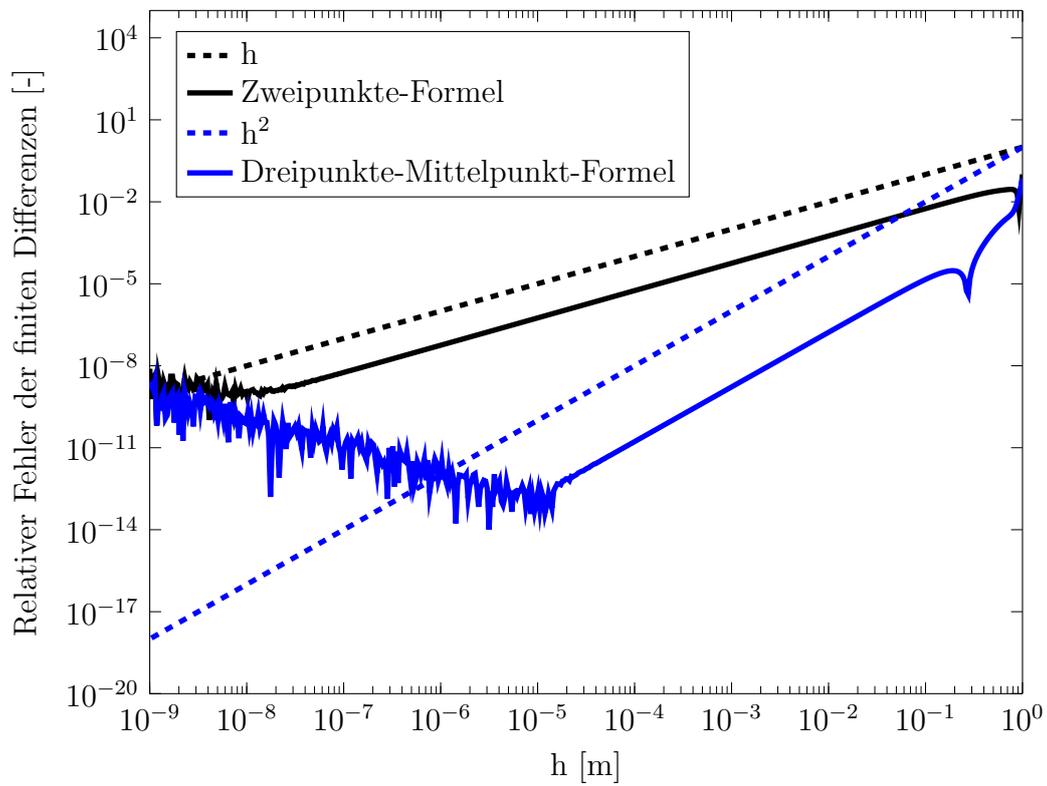


Abbildung 2.7: Konvergenzverhalten der Zweipunkte-Formel und der Dreipunkte-Mittelpunkt-Formel.

2.5.2 Finite-Differenzen-Methode (FDM)

Wie bereits in Kapitel 2.3 erwähnt, sind insbesondere instationäre Problemstellungen im Allgemeinen nicht mehr analytisch lösbar. Deshalb bietet sich eine numerische Berechnung mit der Finite-Differenzen-Methoden (FDM) an [16].

Ein Vorteil dieser Methode ist die Erweiterbarkeit auf nichtlineare Probleme, wie sie beispielsweise bei der Berücksichtigung temperaturabhängiger Stoffwerte auftreten. In diesem Fall ist in jedem Berechnungsschritt ein nichtlineares Gleichungssystem zu lösen, was zum Beispiel mit dem Newton-Raphson-Verfahren auf einem Computer implementiert werden kann [21].

Zur Verbesserung der Lesbarkeit werden im Folgenden alle thermophysikalischen Eigenschaften als konstant angenommen. Dies bedeutet keinesfalls eine Einschränkung, alle genannten FDM-Verfahren können ebenso auf nichtlineare Problemstellungen angewendet werden.

Diskretisierung

Betrachtet man folgendes Anfangs-Randwertproblem für einen Stab der Länge $l = 1$

$$\frac{1}{a} \frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} \quad (2.43)$$

$$T(x, t) = T_0 \quad (\text{Anfangsbedingung}) \quad (2.44)$$

$$T(0, t) = T(1, t) = T_a \quad (\text{Randbedingungen}) \quad (2.45)$$

so wird der kontinuierliche Temperaturverlauf, der sich einstellen wird, in jedem Punkt x eine Lösung der Gleichung (2.43) sein.

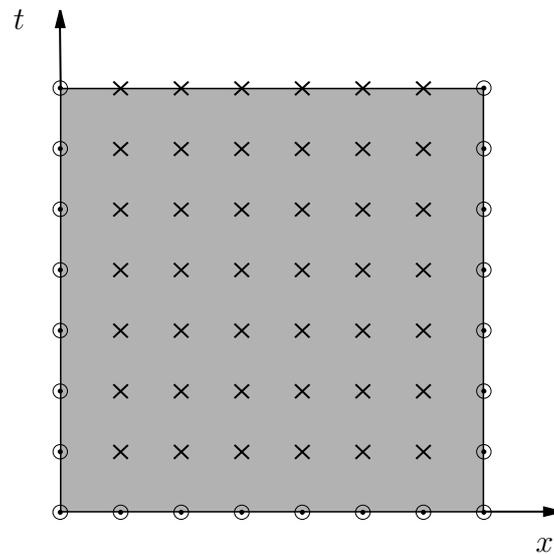
Wie in Abbildung 2.8 dargestellt, überzieht man nun das Berechnungsgebiet, in diesem Fall $0 \leq x \leq 1, t \geq 0$, mit einem regelmäßigen Gitter endlich vieler Punkte. So kann der Temperaturverlauf an diesen diskreten Stützstellen ausgewertet werden. Dazu muss die Gleichung (2.43) ebenfalls diskretisiert werden, welches mit Hilfe der im Kapitel 2.5.1 vorgestellten finiten Differenzen gelingt [16].

Dieses Vorgehen, d.h. die Diskretisierung sowohl entlang der Ortskoordinate x als auch der Zeitkoordinate t , bezeichnet man auch als **globale Diskretisierung**. Die Methode ist allerdings nur für den eindimensionalen Fall sinnvoll, da man in der Ebene oder im Raum zu große Gleichungssysteme erhält. In solchen Fällen ist es angebracht eine andere Diskretisierung zu wählen, z.B. nur die Ortskoordinaten zu diskretisieren und ein DGL-System in der Zeitkoordinate zu lösen [12].

FTCS-Schema

Wendet man beispielsweise die Zweipunkte-Formel (2.40) auf die Zeitableitung und die Dreipunkte-Mittelpunkt-Formel für die zweite Ableitung (2.42) auf die Ortskoordinate in Gleichung (2.43) an

$$\frac{1}{a} \frac{T_j^{i+1} - T_j^i}{\Delta t} = \frac{T_{j-1}^i - 2T_j^i + T_{j+1}^i}{(\Delta x)^2} \quad (2.46)$$



| |
|---|
| <p>⊙ Knoten mit bekannter Temperatur</p> <p>× Knoten mit zu berechnender Temperatur</p> |
|---|

Abbildung 2.8: Diskretisierung des Berechnungsgebietes, nach [2].

und löst nach der Temperatur an dem noch unbekanntem Knoten auf, so erhält man eine direkte Berechnungsvorschrift für die gesuchten Temperaturen. Auf die Gleichung wurde das in Abbildung 2.9 dargestellte **FTCS-Schema** (forward time, centered space) angewendet.

$$T_j^{i+1} = a\Delta t \left(\frac{T_{j-1}^i - 2T_j^i + T_{j+1}^i}{(\Delta x)^2} \right) + T_j^i \quad (2.47)$$

Durch ein „Abrastern“ des gesamten Gebietes, indem man eine Berechnungsschleife über alle Orts- und Zeitschritte laufen lässt, können Informationen über die Gitterpunkte hinweg ausgetauscht werden und der approximierter Temperaturverlauf zu jedem Zeitpunkt berechnet werden [16].

Explizite Verfahren neigen dazu sehr schnell zu divergieren, sobald die Zeitschrittweite zu groß gewählt wird [16]. Ein geeignetes Stabilitätskriterium lautet wie folgt:

$$a \frac{\Delta t}{(\Delta x)^2} \leq 0.5. \quad (2.48)$$

Dieses Konvergenzproblem gibt es bei der impliziten Berechnung nach dem **BTCS-Schema** (backward time, centered space) nicht. Nach den drei unbekanntem Größen aufgelöst ergibt sich für das implizite Schema folgende Berechnungsvorschrift:

$$\frac{a}{(\Delta x)^2} T_{j+1}^{i+1} + T_j^{i+1} \left(-\frac{2a}{(\Delta x)^2 - \frac{1}{\Delta t}} \right) + T_{j+1}^{i+1} \frac{a}{(\Delta x)^2} = -\frac{1}{\Delta t} T_j^i \quad (2.49)$$

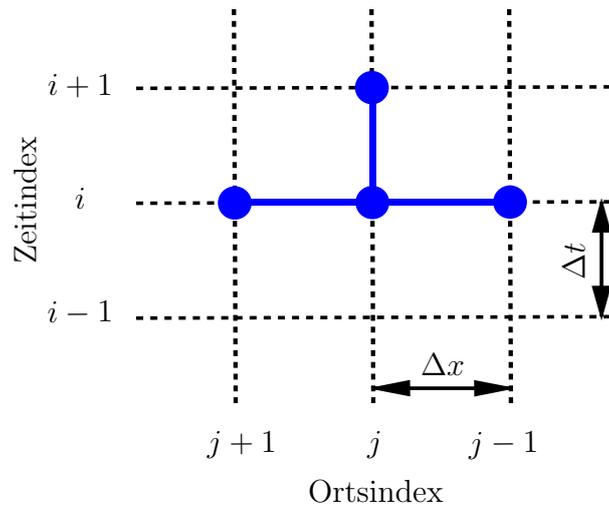


Abbildung 2.9: FTCS-Berechnungsschema, nach [2, 16].

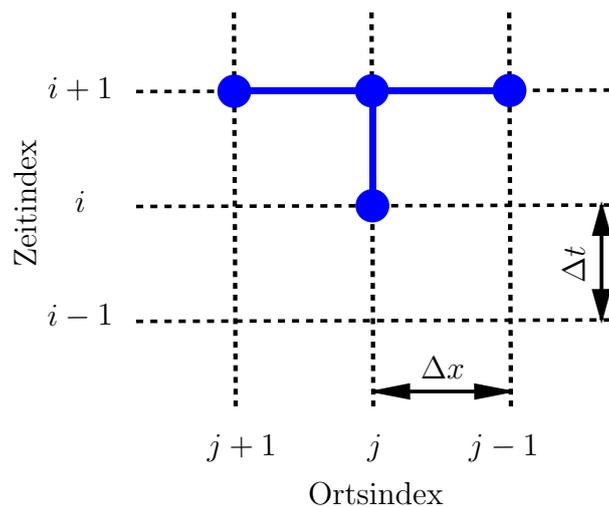


Abbildung 2.10: BTCS-Berechnungsschema, nach [2, 16].

Allerdings ist hier in jedem Berechnungsschritt ein Gleichungssystem zu lösen, weil lediglich die Koeffizienten der gesuchten Werte gegeben sind. Die Berechnung wird damit deutlich aufwendiger, aber die Wahl des Zeitschritts ist nicht durch ein Stabilitätskriterium eingeschränkt. Um ein geeignetes Verfahren auszuwählen müssen für jede Berechnung Vor- und Nachteile der jeweiligen Verfahren abgewogen werden [16].

Anwendung auf inverse Probleme

Mit dem in Kapitel 2.5.2 vorgestellten Verfahren ist es möglich eine approximierte Lösung eines wohlgestellten, direkten Problems zu berechnen. Dazu wird zu jedem diskreten Zeitschritt der Temperaturverlauf an den Ortsknoten bestimmt. Ist bei einem inversen, schlecht

gestellten Problem sowohl Temperatur als auch Wärmestrom am Rand zu jeder Zeit bekannt, liegt es nahe, zunächst die Temperaturen zu jeder Zeit am benachbarten Ortsknoten zu berechnen. Das inverse Randwertproblem wird so zu einem direkten Anfangswertproblem [2].

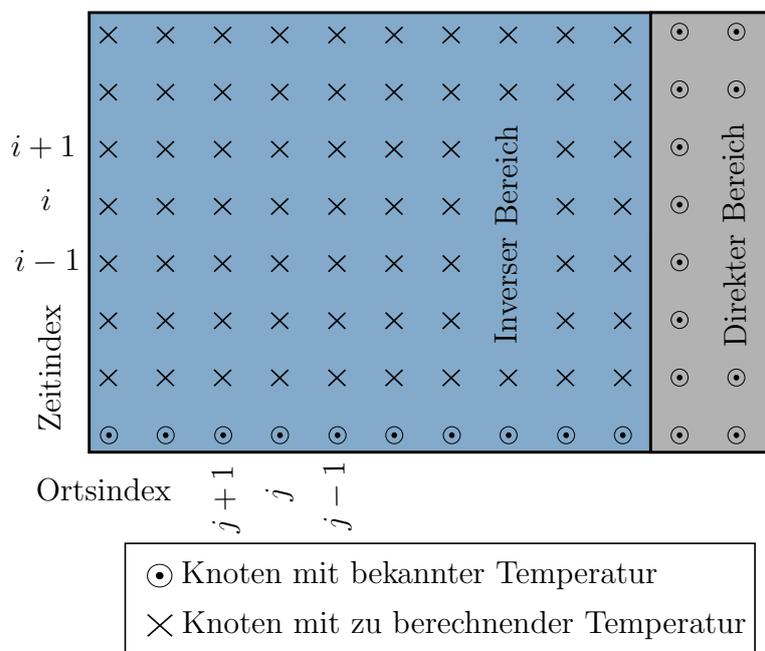


Abbildung 2.11: Berechnungsgebiet mit direktem und inversem Bereich, nach [2].

Wie in Abbildung 2.11 gezeigt, kann man für das IHCP das Berechnungsgebiet aufteilen in einen direkten und einen inversen Bereich. Im direkten Bereich ist ein Anfangsrandwertproblem zu lösen, welches ähnlich gestellt ist wie das Beispiel in Kapitel 2.5.2. Nach dem Bestimmen des gesamten Temperaturfeldes kann man durch dessen Differentiation den Wärmestrom berechnen und erhält beide für die Lösung des inversen Problems notwendigen Randbedingungen.

Im Folgenden werden einige exemplarische Berechnungsstrategien für das inverse Gebiet vorgestellt. Obwohl die Vorstellung anhand einer zylindrischen Geometrie erfolgt, sind alle Verfahren auch für die Geometrie von Platte und Kugel uneingeschränkt geeignet. Außerdem ist eine Erweiterung auf temperaturabhängige, veränderliche Stoffwerte möglich, worauf der Übersichtlichkeit wegen verzichtet wird.

Methode von D'Souza

Die Idee des von D'Souza [6] entwickelten Verfahrens ist die numerische Extrapolation der Lösung des direkten Problems auf den inversen Bereich. Dazu wird das implizite BTCS-Schema zur Diskretisierung der eindimensionalen Gleichung (2.19) verwendet, d.h. die Zeitableitung wird durch einen absteigenden Differenzenquotient und die Ortsableitung

gen durch die Dreipunkte-Mittelpunkt-Formel (2.41) ersetzt. Eine Darstellung des Berechnungsnetzes findet sich in Abbildung 2.10.

$$\frac{1}{a} \frac{T_j^i - T_j^{i-1}}{\Delta t} = \frac{T_{j-1}^i - 2T_j^i + T_{j+1}^i}{(\Delta r)^2} + \frac{1}{r} \frac{T_{j-1}^i - T_{j+1}^i}{2\Delta r} \quad (2.50)$$

Wie bereits vorgestellt, wird aus der impliziten Berechnungsvorschrift bei der Anwendung auf das IHCP eine explizite Formel. Löst man die Gleichung nach der noch unbekanntem Temperatur auf, so erhält man folgende Berechnungsvorschrift:

$$\begin{aligned} T_{j+1}^i &= A_S (T_j^i - T_j^{i-1}) \\ &\quad + B_S (2T_j^i - T_{j-1}^i) \\ &\quad + C_S T_{j-1}^i, \end{aligned} \quad (2.51)$$

wobei

$$A_S = \frac{\frac{1}{a\Delta t}}{\frac{1}{(\Delta r)^2} - \frac{1}{2r\Delta r}} \quad (2.52)$$

$$B_S = \frac{\frac{1}{\Delta r}}{\frac{1}{(\Delta r)^2} - \frac{1}{2r\Delta r}} \quad (2.53)$$

$$C_S = \frac{\frac{1}{2r\Delta r}}{\frac{1}{(\Delta r)^2} - \frac{1}{2r\Delta r}}. \quad (2.54)$$

Auf diese Weise erhält man Schritt für Schritt die Temperaturverteilung am jeweils nächsten Knoten bis zum gesuchten inneren Rand.

Methode von Weber

Im Verfahren nach Weber [22] wird die Wärmeleitungsgleichung durch eine approximierende, hyperbolische Gleichung ersetzt. Für Gleichungen hyperbolischen Typs sind die Bedingungen für die Wohlgestelltheit des Problems (siehe Kapitel 2.3) erfüllt.

$$\sigma \frac{\partial^2 T}{\partial t^2} + \frac{\partial T}{\partial t} = a \left[\frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} \right] \quad (2.55)$$

Dies ist ein Sonderfall der Telegrafengleichung deren Lösungen an fortschreitende Wellen erinnern, wobei $\sigma > 0$ deren Wellengeschwindigkeit beschreibt. In Gleichung (2.55) werden alle Differentialquotienten mit Hilfe der Dreipunkte-Mittelpunkt-Formel (2.41) diskretisiert, das Berechnungsschema ist in Abbildung 2.12 dargestellt.

$$\begin{aligned} \frac{\sigma}{(\Delta t)^2} (T_j^{i+1} - 2T_j^i + T_j^{i-1}) + \frac{1}{2\Delta t} (T_j^{i+1} - T_j^{i-1}) = \\ \frac{a}{(\Delta r)^2} (T_{j+1}^i - 2T_j^i + T_{j-1}^i) + \frac{a}{2r\Delta r} (T_{j+1}^i - T_{j-1}^i) \end{aligned} \quad (2.56)$$

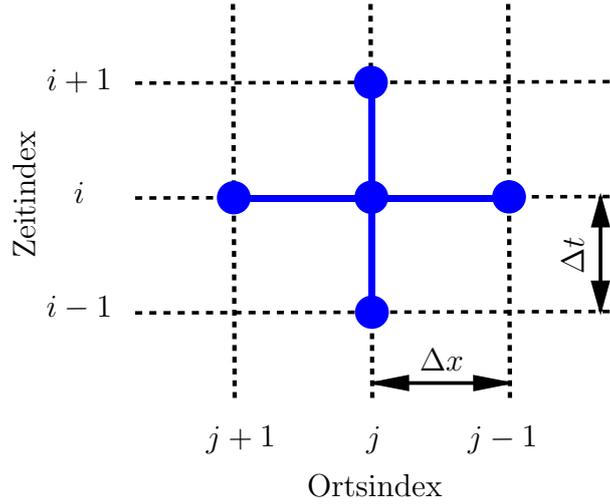


Abbildung 2.12: Berechnungsschema nach Weber, nach [2, 16].

Aufgelöst nach der gesuchten Temperatur T_{j+1}^i erhält man folgende Berechnungsvorschrift:

$$\begin{aligned} T_{j+1}^i = & A_W (T_j^{i+1} - 2T_j^i + T_j^{i-1}) \\ & + B_W (T_j^{i+1} - T_j^{i-1}) \\ & + C_W (2T_{j-1}^i) \\ & + D_W (T_{j-1}^i), \end{aligned}$$

mit den Koeffizienten

$$A_W = \frac{\frac{\sigma}{(\Delta t)^2}}{\frac{a}{(\Delta r)^2} + \frac{a}{2r\Delta r}} \quad (2.57)$$

$$B_W = \frac{\frac{1}{2\Delta t}}{\frac{a}{(\Delta r)^2} + \frac{a}{2r\Delta r}} \quad (2.58)$$

$$C_W = \frac{\frac{a}{(\Delta r)^2}}{\frac{a}{(\Delta r)^2} + \frac{a}{2r\Delta r}} \quad (2.59)$$

$$D_W = \frac{\frac{a}{2r\Delta r}}{\frac{a}{(\Delta r)^2} + \frac{a}{2r\Delta r}}. \quad (2.60)$$

Hierbei ist zu beachten, dass man für jeden Ortsschritt einen Wert des Temperaturverlaufs (mit Länge n) „verliert“, weil der Wert der Temperatur T mit den Indizes j^{i+1} nicht definiert ist. Der Temperaturverlauf am gesuchten Rand ist damit um die Anzahl der Ortsknoten kürzer. Ist die Temperatur am Intervallende gesucht müssen, wie in Abbildung 2.13 dargestellt, entsprechend $M + N$ Messwerte aufgenommen werden.

