

Simulation und Messung der Temperaturverteilung einer Wolframionenquelle

Bachelorarbeit in Physik

angefertigt am
Helmholtz-Institut für Strahlen- und Kernphysik

vorgelegt der

Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

von
Alexander Schäbe

September 2012

Hiermit versichere ich, dass ich die Arbeit selbstständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt, sowie Zitate kenntlich gemacht habe.

- 1. Gutachter:** Priv. Doz. Dr. Vianden
- 2. Gutachter:** Priv. Doz. Dr. Eversheim

Inhaltsverzeichnis

1	Einleitung & Motivation	1
2	Problemstellung	3
3	Theoretischer Hintergrund	5
3.1	Wärmeleitung	5
3.2	Wärmestrahlung	7
4	Lösungen	9
4.1	Analytische Lösung	9
4.2	Numerische Lösung	11
4.2.1	Berechnung der Temperaturverteilung eines Zylinders	11
4.2.2	Berechnung der Temperaturverteilung eines Kegelstumpfes	14
4.2.3	Berechnung der Temperaturverteilung des vollständigen Ofens	15
4.2.4	Anfangswerte	16
4.2.5	Abbruchkriterium	16
5	Verhalten der Simulation	17
5.1	Abhängigkeit von physikalischen Größen	17
5.1.1	Temperaturverteilung bei unterschiedlichen Querschnittsflächen	17
5.1.2	Temperaturverteilung bei unterschiedlichen Außenflächen	18
5.1.3	Temperaturverteilung bei unterschiedlichen Längen	19
5.1.4	Temperaturverteilung bei unterschiedlichen Umgebungstemperaturen	20
5.1.5	Temperaturverteilung bei unterschiedlichen Emissionskoeffizienten	21
5.1.6	Unabhängigkeit von den Anfangsbedingungen	21
5.1.7	Berechnung der Temperaturverteilung bei unterschiedlichen Anfangstemperaturen	22
5.2	Iterationsverhalten	23
5.3	Konvergenzverhalten	24
5.4	Beispielsimulation für einen vollständigen Ofen	24
6	Versuchsaufbau	27
7	Ergebnisse	29
7.1	Ergebnisse der Messungen	29
7.1.1	Messung mit Pyrometer und Thermoelement	29

7.1.2	Messung mit zwei Thermoelementen	30
7.2	Bestimmung des Emissionskoeffizienten	31
7.3	Ergebnisse der Simulation und analytischen Lösung	32
7.4	Messung und Simulation des vollständigen Ofens	33
8	Fazit	35
	Abbildungsverzeichnis	37
	Tabellenverzeichnis	39
	Literatur	41
	Programmcode für den vollständigen Ofen	43
	Danksagung	59

1 Einleitung & Motivation

Mit dem Bonner Isotopenseparator (BONIS) können Isotope voneinander getrennt und in anderen Materialien implantiert werden, was für die Materialforschung eine wichtige Rolle spielt.

Um Rubidium in andere Materialien zu implantieren, muss es zunächst verdampft und anschließend ionisiert werden. Hierfür verwendet man Wolframöfen, die zum einen Teil aus einem Hohlzylinder und zum anderen Teil aus einem Zylinderstumpf bestehen.

In Abbildung 1.1 ist die Oberflächenionisationsquelle dargestellt, in der sich ein solcher Wolframofen befindet. Den Ofen umgeben zwei Glühwendeln, die eine Elektronenstoßheizung bilden, und den Wolframofen erhitzen sollen.

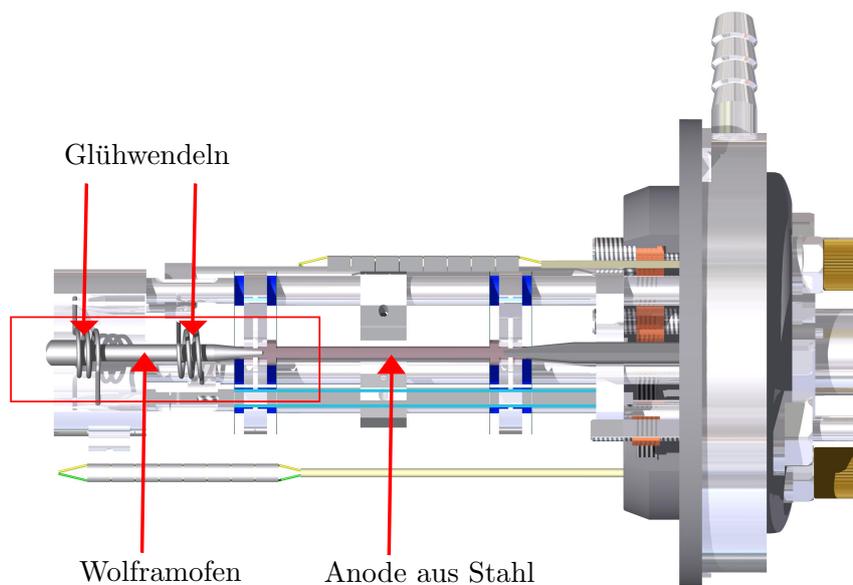


Abbildung 1.1: Querschnitt durch den vorderen Teil der Oberflächenionisationsquelle, mit der die Implantation von Rubidium durchgeführt wird. Der Ofen und die Glühwendeln sind hier rot umrandet. Im Folgenden wird auch nur dieser Bereich betrachtet und simuliert (Abbildung gezeichnet von Christian Kotz).

Dafür steht der konischen Teil des Wolframofens in Kontakt mit einem Stahlrohr, was auf einem positiven Potential von 500 bis 1000 V liegt. Die Glühwendeln hingegen befinden sich auf einem negativen Potential und bei ausreichend hoher Differenz beider Potentiale werden die Elektronen im bestehenden Feld beschleunigt und treffen auf den Wolframofen. Durch die Stöße der Elektronen erhitzt sich der Wolframofen.

Der vordere Teil des Ofens soll eine Temperatur von rund 3000 K erreichen, woraus ein hoher Wärmeverlust durch Strahlung resultiert. Das Abschätzen einer Temperaturverteilung wird durch diesen Sachverhalt erschwert.

Das zuvor beschriebene System wird jedoch auf die vordere Glühwendel reduziert, da die Kenntnis der Temperaturverteilung des Wolframofens ohne Heizen der zweiten Glühwendel von Bedeutung ist. Zwar kann das Problem teilweise analytisch gelöst werden, jedoch vernachlässigt diese Lösung kompliziertere Abhängigkeiten der Wärmeleitfähigkeit von der Temperatur und kann die Geometrie des eigentlichen Ofens nicht berücksichtigen.

In dieser Bachelorarbeit soll dieses Problem numerisch unter Berücksichtigung aller Abhängigkeiten außer dem Stahlkontakt gelöst werden. Mit einer Messung sollen die Ergebnisse der Simulation verifiziert werden.

2 Problemstellung

Der Wolframofen besitzt eine besondere Geometrie, die die Berechnung einer Temperaturverteilung erschwert. In Abbildung 2.1 ist die physikalische Situation skizziert, mit der sich diese Bachelorarbeit beschäftigt. Der Ofen wird an einer Seite mit der konstanten

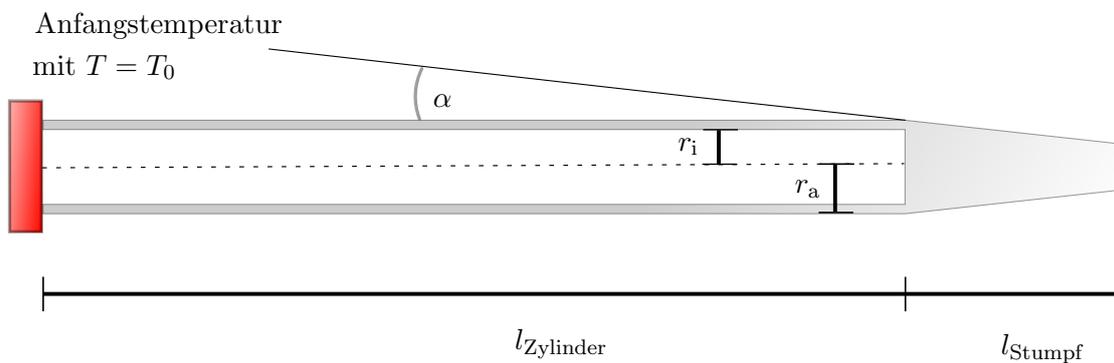


Abbildung 2.1: Querschnitt durch den Wolframofen. Es sind alle bekannten Größen eingezeichnet: r_a und r_i sind jeweils der Außen- und Innenradius des Wolframofens. α ist der Winkel, mit dem der Konus zuläuft. l_{Zylinder} und l_{Stumpf} sind die jeweiligen Längen des zylindrischen und konischen Teils. In rot ist die Wärmequelle mit der konstanten Anfangstemperatur T_0 eingezeichnet, die den Ofen an diesem Ende beheizt.

Temperatur $T = T_0$ erhitzt und nach einer Zeit stellt sich ein stationärer Zustand ein. T_0 kann Werte von rund 3000 K erreichen und befindet sich somit in einem Temperaturbereich, in dem die abgestrahlte Wärme nicht mehr vernachlässigt werden kann. Somit fließt durch den Ofen ein Wärmestrom, der jedoch auch durch die Abstrahlung an der Außenfläche bestimmt wird.