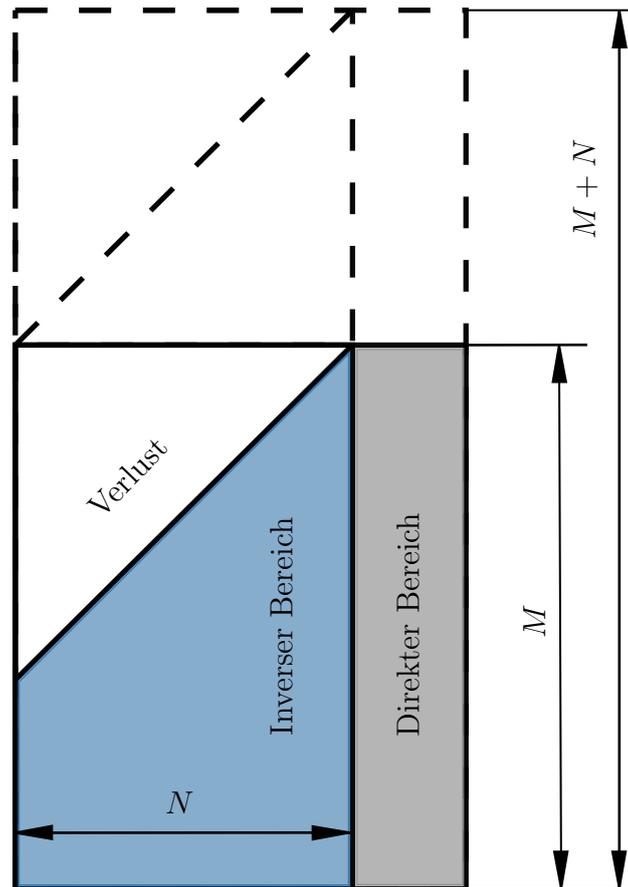


Abbildung 2.13: Datenverlust des Temperaturverlaufs durch Definitionslücken beim Verfahren von Weber [22].

Courant, Fredricks und Lewy [5] haben gezeigt, dass die Gleichung (2.57) konvergiert solange folgende Ungleichung erfüllt ist:

$$\sigma \frac{(\Delta r)^2}{(\Delta t)^2} \leq 1. \quad (2.61)$$

Diese Ungleichung (2.61) ist bereits erfüllt, wenn σ sehr klein gewählt wird. Dies hat Weber zur Bedingung gemacht, ohne eine genaue Größenordnung für den Parameter σ zu nennen. Der Algorithmus liefert sogar für $\sigma = 0$ Ergebnisse [2].

Methode von Hills & Hensel

Der Ansatz von Hills & Hensel [9] besteht darin die Wärmeleitungsgleichung

$$\frac{1}{a} \frac{\partial T}{\partial t} = \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{\partial T}{\partial r} \right) \quad (2.62)$$

durch ein System von Differentialgleichungen erster Ordnung zu ersetzen.

$$\dot{q} = -\lambda \frac{\partial T}{\partial r} \quad (2.63)$$

$$\rho c \frac{\partial T}{\partial t} = -\frac{1}{r} \frac{\partial r \dot{q}}{\partial r} \quad (2.64)$$

Wendet man nun die Zweipunkte-Formel (2.40) auf die Ortsableitungen an und die Dreipunkte-Mittelpunkt-Formel (2.41) auf die Zeitableitung und sortiert um, erhält man

$$T_{j+1}^i = -\frac{\Delta r}{\lambda} \dot{q}_{j+1}^i + T_j^i \quad \text{und} \quad (2.65)$$

$$\dot{q}_{j+1}^i = \frac{r_j}{r_{j+1}} \dot{q}_j^i + \frac{r_j}{r_{j+1}} \frac{\rho c \Delta r}{2 \Delta t} (-T_j^{i+1} + T_j^{i-1}). \quad (2.66)$$

Einsetzen von Gleichung (2.66) in (2.65) liefert

$$T_{j+1}^i = -\left(\frac{r_j}{r_{j+1}} \right) \frac{\Delta r}{\lambda} \dot{q}_j^i + T_j^{i-1} \left[-\left(\frac{r_j}{r_{j+1}} \right) \frac{\Delta r^2}{\lambda} \right] + T_j^i + T_j^{i+1} \left[\left(\frac{r_j}{r_{j+1}} \right) \frac{\Delta r^2}{\lambda} \right] \quad (2.67)$$

Mit Gleichungen (2.66) und (2.67) kann sowohl die Temperatur als auch der Wärmestrom am Knoten $j + 1$ berechnet werden, wenn beides am Knoten j bereits bekannt ist.

Zur Berechnung des ersten und letzten Zeitschritts sind besondere Operationen notwendig. Nimmt man beispielsweise als Anfangsbedingung an, dass sich das System in einem stationären Zustand befindet, so ist der Wärmefluss zum Anfangszeitpunkt konstant. Die Gleichungen (2.65) und (2.66) können deshalb mit

$$r_{j+1} \dot{q}_{j+1}^1 = r_j \dot{q}_j^1 \quad (2.68)$$

$$T_{j+1}^i = -\frac{\Delta r}{\lambda} \dot{q}_{j+1}^1 + T_j^1 \quad (2.69)$$

angenähert werden. Zum letzten Zeitschritt $i = M$ kann Gleichung (2.67) nicht angewendet werden, da die Temperatur für den Zeitschritt $M + 1$ nicht definiert ist. Daher wird die Gleichung (2.64) in diesem Fall durch eine absteigende Zweipunkte-Formel (2.40) diskretisiert, welches auf

$$\dot{q}_{j+1}^M = \frac{r_j}{r_{j+1}} \dot{q}_j^M + \frac{r_j}{r_{j+1}} \frac{\rho c \Delta r}{\Delta t} (T_j^{M-1} + T_j^M) \quad (2.70)$$

führt. Setzt man nun Gleichung (2.70) in (2.65) ein, so erhält man

$$T_{j+1}^M = - \left(\frac{r_j}{r_{j+1}} \right) \frac{\Delta r}{\lambda} \dot{q}_j^M + T_j^{M-1} \left[- \left(\frac{r_j}{r_{j+1}} \right) \frac{a \Delta r^2}{\Delta t} \right] + T_j^M \left[1 + \left(\frac{r_j}{r_{j+1}} \right) \frac{a \Delta r^2}{\Delta t} \right] \quad (2.71)$$

Durch die Verwendung eines anderen Berechnungsnetzes für den finalen Zeitschritt kann ein Verlust von Werten im Temperaturverlauf, wie es bei Weber [22] der Fall ist, vermieden werden.

Der Algorithmus von Hills & Hensel kann gut in Matrizenform dargestellt werden. Dazu muss zunächst ein erweiterter Wärmefluss-Temperatur-Vektor wie folgt definiert werden:

$$\underline{\vartheta}_j = [q_j^1 q_j^2 \dots q_j^M, T_j^1 T_j^2 \dots T_j^M]^T. \quad (2.72)$$

Nun können die Gleichungen (2.66) bis (2.71) in Matrizenform zusammengefasst werden.

$$\underline{\vartheta}_{j+1} = \underline{E}_j \underline{\vartheta}_j \quad (2.73)$$

wobei die Matrix \underline{E}_j aus folgenden Teilmatrizen besteht:

$$\underline{E}_j = \begin{bmatrix} \underline{A}_j & \underline{B}_j \\ \underline{C}_j & \underline{D}_j \end{bmatrix} \quad (2.74)$$

mit

$$\underline{A}_j = \frac{r_j}{r_{j+1}} \underline{I} \quad (2.75)$$

(\underline{I} ist eine $M \times M$ -Einheitsmatrix)

$$\underline{B}_j = \begin{bmatrix} 0 & & & & & 0 \\ b_j^2 & 0 & -b_j^2 & & & \\ & b_j^3 & 0 & -b_j^3 & & \\ & & \ddots & \ddots & \ddots & \\ 0 & & & b_j^{M-1} & 0 & -b_j^{M-1} \\ & & & & 2b_j^M & -2b_j^M \end{bmatrix} \quad (2.76)$$

$$\underline{C}_j = - \frac{r_j \Delta r}{r_{j+1} \lambda} \underline{I} \quad (2.77)$$

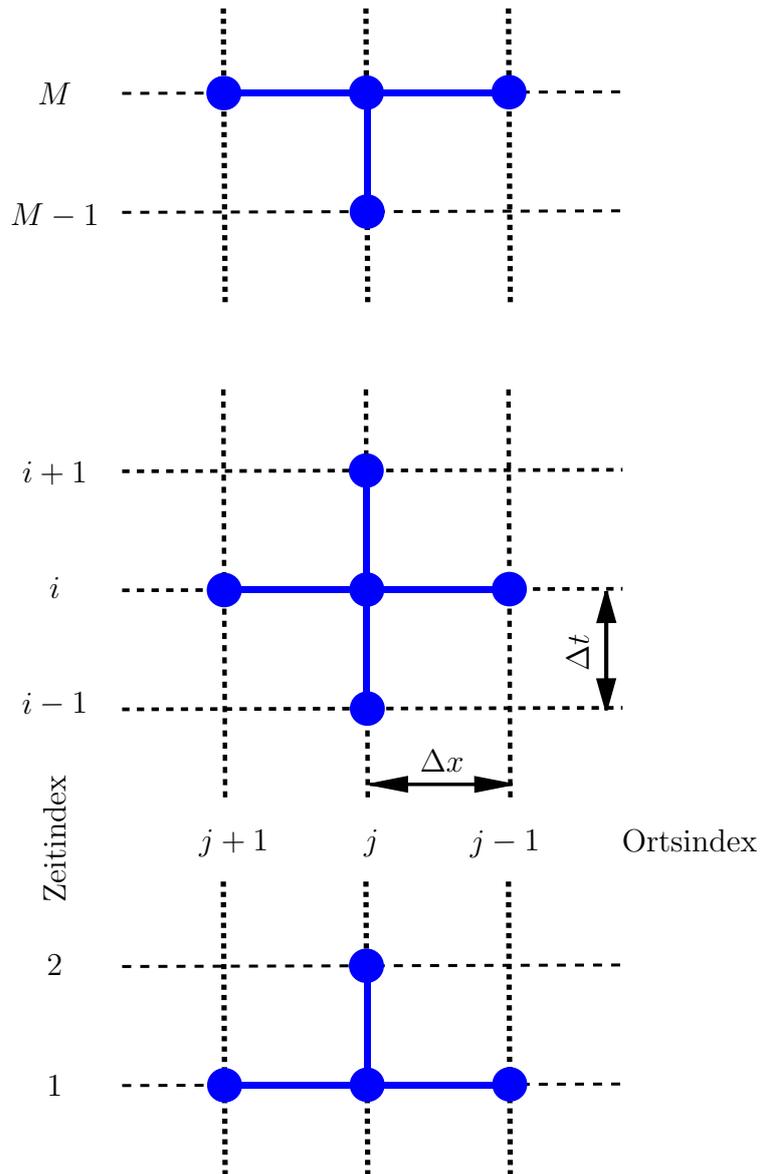


Abbildung 2.14: Berechnungsschema von Hills & Hensel, nach [2, 16].

$$\underline{\underline{D}}_j = \begin{bmatrix} 1 & 0 & & & & 0 \\ -b_j^2 \frac{\Delta r}{\lambda^2} & 1 & b_j^2 \frac{\Delta r}{\lambda^2} & & & \\ 0 & -b_j^3 \frac{\Delta r}{\lambda^3} & 1 & b_j^3 \frac{\Delta r}{\lambda^3} & & \\ & & \ddots & \ddots & \ddots & \\ & & & -b_j^{M-1} \frac{\Delta r}{\lambda^{M-1}} & 1 & b_j^{M-1} \frac{\Delta r}{\lambda^{M-1}} \\ 0 & & & & b_j^M \frac{\Delta r}{\lambda^M} & [1 + 2b_j^M \frac{\Delta r}{\lambda^M}] \end{bmatrix} \quad (2.78)$$

und

$$b_j^i = \frac{\rho c r_j \Delta r}{2r_{j+1} \Delta t} \quad (2.79)$$

Ein Vorteil der Matrix-Schreibweise ist die vergleichsweise einfache Integration eines Filters in den Algorithmus. Bei der Filterung der Ausgangsdaten ist insbesondere bei nicht-linearen Berechnungen problematisch, dass der Algorithmus die vorhandenen Schwankungen sogar noch verstärken kann. Damit können die Daten, die für den Filterprozess zur Verfügung stehen, erheblich größere Abweichungen enthalten. Die von Hills & Hensel [9] vorgestellte Methode, eine Filtermatrix in die Berechnung zu integrieren, ist eine einfache Möglichkeit dies zu verhindern. Außerdem wird dadurch die Stabilität der Ergebnisse bei Schwankungen optimiert.

Benennt man die Filtermatrix mit F , so ist diese von folgender Form:

$$\underline{\underline{F}} = \begin{bmatrix} \underline{\underline{F}}_s & 0 \\ 0 & \underline{\underline{F}}_s \end{bmatrix}. \quad (2.80)$$

Wobei die hier betrachtete Teilmatrix $\underline{\underline{F}}_s$ aus einem einfachen, mittelnden Filter besteht, deren Filterkoeffizienten folgendermaßen definiert sind:

$$a_n = \begin{cases} 0,5 & n = -m \\ 1 & n = -(m-1), \dots, m-1 \\ 0,5 & n = m \end{cases} \quad (2.81)$$

Mit m ist hierbei die Anzahl der vorherigen, bzw. folgenden Zeitschritte bezeichnet. Nachdem alle Koeffizienten ausgewertet sind, werden sie jeweils normiert, d.h. durch die Summe der Koeffizienten geteilt.

$$\Sigma a = a_0 + \sum_{n=1}^m 2a_n \quad (2.82)$$

Damit sichergestellt ist, dass die Zeilensumme der Teilmatrix immer gleich eins ist, werden die Koeffizienten am Ende und Anfang des diskretisierten Intervalls auf Knoten außerhalb des Intervalls projiziert, die eigentlich nicht existieren. Dazu werden die Koeffizienten, die zu diesen (imaginären) Knoten gehören, zu den Koeffizienten am Rand hinzuaddiert.

3 Anwendung

Um die in Kapitel 2.4 und 2.5.2 vorgestellten Lösungsverfahren auf Rohrleitungen anwenden zu können, muss die Realität hinreichend genau in einem mathematischen Modell abgebildet werden. Weiterhin wird mit dem Wissen aus der Modellbildung die in Kapitel 2.4 hergeleitete Übertragungsfunktion näher untersucht. Schließlich werden die exemplarischen Transienten definiert, mit deren Hilfe die Auswahl eines geeigneten Lösungsverfahrens ermöglicht werden soll.

3.1 Modellbildung

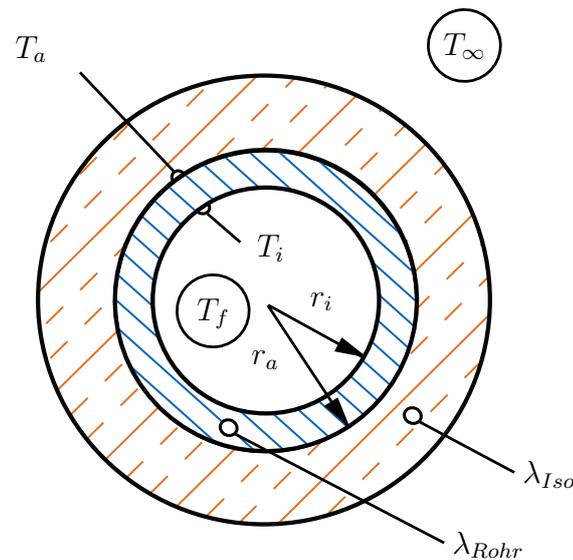


Abbildung 3.1: Schnitt durch eine Rohrleitung mit thermisch isolierter Außenoberfläche.

In Abbildung 3.1 ist eine Rohrleitung, wie sie in Kraftwerken und Anlagen zum Einsatz kommt, im Schnitt dargestellt. Die Eigenschaften des Fluides, welches die Rohrleitung mit der Temperatur T_f durchströmt, werden als homogen vorausgesetzt. Infolge der in ausreichendem Abstand vom Einlauf des Rohres ausgebildeten Strömung bildet sich zwischen dem strömenden Fluid und der Innenwand der Rohrleitung eine thermische Grenzschicht

aus. Durch den konvektiven Wärmeübergang vom Fluid auf das Rohr nimmt dies am Innenradius die Temperatur T_i an. Aufgrund von Wärmeleitung in radialer Richtung stellt sich an der Außenseite die Temperatur T_a ein. Diese ist für eine geeignete Messeinrichtung leicht zugänglich, welche auf der Rohroberfläche innerhalb des Isolationsmaterials installiert ist. Die Installationen unterscheiden sich untereinander, je nachdem welche Firma diese eingerichtet hat. Eine Beschreibung ausgewählter Installationsarten findet sich im Anlage B.

Anfangs- und Randbedingungen

Da im Inneren der Rohrleitung ein Fluid mit rotationssymmetrischem Profil und homogener Temperatur T_f strömt, kann zwischen zwei stationären Zuständen von einer im Kapitel 2.1 beschriebenen Kolbenströmungstransiente ausgegangen werden.

$$\left. \frac{dT}{d\varphi} \right|_{r=r_i} = 0 \quad (3.1)$$

Die Isolationsschicht an der Außenseite des Rohres wird in guter Näherung als wärmedicht angenommen, d.h. $\lambda_{iso} \rightarrow 0$. Woraus folgt, dass

$$\dot{q}|_{r=r_a} = 0. \quad (3.2)$$

Somit sind an der Außenoberfläche der Rohrleitung sowohl Temperatur T_a als auch Wärmefluss \dot{q}_a bekannt. Dies hat den Vorteil, dass bereits ein Messelement zur Bestimmung der äußeren Randbedingungen ausreicht und damit auch die Berechnung des direkten Temperaturfelds entfällt. Die Rohrleitung entspricht demnach dem inversen Berechnungsgebiet in Abbildung 2.11 und die Isolationsschicht dem direkten Berechnungsgebiet.

Als Anfangsbedingung kann eine Temperaturverteilung aus einem stationären Zustand angenommen werden. Zusammen mit der Adiabasie der Rohrleitung an der Außenseite kann damit eine homogene Temperaturverteilung über den Radius angenommen werden.

$$\frac{dT}{dt} = 0 \Rightarrow T(r) = \text{const.} = T_0 \quad (3.3)$$

Materialeigenschaften

Die Stoffwerte

- Wärmeleitfähigkeit λ
- Dichte ϱ
- spez. Wärmekapazität c

sind im Allgemeinen eine Funktion der Temperatur. Im Fall einer analytischen Betrachtung müssen diese als konstant, homogen und isotrop angenommen werden. Für eine konservative Abschätzung wird der Wert für die Temperaturleitfähigkeit $a = \frac{\lambda}{\rho c}$ als konstant angenommen, bei dem die größte Schädigung des Bauteils infolge der Temperaturänderung zu erwarten ist. Dies ist der Fall wenn sich Temperaturgradienten nur sehr langsam innerhalb eines Bauteils ausbreiten, d.h. die Temperaturleitfähigkeit niedrig ist. In diesem Fall sind die Zeitanteile mit hohen Temperaturdifferenzen und damit verbundenen mechanischen Spannungen größer als bei vergleichbaren Bauteilen mit hoher Temperaturleitfähigkeit. Da bei dieser Abschätzung die angenommene Ermüdung des betrachteten Bauteils höher ist als die tatsächliche, ist diese Annahme berechtigt.

Eindimensionalisierung der Problemstellung

Da die Länge der Rohrleitung L viel größer als der Radius r ist,

$$L \gg r \quad (3.4)$$

darf die Strömung innerhalb der Rohrleitung als voll ausgebildet betrachtet werden. Zusammen mit der Vernachlässigung potentieller Energie und der Adiabasie der Rohrleitung ist dadurch der Temperaturgradient in axialer Richtung dT/dz gegenüber dem radialen Temperaturgradienten dT/dr vernachlässigbar klein. Außerdem werden Kolbenströmungen vorausgesetzt, infolge derer es zu keiner Schichtung des Fluids innerhalb der Rohrleitung kommen kann. Das Problem darf demnach als rotationssymmetrisch angesehen werden ($dT/d\varphi = 0$). Weiterhin ist die Freisetzung von thermischer Energie innerhalb der Rohrwand von der Berechnung ausgeschlossen, deshalb entfällt der Quellterm $\dot{\omega}/\lambda$ ebenfalls. Damit vereinfacht sich Gleichung (2.19) zu

$$\frac{1}{a} \frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r}. \quad (3.5)$$

Mathematisches Modell

Die in Kapitel 2.2 hergeleitete Fourier'sche DGL (2.19) muss in jedem Punkt erfüllt sein. Zusammen mit den vorherigen Annahmen über die Anfangs- und Randbedingungen, Stoffwerte und Anzahl der Dimensionen ergibt sich folgendes Modell:

$$\frac{1}{a} \frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} \quad (3.6)$$

$$T(r, 0) = T_0 \quad (\text{Anfangsbedingung}) \quad (3.7)$$

$$T(r = r_a, t) = T_a \quad (\text{Randbedingung}) \quad (3.8)$$

$$\dot{q}_{r=r_a} = 0 \quad (\text{Randbedingung}) \quad (3.9)$$

3.2 Eigenschaften der Übertragungsfunktion

Um aus den gemessenen Außentemperaturen mit Hilfe der in Kapitel 2.4 hergeleiteten Übertragungsfunktion belastbare Aussagen über die Innentemperatur zu erhalten, muss deren Frequenzantwortverhalten bekannt sein. Dies ergibt sich direkt aus der Frequenzabhängigkeit.

Ist der äußere Wärmestrom \dot{q}_a gemäß Gleichung (3.2) bekannt, so vereinfacht sich Gleichung (2.33) zu

$$\hat{T}_i = \bar{r}_a [I_0(\bar{r}_i) K_1(\bar{r}_a) + K_0(\bar{r}_i) I_1(\bar{r}_a)] \hat{T}_a. \quad (3.10)$$

Für den Fall des direkten Problems weist die Übertragungsfunktion für die eindimensionale Wärmeleitung ein Tiefpassverhalten auf, d.h. die Amplituden der wellenförmigen Temperaturverläufe werden gedämpft und phasenverschoben übertragen. Dabei werden hochfrequente Temperaturwellen wesentlich stärker gedämpft als solche mit einer niedrigen Frequenz.

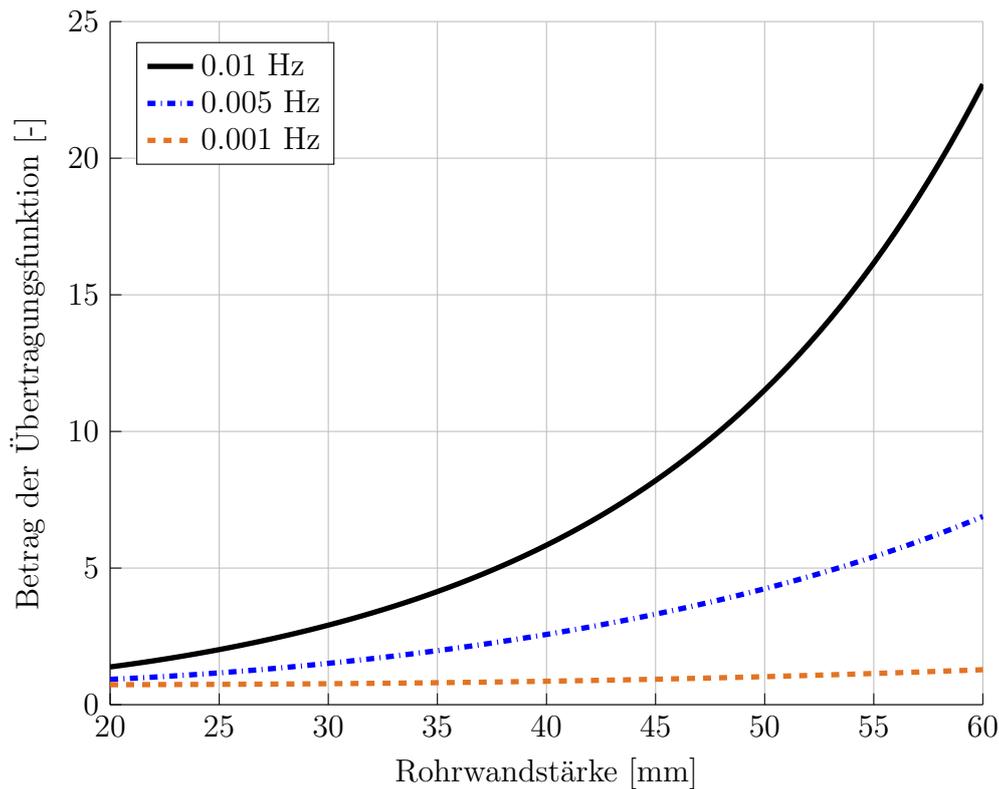


Abbildung 3.2: Betrag der Übertragungsfunktion der analytischen Lösungsmethode aufgetragen über der Wandstärke für verschiedene Frequenzen. Das Radienverhältnis r_i/r_a beträgt 0.85.

Im Fall eines inversen Problems ist deshalb mit einem Hochpassverhalten der Übertra-

gungsfunktion zu rechnen. Die Amplituden werden insbesondere von hochfrequenten Temperaturwellen verstärkt, wie in Abbildung 3.2 deutlich erkennbar ist.

Legt man die Frequenz fest und variiert das charakteristische Längenmaß der Wärmeleitung, so stellt man ein ähnliches Verhalten wie bei steigender Frequenz fest. Die Länge der Strecke, über die der Wärmetransport erfolgt, ist entscheidend für die Dämpfung im Fall eines direkten Problems. Daher verschiebt sich das Spektrum der inakzeptabel verstärkten Frequenzen für größere Wandstärken im inversen Fall hin zu kleineren Frequenzen.

3.3 Definition der Testfälle

Die in Kapitel 2.4 und Kapitel 2.5.2 vorgestellten Lösungsverfahren werden auf eine beispielhafte Rohrleitung aus warmfestem, legiertem Stahl (X10CrMoVNb9-1) mit folgenden Bauteildaten angewendet:

- Außendurchmesser $D = 504.6 \cdot 10^{-3} [m]$
- Wandstärke $s = 39.4 \cdot 10^{-3} [m]$
- Temperaturleitfähigkeit $a = 6.84 \cdot 10^{-6} [\frac{m^2}{s}]$
- Temperaturdifferenz der Transiente $\Delta T_{Trans} = 400 K$ für Sinus- und Rechteck-Impuls und $\Delta T_{Trans} = 390 K$ für den Dreieck-Impuls

Dazu werden ein Sinus-, Dreieck- und Rechteck-Impuls (siehe Abbildung 3.3) exemplarisch als zeitlicher Temperaturverlauf auf der Rohrrinnenseite aufgegeben. Die Ergebnisse der Berechnung mit dem MATLAB `pdepe`-Löser, die resultierenden Außentemperaturverläufe, dienen den verschiedenen Berechnungsverfahren als Eingangsdaten. Die Abweichung von berechnetem und ursprünglich aufgegebenem Temperaturverlauf soll die Effektivität der Verfahren zeigen. Zur besseren Vergleichbarkeit wird folgender relativer Fehler definiert:

$$\epsilon_{rel} = \frac{|\Delta T_i|_{max}}{\Delta T_{Trans}} \quad (3.11)$$

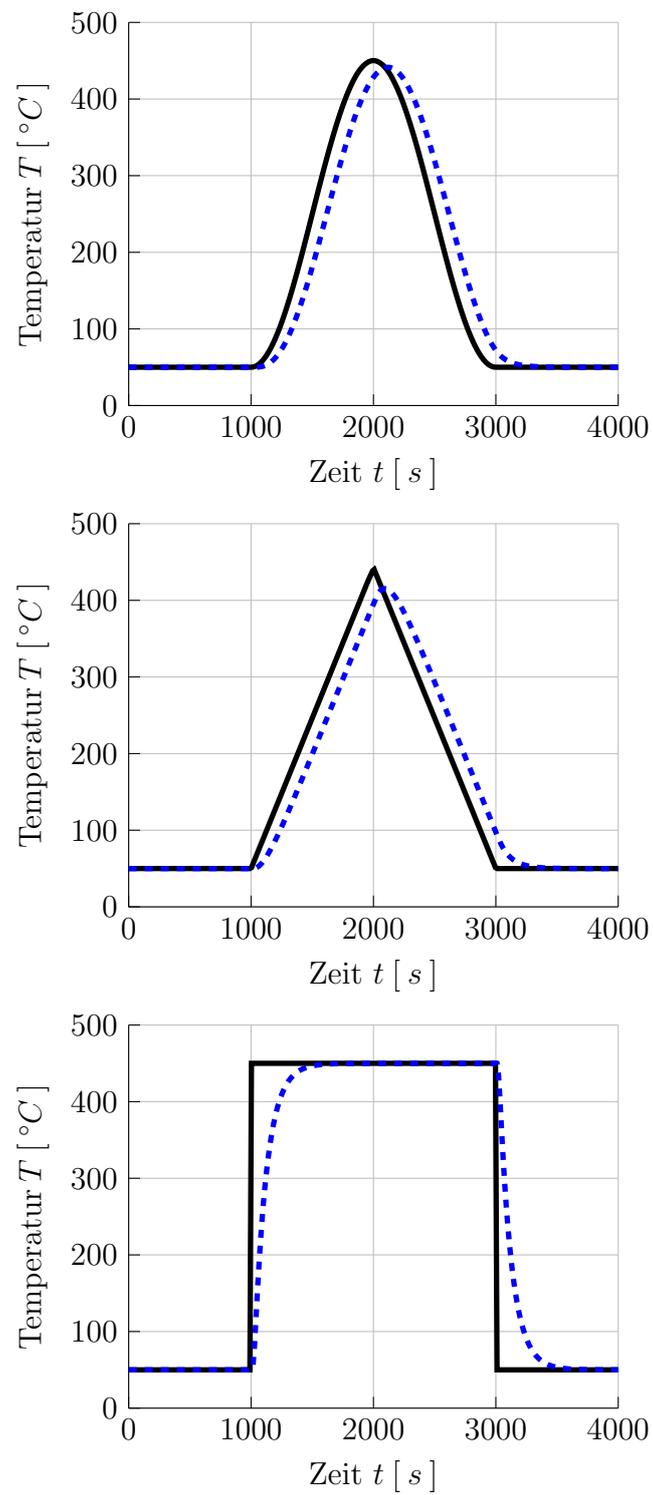


Abbildung 3.3: Definierte Testfälle, die auf eine exemplarische Rohrleitung angewendet werden. Dargestellt sind die aufgegebene Innentemperaturverlauf T_i (—) und der daraus resultierende Außentemperaturverlauf T_a (-.-.-).

4 Numerische Ergebnisse

Ergebnisse der analytischen Lösung

Wie bereits in Abbildung 3.2 dargestellt, nimmt der Betrag der Übertragungsfunktion des analytischen Lösungsverfahrens im betrachteten Bereich der Rohrwandstärke stark zu.

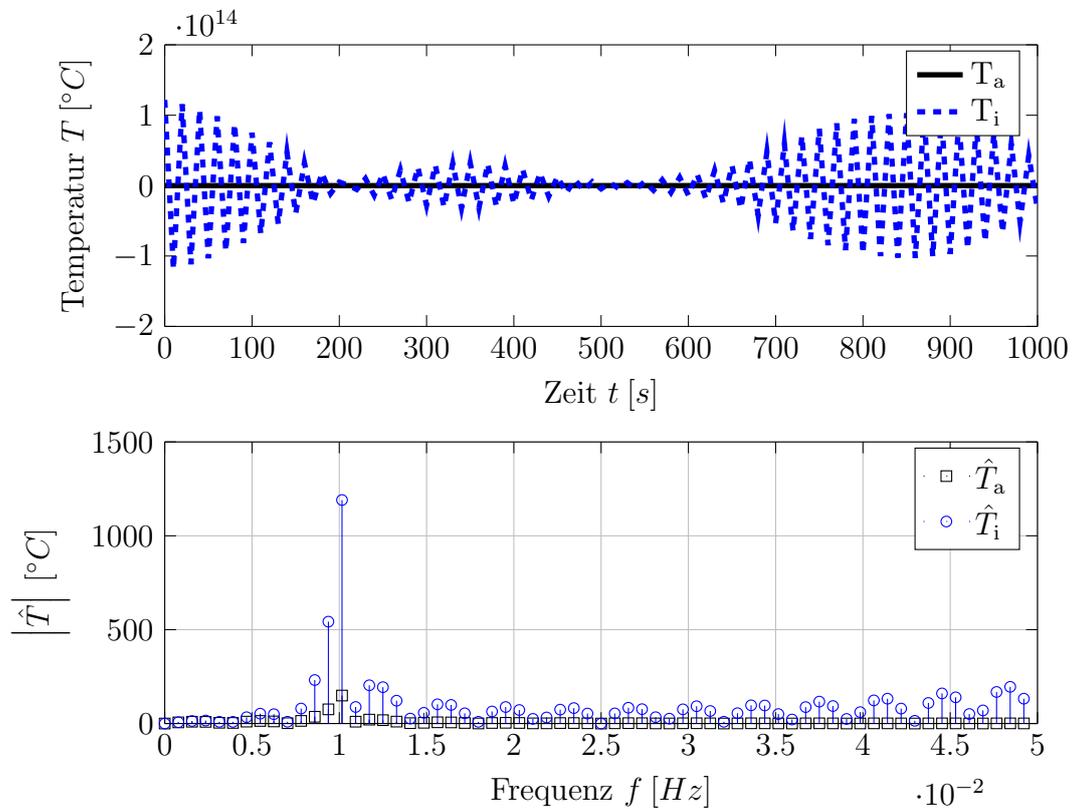


Abbildung 4.1: Ergebnisse der analytischen Lösung für einen sinusförmigen Temperaturverlauf mit einer Amplitude von 200°C und einer Frequenz von 0.01 Hz . Die Geometrie und die materialabhängigen Stoffwerte entsprechen dem Testbeispiel aus Kapitel 3.3.

Durch numerische Effekte der schnellen Fourier-Transformation sind die Fourierkoeffizienten der höheren Erregerordnungen nur sehr klein, aber nicht gleich Null. Bei der Multiplikation mit der betragsmäßig sehr großen Übertragungsfunktion werden diese Fou-

rierkoeffizienten so stark gewichtet, dass die Rücktransformation zu einem instabilen, oszillierenden Innentemperaturverlauf $T_i(t)$ führt.

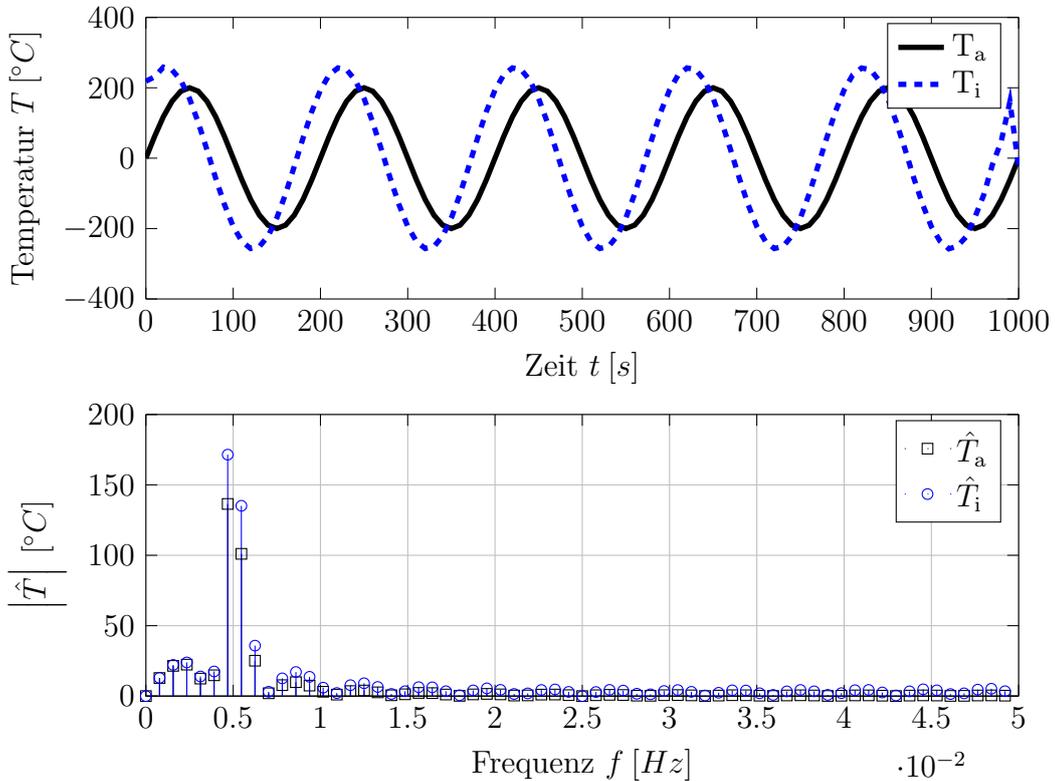


Abbildung 4.2: Ergebnisse der analytischen Lösung für einen sinusförmigen Temperaturverlauf mit einer Amplitude von $200\text{ }^\circ\text{C}$ und einer Frequenz von 0.005 Hz . Der Außendurchmesser beträgt $D = 120 \cdot 10^{-3}\text{ [m]}$, die Wandstärke $s = 20 \cdot 10^{-3}\text{ [m]}$ und die Temperaturleitfähigkeit $a = 6.84 \cdot 10^{-6}\text{ [}\frac{\text{m}^2}{\text{s}}\text{]}$.

Wie in Abbildung 4.2 dargestellt, ist es möglich eine physikalisch sinnvolle Lösung herbeizuführen, insbesondere wenn sich der Außentemperaturverlauf $T_a(t)$ durch niedrigfrequente, harmonische Schwingungen darstellen lässt. Allerdings sind diese Bedingungen in der Praxis nicht immer erfüllt, z.B. kommt es bei der Fourier-Transformation von Signalen mit konstantem Anteil zur Anregung höherer Erregerordnungen. Filtert man diese Koeffizienten aus dem Spektrum heraus, ist die Lösung zwar stabil, aber unter Umständen nicht mehr ursächlich für den gemessenen Außentemperaturverlauf. Es empfiehlt sich daher die Lösung des IHCP für eine Rohrleitung numerisch zu berechnen. Da der Anwendungsbereich dieses Lösungsverfahrens stark eingeschränkt ist und nicht im gesamten Bereich der signifikanten Schädigung durch transiente Vorgänge belastbaren Ergebnisse liefert, wird auf eine weitere Analyse der analytischen Methode im Folgenden verzichtet.

Ergebnisse der FDM-Verfahren

Berechnungsaufwand bei steigender Wandstärke

In Abbildung 4.3 ist die Stabilität der FDM-Verfahren bei vergrößertem Ortsschritt dargestellt. Da es sich um explizite Berechnungsverfahren handelt ist die Schrittweite begrenzt, wie bereits in Kapitel 2.5.2 für das FTCS-Verfahren erläutert. Während das Verfahren nach D'Souza [6] bereits ab einer Schrittweite von $\Delta r > 0.85 \text{ mm}$ instabil wird, ist die Grenze beim Verfahren von Weber [22] erst bei $\Delta r > 2.25 \text{ mm}$ erreicht. Das Verfahren von Hills & Hensel [9] bewahrt das stabile Verhalten im Bereich $\Delta r < 2.5 \text{ mm}$, wobei auch bei diesem Verfahren der relative Fehler ϵ_{rel} insgesamt mit steigender Schrittweite zunimmt.

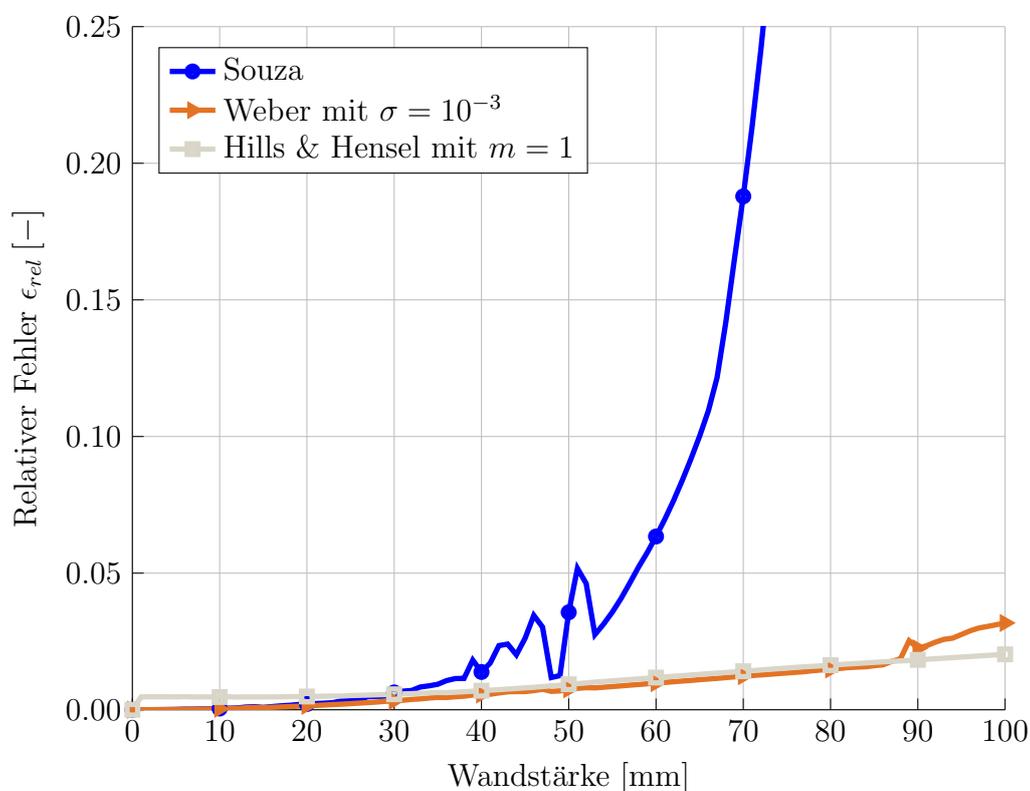


Abbildung 4.3: Relativer Fehler der FDM-Verfahren aufgetragen über der Wandstärke der Rohrleitung. Das Radienverhältnis r_i/r_a beträgt 0.85, für jedes Verfahren ist die Rohrwand mit 40 Ortsknoten diskretisiert. Es wurde der Sinus-Impuls (siehe Abbildung 3.3) rekonstruiert.

Um belastbare Ergebnisse zu erhalten, muss demnach die Ortsschrittweite beschränkt werden, d.h. das Gitter der Ortsdiskretisierung feiner gewählt werden. Dies führt bei steigenden Wandstärken, insbesondere für das FDM-Verfahren von D'Souza [6], zu erhöhtem Berechnungsaufwand.