Ein zusätzliches Problem ist die Geometrie des Ofens. Der vordere Teil des Ofens ist ein Hohlzylinder und am Ende befindet sich ein Kegelstumpf, was eine vollständige analytische Lösung der Problematik kompliziert macht. Erschwerend kommt das Abstrahlen der Wärme an der Stirnfläche hinzu.

Es sind nur die oben eingezeichneten Größen gegeben: die Länge des zylindrischen und konischen Teils (l_{Zylinder} und l_{Stumpf}), der Innen- und Außenradius des Hohlzylinders (r_i und r_a), der Winkel α , mit dem der Konus zuläuft, und die konstant gehaltene Anfangstemperatur T_0 . Randbedingungen, wie z.B. die Endtemperatur sind nicht gegeben.

3 Theoretischer Hintergrund

Insgesamt gibt es drei Arten von Wärmetransport: Konvektion, Leitung und Strahlung. Da der Wolframofen auf eine Temperatur in einem Bereich erhitzt wird, in dem die abgestrahlte Wärme nicht vernachlässigt werden kann, ist für die Betrachtung der Temperaturverteilung die Abstrahlung interessant, da durch sie das System Wärme verliert. Weiterhin spielt die Wärmeleitung für den Wärmefluss im Zylinder eine große Rolle. Da sich nach dem Erhitzen des Ofens ein Temperaturgleichgewicht einstellt, gibt es keinerlei Zeitabhängigkeit, die berücksichtigt werden muss. Zusätzlich befindet sich das Wolframobjekt im Vakuum und somit schließt die physikalische Situation Konvektion aus. Diese Art des Wärmetransportes wird daher nicht betrachtet.

3.1 Wärmeleitung

Nach dem zweiten Hauptsatz der Thermodynamik fließt Wärme immer von der höheren zur niedrigeren Temperatur. Dieser Wärmefluss ist für ein Objekt mit der Querschnittsfläche A , Wärmeleitfähigkeit λ und Abstand s zwischen den Temperaturen T_A und T_B definiert als

$$\Phi_L = \dot{Q} = \lambda \cdot \frac{A}{s} (T_A - T_B). \quad (3.1)$$

Alternativ kann auch eine Wärmestromdichte über einen Temperaturgradienten definiert werden [Stö00].

$$\dot{q} = -\lambda \frac{\partial T}{\partial z} \quad (3.2)$$

Weiterhin kann die Wärmeleitung auch über die Wärmeleitungsgleichung

$$\frac{\partial}{\partial t} T(\vec{x}, t) - \lambda \cdot \Delta T(\vec{x}, t) = f(\vec{x}, t) \quad (3.3)$$

beschrieben werden.

Hierbei berücksichtigt $f(\vec{x}, t)$ Wärmequellen in dem zu berechnenden Objekt und $\frac{\partial}{\partial t} T(\vec{x}, t)$ die Zeitabhängigkeit der Temperaturverteilung. Bei dem Wolframofen handelt es sich um einen quellenfreien und stationären Fall und Gleichung 3.3 vereinfacht sich so zu einer Laplace-Gleichung [Mei78].

Die Wärmeleitfähigkeit λ ist von der Temperatur abhängig. Für Wolfram ist die Abhängigkeit in Abbildung 3.1 dargestellt [Smi93][IAE91][Inc85].

Unter Verwendung eines Polynoms ergibt sich durch Anpassung an die Messdaten für die Leitfähigkeit

$$\lambda(T) = 240,51 \frac{\text{W}}{\text{m} \cdot \text{K}} - 0,2899 \frac{\text{W}}{\text{m} \cdot \text{K}^2} T + 2,5403 \cdot 10^{-4} \frac{\text{W}}{\text{m} \cdot \text{K}^3} T^2 - 1,0263 \cdot 10^7 \frac{\text{W}}{\text{m} \cdot \text{K}^4} T^3 + 1,5238 \cdot 10^{11} \frac{\text{W}}{\text{m} \cdot \text{K}^5} T^4 \quad (3.4)$$

Die Anpassung des Polynoms ist jedoch nur für den Bereich der Messdaten (293 K bis 2500 K) gültig [ER12]. Das Polynom, das die Leitfähigkeit beschreibt, steigt ab Temperaturen von 2500 K wieder an. Da das Verhalten der Messwerte im Bereich von mehr als 1000 K einem linearen ähnelt, wird der Verlauf für Temperaturen über 1000 K dementsprechend extrapoliert. Über die Geradenanpassung ergibt sich

$$\lambda(T) = -1,4(9) \cdot 10^{-3} \frac{\text{W}}{\text{m} \cdot \text{K}^2} \cdot T + 130(2) \frac{\text{W}}{\text{m} \cdot \text{K}}. \quad (3.5)$$

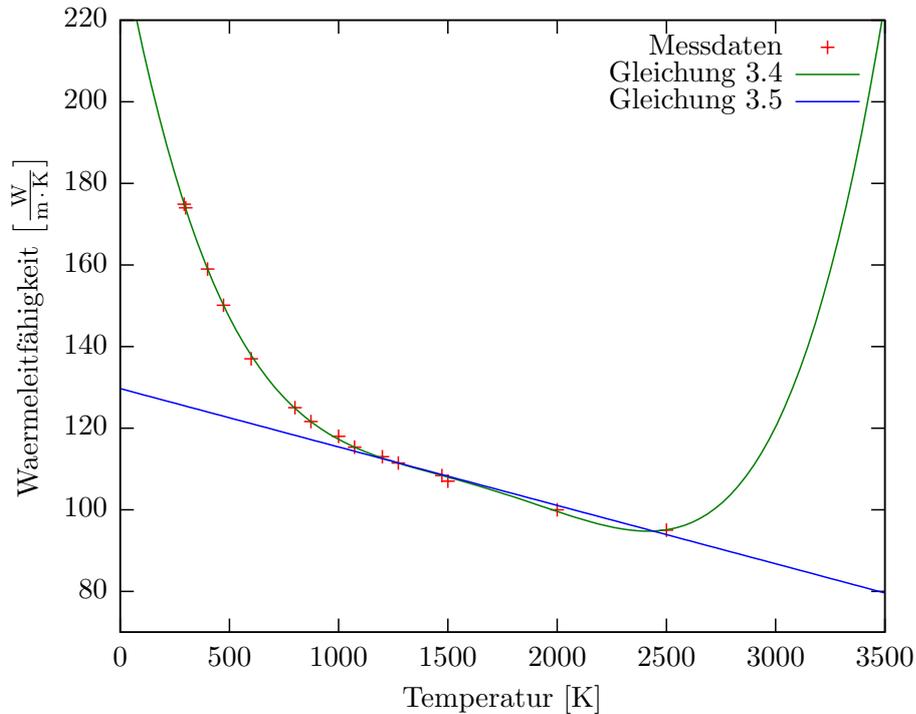


Abbildung 3.1: Wärmeleitfähigkeit von Wolfram in Abhängigkeit von der Temperatur. Gezeigt sind die Messdaten aus [IAE91] [Inc85] [Smi93], der Fit mit einem Polynom für den Temperaturbereich von 293 K bis 2500 K und der Fit für Temperaturen ab 1000 K, für den ein lineares Verhalten angenommen wird.

3.2 Wärmestrahlung

Die Strahlungsleistung eines Körpers wird durch das Stefan-Boltzmann-Gesetz beschrieben.

$$P = \Phi_A = \varepsilon \cdot \sigma \cdot A \cdot T^4 \quad (3.6)$$

ε ist der Emissionskoeffizient, A die Oberfläche des strahlenden Körpers und σ die Stefan-Boltzmann-Konstante. Bemerkenswert ist die Abhängigkeit von T^4 .

Der Emissionskoeffizient ε gibt die Fähigkeit eines Körpers an, in Relation zu einem schwarzen Körper Wärme abzustrahlen. Ein schwarzer Körper ist ein idealer Strahler, der die maximale Abstrahlung und somit den höchsten Wert von $\varepsilon = 1$ besitzt. Der Emissionskoeffizient ist abhängig von Temperatur, Material und Oberfläche.

In Abbildung 3.2 ist für Stahl die Emissivität gegen die Rauheit der Oberfläche aufgetragen. Nach [Sko10] besteht für Wolframdraht ein vergleichbarer Zusammenhang zwischen ε und der Rauheit.

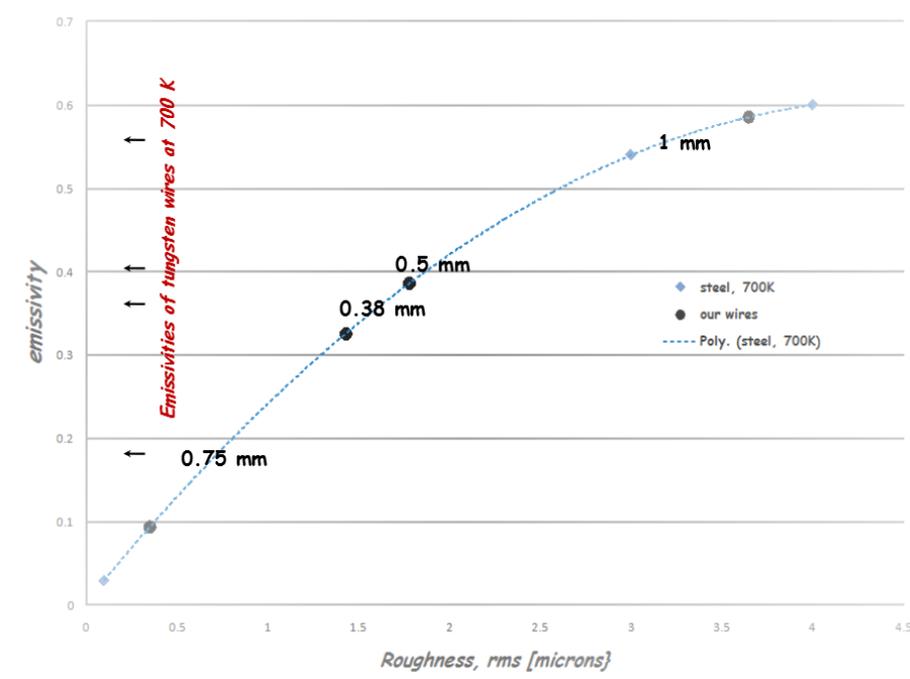


Abbildung 3.2: Emissivität in Abhängigkeit von der Rauheit der Oberfläche (Effektivwert (rms) für die mittlere Tiefe des Oberflächenprofils). Die schwarzen Messpunkte sind Daten für Wolframdraht mit verschiedenen Dicken und liegen gut auf dem Graphen für Stahl. [Sko10]

Die Rauheit ist ein Maß für die mittlere Größe der Vertiefungen auf der Oberfläche des Wolframobjekts.