Stabilität bei Sprüngen und Knicken

Im Folgenden wird die Stabilität der numerischen Berechnungsverfahren bei Knicken und Sprüngen im Temperaturverlauf untersucht. Dazu wird im ersten Schritt ein Dreieck-Impuls rekonstruiert, dessen Steigung sukzessive erhöht wird. Der sich ergebende relative Fehler wird in Abbildung 4.4 über der Steigung des Dreiecks aufgetragen.

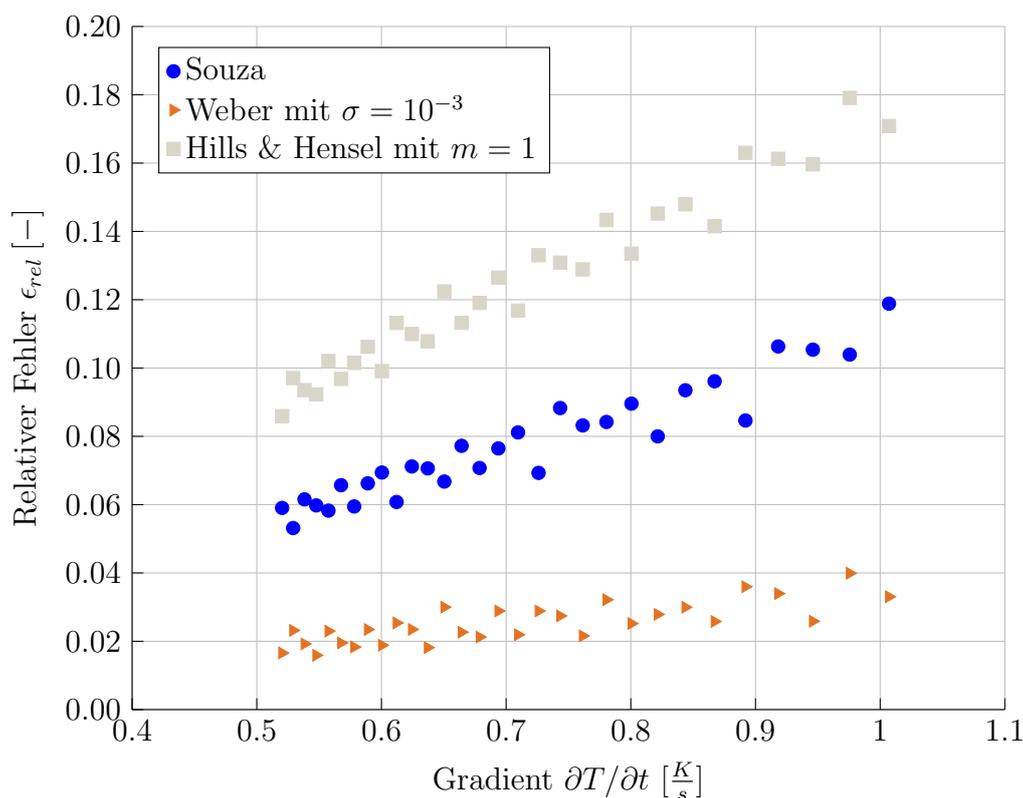


Abbildung 4.4: Dreieck-Impuls rekonstruiert durch die FDM-Verfahren. Der relative Fehler ϵ_{rel} ist über dem Temperaturgradienten des Dreieck-Impuls aufgetragen.

Beim Verfahren von D’Souza [6] lässt sich ein starker Anstieg des relativen Fehlers bei steigenden Gradienten beobachten. Da im Verfahren von Hills & Hensel [9] der integrierte Filter den Innentemperaturverlauf glättet, kann die Spitze des Dreiecks nicht genau ermittelt werden. Daher liegt der Fehler des Verfahrens in diesem Fall eine Größenordnung höher als beim Verfahren von Weber [22]. Dies zeigt, dass jede Art der Stabilisierung durch Filterung beim inversen Problem einen Verlust an Informationen bedeutet.

Es ist daher genauestens abzuwägen, ob eine Maßnahme den Gesamtfehler durch Stabilisation noch weiter senken kann oder ob der Gesamtfehler durch den Informationsverlust bereits wieder ansteigt. Dies bedeutet für den zweiten Schritt, dass im Verfahren von Hills & Hensel [9] kein Überschwingen des berechneten Temperaturverlaufs durch numerische Effekte auftritt, siehe Abbildung 4.5. Während das Überschwingen beim Verfahren von

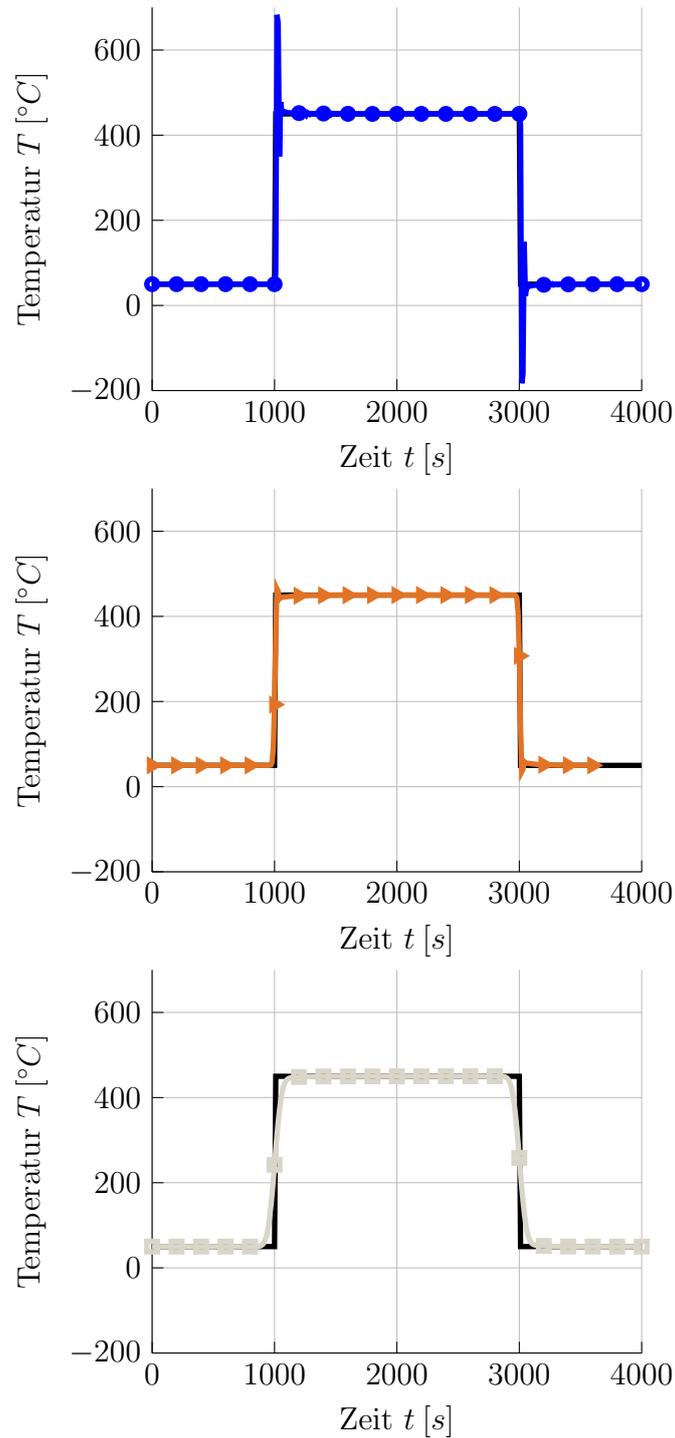


Abbildung 4.5: Rechteck-Impuls (—) (siehe Abbildung 3.3) rekonstruiert durch die vorgestellten FDM-Verfahren von Souza [6] (—●—), Weber [22] (—▶—) und Hills & Hensel [9] (—■—).

Weber [22] kurzzeitig mit geringen thermischen Amplituden erfolgt, lässt sich bei der Rekonstruktion des Innentemperaturverlaufs mit der Methode von D'Souza [6] eine sehr starke Abweichung vom ursprünglich aufgegebenen Testfall feststellen.

Anforderungen an den Filterprozess

Abbildung 4.6 zeigt die Sensitivität, mit welcher die betrachteten FDM-Verfahren auf fehlerhafte Eingangsdaten reagieren. Dazu wird zu den Eingangsdaten ein normalverteiltes Rauschen addiert, bevor der ursprünglich aufzugebene Sinus-Impuls mit Hilfe der FDM-Verfahren wiederhergestellt wird. Der relative Fehler der FDM-Verfahren wird zur besseren Vergleichbarkeit über der maximalen Messabweichung ΔT [K] aufgetragen.

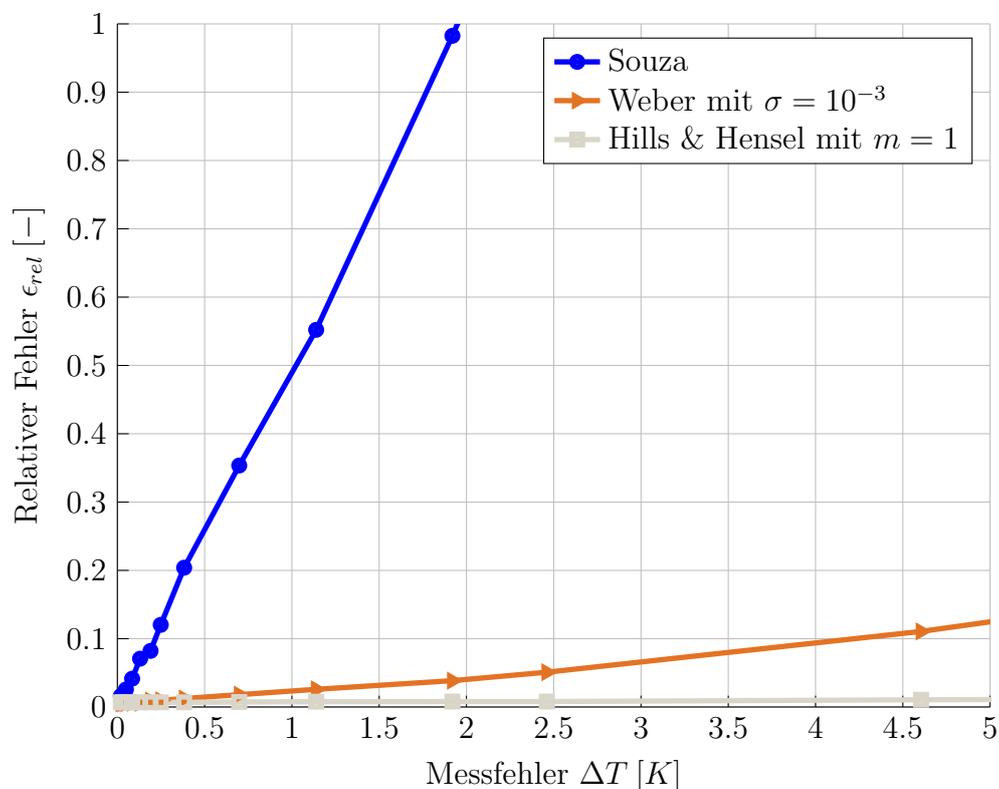


Abbildung 4.6: Einfluss eines Messfehlers ΔT auf den relativen Fehler ϵ_{rel} . Es wurde der Sinus-Impuls (siehe Abbildung 3.3) rekonstruiert.

Es fällt auf, dass bereits für geringe Abweichungen der relative Fehler der Methode nach D'Souza [6] divergiert. Auch der relative Fehler des Weber-Algorithmus steigt bei Messfehlern im Bereich $\Delta T = 5$ K signifikant an. Lediglich das Verfahren von Hills & Hensel [9] zeigt bei verrauschten Eingangsdaten durch den integrierten Filter ein stabiles Verhalten. Das Messrauschen in einzelnen Punkten wird über die umgebenen Zeitschritte gemittelt und kann sich deshalb nicht verstärkt auf den gesuchten Innentemperaturver-

lauf auswirken. Eine Plausibilisierung des gemessenen Außentemperaturverlaufs lässt sich dadurch allerdings nicht vollständig vermeiden.

Einfluss der eingeführten Parameter

Wie bereits in Kapitel 2.5.2 beschrieben, nannte Weber [22] als einzige Einschränkung für den Parameter σ , dass er sehr klein gewählt werden soll. In Abbildung 4.7 ist der relative Fehler ϵ_{rel} über σ für die Rekonstruktion eines Sinus-Impuls aufgetragen. Es ist zu beobachten, dass der relative Fehler ϵ_{rel} für $\sigma < 10^{-3}$ nicht weiter sinkt.

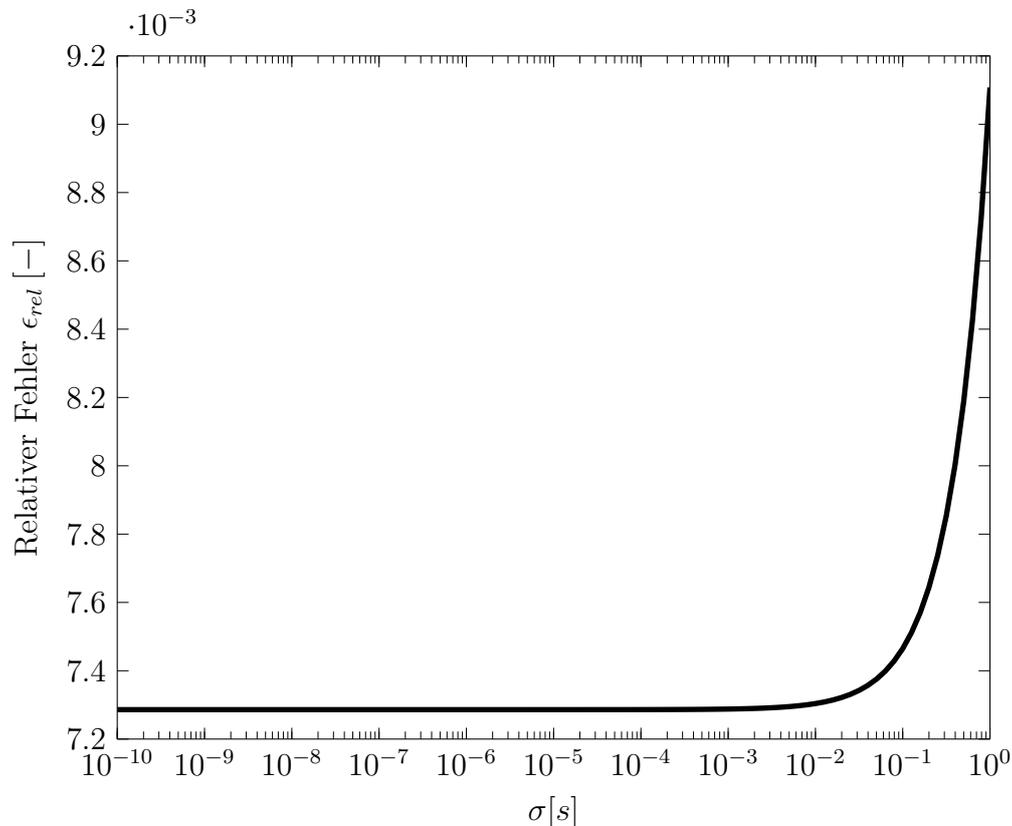


Abbildung 4.7: Einfluss des Parameters σ auf den relativen Fehler ϵ_{rel} . Wiederhergestellt wurde der Dreieck-Impuls (siehe Abbildung 3.3).

Theoretisch kann σ demnach gleich 0 gesetzt werden. Allerdings ist zu beachten, dass dann für die Berechnung der Temperatur am nächsten Ortsschritt weniger bekannte Stützstellen ausgewertet werden. Dies kann ursächlich für Instabilitäten sein, insbesondere bei ungefilterten Eingangsdaten.

Der Parameter m im Algorithmus von Hills & Hensel [9] gibt an, wie viele Stützstellen in die Gewichtung des Filterprozesses einbezogen werden. Je größer m gewählt wird, desto stabiler sind die resultierenden Ergebnisse. Die Mittelwertbildung bedingt auch einen Fehler, da Knicke „verrundet“ werden.

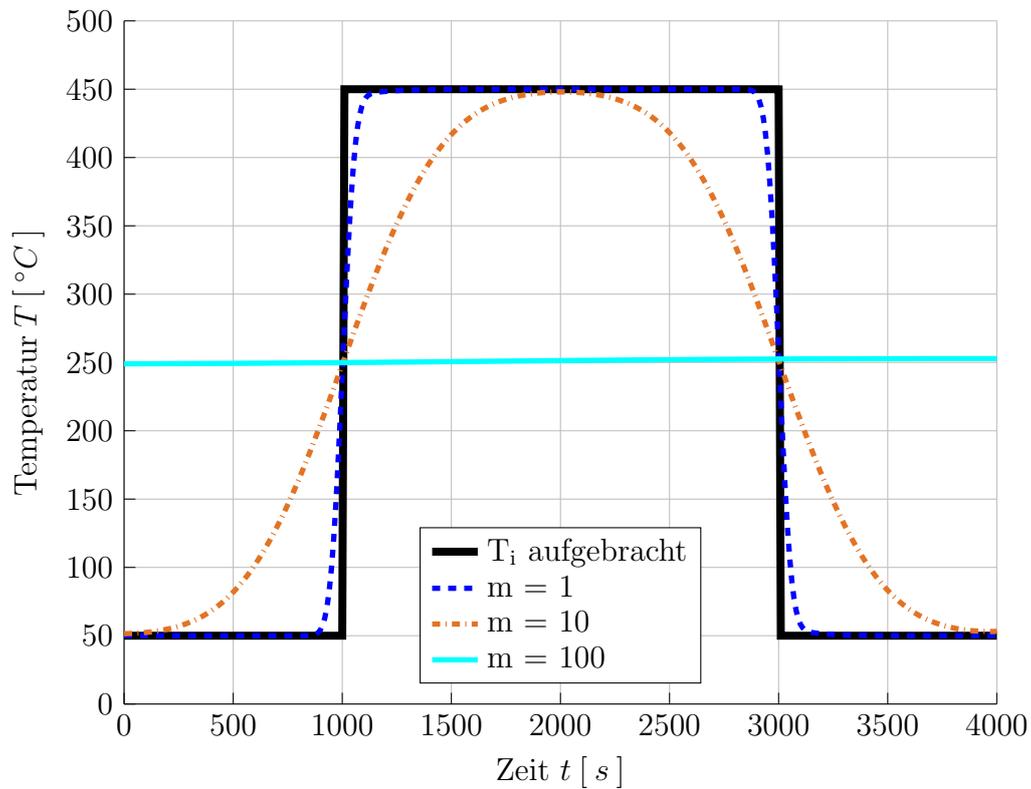


Abbildung 4.8: Einfluss des Parameters m auf den rekonstruierten Temperaturverlauf des Rechteck-Impuls (siehe Abbildung 3.3).

Dies wird bei der Rekonstruktion eines Rechteck-Impulses in Abbildung 4.8 besonders deutlich. Wird m zu groß gewählt, führt dies zu einem erheblichen Informationsverlust. In Abbildung 4.8 ist zu sehen, dass für $m = 100$ bereits der gesamte transiente Vorgang auf einen konstanten Temperaturverlauf abgebildet wird.

5 Fazit

In dieser Arbeit wurden FDM-Berechnungsverfahren und eine analytische Lösungsmethode mit einer Übertragungsfunktion auf das instationäre, inverse Wärmeleitungsproblem für eine Rohrgeometrie angewendet. Während die analytische Lösung auf Anwendungen mit begrenzter Wandstärke und niederfrequenten, harmonischen Temperaturverläufen beschränkt ist, sind die FDM-Verfahren auf ein breites Spektrum von Problemstellungen anwendbar.

Die Methode nach Hills & Hensel [9] mit einem integrierten Filterprozess hat sich dabei als besonders stabil erwiesen. Messfehler werden nicht verstärkt bis zur Innenoberfläche weitergegeben und zusätzlich entstehen keine numerischen Artefakte, d.h. Temperaturverläufe, die aus der numerischen Berechnung resultieren und nicht physikalischen Ursprungs sind. Außerdem ist das Verfahren in jedem Software-Paket handzuhaben, welches Berechnungen mit Vektoren und Matrizen beherrscht.

Damit ist es möglich aus gemessenen Außentemperaturen den Innentemperaturverlauf zu bestimmen und bestehende technische Regelwerke anzuwenden. So kann eine genaue Ermüdungsbewertung von Rohrleitungsanlagen in konventionellen Kraftwerken durchgeführt werden. Die Kraftwerksbetreiber und technischen Prüfgesellschaften können auf diese Weise die Sicherheit bestehender Anlagen auch bei veränderten Betriebszuständen garantieren.

Im nächsten Schritt ist es sinnvoll die Berechnung des inversen Temperaturfeldes auf mehrere Raumdimensionen zu erweitern, um so auch Lastfälle mit einer Schichtung des Fluides innerhalb der Rohrleitung abbilden zu können. Außerdem kann das Verfahren von Hills & Hensel [9] auf nichtlineare Problemstellungen angewendet werden. Dadurch kann eine weitere Abschätzung durch exakte Berechnungsergebnisse ersetzt werden.

Anlagen

A Lagrange-Interpolation

Siehe auch [7],[12] und [21].

Wird ein Polynom dadurch bestimmt, dass es eine stetige Verbindung von definierten Punkten, sogenannten Stützstellen, darstellt, so spricht man von *Polynominterpolation*. Für den einfachsten Fall nimmt man ein Polynom 1. Grades an, welches durch die bekannten Stützstellen (x_0, y_0) und (x_1, y_1) geht, siehe Abbildung A.1.

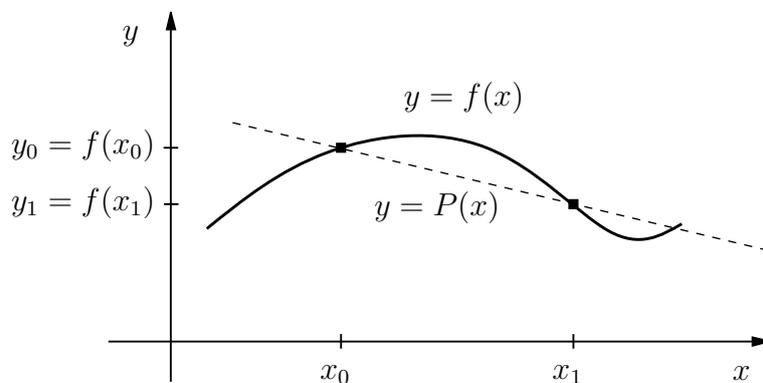


Abbildung A.1: Lagrangepolynom für zwei gegebene Stützstellen, nach [7, 21].

Dafür wird folgendes lineares Polynom angenommen:

$$P(x) = \underbrace{\frac{x - x_1}{x_0 - x_1}}_{=L_{10}(x)} f(x_0) + \underbrace{\frac{x - x_0}{x_1 - x_0}}_{=L_{11}(x)} f(x_1). \quad (\text{A.1})$$

Es lässt sich leicht zeigen, z.B. durch einsetzen, dass $P(x = x_0) = f(x_0)$ und $P(x = x_1) = f(x_1)$. Hierbei gilt für die Lagrange'schen Teilpolynome, dass $L_{10}(x_0) = 1$, $L_{11}(x_0) = 0$, $L_{10}(x_1) = 0$ und $L_{11}(x_1) = 1$.

Im Allgemeinen ist für die Interpolation von $n + 1$ Stützstellen ein Polynom vom Grad n zu bestimmen. An die Lagrange'schen Teilpolynome $L_{ni}(x)$ sind dabei die folgenden Bedingungen gestellt:

$$L_{ni}(x_j) = 0 \text{ für } j \neq i \text{ und } i = 0, 1, \dots, n, \quad (\text{A.2})$$

$$L_{ni}(x_i) = 1 \text{ für } i = 0, 1, \dots, n. \quad (\text{A.3})$$

Erfüllt werden diese von

$$L_{ni}(x) = \frac{(x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)}, \quad (\text{A.4})$$

wobei der konstante Faktor im Nenner sicherstellt, dass $L_{ni}(x_i) = 1$ ist und das Polynom vom Grad n im Zähler dafür sorgt, dass $L_{ni}(x_j) = 0$ ist. Das Lagrange'sche Interpolationspolynom für $n + 1$ gegebene Stützstellen ergibt sich damit zu

$$P_n(x) = \sum_{i=0}^n f(x_i) L_{ni}(x) = \sum_{i=0}^n f(x_i) \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j}. \quad (\text{A.5})$$

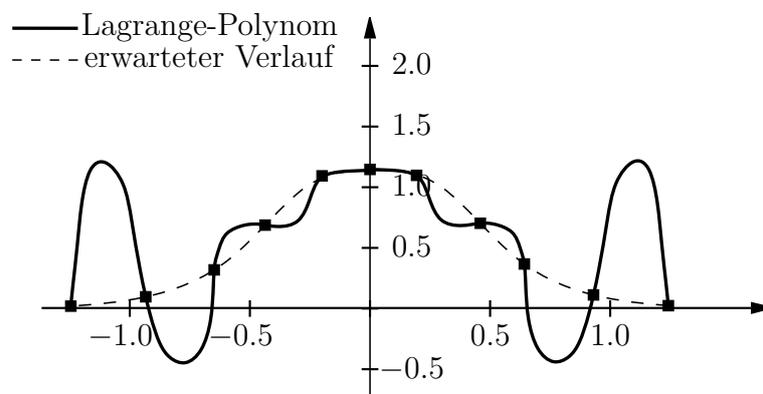


Abbildung A.2: Oszillationen bei Lagrange-Interpolation mit mehreren Stützstellen, nach [7, 21].

Wenn die Anzahl der Stützstellen zunimmt, wird es bei der Lagrange-Interpolation notwendig die stetige Funktion durch Polynome hohen Grades zu interpolieren. Dadurch kann es, wie in Abbildung A.2 zu sehen, zu ungewollten Oszillationen kommen. Dies lässt sich vermeiden, indem die Anzahl der Stützstellen an den Rändern des Intervalls erhöht wird.

B Messtechnik

Wie von Hofstötter [10] genauer betrachtet, werden zur Erschöpfungsüberwachung in Kraftwerken folgende Messinstallationsarten eingesetzt:

- FAMOS (alt)
- FAMOS (neu)
- AMTEC
- TÜV Schellen
- TÜV Polster.

Im Folgenden werden diese Installationsarten kurz erklärt.

FAMOS (alt)

Bei der Messinstallation FAMOS (alt) sind die Thermoelemente durch kleine Schellen auf einem Spannband befestigt, sodass die Köpfe der Elemente überkragen. Diese werden von einer weiteren Schelle auf der Rohroberfläche fixiert. Um den Sensor zu schützen wird um die Installation eine massive Schelle mit Abstandsbolzen gelegt. Siehe Abbildung B.1.

FAMOS (neu)

Wie Abbildung B.2 zeigt, sind bei einer Installation von FAMOS (neu) die Thermoelemente mit kleinen Schellen auf einem breiten Spannband befestigt. Die Köpfe sind mit einer Folie auf der Außenseite des Spannbands befestigt und kragen daher nicht über. Diese Installation wird ebenfalls durch eine massive Schelle geschützt.

AMTEC

Bei einer Installation von AMTEC wird der Kopf eines Thermoelements mit Hilfe eines Polsters und zwei Spannbandern auf der Rohroberfläche fixiert. Außerdem besteht die Anordnung aus einer Positionierhilfe und einer Zugentlastung mittels zweier Spannbander für das Rohr mit dem die Thermoelemente aus der Isolation hinausgeführt werden. Zu sehen ist dies in Abbildung B.3.

TÜV Schellen

Wie in Abbildung B.4 zu sehen, werden in diesem Fall die Thermoelementköpfe direkt auf die Rohroberfläche aufgeschweißt.

TÜV Polster

Diese Installation besteht aus einem Thermoelement, welches unter dem Spannband befestigt ist und mit einem Glasfaserpolster gegen das Spannband isoliert ist. In Abbildung B.5 ist eine solche Anordnung dargestellt.

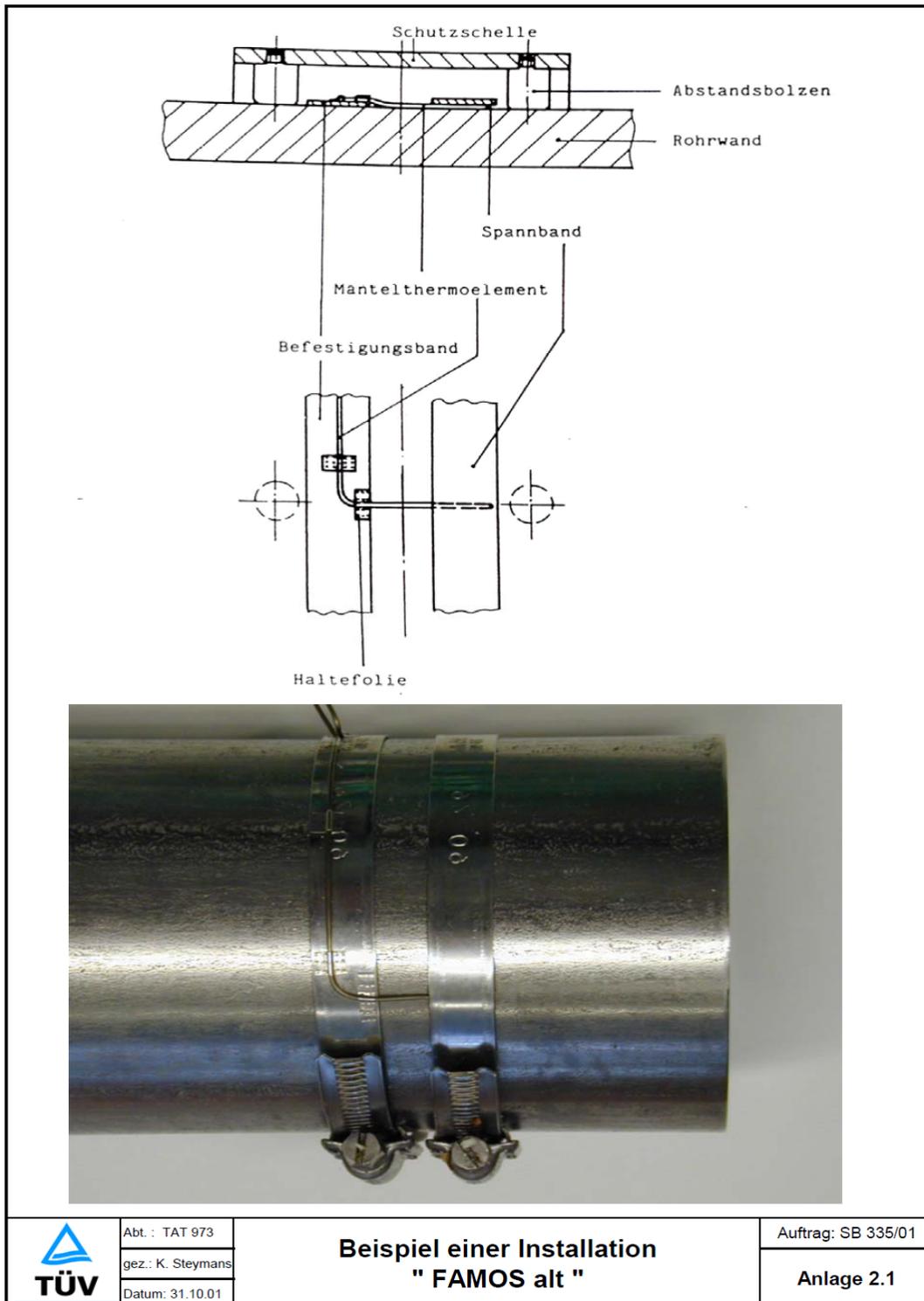


Abbildung B.1: Bild einer Messinstallation FAMOS (alt), übernommen aus [10].

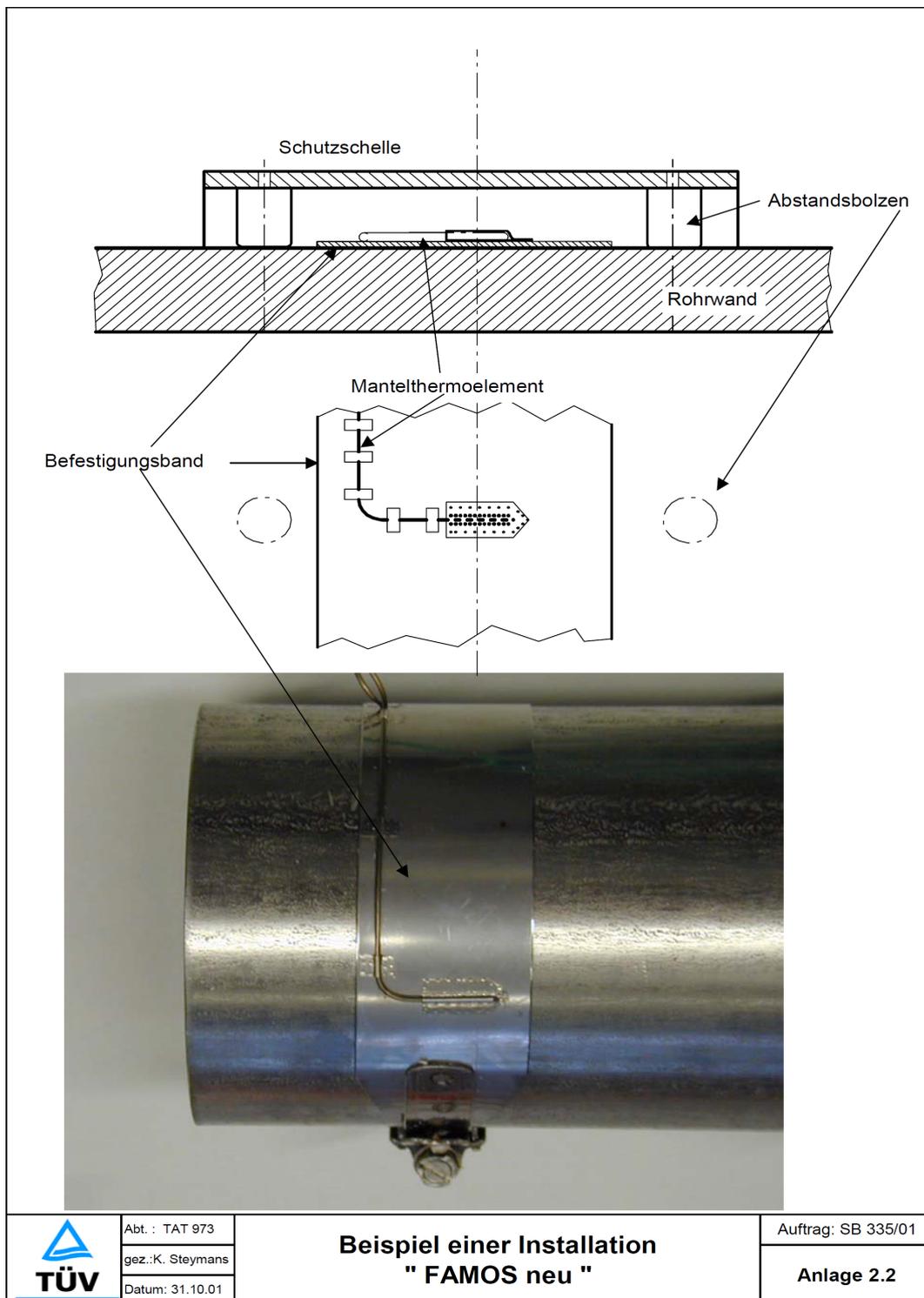
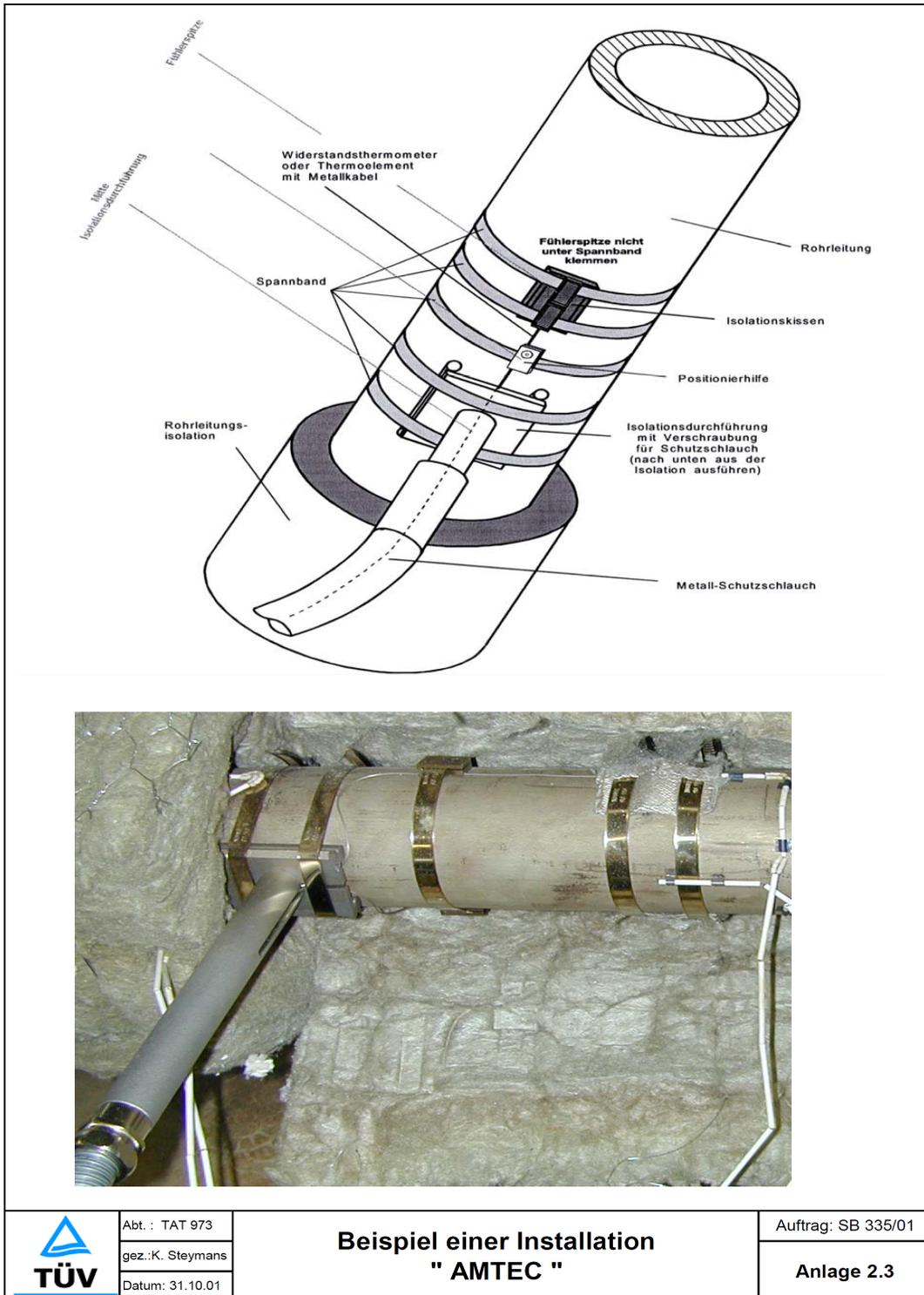


Abbildung B.2: Bild einer Messinstallation FAMOS (neu), übernommen aus [10].



| | | | |
|---|-------------------------------------|--|--------------------|
|  | Abt. : TAT 973 | Beispiel einer Installation " AMTEC " | Auftrag: SB 335/01 |
| | gez.:K. Steymans Datum: 31.10.01 | | Anlage 2.3 |

Abbildung B.3: Bild einer Messinstallation AMTEC, übernommen aus [10].

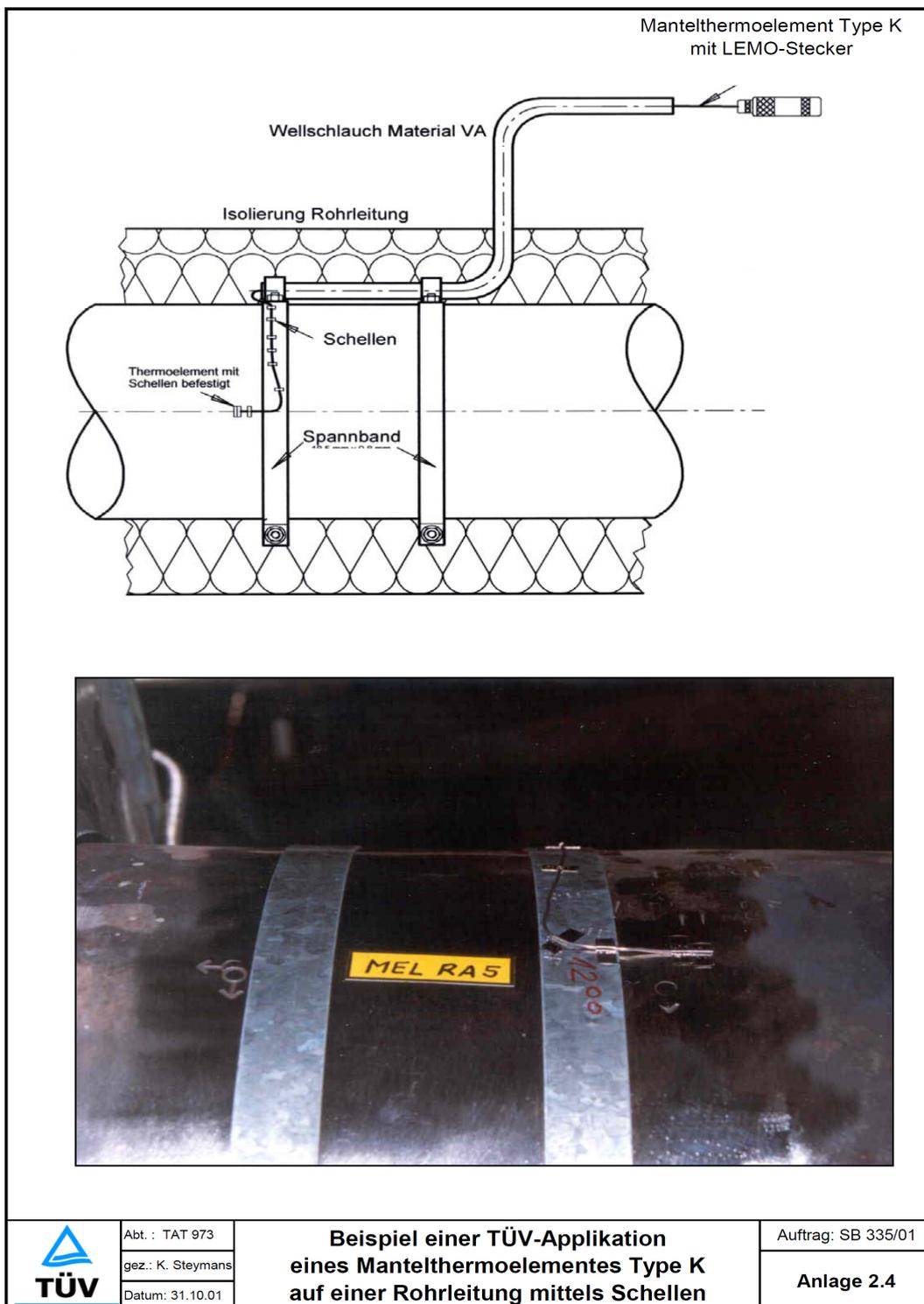


Abbildung B.4: Bild einer Messinstallation TÜV Schelle, übernommen aus [10].