Da der Emissionskoeffizient zwischen 0,1 und 0,6 (siehe Abbildung 3.2) schwanken kann, ist die Kenntnis über die Rauheit unbedingt notwendig. Gerade bei hohen Temperaturen ist der Verlust über Strahlung sehr hoch und deshalb spielt dieser Faktor eine große Rolle. Die Rauheit der Testobjekte wird in Kapitel 7.2 bestimmt.

4 Lösungen

In diesem Kapitel wird sowohl eine analytische als auch eine numerische Lösung für das in Kapitel 2 beschriebene Problem betrachtet.

4.1 Analytische Lösung

Da auch Messungen für einen einfachen Wolframzylinder durchgeführt werden, wird an dieser Stelle ein Hohlzylinder betrachtet, was zudem auch eine analytische Lösung ermöglicht.

Bekannt sind die konstante Anfangstemperatur des Zylinders, die Länge und die Querschnittsfläche.

Wenn die Abstrahlung an der Stirnfläche vernachlässigt und angenommen wird, dass der Emissionskoeffizient $\varepsilon(T)$ die Gestalt $\sum_{i=0}^n c_i \cdot T^{\alpha_i}$ für $\alpha_i \in \mathbb{Z}$ und $c_i \in \mathbb{R}$ besitzt und die Wärmeleitfähigkeit λ nicht temperaturabhängig ist, ist eine analytische Lösung möglich. Der Zylinder wird in Scheiben der Länge dz unterteilt. Für eine derartige Scheibe kann folgende Bilanz der Wärmeströme aufgestellt werden:

$$\Phi_{\text{in}} - \Phi_{\text{out}} = \Phi_{\text{strahl}} \quad (4.1)$$

$$-\lambda \left. \frac{dT}{dz} \right|_z A_Q + \lambda \left. \frac{dT}{dz} \right|_{z+dz} A_Q = \varepsilon(T) \sigma 2\pi r_a T^4 dz \quad (4.2)$$

Hierbei ist A_Q die Querschnittsfläche eines Hohlzylinders, r_a der Aussen- und r_i der Innenradius. Der Wärmeverlust zwischen den Stellen z und $z + dz$ geschieht also über Abstrahlung an der Aussenfläche. Daraus resultiert eine Temperaturdifferenz $\Delta T = T - T'$. Die obige Gleichung kann in

$$\lambda(r_a^2 - r_i^2) \frac{d^2 T}{dz^2} dz - \varepsilon(T) 2\sigma r_a T^4 dz = 0 \quad (4.3)$$

umgeschrieben werden. Es wird für die folgenden Rechnungen $\varepsilon(T) = \varepsilon$ angenommen und $P = \frac{dT}{dz}$ substituiert, woraus sich $\frac{dP}{dz} = \frac{dP}{dT} \frac{dT}{dz} = P' \cdot P$ ergibt.

$$P' \cdot P = \frac{2\varepsilon\sigma r_a}{\lambda(r_a^2 - r_i^2)} T^4 \quad (4.4)$$

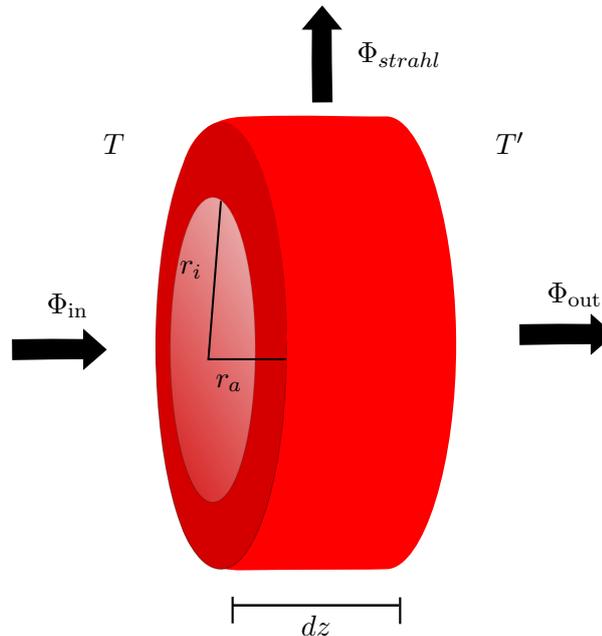


Abbildung 4.1: Skizze zur analytischen Betrachtung. Ein Zylinder mit der Länge dz besitzt am Anfang die Temperatur T und am Ende die Temperatur T' , wobei $T' < T$ gilt. Der Fluss Φ_{out} , der dieses Segment verlässt, setzt sich aus dem Fluss Φ_{in} , der zuvor hineinkommt, und Φ_{strahl} , der durch Strahlung verlorengegangene Wärme zusammen. Die Querschnittsfläche wird durch den Innenradius r_i und den Außenradius r_a bestimmt.

Somit folgt:

$$P = \frac{dT}{dz} = -\sqrt{\frac{4\varepsilon\sigma r_a}{5\lambda(r_a^2 - r_i^2)}} T^{\frac{5}{2}} \quad (4.5)$$

$$\int_{T_0}^{T_E} \frac{dT}{T^{\frac{5}{2}}} = -\sqrt{\frac{4\varepsilon\sigma r_a}{5\lambda(r_a^2 - r_i^2)}} \int_0^l dz \quad (4.6)$$

$$T_E = \frac{1}{\left(3l \cdot \sqrt{\frac{\varepsilon r_a}{5\lambda(r_a^2 - r_i^2)}} + T_0^{-\frac{3}{2}}\right)^{\frac{2}{3}}} \quad (4.7)$$

l ist die Länge des Zylinders und damit die obere Integrationsgrenze für die Temperaturänderung pro Zylinderscheibe.

Die Lösung ist auf kompliziertere Geometrien – wie den Wolframofen – nicht anwendbar und berücksichtigt das Abstrahlen an der Stirnfläche nicht. Zudem gibt diese Lösung keinen Aufschluss über die Temperaturverteilung selbst, sondern nur über die Endtemperatur des Zylinders.

Die analytische Lösung geht zusätzlich davon aus, dass die Wärmeleitfähigkeit λ nicht von der Temperatur abhängig ist. Um später auch Ergebnisse mit der analytischen Lösung zu berechnen, wird eine mittlere Leitfähigkeit der Form

$$\bar{\lambda}(T) = \frac{1}{T_E - T_A} \int_{T_A}^{T_E} \lambda(T) dT \quad (4.8)$$

$$= 7 \cdot 10^{-3} \frac{\text{W}}{\text{m} \cdot \text{K}^2} (T_E + T_A) + 130 \frac{\text{W}}{\text{m} \cdot \text{K}} \quad (4.9)$$

angenommen. Die Mittelung ist für $\lambda(T)$ nur für $T \geq 1000 \text{ K}$ gültig, da hier $\lambda(T)$ aus Gleichung 3.5 verwendet wurde, die nur Gültigkeit für diesen Temperaturbereich besitzt. Eine Alternative wäre, Gleichung 3.3 für den stationären und quellenfreien Fall zu lösen. Eine möglicher Ansatz wäre eine Green'sche Funktion. Dies geht jedoch über den Rahmen dieser Arbeit hinaus.

4.2 Numerische Lösung

In diesem Kapitel wird auf die numerischen Lösungen für die Temperaturverteilung eines Zylinders, eines Kegelstumpfes und eines vollständigen Ofens eingegangen.

4.2.1 Berechnung der Temperaturverteilung eines Zylinders

Zunächst wird lediglich ein Zylinder unter Berücksichtigung der Abstrahlung an der Stirnfläche betrachtet. Die Ausgangssituation ist ansonsten mit der der analytischen Lösung identisch.

Das Prinzip des Programms beruht auf einer immer feiner werdenden Unterteilung des Zylinders. Diese Unterteilung sieht vor, den Zylinder beliebig häufig zu halbieren. Das Programm berechnet bei n Halbierungen die Anfangs- und Endtemperaturen für 2^n kleinere Zylindersegmente. Im Folgenden wird die Berechnung der Temperaturen der einzelnen Segmente erklärt.

Der gesamte Zylinder wird zunächst halbiert. Jede diese Zylinderhälften besitzt eine Anfangs- und Endtemperatur. Die Endtemperatur einer Hälfte ist die Anfangstemperatur der darauffolgenden Hälfte. Anschließend werden beide Hälften erneut in der Mitte geteilt. Der gesamte Zylinder besteht jetzt aus vier Zylindern und es müssen nun alle T_i für $i \leq 4$ bestimmt werden. Durch die Unterteilung wird jedes T_i zu T_{2i} , d.h. allen Temperaturen mit ungeraden Indizes muss ein Wert zugewiesen werden. Dies geschieht über eine einfache Mittelung der benachbarten Temperaturen.

$$T_{2i+1} = \frac{1}{2} (T_{2i} + T_{2i+2}) \quad (4.10)$$

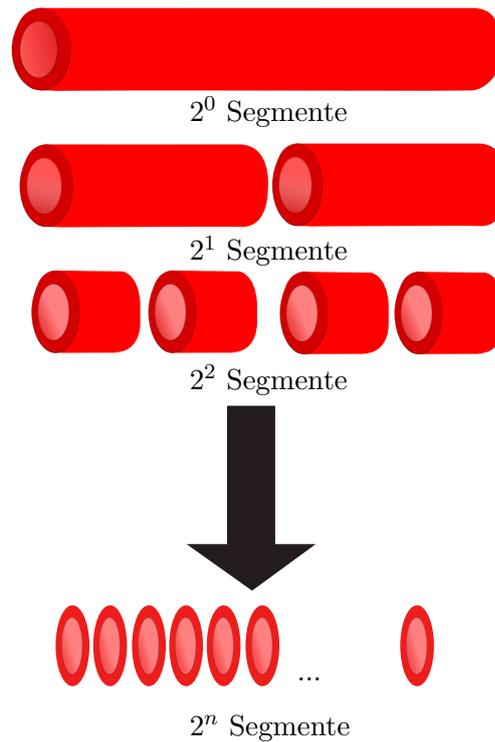


Abbildung 4.2: Segmentierung des Zylinders. Die Anzahl der Segmente hängt von der Häufigkeit der Halbierungen n ab und berechnet sich nach 2^n .

Nach jeder Unterteilung werden die Werte für die einzelnen Temperaturen der Segmente bestimmt. Hierfür wird wie beim analytischen Ansatz die Gleichung 4.1 verwendet, aus der

$$\lambda(T_i) \frac{A_Q}{l} \cdot (T_{i-1} - T_i) - \lambda(T_i) \frac{A_Q}{l} \cdot (T_i - T_{i+1}) = \varepsilon \cdot \sigma \cdot A_M \cdot T_{i,Z}^4, \quad (4.11)$$

folgt. A_Q ist die Querschnittsfläche, A_M die Mantelfläche und T_{i-1} , T_i und T_{i+1} die Temperaturen der Zylindersegmente. Der Wärmestrom der in ein Zylindersegment hineinfließt, spaltet sich in einen Strom für die Abstrahlung und einen anderen für den Abfluss in das nächste Segment auf.

Das Programm berechnet so jedes beliebige T_i aus T_{i+1} und T_{i-1} . Da die Anfangstemperatur T_0 die einzige gegebene Temperatur ist, beginnt das Programm bei $i = 1$ und endet beim letzten Segment mit dem Index 2^n . Es werden nun nacheinander T_1, T_2, \dots, T_N berechnet.

T_i wird unter Verwendung von T_{i+1} und T_{i-1} ermittelt. Im nächsten Schritt wird T_{i+1} mit T_i und T_{i+2} berechnet, wodurch sich die Grundlage für die zuvor durchgeführte Berechnung von T_i ändert. Das Programm muss somit erneut alle Temperaturen T_i berechnen, d.h. die Berechnung für alle T_i muss mit Gleichung 4.11 nacheinander wiederholt werden. Dabei nähern sich die Werte immer weiter den tatsächlichen an. Die Berechnung wird

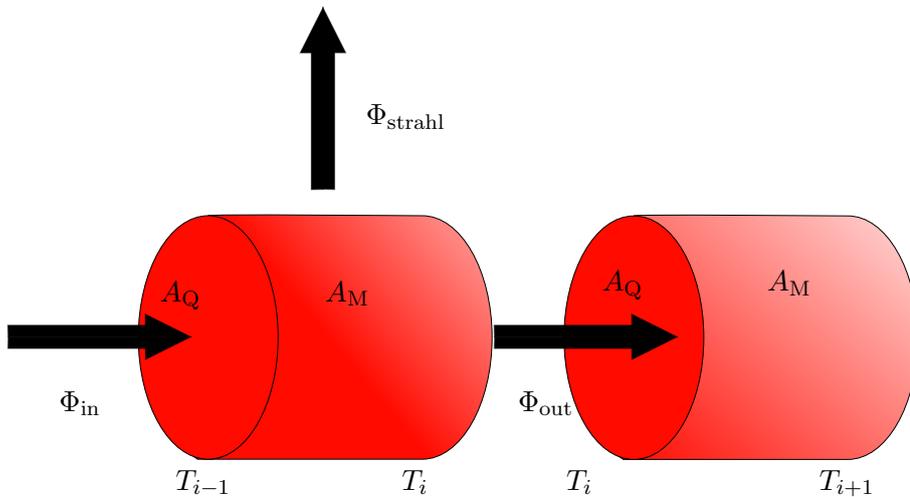


Abbildung 4.3: Die Temperaturströme zweier Zylindersegmente. A_Q ist die Querschnittsfläche und A_M die Mantelfläche eines Segmentes. T_{i-1} , T_i und T_{i+1} die jeweiligen Temperaturen der Zylindersegmente. Die Differenz zwischen den Wärmeströmen, die hinein- und hinausfließen, ist die abgestrahlte Wärme.

abgebrochen, bis sich die Endtemperaturen T_N zwischen den Iterationsschritten nicht mehr um den eingestellten Betrag ändern (siehe Kapitel 4.2.5).

Die Berechnung der Temperatur an der Stirnfläche des Zylinders muss gesondert betrachtet werden, da das letzte Segment des Zylinders kein nachfolgendes Segment besitzt, in das Wärme abfließen kann. Zusätzlich muss berücksichtigt werden, dass dieses letzte Segment nicht nur Wärme über die Mantelfläche abstrahlt, sondern auch über die Stirnfläche. Die Endtemperatur T_N berechnet sich somit aus

$$\varepsilon\sigma(A_M + A_S)T_N^4 = \lambda(T_{N-1})\frac{A_Q}{l}(T_{N-1} - T_N), \quad (4.12)$$

wobei A_M die Mantelfläche, A_S die Stirnfläche und A_Q die Querschnittsfläche ist.

Nach mehreren Iterationsschritten sind nun alle Temperaturen für vier Zylindersegmente bekannt.

Bei Gleichung 4.11 und 4.12 wird angenommen, dass die Wärmestrahlung bei einer konstanten Temperatur T_i stattfindet. Dabei wird jedoch nicht berücksichtigt, dass die Abstrahltemperatur zwischen T_{i-1} und T_i , also der Anfangs- und Endtemperatur des aktuellen Segmentes, liegen muss. Um den Fehler dieser Annahme möglichst gering zu halten, ist es notwendig, die Segmente im Verlauf des Programms immer weiter zu verkleinern.

Daher wird – falls eingestellt – jeder dieser vier Zylinder erneut halbiert und die dabei neuen Temperaturen wie schon oben beschrieben durch die Mittelung der benachbarten Temperaturen genähert. Mit diesen Temperaturen beginnt das Programm unter Verwen-

dung von Gleichung 4.11 von neuem zu iterieren bis die eingestellte Genauigkeit erreicht worden ist (siehe Kapitel 4.2.5).

4.2.2 Berechnung der Temperaturverteilung eines Kegelstumpfes

Der Wolframofen besteht nicht nur aus einem Hohlzylinder, sondern besitzt noch einen massiven Kegelstumpf, der gesondert betrachtet werden muss (siehe Abbildung 2.1). Das Prinzip der Segmentierung, das zuvor schon für den Zylinder verwendet wurde, wird auch auf den Kegelstumpf angewendet.

Es gibt jedoch drei Dinge die berücksichtigt werden müssen: die Verringerung der Querschnittsfläche und der Mantelfläche mit größer werdender z -Komponente und die Tatsache, dass der Zylinderstumpf des späteren Wolframofen nicht hohl ist. Die Querschnitts- und Mantelfläche hängen vom Radius ab, dessen Abnahme vom Winkel relativ zur Grundfläche des Kegelstumpfes bestimmt wird. Der Radius berechnet sich mit dem Anfangsradius r_0 über

$$r(z) = r_0 - \tan(\alpha) \cdot z, \quad (4.13)$$

wobei α der Winkel ist, mit dem der Konus zuläuft. Für jedes kleinere Segment des Konus, welches ebenfalls ein Konus ist, wird für den Wärmefluss durch die Querschnittsfläche eine mittlere Querschnittsfläche angenommen, die sich aus der Mittelung der Radien von Grundfläche und der Deckelfläche ergibt. Das Programm verwendet eine Bilanzgleichung wie in Gleichung 4.11:

$$\lambda(T_i) \frac{\bar{A}_{Q,i}}{l} (T_{i-1} - T_i) - \lambda(T_i) \frac{\bar{A}_{Q,i+1}}{l} (T_i - T_{i+1}) = \varepsilon \sigma A_{M,i} T_{i,Z}^4. \quad (4.14)$$

Hier ist \bar{A}_Q die mittlere Querschnittsfläche des jeweiligen Segmentes, l die Länge eines einzelnen Segmentes, $\lambda(T)$ die Wärmeleitfähigkeit und A_M die Mantelfläche, über die die Abstrahlung stattfindet. Bei der Berechnung der Abstrahlfläche wird auch hier die Abnahme des Radius berücksichtigt.