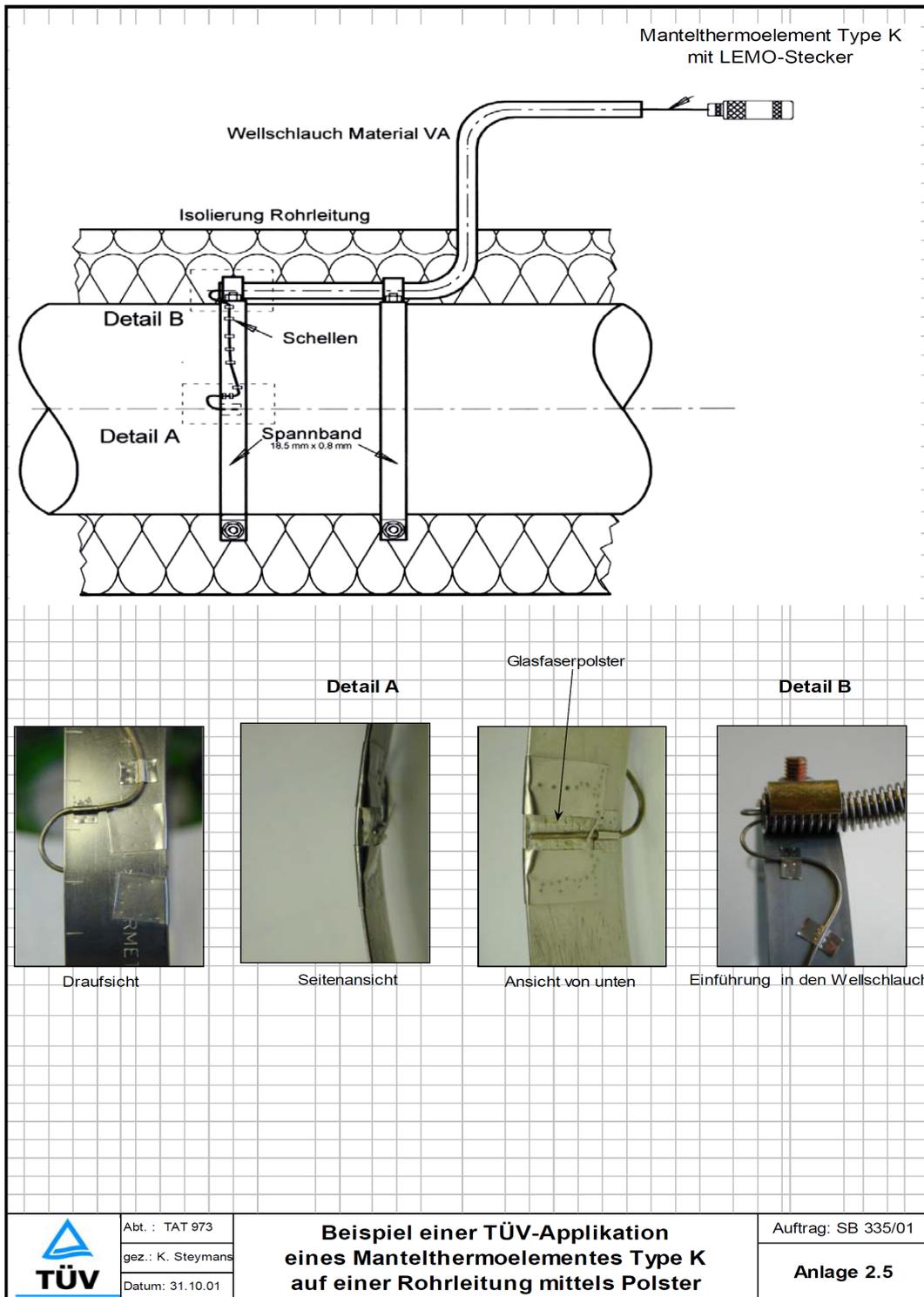


Abbildung B.5: Bild einer Messinstallation TÜV Polster, übernommen aus [10].

C MATLAB-Codes

Skript zur Erstellung von Abbildung 2.7

```
1 %% Konvergenz der FD-Verfahren
2
3 % plottet das Konvergenzverhalten den Zweipunkte-Mittelpunkt-Formel und der
4 % Zweipunkte-Formel fuer Kapitel 2.5.2
5
6 clc; %Vorbereiten der Arbeitsumgebung
7 clear all;
8 close all;
9 format long; %Rechnung mit 64 Bit Genauigkeit
10
11 %% Initialisieren
12 n=500; %Anzahl der Stuetzstellen
13 f=@(x) (x./(x+1)).^5; %Beispielfunktion
14 dfdx=@(x) (5*x.^4)/((x+1).^6); %Ableitung der Beispielfunktion
15 x0=0.6; %auszuwertende Stelle
16 h=logspace(0,-9,n); %logarithmischer Vektor der Laenge n
17 e=zeros(length(h),1); %Vektor fuer Fehlereintraege
18
19 %% Plot
20 figure(1)
21 loglog(h,h,'k-', 'LineWidth',2); %Plotten der Konvergenzordnung der 2PF
22 hold on;
23 for i=1:n %Berechnen des Fehlers der Zweipunkte Formel
24 e(i)=abs(dfdx(x0)-(1/h(i))*(f(x0)-f(x0-h(i)))));
25 end
26 loglog(h,e,'k-', 'LineWidth',2); %Plotten des Fehlers der Zweipunkte Formel
27 hold on;
28 loglog(h,h.^2,'b-', 'LineWidth',2); %Plotten der Konvergenzordnung der 3PM
29 hold on;
30 for i=1:n %Berechnen des Fehlers der Dreipunkte-Mittelpunkt-Formel
31 e(i)=abs(dfdx(x0)-(1/(2*h(i)))*(f(x0+h(i))-f(x0-h(i)))));
32 end
33 loglog(h,e,'b-', 'LineWidth',2); %Plotten des Fehlers der Dreipunkte-Mittelpunkt-Formel
34 hold on;
35 axis([1e-9,1e0,1e-20,1e5]);
36 xlabel('h [m]');
37 ylabel('Relativer Fehler der finiten Differenzen');
38 legend('h', 'Zweipunkte-Formel', 'h^2', 'Dreipunkte-Mittelpunkt-Formel', 'Location', 'NW');
```

Skript zur Erstellung von Abbildung 3.2

```
1 %% Frequenzabhaengigkeit des Betrags der Uebertragungsfunktion (TF)
2 % berechnet den Betrag der Uebertragungsfunktion fuer ein Rohr
3
4 clc; %Vorbereiten der Arbeitsumgebung
5 clear all;
6 close all;
7
8 %% Initialisieren
9 a = 6.83; %Temperaturleitfaehigkeit in mm^2/s
10
11 f = [0.001,0.005,0.01]; %Frequenzvektor
12 dr = linspace(20,60,81); %Laengenvektor
13 qr = [0.75 0.85 0.95]; %Radienverhaeltisvektor
14
15 TF = zeros(length(dr),length(f)); %Uebertragungsfunktion
16
17 %% Berechnen und Plotten
18
19 k = sqrt(2*pi*f./a); %Koeffizient -> Carslaw/Jaeger
20
```

```

21 for i = 1:length(f) %Schleife ueber alle Frequenzen
22     r_a = dr./(1-qr(2)); %Berechnung des Aussenradius
23     r_i = qr(2).*r_a; %Berechnung des Innenradius
24
25     ra_c = r_a*sqrt(1i).*k(i); %Komplexe Argumente fuer Besselfunktionen
26     ri_c = r_i*sqrt(1i).*k(i); %Komplexe Argumente fuer Besselfunktionen
27
28     B0i = besseli(0,ri_c); %Auswertung der Besselfunktionen
29     B0k = besselk(0,ri_c);
30     B1i = besseli(1,ra_c);
31     B1k = besselk(1,ra_c);
32
33     D = (B0i.*B1k + B0k.*B1i).*ri_c; %Matrixeintrag der Uebertragungsfkt.
34     G = qr(2).*D; %Eintrag der inversen Matrix
35     TF(:,i) = G; %Speichern in Uebertragungsfunktion
36
37 end
38 %% Plotten
39
40 figure()
41 hold on;
42 semilogy(dr,abs(TF(:,3)),'-k','LineWidth',2);
43 semilogy(dr,abs(TF(:,2)),'-b','LineWidth',2);
44 semilogy(dr,abs(TF(:,1)),'--','Color',[0.89 0.45 0.14],'LineWidth',2);
45
46 legend('0.01 Hz','0.005 Hz','0.001 Hz','Location','NorthWest')
47 grid on;
48 xlabel('Rohrwandst\''arke [mm]')
49 ylabel('Betrag der \''Ubertragungsfunktion')

```

Funktionen zur Berechnung der in Kapitel 3.3 definierten Testfälle

Sinus-Impuls

```

1 function [ T ] = pdepeRohrSin( s, t )
2 %pdepeRohrSin berechnet das direkte Problem als Eingangsdaten fuer die inversen
3 %Loesungsalgorithmen
4 % Als Innentemperaturverlauf wurde ein Sinus-Impuls gewaehlt
5
6 %% Input:
7 % s Radienvektor
8 % t Zeitvektor
9
10 %% Output:
11 % T [length(t) x length(s)] Temperaturfeld in globaler Diskretisierung
12
13 %% Berechnen des direkten Problems
14 m = 1; %Konstante fuer Platte, Zylinder, Kugel
15 sol = pdepe(m,@pdepde,@pdeic,@pdebc,s,t); %Loesungsarray berechnen
16 T = sol(:, :, 1); %Temperaturfeld auslesen
17 end
18
19 function [c,f,s] = pdepde(~,~,~,DuDx) %Definieren der PDGL
20 lambda = 0.033; %Waermeleitfaehigkeit in W/(mmK)
21 rho = 7.76e-6; %Dichte in kg/mm^3
22 cv = 622; %spez. Waermekapazitaet in J/(kgK)
23 c = rho*cv; %Koeffizienten fuer Fourier'sche DGL
24 f = lambda*DuDx;
25 s = 0;
26 end
27
28 function u0 = pdeic(s) %Definieren der Anfangsbedingungen
29 a = 50; %Anfangstemperatur in Grad Celsius
30 u0 = ones(1,length(s)).*a; %Vektor mit konstanter Temperaturverteilung
31 end
32
33 function [pl,ql,pr,qr] = pdebc(~,ul,~,~,t) %Definieren der Randbedingungen
34 D = 1000; %Zeitverzug
35 T = 2000; %Periodendauer
36 A = 200; %Amplitude
37
38 if t>D && t<(D+T) %Auswahl des Zeitbereichs mit Impuls-Funktion
39     pl = ul-250+A*sin(2*pi/T*(t-500));
40 else
41     pl = ul-50; %Temperatur des Vor- und Nachlaufs
42 end

```

```

43 q1 = 0; %Koeffizienten zum Festlegen
44 pr = 0; % der uebrigen Randbedingungen
45 qr = 1;
46 end

```

Dreiecks-Impuls

```

1 function [ T ] = pdepeRohrTriang( s , t )
2 %pdepeRohrTriang berechnet das direkte Problem als Eingangsdaten fuer die inversen
3 %Loesungsalgorithmen
4 % Als Innentemperaturverlauf wurde ein Dreiecks-Impuls gewaehlt
5
6 %% Input:
7 % s Radienvektor
8 % t Zeitvektor
9
10 %% Output:
11 % T [30x40] Temperaturfeld in globaler Diskretisierung
12
13 %% Berechnen des direkten Problems
14 m = 1; %Konstante fuer Platte , Zylinder , Kugel
15 sol = pdepe(m,@pdepde,@pdeic,@pdebc ,s ,t); %Loesungsarray berechnen
16 T = sol (: ,: ,1); %Temperaturfeld auslesen
17 end
18
19 function [c,f,s] = pdepde(~ ,~ ,~ ,DuDx) %Definieren der PDGL
20 lambda = 0.033; %Waermeleitfaehigkeit in W/(mmK)
21 rho = 7.76e-6; %Dichte in kg/mm^3
22 cv = 622; %spez. Waermekapazitaet in J/(kgK)
23 c = rho*cv; %Koeffizienten fuer Fourier 'sche DGL
24 f = lambda*DuDx;
25 s = 0;
26 end
27
28 function u0 = pdeic(s) %Definieren der Anfangsbedingungen
29 a = 50; %Anfangstemperatur in Grad Celsius
30 u0 = ones(1,length(s)).*a; %Vektor mit konstanter Temperaturverteilung
31 end
32
33 function [pl,q1,pr,qr] = pdebc(~ ,ul,~ ,~ ,t) %Definieren der Randbedingungen
34 D = 1000; %Zeitverzug
35 T = 2000; %Periodendauer
36 A = 250; %Amplitude
37
38 if t>D && t<(D+T) %Auswahl des Zeitbereichs mit Impuls-Funktion
39 pl = ul-50-abs(A*asin(sin(pi/T*(t-D))));
40 else
41 pl = ul-50; %Temperatur des Vor- und Nachlaufs
42 end
43 q1 = 0; %Koeffizienten zum Festlegen
44 pr = 0; % der uebrigen Randbedingungen
45 qr = 1;
46 end

```

Rechtecks-Impuls

```

1 function [ T ] = pdepeRohrRectwin( s , t )
2 %pdepeRohrRectwin berechnet das direkte Problem als Eingangsdaten fuer die inversen
3 %Loesungsalgorithmen
4 % Als Innentemperaturverlauf wurde ein Rechtecks-Impuls gewaehlt
5
6 %% Input:
7 % s Radienvektor
8 % t Zeitvektor
9
10 %% Output:
11 % T [30x40] Temperaturfeld in globaler Diskretisierung
12
13 %% Berechnen des direkten Problems
14 m = 1; %Konstante fuer Platte , Zylinder , Kugel
15 sol = pdepe(m,@pdepde,@pdeic,@pdebc ,s ,t); %Loesungsarray berechnen
16 T = sol (: ,: ,1); %Temperaturfeld auslesen
17 end
18
19 function [c,f,s] = pdepde(~ ,~ ,~ ,DuDx) %Definieren der PDGL
20 lambda = 0.033; %Waermeleitfaehigkeit in W/(mmK)

```

```

22 rho = 7.76e-6; %Dichte in kg/mm^3
23 cv = 622; %spez. Waermekapazitaet in J/(kgK)
24 c = rho*cv; %Koeffizienten fuer Fourier 'sche DGL
25 f = lambda*DuDx;
26 s = 0;
27 end
28
29 function u0 = pdeic(s) %Definieren der Anfangsbedingungen
30 a = 50; %Anfangstemperatur in Grad Celsius
31 u0 = ones(1,length(s)).*a; %Vektor mit konstanter Temperaturverteilung
32 end
33
34 function [pl,ql,pr,qr] = pdebc(~,ul,~,~,t) %Definieren der Randbedingungen
35 pl = ul-50 -400*((t>1000)*(t-1000)-(t>1001)*(t-1001))...
36 +400*((t>3000)*(t-3000)-(t>3001)*(t-3001));
37 %Aufgabe des Rechtecks-Impuls
38 ql = 0; %Koeffizienten zum Festlegen
39 pr = 0; % der uebrigen Randbedingungen
40 qr = 1;
41 end

```

Skript zur Erstellung von Abbildung 4.1

```

1 %% TestFourierAnsatzSinus
2
3 % Anwendung der Uebertragungsfunktionen nach Carslaw/Jaeger
4
5 clear; %Vorbereiten der Arbeitsumgebung
6 clc;
7 close all;
8
9 %% Initialisieren
10
11 fs = 0.1; %Abtastfrequenz in [Hz]
12 t = 0:1/fs:1000; %Zeitintervall in [s]
13
14 %% Aussentemperatur-Signal erzeugen
15
16 A = 200; %Amplitude
17 freq = 0.01; %Frequenz
18
19 Ta = A*sin(2*pi*freq*t); %Signal
20
21 %% Darstellung des zeitlichen Temperaturverlaufs
22
23 figure()
24 grid on;
25 subplot(2,1,1);
26 plot(t,Ta,'-k','Linewidth',2);
27 xlabel('Zeit $t$ [s]$ ');
28 ylabel('Temperatur $T\text{[}^\circ\text{C]}$ ');
29
30
31 %% Mittels FFT in den Frequenzbereich transformieren
32
33 nfft = 2^nextpow2(length(Ta)); %Ermitteln der Zweierpotenz fuer FFT
34 X = fft(Ta,nfft); %Mittels FFT in den Frequenzbereich transformieren
35
36 n_pos = nfft/2 + 1; %%diskrete Frequenzstellen im pos. Bereich
37 Ts = diff(t(1:2)); %Abtastzeit berechnen
38 fl = [0:floor((nfft-1)/2)] / (nfft*Ts); %Vektor mit Frequenzen im
39 % positiven Spektrum erzeugen
40
41 Ta_pos0 = abs(X(1:n_pos)); %Bilden des Betrags des positiven Spektrums
42 magTa = [Ta_pos0(1)/nfft Ta_pos0(2:n_pos-1)/(nfft/2) ]; %Skalierung des
43 % Spektrums
44
45 %% Darstellung des Spektrums
46
47 subplot(2,1,2);
48 stem(fl, magTa, '-k', 'Marker', 's' );
49 xlabel('Frequenz $f$ [Hz]$ ');
50 ylabel('T');
51
52 %% Frequenzvektor fuer die frequenzabhaengige Uebertragungsfunktion bestimmen
53
54 df = fs/nfft; %Aufloesung des Spektrums berechnen
55 f = -fs/2:df:fs/2-df; %Frequenzvektor erstellen
56 f = ifftshift(f); %Umordnung der Frequenzen, damit die Reihenfolge
57 % zur Ausgabe der FFT passt
58

```

```

59 %% Uebertragungsfunktion
60
61 a = 6.839e-6;           %Temperaturleitfaehigkeit in m^2/s
62 ra = 0.252;           %Aussenradius in m
63 sb = 0.0395;         %Wandstaerke des Rohrs in m
64
65 ri = ra-sb;           %Innenradius berechnen
66 w = 2*pi.*f(2:end);   %Kreisfrequenzvektor erstellen
67 k = sqrt(w./a);       %Koeffizient k aus Carslaw/Jaeger
68 ra_c = ra*sqrt(1i).*k; %Komplexe Argumente fuer Besselfunktionen
69 ri_c = ri*sqrt(1i).*k; %Komplexe Argumente fuer Besselfunktionen
70
71 B0i = besseli(0,ri_c); %Berechnen der Werte der Besselfunktionen
72 B0k = besselk(0,ri_c);
73 B1i = besseli(1,ra_c);
74 B1k = besselk(1,ra_c);
75
76 D = ri_c.*(B0i.*B1k + B0k.*B1i); %Matrizeintrag D aus Kapitel 2.4
77 G = ra/ri.*D;         %Eintrag der inversen Matrix,
78 % entspricht der komplexen Uebertragungsfunktion
79
80 %% Ausgangssignal im Frequenzbereich
81
82 Y = zeros(1,length(X)); %Initialisieren des neuen Spektrums
83 Y(1) = X(1);          %Fourierkoeffizient fuer konstanten Offset
84 % bleibt erhalten
85
86 Y(2:end) = G.*X(2:end); %Berechnen des Spektrums
87
88 Ti_pos0 = abs(Y(1:n_pos)); %Bilden des Betrags
89 magTi = [Ti_pos0(1)/nfft Ti_pos0(2:n_pos-1)/(nfft/2) ]; %Skalierung
90
91 %% Darstellung des veraenderten Spektrums
92
93 subplot(2,1,2)
94 hold on;
95 stem(f1,magTi,'-b')
96 grid on;
97 legend('\hat{T}_{a}','\hat{T}_{i}')
98
99 %% Ausgangssignal in den Zeitbereich transformieren
100
101 Ti = ifft(Y,nfft); %Mittels IFFT in den Zeitbereich zuruecktransformieren
102
103 %% Nicht benoetigte Werte abschneiden und Realteil bilden
104
105 Ti = Ti(1:length(Ta));
106 Ti_real = real(Ti);
107
108 %% Darstellung des wiederhergestellten Zeitsignals
109
110 subplot(2,1,1)
111 hold on;
112 plot(t,Ti_real,'--b','LineWidth',2)
113 legend('T_{a}','T_{i}')