Das Programm für den Kegelstumpf mit einem Winkel von 0° und einem Anfangsradius von r_0 und der Gesamtlänge l_0 liefert das gleiche Ergebnis wie das Programm für den Zylinder mit den identischen Parametern.

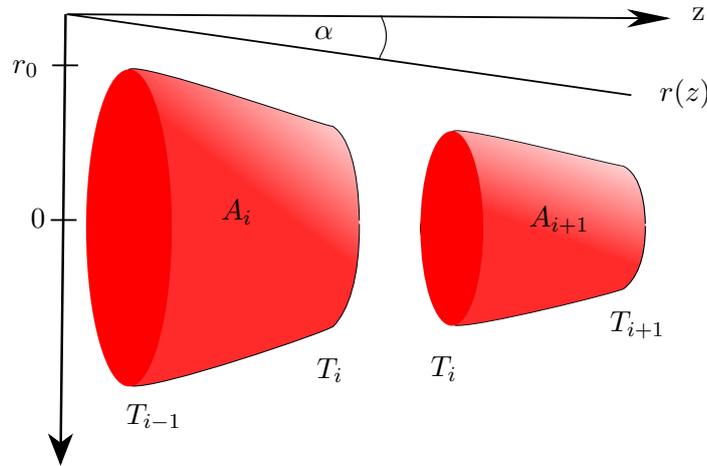


Abbildung 4.4: Unterteilung des Zylinderstumpfes. A_i und A_{i+1} sind die Mantelflächen des i -ten und $(i+1)$ -ten Segmentes. T_{i-1} , T_i und T_{i+1} sind die jeweiligen Temperaturen an den Grundflächen der Stümpfe. r_0 ist der Anfangsradius und $r(z)$ ist der Radius abhängig von der z -Koordinate. α ist der Winkel, mit dem der Konus zuläuft.

4.2.3 Berechnung der Temperaturverteilung des vollständigen Ofens

Um nun die komplette Geometrie des Ofens berücksichtigen zu können, müssen die Programme für den Kegelstumpf und für den Zylinder zusammengefügt werden. Lediglich der Übergang vom zylindrischen zum konischen Teil muss gesondert betrachtet werden. Hierfür wird die Gleichung

$$\lambda(T_{i,Z}) \frac{A_{Q,Z}}{l} \cdot (T_{i-1,Z} - T_{i,Z}) = \lambda(T_{i,Z}) \frac{A_{Q1,S}}{l} \cdot (T_{i,Z} - T_{1,S}) + \varepsilon \cdot \sigma \cdot A_{M,1} \cdot T_{i,Z}^4 \quad (4.15)$$

verwendet.

Hierbei ist $A_{Q,Z}$ die Querschnittsfläche des Zylinders, $T_{i,Z}$ ist die Endtemperatur des Zylinders und zugleich die Anfangstemperatur des Kegelstumpfes, $A_{Q1,S}$ die erste mittlere Querschnittsfläche und $A_{M,1}$ die erste Mantelfläche des Kegelstumpfes.

Auch hier ist der Algorithmus derselbe wie für den Zylinder und den Konus. Die Anzahl der Segmente ist für den Zylinder und den Stumpf identisch.

4.2.4 Anfangswerte

Um die Zahl der Iterationsschritte zu Beginn des Programms gering zu halten, werden die Anfangswerte für die Temperaturen nach der ersten Halbierung gesetzt. Unter Vernachlässigung des Abflusses der Wärme in das nachfolgende Segment wird die Endtemperatur der zwei Hälften berechnet. Es müssen zunächst zwei neue Temperaturen bestimmt werden. Dies geschieht über

$$\Phi_{\text{in}} = \Phi_{\text{strahl}} \quad (4.16)$$

$$\lambda(T_0) \frac{A_Q}{l'} (T_i + T_{i+1}) = \varepsilon \sigma A_M T_i^4, \quad (4.17)$$

wobei $i = 0, 1$ sein kann. Für $i = 0$ sind alle möglichen Parameter gegeben, um T_1 zu berechnen. T_0 ist die konstant gehaltene Anfangstemperatur, A_Q die Querschnittsfläche, $\lambda(T_0)$ die Wärmeleitfähigkeit für $T = T_0$ und l' die Länge des Zylindersegmentes. Diese nichtlineare Gleichung wird über eine Nullstellenbestimmung mittels Bisektion gelöst. Mit dem bestimmten T_1 kann im nächsten Schritt T_2 bestimmt werden. Es sind nun grob abgeschätzte Werte für T_1 und T_2 bekannt.

Für vier Segmente werden nach der Abschätzung für zwei Segmente die fehlenden Temperaturen durch Mittelung nach Formel 4.10 berechnet.

4.2.5 Abbruchkriterium

Das Programm bricht die Iteration ab, wenn sich die Endtemperaturen zweier Iterationsschritte nicht mehr als um einen festgelegten Betrag unterscheiden.

Dieser kann beliebig eingestellt werden. Da sich bei einer feineren Unterteilung die benachbarten Temperaturen nur noch geringfügig unterscheiden, wurde ein segmentabhängiges Abbruchkriterium gewählt. Somit wird gewährleistet, dass es nicht zu einem vorzeitigen Abbruch kommt. Für die nachfolgenden Ergebnisse wurde ein Betrag von $\frac{1}{N}$ in Kelvin eingestellt, wobei N die Zahl der Segmente ist, mit der aktuell in der jeweiligen Iteration gerechnet wird.

5 Verhalten der Simulation

In diesem Kapitel wird das Programm bezüglich seines Verhaltens bei Änderung verschiedener Parameter, Konvergenz und Iteration untersucht.

5.1 Abhängigkeit von physikalischen Größen

Um das Programm auf Verhalten in Abhängigkeit von unterschiedlichen Parametern zu untersuchen, wird es verschiedenen Plausibilitätsprüfungen unterzogen. Die nachfolgenden Untersuchungen beschränken sich ausschließlich auf einen Zylinder.

Die Simulation geht von folgender Situation aus:

- Das Objekt befindet sich im Vakuum.
- Es handelt sich um ein stationäres System, d.h. alle Wärmeströme befinden sich in einem Gleichgewicht.
- Es wird ein Strahlungsgleichgewicht im Inneren des Hohlzylinders angenommen.
- Eine Grundfläche wird beheizt und befindet sich auf konstanter Temperatur, während alle anderen Temperaturen keine festen Werte zugewiesen worden sind.
- Der Wärmeabfluss durch die Anode aus Stahl wird vernachlässigt.

5.1.1 Temperaturverteilung bei unterschiedlichen Querschnittsflächen

Die Größe der Querschnittsfläche eines Hohlzylinders richtet sich nach dem Innen- und Außenradius, d.h. mit zunehmendem Innenradius verringert sich die Querschnittsfläche. Deswegen hat ein Hohlzylinder einen größeren Wärmewiderstand.

Durch den höheren Wärmewiderstand kann die Wärme schlechter in den hinteren Teil des Zylinders abfließen und sorgt somit für eine stärkere Abstrahlung im vorderen Teil des Zylinders. Die Wärmeverlust ist bei größeren Wärmewiderstand höher.

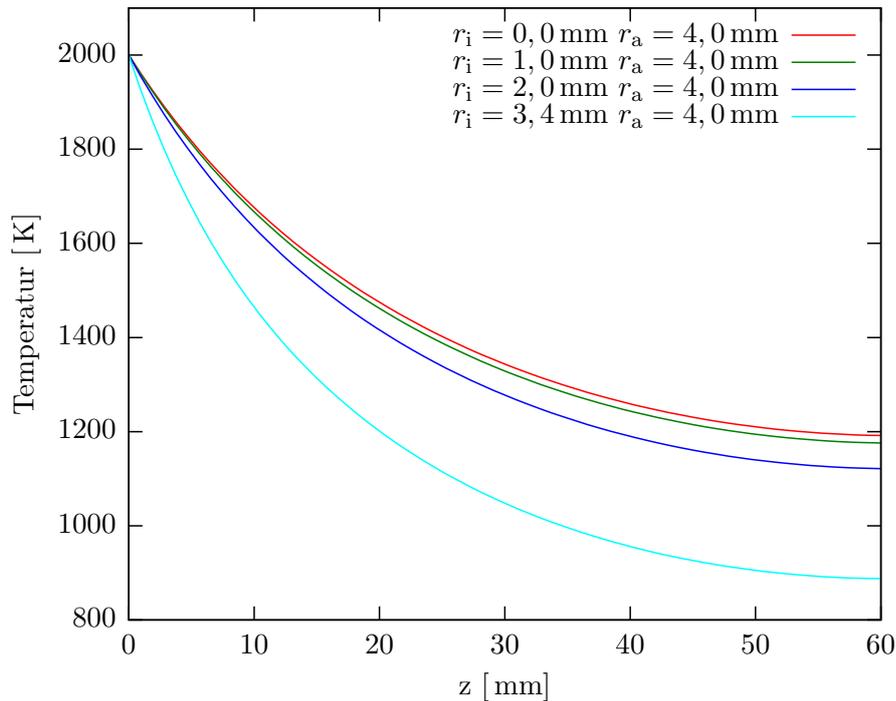


Abbildung 5.1: Temperaturverteilung bei unterschiedlicher Querschnittsfläche und gleicher Mantelfläche. Mit abnehmender Querschnittsfläche sinkt der Wärmeabfluss und dadurch wird mehr Wärme zu Beginn abgestrahlt.

In Abbildung 5.1 sind die Ergebnisse der Simulation dargestellt. Hierfür wurde die Temperatur gegen die z -Koordinate eines Zylinders mit $l = 60$ mm bei einer Anfangstemperatur von 2000 K aufgetragen. Mit abnehmenden Innenradius nimmt der Wärmeverlust im vorderen Teil des Zylinders ab und führt somit zu einer höheren Endtemperatur. Das Ergebnis der Simulation entspricht den Erwartungen.

5.1.2 Temperaturverteilung bei unterschiedlichen Außenflächen

Über die Mantelfläche des Zylinders findet die Abstrahlung statt, d.h. je größer der Außenradius bei gleichbleibender Querschnittsfläche ist, desto höher muss der Wärmeverlust durch Abstrahlung sein. Es ist also bei steigendem Aussenradius ein stärkerer Abfall der Temperaturverteilung entlang des Zylinders zu erwarten.

In Abbildung 5.2 ist das Ergebnis der Simulation für einen Zylinder mit fester Länge, Querschnittsfläche und Anfangstemperatur, aber unterschiedlicher Oberfläche zu sehen. Die Simulation zeigt, dass der Wärmeverlust mit steigendem Außenradius zunimmt.

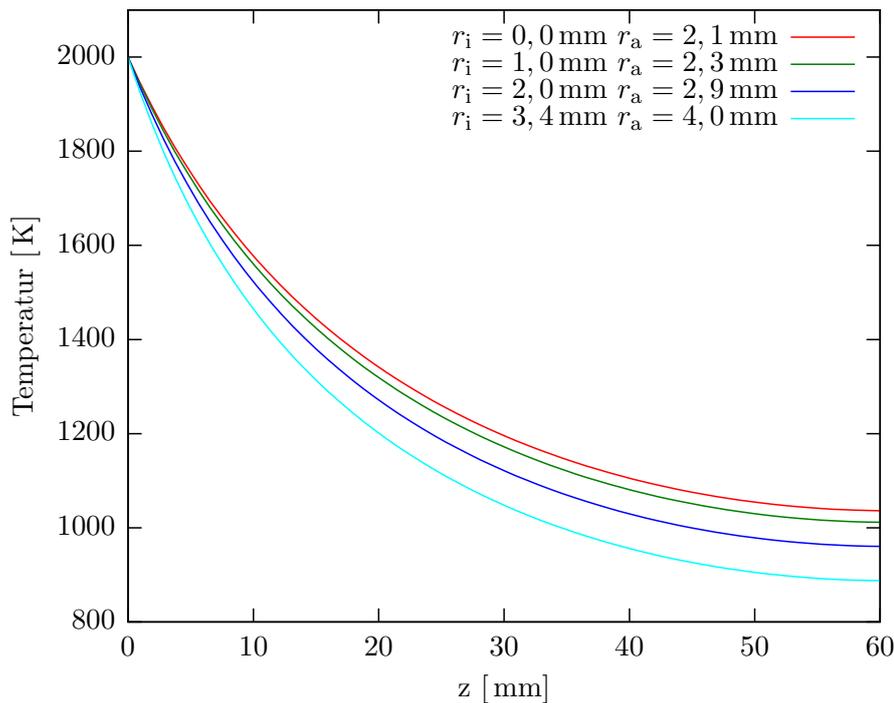


Abbildung 5.2: Temperaturverteilung bei unterschiedlicher Abstrahlfläche, aber gleichbleibender Querschnittsfläche. Ein größerer Außenradius r_a bewirkt einen größeren Temperaturverlust.

5.1.3 Temperaturverteilung bei unterschiedlichen Längen

Die Temperatur an einem Punkt im Zylinder wird durch die Gesamtlänge des Zylinders selbst bestimmt. Je länger der Zylinder ist, desto besser kann die Wärme vom beheizten Ende abfließen, da über die komplette Länge des Zylinders mehr Fläche zur Abstrahlung zur Verfügung steht.

Seien l_1 und l_2 die Längen zweier Zylinder und es gelte $l_1 < l_2$, so sollte die Temperatur an der Stelle z' für den Zylinder mit der Länge l_2 kälter als für l_1 sein.

In Abbildung 5.3 ist das Ergebnis der Simulation dargestellt und es ist zu erkennen, dass auch diese Bedingung erfüllt wird, da für $z' = 30$ mm Temperatur des längsten Zylinders am niedrigsten und die des kürzesten am höchsten ist.

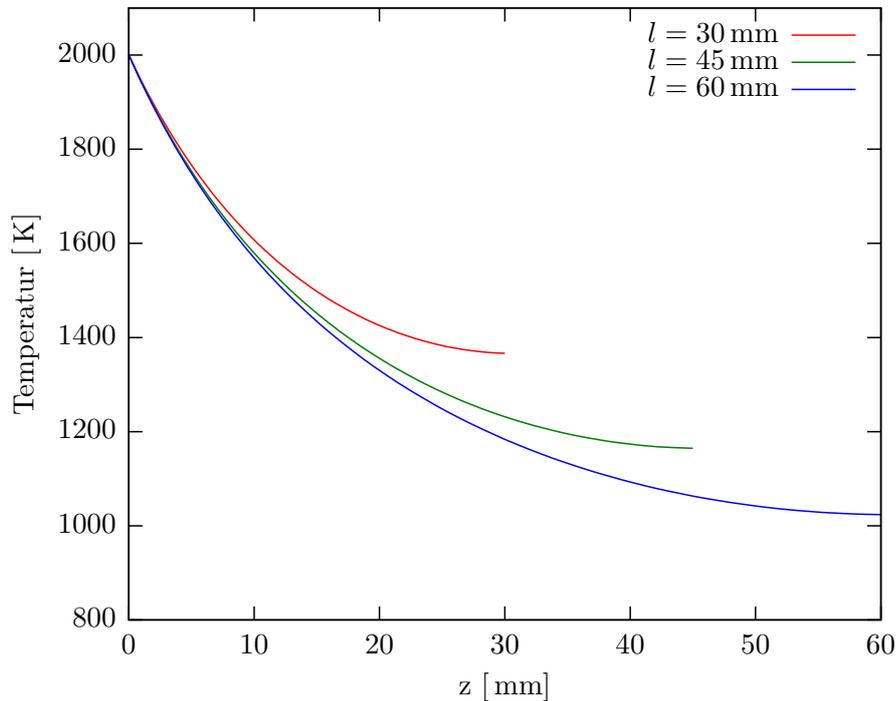


Abbildung 5.3: Temperaturverteilung bei unterschiedlicher Länge. Je länger der Zylinder ist, desto besser kann die Wärme nach hinten abfließen. Ein längerer Zylinder ist an der Stelle z' im vorderen Teil kühler als ein kürzerer.

5.1.4 Temperaturverteilung bei unterschiedlichen Umgebungstemperaturen

Zuvor wurde von einer Umgebungstemperatur von $T_U = 0$ K ausgegangen. Ist $T_U \neq 0$ K, so verändern sich die Strahlungsterme in allen Gleichungen zu

$$\Phi_{\text{Strahl}} = \varepsilon A \sigma (T^4 - T_U^4). \quad (5.1)$$

Durch eine Umgebungstemperatur verringert sich die gesamte Strahlungsleistung. Gleichung 5.1 repräsentiert einen Wärmestrom vom strahlenden Körper und der Umgebung, der sich bei höherer Umgebungstemperatur verringert.

Bei einem Zylinder müsste dies zu einer höheren Endtemperatur führen, da die Abgabe der Wärme an die Umgebung verringert wird.

Abbildung 5.4 zeigt die Temperaturverteilungen für einen Vollzylinder mit $T_U = 2000$ K, $l = 60$ mm und vier unterschiedlichen Umgebungstemperaturen. Wie erwartet führt eine höhere Umgebungstemperatur zu weniger Verlust durch Abstrahlung und somit zu einer höheren Endtemperatur.

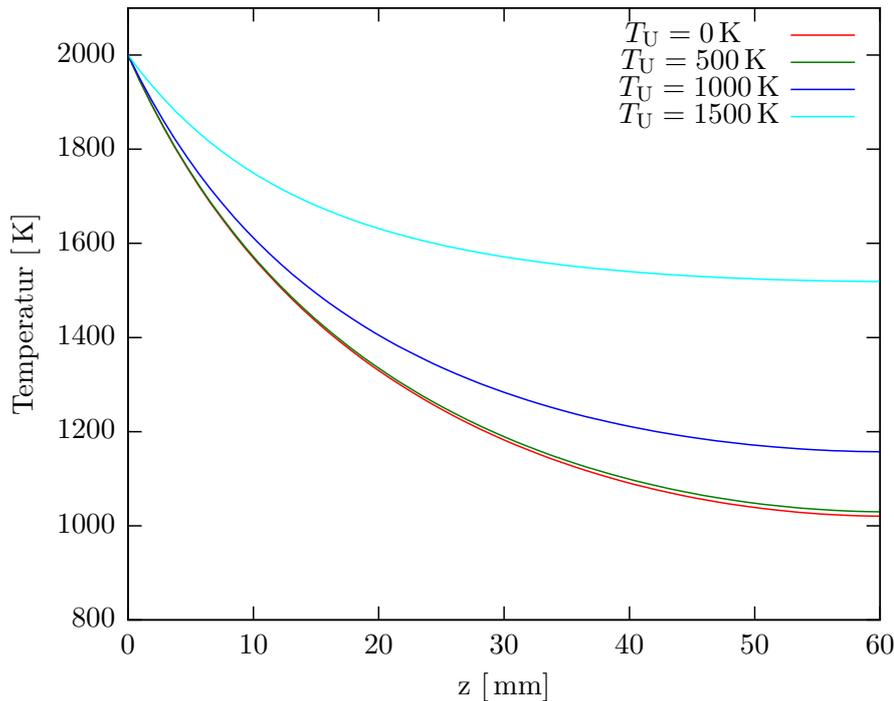


Abbildung 5.4: Temperaturverteilung bei verschiedenen Umgebungstemperaturen. Die Verluste durch Abstrahlung sind bei höheren Umgebungstemperaturen geringer, was eine höhere Endtemperatur zur Folge hat.