```

Skript zur Erstellung von Abbildung 4.2

```

1 %% TestFourierAnsatzSinusPossible
2
3 % Anwendung der Uebertragungsfunktionen nach Carslaw/Jaeger
4
5 clear;           %Vorbereiten der Arbeitsumgebung
6 clc;
7 close all;
8
9 %% Initialisieren
10
11 fs = 0.1;       %Abtastfrequenz
12 t = 0:1/fs:1000; %Zeitintervall
13
14 %% Signal erzeugen
15
16 A = 200;        %Amplitude
17 freq = 0.005;   %Frequenz
18
19 Ta = A*sin(2*pi*freq*t); %Signal
20
21 %% Darstellung des Zeitsignals
22
23 figure()

```

```

24 subplot(2,1,1)
25 plot(t,Ta,'-k','Linewidth',2);
26 xlabel('Zeit $t\:[s]$');
27 ylabel('Temperatur $T\:[^\circ C]$');
28
29 %% Mittels FFT in den Frequenzbereich transformieren
30
31 nfft = 2^nextpow2(length(Ta)); %Ermitteln der Zweierpotenz fuer FFT
32 X = fft(Ta,nfft); %Mittels FFT in den Frequenzbereich transformieren
33
34 n_pos = nfft/2 + 1; %#diskrete Frequenzstellen im pos. Bereich
35 Ts = diff(t(1:2)); %Abtastzeit berechnen
36 fl = [0:floor((nfft-1)/2)] / (nfft*Ts); %Vektor mit Frequenzen im
37 % positiven Spektrum erzeugen
38
39 Ta_pos0 = abs(X(1:n_pos)); %Bilden des Betrags des positiven Spektrums
40 magTa = [Ta_pos0(1)/nfft Ta_pos0(2:n_pos-1)/(nfft/2) ]; %Skalierung des
41 % Spektrums
42
43 %% Darstellung des Spektrums
44
45 subplot(2,1,2)
46 stem(fl, magTa, '-k', 'Marker', 's' );
47 xlabel('Frequenz $f\:[Hz]$');
48 ylabel('$\hat{T}$');
49
50 %% Frequenzvektor fuer die frequenzabhaengige Uebertragungsfunktion bestimmen
51
52 df = fs/nfft; %Aufloesung des Spektrums berechnen
53 f = -fs/2:df:fs/2-df; %Frequenzvektor erstellen
54 f = ifftshift(f); %Umordnung der Frequenzen, damit die Reihenfolge
55 % zur Ausgabe der FFT passt
56
57 %% Uebertragungsfunktion
58
59 a = 6.83; %Temperaturleitfaehigkeit in mm^2/s
60 ra = 120; %Aussenradius in mm
61 sb = 20; %Wandstaerke des Rohrs in mm
62
63 ri = ra-sb; %Innenradius
64 w = 2*pi.*f(2:end); %Kreisfrequenzvektor
65 k = sqrt(w./a); %Koeffizient k aus Carslaw/Jaeger
66 ra_c = ra*sqrt(1i).*k; %Komplexe Argumente fuer Besselfunktionen
67 ri_c = ri*sqrt(1i).*k; %Komplexe Argumente fuer Besselfunktionen
68
69 B0i = besseli(0,ri_c); %Berechnen der Werte der Besselfunktionen
70 B0k = besselk(0,ri_c);
71 B1i = besseli(1,ra_c);
72 B1k = besselk(1,ra_c);
73
74 D = ri_c.*(B0i.*B1k + B0k.*B1i); %Matrizeintrag D aus Kapitel 2.4
75 G = ra/ri.*D; %Eintrag der inversen Matrix,
76 % entspricht der komplexen Uebertragungsfunktion
77
78
79 %% Ausgangssignal im Frequenzbereich
80
81 Y = zeros(1,length(X)); %Initialisieren des neuen Spektrums
82 Y(1) = X(1); %Fourierkoeffizient fuer konstanten Offset
83 % bleibt erhalten
84
85 Y(2:end) = G.*X(2:end); %Berechnen des Spektrums
86
87 Ti_pos0 = abs(Y(1:n_pos)); %Bilden des Betrags
88 magTi = [Ti_pos0(1)/nfft Ti_pos0(2:n_pos-1)/(nfft/2) ]; %Skalierung
89
90
91 %% Darstellung des veraenderten Spektrums
92
93 subplot(2,1,2)
94 hold on;
95 stem(fl,magTi,'-b')
96 grid on;
97 legend('\hat{T}_{-a}','\hat{T}_{-i}')
98
99 %% Ausgangssignal in den Zeitbereich transformieren
100
101 Ti = ifft(Y,nfft); %Mittels IFFT in den Zeitbereich zuruecktransformieren
102
103 %% Nicht benoetigte Werte abschneiden und Realteil bilden
104
105 Ti = Ti(1:length(Ta));
106 Ti_real = real(Ti);
107
108 %% Darstellung des wiederhergestellten Zeitsignals
109
110 subplot(2,1,1)
111 hold on;

```

```
112 plot(t, Ti_real, '--b', 'LineWidth', 2)
113 legend('T_{a}', 'T_{i}')
```

Vorgestellte FDM-Verfahren

Methode von D'Souza

```
1 function [ T ] = numSouza( rho, c, lambda, r, Ta )
2 %numSouza berechnet eine numerische Loesung des IHCP fuer eine Rohrgeometrie
3 % Angewendet wird der Algorithmus von D'Souza aus dem ASME-Paper No. 75-WA/HT-81
4
5 %INPUT: rho      []      Dichte des Werkstoffs
6 %         c      []      spez. Waermekapazitaet
7 %         lambda  []      Waermeleitfaehigkeit
8 %         r      [1xN]   Vektor mit Radienwerten
9 %         Ta     [Mx1]   Aussentemperaturschrieb
10 %OUTPUT: T      [MxN]   Temperaturfeld
11
12 %% Initialisieren
13 a = lambda/(rho*c); %Berechnen der Temperaturleitfaehigkeit
14 t = linspace(10,10*length(Ta),length(Ta)); %Zeitstempel zum Temperaturlastschrieb
15 dr = r(1)-r(2); %Ortschrittweite
16 dt = t(2)-t(1); %Zeitschrittweite
17
18 T = zeros(length(t),length(r)); %Initialisieren des Temperaturfelds
19 T0 = Ta(1); %Anfangstemperatur
20 T(:,1) = Ta; %Randbedingung: Aussentemperatur
21 T(1,:) = T0.*ones(1,length(r)); %Anfangsbedingung: stationaere Temperaturverteilung zu
    Beginn
22 T = [Ta,T]; %Randbedingung: Messung an isoliertem Rand
23
24 %% Berechnen
25
26 for j = 2:length(r)-1 %Schleife ueber alle Orstknoten
27     A = (1/(a*dt))/(1/dr^2-1/(2*r(j)*dr)); %Berechnung der Koeffizienten
28     B = (1/(dr^2))/(1/dr^2-1/(2*r(j)*dr));
29     C = (1/(2*r(j)*dr))/(1/dr^2-1/(2*r(j)*dr));
30     for i = 2:length(Ta) %Schleife ueber alle Orstschritte
31         T(i,j+1) = A*(T(i,j)-T(i-1,j))...%Berechnung der Temperatur am aktuellen Knoten
32                 + B*(2*T(i,j)-T(i,j-1))...
33                 - C*(T(i,j-1));
34     end
35 end
36 T = T(:,1:length(r)); %Loeschen der zusaetzlichen Knoten
37 end
```

Methode von Weber

```
1 function [ T ] = numWeber( sigma, rho, c, lambda, r, Ta )
2 %numWeber berechnet eine numerische Loesung des IHCP fuer eine Rohrgeometrie
3 % Angewendet wird der Algorithmus von Weber veroeffentlicht im Int.
4 % Journal of Heat and Mass Transfer, 1981
5
6 %INPUT: sigma    []      Stabilitaetsparameter
7 %         rho    []      Dichte des Werkstoffs
8 %         c      []      spez. Waermekapazitaet
9 %         lambda  []      Waermeleitfaehigkeit
10 %         r      [1xN]   Vektor mit Radienwerten
11 %         Ta     [Mx1]   Aussentemperaturen
12 %OUTPUT: T      [MxN]   Temperaturfeld
13
14 %% Initialisieren
15 a = lambda/(rho*c); % Berechnen der Temperaturleit-
16 % faehigkeit
17
18 dr = r(2)-r(1); %Ortschrittweite, berechnet
19 % aus Differenz des ersten und zweiten Eintrags des Radienvektors
20
21 t = linspace(10,10*length(Ta),length(Ta)); %Zeitstempel: Das Intervall
22 % von 10 s bis 10*Anzahl der Messwerte wird in einem Vektor in so viele
23 % lineare Intervalle unterteilt wie es Messpunkte gibt.
24
25 dt = t(2)-t(1); %Zeitschrittweite,
26 % berechnet aus Differenz des ersten und zweiten Eintrags des Zeitstempels
```

```

27
28 T = zeros(length(Ta),length(r));           %Temperaturfeld: Matrix mit
29 % Nulleintraegen, die Zeilenanzahl ist gleich der Anzahl der Zeitschritte
30 % und Spaltenanzahl entspricht der Anzahl der Ortsknoten
31
32 T0 = Ta(1);                               %Auslesen der Anfangstemperatur
33 T(:,1) = Ta;                               %Randbedingung: gemessene
34 % Aussentemperaturen werden in die erste Spalte (und jede Zeile) des
35 % Temperaturfeldes gespeichert
36
37 T = [Ta,T];                               %Randbedingung: Messung an
38 % isoliertem Rand, zum Ergaenzen der waagerechten Tangente wird der Aussen-
39 % temperaturvektor an einem zusaetzlichen, fiktiven Knoten eingefuegt
40
41 T(1,:) = ones(1,length(r)+1).*T0;         %Anfangsbedingung: stationaere
42 % Temperaturverteilung zu Beginn, die gesamte erste Zeile des erweiterten
43 % Temperaturfelds wird mit einem Zeilenvektor mit der Anfangstemperatur T0
44 % als Eintraegen gefuehlt
45 k = 1:length(t);                           %Hilfsvariable fuer
46 % Schleifenkonstruktion
47
48 %% Berechnen
49 for j = 2:length(r)                       %Schleife ueber alle Ortsknoten
50     A = (sigma/dt^2)/(a/(dr^2)+a/(2*r(j)*dr)); %Setzen der Variablen fuer
51     % jeden Schritt der Berechnungsschleife
52     B = (1/(2*dt))/(a/(dr^2)+a/(2*r(j)*dr));
53     C = (a/dr^2)/(a/(dr^2)+a/(2*r(j)*dr));
54     D = (a/(2*r(j)*dr))/(a/(dr^2)+a/(2*r(j)*dr));
55
56     for i = 2:length(k)-1                 %Schleife ueber alle Zeitschritte
57         T(i,j+1) = A*(T(i+1,j)-2*T(i,j)+T(i-1,j))...
58                 + B*(T(i+1,j)-T(i-1,j))...
59                 + C*(2*T(i,j)-T(i,j-1))...
60                 + D*(T(i,j-1));
61     end %durch ... wird die Berechnung ueber den Zeilenumbruch fortgesetzt
62     k = 1:length(k)-1;                   %Aendern des Zeitbereichs
63 end
64 T = T(:,1:length(r));                   %Loeschen der zusaetzlichen Knoten
65 end

```

Methode von Hills & Hensel

```

1 function [ T ] = numHillsHensel( m, rho, c, lambda, r, Ta )
2 %numHillsHensel berechnet die Temperaturverteilung fuer eine Rohrgeometrie
3 % Angewendet wird der Algorithmus von Hills & Hensel veroeffentlicht in
4 % Numerical Heat Transfer, 1986
5
6 %INPUT: m      []      Anzahl der gewichteten Knoten im Filterprozess
7 % rho        []      Dichte des Werkstoffs
8 % c          []      spez. Waermekapazitaet
9 % lambda     []      Waermeleitfaehigkeit, die Stoffwerte sind in
10 % konsistenten Einheiten anzugeben!
11 % r          [1xN]   Vektor mit Radienwerten von r_a nach r_i in [mm]
12 % Ta        [Mx1]   Aussentemperaturschrieb mit Messwerten in [ $\circ$C]
13 %OUTPUT: T    [MxN]   Temperaturfeld
14
15 %% Initialisieren
16
17 t = linspace(10,10*length(Ta),length(Ta)); %Zeitstempel: Das Intervall
18 % von 10 s bis 10*Anzahl der Messwerte wird in einem Vektor in so viele
19 % lineare Intervalle unterteilt wie es Messpunkte gibt.
20
21 dr = r(1)-r(2); %Ortschrittweite, berechnet
22 % aus Differenz des ersten und zweiten Eintrags des Radienvektors
23
24 dt = t(2)-t(1); %Zeitschrittweite,
25 % berechnet aus Differenz des ersten und zweiten Eintrags des Zeitstempels
26
27 T = zeros(length(t),length(r)); %Temperaturfeld: Matrix mit
28 % Nulleintraegen, die Zeilenanzahl ist gleich der Anzahl der Zeitschritte
29 % und Spaltenanzahl entspricht der Anzahl der Ortsknoten
30
31 T(:,1) = Ta; %Randbedingung: gemessene
32 % Aussentemperaturen werden in die erste Spalte (und jede Zeile) des
33 % Temperaturfeldes gespeichert
34
35 T = [Ta,T]; %Randbedingung: Messung an
36 % isoliertem Rand, zum Ergaenzen der waagerechten Tangente wird der Aussen-
37 % temperaturvektor an einem zusaetzlichen, fiktiven Knoten eingefuegt
38
39 q = zeros(length(Ta),1); %Randbedingung: Messung an
40 % adiabatem Rand, erzeugt einen Spaltenvektor mit Nullen, mit Laenge des

```

MATLAB-Codes

```

41 % Aussentemperaturvektors
42
43 theta = [q;Ta]; %Waermestrom-Temperatur-
44 % Vektor erzeugen, durch Verbinden der Spaltenvektoren uebereinander
45
46 %% Filtermatrix initialisieren
47
48 f_0 = ones(1,length(Ta)); %Zeilenvektor mit Nullen als Eintraegen
49 % der Laenge des Aussentemperaturvektors erzeugen
50
51 F_s = diag(f_0); %Erzeugen einer Matrix mit den
52 % Eintraegen von f_0 auf der Diagonalen
53
54 f_i = ones(length(Ta)-1,1); %Spaltenvektor mit Einsen als Eintraegen
55 % mit der Laenge des Aussentemperaturvektors minus Eins erzeugen
56
57 %Super- und Subdiagonalen der Filtermatrix mit Einsen fuellen
58 for i = 1:m-1 %Schleife ueber alle Diagonalen, die mit Einsen gefuellt
59 % werden
60 F_s = F_s+diag(f_i,i);
61 F_s = F_s+diag(f_i,-i);
62 f_i = ones(length(f_i)-1,1); %Anpassung der Laenge des Diagonalenvektors
63 % fuer die naechste Iteration
64 end
65
66 %m-te Diagonalen mit Eintraegen 1/2 fuellen (letzter Iterationsschritt der
67 % Schleife)
68 f_m = 0.5.*f_i;
69 F_s = F_s+diag(f_m,m);
70 F_s = F_s+diag(f_m,-m);
71
72 %Projizieren ueberzaehlicher Knoten auf den Rand
73 for i = 1:m %Schleife ueber alle gewichteten Knoten
74 f = 1.5+(m-i); % Berechnen des Summengliedes
75 F_s(i,1) = f; % Ueberschreiben der indizierten Stelle mit dem
76 % Summenglied
77 F_s(length(Ta)-i+1,length(Ta)) = f; % Ueberschreiben der indizierten
78 % Stelle mit dem Summenglied
79 end
80
81 %Anfangswerte gewichten
82 F_s(length(Ta),:) = [zeros(1,(length(Ta)-2*m-1)) 0.5 ones(1,(2*m-1)) 0.5];
83 % Die letzte Zeile der Teilmatrix F_s wird mit den Werten ueberschrieben,
84 % die Zeilenvektoren in der eckigen Klammer werden zu einer Zeile zusammen
85 % gefasst
86
87 %Normieren, damit Zeilensumme gleich 1
88 F_s = F_s./(2*m); %Alle Eintraege der Matrix F_s werden komponentenweise
89 % (Operator: ./) durch die Zeilensumme geteilt.
90
91 %Kombination der Teilmatrizen
92 F = [F_s zeros(length(Ta));zeros(length(Ta)) F_s]; %Die Teilmatrizen F_s
93 % werden auf der Diagonalen der Filtermatrix angordnet und mit Nullmatrizen
94 % aufgefuellt
95
96 %% Berechnen
97 for i = 2:length(r) %Schleife ueber alle Ortschritte
98 %Berechnen der Koeffizienten und Teilmatrizen, fuer aequidistante
99 % Ortsdiskretisierung auch ausserhalb der Schleife zu realisieren
100
101 a_i = r(i-1)/r(i); %Radienverhaeltnis der Ortsknoten
102 A = eye(length(Ta)).*a_i; %Einheitsmatrix [length(Ta)xlength(Ta)]
103 % wird komponentenweise mit dem Koeffizienten a_i multipliziert
104
105 b_i = a_i*0.5*dr*rho*c/dt; %Berechnen des Koeffizienten b_i
106 b = ones(1,length(Ta)-1).*b_i; %Erzeugen eines Zeilenvektors um einen
107 % Eintrag kuerzer als der Aussentemperaturvektor, der mit
108 % komponentenweise mit dem Koeffizient b_i multipliziert wird
109
110 B = zeros(length(Ta))-diag(b,1)+diag(b,-1); %Erzeugen einer Nullmatrix
111 % [length(Ta)xlength(Ta)] mit dem Vektor b auf der Super- und
112 % Subdiagonalen
113
114 B(1,2) = 0; %Ueberschreiben der indizierten
115 % Stelle mit 0
116 B(length(Ta),length(Ta)) = -2*b_i; %Ueberschreiben der indizierten
117 % Stelle mit -2*b_i
118 B(length(Ta),length(Ta)-1) = 2*b_i; %Ueberschreiben der indizierten
119 % Stelle mit 2*b_i
120
121 c_i = -a_i*dr/lambda; %Berechnen des Koeffizienten c_i
122 C = eye(length(Ta)).*c_i; %Einheitsmatrix [length(Ta)xlength(Ta)]
123 % wird komponentenweise mit dem Koeffizienten c_i multipliziert
124
125 d_i = b_i*dr/lambda; %Berechnen des Koeffizienten d_i
126 d = ones(1,length(Ta)-1).*d_i; %Erzeugen eines Zeilenvektors um einen
127 % Eintrag kuerzer als der Aussentemperaturvektor, der mit
128 % komponentenweise mit dem Koeffizient d_i multipliziert wird

```

```

129
130 D = diag(ones(1,length(Ta))-diag(d,-1)+diag(d,1); %Erzeugen einer
131 % Matrix [length(Ta)xlength(Ta)] mit Einsen auf der Diagonalen und
132 % mit dem Vektor d auf der Super- und Subdiagonalen
133
134 D(1,2) = 0; %Ueberschreiben der indizierten
135 % Stelle mit 0
136 D(length(Ta),length(Ta)) = 1+2*d_i; %Ueberschreiben der indizierten
137 % Stelle mit 1+2*d_i
138 D(length(Ta),length(Ta)-1) = -2*d_i; %Ueberschreiben der indizierten
139 % Stelle mit -2*d_i
140
141 E =[A B;C D]; %Kombination der Teilmatrizen
142
143 theta = E*theta; %Berechnen der Temperaturen zu jedem Zeitschritt
144 % durch Matrix-Vektor-Multiplikation
145 theta = F*theta; %Multiplikation mit Filtermatrix
146
147 T(:,i+1) = theta(length(Ta)+1:end); %Speichern der Temperaturen spaltenweise in das Temperaturfeld
148 end
149 T = T(:,1:length(r)); %Loeschen der zusaetzlichen ,
150 % fiktiven Knoten
151 end

```

Skript zur Erstellung von Abbildung 4.3

```

1 %% TestWandstaerke
2 %Erstellt einen Plot zur Auswertung des relativen Fehlers ueber der
3 %Wandstaerke, dem charakteristischen Mass der Waermeleitung.
4 %Das Radienverhaeltnis wird dabei festgehalten, ebenso ist die
5 %Ortsdiskretisierung identisch.
6
7 close all; %Vorbereiten der Arbeitsumgebung
8 clear;
9 clc;
10
11 %% Initialisieren
12
13 rho = 7.76e-6; %Dichte des Beispielwerkstoffs in
14 % kg/m^3
15 c = 622; %spez. Waermekapazitaet des
16 % Beispielwerkstoffs in J/(kgK)
17 lambda = 0.033; %Waermeleitfaehigkeit des
18 % Beispielwerkstoffs in W/(mK)
19
20 sigma = 1e-3; %Stabilitaetsparameter fuer
21 % Weber-Algorithmus
22 m = 1; %Anzahl der gewichteten Knoten im
23 % Filter von Hills und Hensel
24
25 t = linspace(0,4000,401); %Zeitvektor des Testproblems
26
27 dr = linspace(0,100,101); %Laengenvektor
28 qr = [0.75 0.85 0.95]; %Radienverhaeltnisvektor
29
30 i = 2; %Variable zur Auswahl des
31 % Radienverhaeltnis
32
33 ra = dr./(1-qr(i)); %Berechnung des Aussenradius
34 ri = qr(i).*ra; %Berechnung des Innenradius
35
36 E = zeros(length(dr),5); %Fehlermatrix erstellen
37 E(:,1) = dr; %Eintragen der Wandstaerken
38
39 %% Berechnen
40
41 for i = 2:length(dr)
42 s = linspace(ri(i),ra(i),40); %Vektor mit Radienwerten
43 [ T ] = pdepeRohrSin( s, t ); %Berechnung des Testproblems
44 Ta = T(:,length(s)); %Auslesen der Aussentemperatur-
45 % Messdaten
46 r = fliplr(s); %Drehen der Radienwerte, da beim
47 % inversen Problem von Aussen nach Innen gerechnet wird
48 [ T_S ] = numSouza( rho, c, lambda, r, Ta ); %Rueckrechnung mit FDM
49 [ T_W ] = numWeber(sigma, rho, c, lambda, r, Ta );
50 [ T_MG ] = numMurioGuo( rho, c, lambda, r, Ta, eps );
51 [ T_HH ] = numHillsHensel( m, rho, c, lambda, r, Ta );
52 E(i,2) = max(abs(T_S(1:(length(t)-length(s)),length(s))-...
53 T(1:(length(t)-length(s)),1)))/400.125; %Eintragen des relativen
54 % Fehlers in die Matrix
55 E(i,3) = max(abs(T_W(1:(length(t)-length(s)),length(s))-...