5.1.5 Temperaturverteilung bei unterschiedlichen Emissionskoeffizienten

Der Emissionskoeffizient ε spielt – wie die Fläche – bei der Abstrahlung eine große Rolle. Ein höherer Emissionskoeffizient muss auch für eine höhere Abstrahlung sorgen und somit auch für niedrigere Temperaturen entlang des Zylinders. In Abbildung 5.5 ist das Ergebnis der Simulation für drei verschiedene Emissionskoeffizienten dargestellt. Auch hier stimmen die Resultate mit den Erwartungen überein.

5.1.6 Unabhängigkeit von den Anfangsbedingungen

Wie schon zuvor erwähnt, arbeitet das Programm anfangs mit einer Abschätzung und legt mit ihr Anfangswerte für die Temperaturen fest, die es für die erste Iteration verwendet. Die Simulation sollte unabhängig von diesen Anfangswerten immer das gleiche Ergebnis liefern.

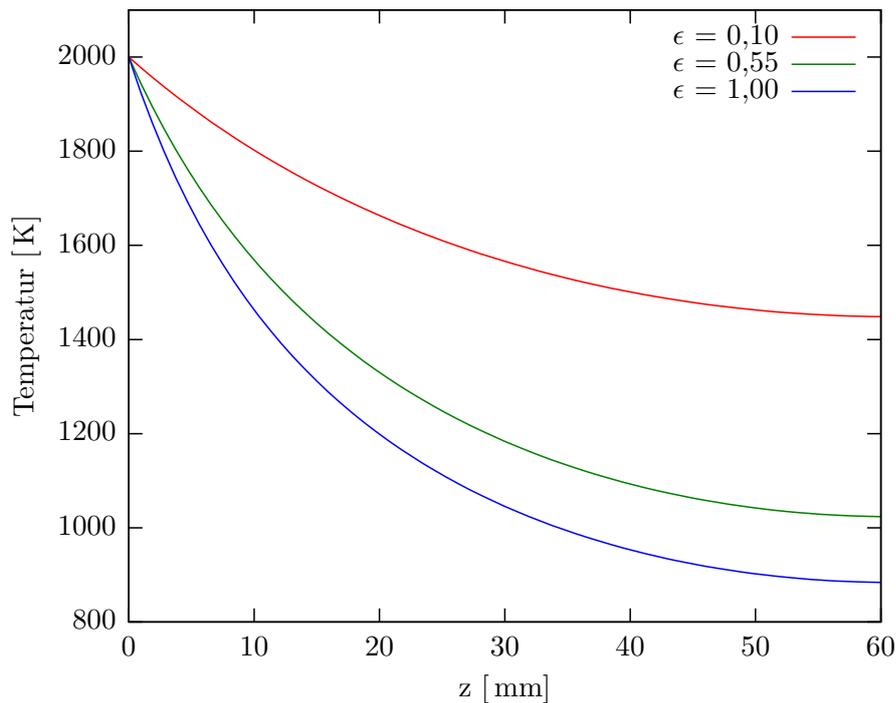


Abbildung 5.5: Simulierte Temperaturverteilungen bei unterschiedlichen Emissionskoeffizienten. Je größer der Emissionskoeffizient, desto größer ist der Wärmeverlust.

Um dies zu prüfen, wurden statt der Abschätzung zu Beginn des Programms alle anderen Werte für die Temperatur – außer die Anfangstemperatur – auf Null gesetzt. Das Programm lieferte unabhängig von den Anfangsbedingungen die gleiche Endtemperatur. Es benötigt jedoch mehr Iterationsschritte, da die abgeschätzten Temperaturen den tatsächlichen Werten näher kommen.

5.1.7 Berechnung der Temperaturverteilung bei unterschiedlichen Anfangstemperaturen

Da der Wärmestrom für die Abstrahlung proportional zur vierten Potenz der Temperatur ist, müsste bei steigender Anfangstemperatur auch der Wärmeverlust erheblich größer werden. Die Differenz von Anfangs- und Endtemperatur sollte bei hohen Temperaturen größer sein. In Abbildung 5.6 sind die Ergebnisse der Simulation für einen Vollzylinder mit $l = 60$ mm bei unterschiedlichen Anfangstemperaturen zu sehen. Die Ergebnisse der Simulation zeigen das zuvor genannte Verhalten.

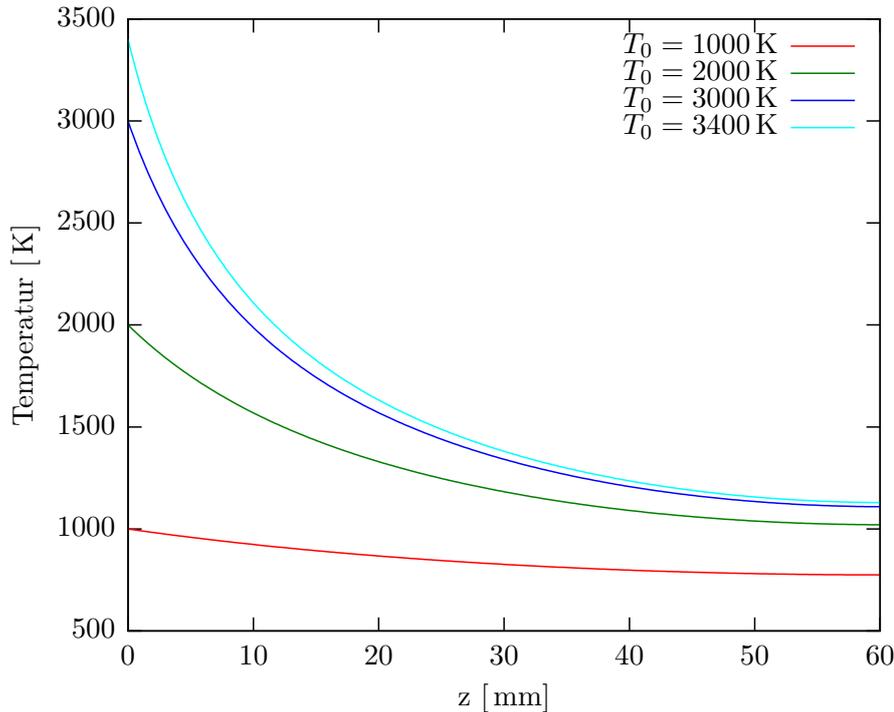


Abbildung 5.6: Simulierte Temperaturverteilungen bei unterschiedlichen Anfangstemperaturen T_0 . Je höher die Anfangstemperatur ist, desto größer ist der Wärmeverlust.

5.2 Iterationsverhalten

Die Temperaturverteilung für vier und acht Segmente nach jeweils einer und fünf Iterationen ist in Abbildung 5.7 dargestellt. Es ist gut zu erkennen, dass sich für vier Segmente die Temperaturverteilung zwischen erster und fünfter Iteration noch erheblich ändert. Bei acht Segmenten ist der Unterschied der Temperaturverteilungen nach der ersten und fünften Iteration minimal.

Das Verhalten bei unterschiedlichen Iterationsschritten für vier Segmente ist auf die Abhängigkeit der Temperatur T_i von T_{i+1} und T_{i-1} zurückzuführen. T_i wird auf der Grundlage von T_{i-1} und T_{i+1} berechnet, welche in diesem Fall zu Beginn abgeschätzt wurden. Im Fall, der in Abbildung 5.7 gezeigt ist, liefert die Abschätzung für vier Segmente zu hohe Werte. Nach der fünften Iteration wurde dies schon erheblich korrigiert.

Für acht Segmente ist der Unterschied zwischen den einzelnen Iterationen nur noch gering, da hier für die erste Iteration die Temperaturwerte der letzten Iteration für vier Segmente verwendet worden sind. Die Berechnung für acht Segmente beginnt mit Werten, die näher an den korrekten Werten liegen.