```

```

56     T(1:(length(t)-length(s),1))/400.125;
57     E(i,4) = max(abs(T_HH(1:(length(t)-length(s)),length(s))-...
58     T(1:(length(t)-length(s),1))/400.125;
59 end
60
61 %% Plot
62
63 figure()
64 hold on;
65 grid on;
66 plot(E(:,1),E(:,2),'-b','Marker','o','LineWidth',2);
67 plot(E(:,1),E(:,3),'Color',[0.89 0.45 0.14],'Marker','>','LineWidth',2);
68 plot(E(:,1),E(:,4),'Color',[0.85 0.84 0.79],'Marker','s','LineWidth',2);
69 axis([0 dr(length(dr)) 0 0.25]);
70 xlabel('Wandstärke [mm]')
71 ylabel('$\epsilon_{rel}$')
72 grid on;
73 hold off;
74 legend('Souza','Weber','Murio & Guo','Hills & Hensel','Location','NorthWest')

```

Skript zur Erstellung von Abbildung 4.4

```

1 %% TestGradientDreieck
2
3 % testet die Finite-Differenzen-Approximationen fuer das 1D-IHCP einer
4 % Rohrleitung und berechnet eine Matrix mit dem relativen Fehler der
5 % Verfahren als Eintraegen
6
7 % clear;
8 clc;
9 close all;
10
11 %% Initialisieren
12
13 rho = 7.76e-6; %Dichte des Beispielwerkstoffs in kg/m^3
14 c = 622; %spez. Waermekapazitaet des Beispielwerkstoffs in J/(kgK)
15 lambda = 0.033; %Waermeleitfaehigkeit des Beispielwerkstoffs in W/(mK)
16
17 sigma = 1e-6; %Stabilitaetsparameter fuer Weber-Algorithmus
18 eps = 1e-3; %Messabweichung fuer MurioGuo-Algorithmus
19 m = 1; %Anzahl der gewichteten Knoten im Filter von Hills und Hensel
20
21 t = linspace(0,4000,401); %Zeitvektor des Testproblems
22
23 T_p = linspace(50,1500,32); %Vektor mit Periodendauer um Gradient zu variieren
24 DTDt = 390.237./(0.5*T_p); %Gradientenvektor
25
26 da = 504.6; %Aussendurchmesser in mm
27 sb = 39.5; %Wanddicke in mm
28 ra = 0.5*da; %Aussenradius in mm
29 ri = ra-sb; %Innenradius in mm
30 s = linspace(ri,ra,round(ra-ri)); %Vektor mit Radienwerten
31 r = fliplr(s); %Drehen der Radienwerte, da beim inversen Problem von Aussen nach
32 %Innen gerechnet wird
33 %nur im ersten Schritt, danach auskommentieren
34 % E = zeros(length(DTDt),5); %Fehlermatrix
35 % E(:,1) = fliplr(DTDt); %Gradientenvektor eintragen
36
37 %% Berechnen
38 i = 32; %Laufvariable, es sind length(T_p)
39 %Durchgaenge erforderlich
40 [ T ] = pdepeRohrTriang( s, t ); %Berechnung des Testproblems
41 Ta = T(:,length(s)); %Auslesen der Aussentemperatur-Messdaten
42 [ T_S ] = numSouza( rho, c, lambda, r, Ta );
43 [ T_W ] = numWeber(sigma, rho, c, lambda, r, Ta );
44 [ T_MG ] = numMurioGuo( rho, c, lambda, r, Ta, eps );
45 [ T_HH ] = numHillsHensel( m, rho, c, lambda, r, Ta );
46
47 %% Ausgabe
48 %Speichern des relativen Fehlers in die Fehlermatrix
49 E(i,2) = max(abs(T_S(1:(length(t)-length(s)),length(s))-T(1:(length(t)-length(s),1)))/390.237;
50 E(i,3) = max(abs(T_W(1:(length(t)-length(s)),length(s))-T(1:(length(t)-length(s),1)))/390.237;
51 E(i,4) = max(abs(T_MG(1:(length(t)-length(s)),length(s))-T(1:(length(t)-length(s),1)))/390.237;
52 E(i,5) = max(abs(T_HH(1:(length(t)-length(s)),length(s))-T(1:(length(t)-length(s),1)))/390.237;
53
54 %% Plot
55
56 figure()
57 hold on;

```

```

59 scatter(Eps(:,1),Eps(:,2),s,'b','fill');
60 scatter(Eps(:,1),Eps(:,3),s,[0.89 0.45 0.14], '>', 'fill');
61 scatter(Eps(:,1),Eps(:,5),s,[0.85 0.84 0.79], 's', 'fill');
62 axis([0.4 1.1 0 0.2]);
63 xlabel('Gradient dT/dt [K/s]')
64 ylabel('Relativer Fehler \epsilon_{rel} [-]')
65 grid on;
66 hold off;
67 legend('Souza', 'Weber', 'Hills & Hensel', 'Location', 'NorthWest')

```

Skript zur Erstellung von Abbildung 4.5

```

1 %% TestRechteck
2 %Erstellt den Plot zur Auswertung des Grenzwerts des Gradienten gegen
3 %Unendlich
4
5 close all; %Vorbereiten der Arbeitsumgebung
6 clear;
7 clc;
8
9 %% Initialisieren
10
11 rho = 7.76e-6; %Dichte des Beispielwerkstoffs
12 % in kg/m^3
13 c = 622; %spez. Waermekapazitaet des
14 % Beispielwerkstoffs in J/(kgK)
15 lambda = 0.033; %Waermeleitfaehigkeit des
16 % Beispielwerkstoffs in W/(mK)
17
18 sigma = 1e-6; %Stabilitaetsparameter fuer
19 % Weber-Algorithmus
20 m = 1; %Anzahl der gewichteten Knoten im
21 % Filter von Hills und Hensel
22
23 t = linspace(0,4000,401); %Zeitvektor des Testproblems
24
25 da = 504.6; %Aussendurchmesser in mm
26 sb = 39.5; %Wanddicke in mm
27 ra = 0.5*da; %Aussenradius in mm
28 ri = ra-sb; %Innenradius in mm
29 s = linspace(ri,ra,round(ra-ri)); %Vektor mit Radienwerten
30 r = fliplr(s); %Drehen der Radienwerte, da beim
31 % inversen Problem von Aussen nach Innen gerechnet wird
32
33 %% Berechnen
34
35 [ T ] = pdepeRohrRectwin( s, t ); %Berechnung des Testproblems
36 Ta = T(:,length(s)); %Auslesen der Aussentemperatur-
37 % Messdaten
38 [ T_S ] = numSouza( rho, c, lambda, r, Ta ); %Rueckrechnung mit FDM
39 [ T_W ] = numWeber(sigma, rho, c, lambda, r, Ta );
40 [ T_HH ] = numHillsHensel( m, rho, c, lambda, r, Ta );
41
42 %% Plot
43
44 figure()
45 hold on;
46 plot(t,T(:,1), '-k', 'LineWidth',2);
47 plot(t,T_S(:,length(s)), '-b', 'Marker', 'o', 'LineWidth',2);
48 xlabel('Zeit $t\:[s]$')
49 ylabel('Temperatur $T\:[^{\circ}C]$')
50 grid on;
51 hold off;
52 legend('T_{i} aufgegeben', 'Souza', 'Location', 'NorthWest')
53
54 figure()
55 hold on;
56 plot(t,T(:,1), '-k', 'LineWidth',2);
57 plot(t(1:(length(t)-length(s))),T_W(1:(length(t)-length(s)),...
58 length(s)), 'Color',[0.89 0.45 0.14], 'Marker', '>', 'LineWidth',2);
59 xlabel('Zeit $t\:[s]$')
60 ylabel('Temperatur $T\:[^{\circ}C]$')
61 grid on;
62 hold off;
63 legend('T_{i} aufgegeben', 'Weber', 'Location', 'NorthWest')
64
65 figure()
66 hold on;
67 plot(t,T(:,1), '-k', 'LineWidth',2);
68 plot(t,T_HH(:,length(s)), 'Color',[0.85 0.84 0.79],...
69 'Marker', 's', 'LineWidth',2);

```

```

70 xlabel('Zeit $t\:[s]$\')
71 ylabel('Temperatur $T\:[^\circ C]$\')
72 grid on;
73 hold off;
74 legend('T_{i} aufgegeben','Hills & Hensel','Location','NorthWest')

```

Skript zur Erstellung von Abbildung 4.6

```

1 %% TestNoiseFDM
2 % testet die Finite-Differenzen-Approximationen fuer das 1D-IHCP einer
3 % Rohrleitung bei verrauschten Eingangsdaten
4 % und berechnet eine Matrix mit dem relativen Fehler der
5 % Verfahren als Eintraegen
6
7 clear; %Vorbereiten der Arbeitsumgebung
8 clc;
9 close all;
10
11 %% Initialisieren
12
13 rho = 7.76e-6; %Dichte des Beispielwerkstoffs
14 % in kg/m^3
15 c = 622; %spez. Waermekapazitaet des
16 % Beispielwerkstoffs in J/(kgK)
17 lambda = 0.033; %Waermeleitfaehigkeit des
18 % Beispielwerkstoffs in W/(mK)
19
20 sigma = 1e-3; %Stabilitaetsparameter fuer
21 % Weber-Algorithmus
22 m = 1; %Anzahl der gewichteten Knoten im
23 % Filter von Hills und Hensel
24
25 t = linspace(0,4000,401); %Zeitvektor des Testproblems
26 snr = logspace(9,4,14); %Signal-Rauschen-Verhaeltnis
27
28 da = 504.6; %Aussendurchmesser in mm
29 sb = 39.5; %Wanddicke in mm
30 ra = 0.5*da; %Aussenradius in mm
31 ri = ra-sb; %Innenradius in mm
32 s = linspace(ri,ra,round(ra-ri)); %Vektor mit Radienwerten
33 r = fliplr(s); %Drehen der Radienwerte, da beim
34 % inversen Problem von Aussen nach Innen gerechnet wird
35
36 [ T ] = pdepeRohrSin( s, t ); %Berechnung des Testproblems
37 E = zeros(length(snr),4); %Fehlermatrix
38 % E = [snr ' E'];
39
40 %% Berechnen
41
42 for i = 1:length(snr) %Schleife ueber alle Verhaeltnisse
43     Ta = T(:,length(s)); %Auslesen der Aussentemperatur-
44     % Messdaten
45     Ta = awgn(Ta,snr(i),'measured','linear');
46     [ T_S ] = numSouza( rho, c, lambda, r, Ta );
47     [ T_W ] = numWeber(sigma, rho, c, lambda, r, Ta );
48     [ T_HH ] = numHillsHensel( m, rho, c, lambda, r, Ta );
49
50 %% Ausgabe
51 %Speichern des relativen Fehlers in die Fehlermatrix
52 E(i,1) = max(abs(Ta(1:(length(t)-length(s)))-...
53     T(1:(length(t)-length(s)),length(s))));
54 E(i,2) = max(abs(T_S(1:(length(t)-length(s)),length(s))-...
55     T(1:(length(t)-length(s)),1)))/400.125;
56 E(i,3) = max(abs(T_W(1:(length(t)-length(s)),length(s))-...
57     T(1:(length(t)-length(s)),1)))/400.125;
58 E(i,4) = max(abs(T_HH(1:(length(t)-length(s)),length(s))-...
59     T(1:(length(t)-length(s)),1)))/400.125;
60
61 end
62
63 %% Plot
64 figure()
65 hold on;
66 semilogx(E(:,1),E(:,2),'-b','Marker','o','LineWidth',2);
67 semilogx(E(:,1),E(:,3),'Color',[0.89 0.45 0.14],'Marker','>','LineWidth',2);
68 semilogx(E(:,1),E(:,4),'Color',[0.85 0.84 0.79],'Marker','s','LineWidth',2);
69 axis([0 5 0 1]);
70 xlabel('Messfehler \Delta T')
71 ylabel('Relativer Fehler')
72 grid on;
73 hold off;

```

```
74 legend('Souza','Weber','Hills & Hensel','Location','NorthEast')
```

Skript zur Erstellung von Abbildung 4.7

```
1 %% TestWeberSigma
2 % testet den Algorithmus von Weber auf Abhaengigkeit vom Parameter sigma
3
4 clear; %Vorbereitung der Arbeitsumgebung
5 clc;
6 close all;
7
8 %% Initialisieren
9
10 sigma = logspace(-10,0,101); %Parameter sigma, variiert
11 % von 2e-10 bis 1
12 rho = 7.76e-6; %Dichte des Beispielwerkstoffs
13 % in kg/mm^3
14 c = 622; %spez. Waermekapazitaet des
15 % Beispielwerkstoffs in J/(kgK)
16 lambda = 0.033; %Waermeleitfaehigkeit des
17 % Beispielwerkstoffs in W/(mmK)
18
19 t = linspace(0,4000,401); %Zeitvektor des Testproblems
20
21 da = 504.6; %Aussendurchmesser in mm
22 sb = 39.5; %Wanddicke in mm
23 ra = 0.5*da; %Aussenradius in mm
24 ri = ra-sb; %Innenradius in mm
25 s = linspace(ri,ra,round(ra-ri)); %Vektor mit Radienwerten
26
27 [ T ] = pdepeRohrTriang( s, t ); %Berechnung des Testproblems
28
29 Ta = T(:,length(s)); %Auslesen der Aussentemperatur-Messdaten
30 r = fliplr(s); %Drehen der Radienwerte, da beim
31 % inversen Problem von aussen nach innen gerechnet wird
32
33 e = zeros(length(sigma),1); %Fehlervektor initialisieren
34
35
36 %% Berechnen
37
38 for i = 1:length(sigma)
39 %Anwendung des Weber-Verfahrens
40 [ X ] = numWeber(sigma(i), rho, c, lambda, r, Ta );
41 %Auswerten des relativen Fehlers fuer jedes Element von sigma
42 e(i) = max(abs(X(1:(length(t)-length(s)),length(s))-...
43 T(1:(length(t)-length(s),1)))/390.237;
44
45 end
46
47 %% Plot
48
49 figure()
50 grid on;
51 semilogx(sigma,e,'-k','LineWidth',2);
52 xlabel('\sigma')
53 ylabel('\epsilon_{rel}')
```

Skript zur Erstellung von Abbildung 4.8

```
1 %% TestHillsHenselm
2 % testet den Algorithmus von Weber auf Abhaengigkeit vom Parameter m
3
4 clear; %Vorbereitung der Arbeitsumgebung
5 clc;
6 close all;
7
8 %% Initialisieren
9
10 m = logspace(0,2,3); %Parameter sigma, variiert
11 % von 1 bis 1e2
12 rho = 7.76e-6; %Dichte des Beispielwerkstoffs
13 % in kg/mm^3
14 c = 622; %spez. Waermekapazitaet des
15 % Beispielwerkstoffs in J/(kgK)
```

MATLAB-Codes

```
16 lambda = 0.033; %Waermeleitfaehigkeit des
17 % Beispielwerkstoffs in W/(mmK)
18
19 t = linspace(0,4000,401); %Zeitvektor des Testproblems
20
21 da = 504.6; %Aussendurchmesser in mm
22 sb = 39.5; %Wanddicke in mm
23 ra = 0.5*da; %Aussenradius in mm
24 ri = ra-sb; %Innenradius in mm
25 s = linspace(ri,ra,round(ra-ri)); %Vektor mit Radienwerten
26
27 [ T ] = pdepeRohrRectwin( s, t ); %Berechnung des Testproblems
28
29 Ta = T(:,length(s)); %Auslesen der Aussentemperatur-
30 % Messdaten
31 r = fliplr(s); %Drehen der Radienwerte, da beim
32 % inversen Problem von aussen nach innen gerechnet wird
33
34 E = zeros(length(m),2); %Fehlermatrix erstellen
35 E(:,1) = m; %
36
37 Ti = zeros(length(Ta),length(m));
38
39 %% Berechnen
40
41 for i = 1:length(m)
42 % m = m(i); %Auswahl des Parameters
43 [ X ] = numHillsHensel( m(i), rho, c, lambda, r, Ta );
44 % m = logspace(0,2,3);
45 E(i,2) = max(abs(X(:,length(s))-T(:,1)))/400;
46 Ti(:,i) = X(:,length(s));
47 end
48
49 Ti = [T(:,1) Ti];
50 %% Plot
51
52 figure()
53 grid on;
54 hold on;
55 plot(t,Ti(:,1),'-k','LineWidth',3);
56 plot(t,Ti(:,2),'--b','LineWidth',2);
57 plot(t,Ti(:,3),'-r','Color',[0.89 0.45 0.14],'LineWidth',2);
58 plot(t,Ti(:,4),'-c','LineWidth',2);
59 xlabel('Zeit t [s]')
60 ylabel('Temperatur T [^\circ C]')
61 legend('T_{i} aufgebracht',[ 'm = ',num2str(m(1))],[ 'm = ',...
62 num2str(m(2))],[ 'm = ',num2str(m(3))],'Location','South');
```

Literaturverzeichnis

- [1] J. V. Beck. Surface heat flux determination using an integral method. Nuclear Engineering and Design, 7(2):170–178, 1968.
- [2] J. V. Beck, B. Blackwell und C. R. St. Clair, Jr. Inverse heat conduction. Wiley-Interscience, New York, 1985.
- [3] O. R. Burggraf. An exact solution of the inverse problem in heat conduction theory and application. Journal of Heat Transfer, 1964.
- [4] H. S. Carslaw und J. C. Jaeger. Conduction of heat in solids. Clarendon Press, Oxford, 1959.
- [5] R. Courant, K. Fredricks und H. Lewy. On the partial differential equations of mathematical physics. IBM Journal, 1967.
- [6] N. D’Souza. Numerical solution of one-dimensional inverse transient heat conduction by finite difference method,. ASME paper No. 75-WA/HT-81, 1975.
- [7] J. D. Faires und R. L. Burden. Numerische Methoden: Näherungsverfahren und ihre praktische Anwendung. Spektrum Akademischer Verlag GmbH, Heidelberg, Berlin, Oxford, 1994.
- [8] I. Frank. An application of least squares method to the solution of the inverse problem of heat conduction. Journal of Heat Transfer, 85(4):378–379, 1963.
- [9] R. G. Hills und E. C. Hensel, Jr. One-dimensional nonlinear inverse heat conduction technique. Numerical Heat Transfer, Part A: Applications, 10(4):369–393, 1986.
- [10] P. Hofstötter. Einsatz von Thermoelementen zur Erfassung der Temperatur von Rohrleitungswandungen im Rahmen der Ermüdungsüberwachung. Schriftenreihe Reaktorsicherheit und Strahlenschutz, 2004.
- [11] F. P. Incropera, D. DeWitt, T. L. Bergman und A. S. Lavine. Fundamentals of heat and mass transfer. John Wiley & Sons, 2007.
- [12] C. Karpfinger. Höhere Mathematik in Rezepten. Springer, 2014.

- [13] N. W. McLachlan. Bessel functions for engineers. Clarendon Press Oxford, 1955.
- [14] K. Miller und G. A. Viano. On the necessity of nearly-best-possible methods for analytic continuation of scattering data. Journal of Mathematical Physics, 14(8):1037–1048, 1973.
- [15] G. P. Mulholland und B. P. Gupta. Inverse problem of heat conduction in composite media. American Society of Mechanical Engineers, 1975.
- [16] W. Polifke. Wärmeübertragung: Grundlagen, analytische und numerische Methoden. Pearson Deutschland GmbH, 2009.
- [17] M. Richter. Inverse Probleme: Grundlagen, Theorie und Anwendungsbeispiele. Springer-Verlag, 2015.
- [18] G. Stolz. Numerical solutions to an inverse problem of heat conduction for simple shapes. Journal of heat transfer, 82(1):20–25, 1960.
- [19] R. Strecker, F. und Feldtkeller. Grundlagen der theorie des allgemeinen vierpols. Elektrische Nachrichtentechnik, 6:93, 1929.
- [20] A. N. Tikhonov und V. Y. Arsenin. Solutions of ill posed problems. Bulletin (New Series) of the American Mathematical Society, 1(3):521–524, 1979.
- [21] W. A. Wall und M. Kronbichler. Numerische Methoden für Ingenieure, 2016.
- [22] C. F. Weber. Analysis and solution of the ill-posed inverse heat conduction problem. International Journal of Heat and Mass Transfer, 24(11):1783–1792, 1981.