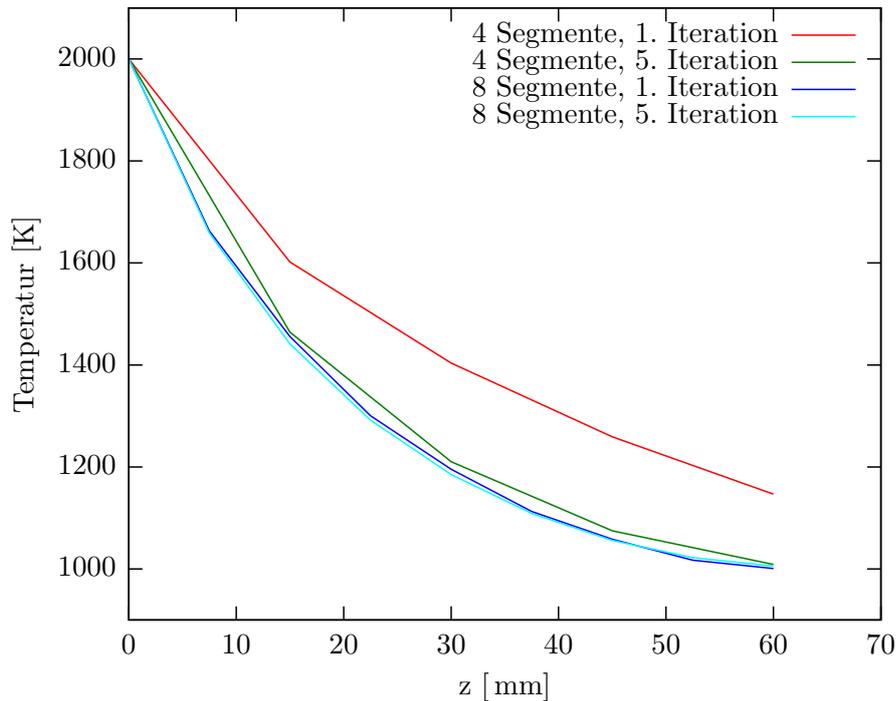


Abbildung 5.7: Iterationsverhalten der Simulation für vier und acht Segmente nach jeweils der ersten und fünften Iteration.

5.3 Konvergenzverhalten

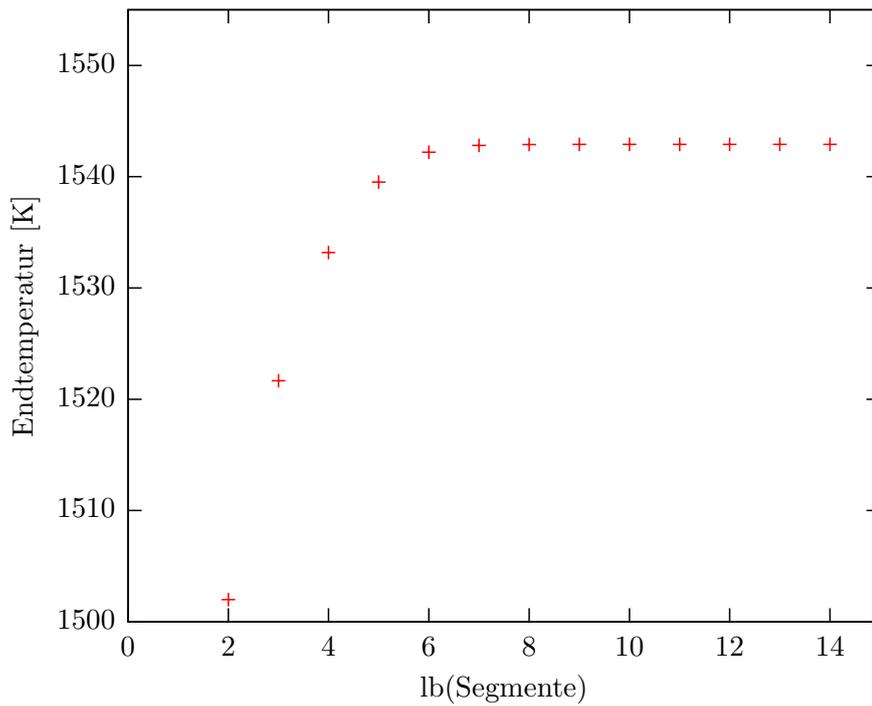
Nun soll das Konvergenzverhalten der Endtemperaturen untersucht werden. Trägt man für einen Vollzylinder mit einer Länge von $l = 32,7$ mm, einem Außenradius von $r_a = 2$ mm und einer Anfangstemperatur von $T_0 = 3400$ K die Endtemperaturen gegen den Logarithmus dualis der Segmente auf, ergibt dies die folgende Kurve in Abbildung 5.8. Anhand der Datenpunkte, die das Programm ausgibt, ist zu erkennen, dass die Änderung der Endtemperatur bei wachsender Segmentzahl immer kleiner wird. Die Änderungen sind ab 2^6 nur noch in einem Bereich von < 1 K. Die Temperaturen verhalten sich eindeutig konvergent.

5.4 Beispielsimulation für einen vollständigen Ofen

In Abbildung 5.9 sieht man eine 3D-Ansicht eines Ofens, wobei die Temperatur farblich dargestellt ist. Für die Simulation wurden die Abmessungen des vollständigen Ofens verwendet: $r_i = 1,7$ mm, $r_a = 2$ mm, $l_{\text{Stumpf}} = 11$ mm und $l_{\text{Zylinder}} = 63$ mm. Die Anfangstemperatur beträgt 2000 K.

Segmente	lb(Segmente)	T_E [K]
4	2	1501,991742
8	3	1521,657096
16	4	1533,183887
32	5	1539,517208
64	6	1542,201867
128	7	1542,806419
256	8	1542,891921
512	9	1542,903653
1024	10	1542,905908
2048	11	1542,906585
4096	12	1542,906776
8192	13	1542,906806
16384	14	1542,906821

Tabelle 5.1: Endtemperaturen für die verschiedene Anzahl von Segmenten

Abbildung 5.8: Konvergenzverhalten des Programms. Es ist eindeutig zu erkennen, dass sich ab einer Anzahl von 2^6 Segmenten die Endtemperatur bei einer feineren Unterteilung nur noch unerheblich ändert.

Der Programmcode befindet sich im Anhang und ist eine Kombination aus den Programmen für den Kegelstumpf und den Zylinder.

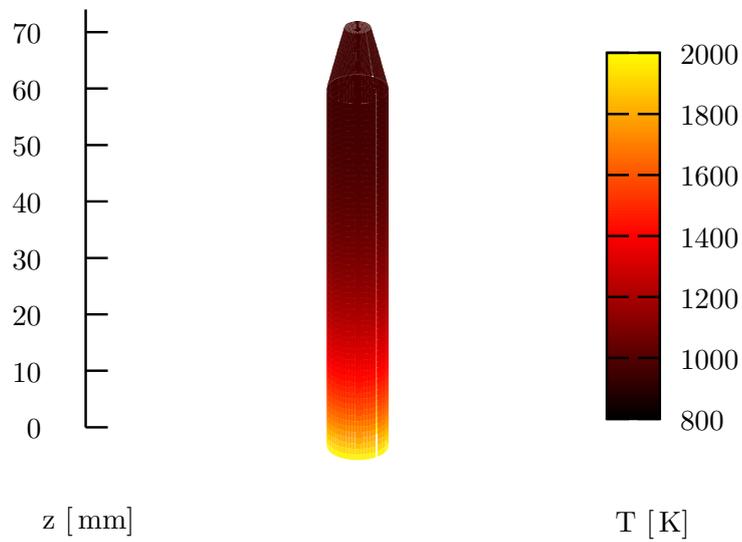


Abbildung 5.9: 3D-Ansicht bei einer Anfangstemperatur von 2000 K für einen Wolframofen. Für die Simulation wurden die Abmessungen eines tatsächlichen Ofens verwendet.

6 Versuchsaufbau

In diesem Kapitel wird der Versuchsaufbau beschrieben, mit dem die Temperaturmessungen für einen Zylinder und Ofen aus Wolfram durchgeführt werden.

Um die Ergebnisse des Programms zu überprüfen, wird zuerst die Endtemperatur eines Wolframvollzylinders gemessen. Hierzu wurde die RTA-Anlage (Rapid Thermal Annealing) verwendet.

Die RTA-Anlage verfügt über zwei Klemmen (in der Abbildung schwarz), in die ein

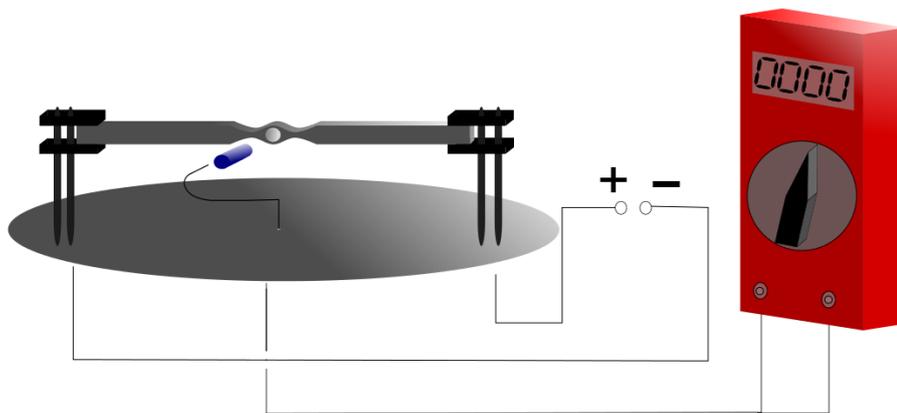


Abbildung 6.1: Schematischer Versuchsaufbau. Das Digitalmultimeter (rot) ist an ein Thermoelement des Typs B angeschlossen, mit dem die Endtemperatur des Testobjekts (blau) gemessen wird. Der Graphitstreifen (grau) ist in Klemmen (schwarz) eingespannt, die den Stromfluss gewährleisten.

Graphitstreifen eingespannt werden kann. Dieser Streifen fungiert als Widerstand und wird somit bei Stromdurchfluss heiß. Die Temperatur hängt dadurch von der Querschnittsfläche des Graphitstreifens ab. Da der hier verwendete Streifen in der Mitte eine Verjüngung besitzt, die den Widerstand erhöht, können hier Temperaturen von mehr als 2500 K erreicht werden. In dem Versuch wurde jedoch eine Temperatur von 2210 K nicht überschritten, um ein Überhitzen der Kontakte an den Klemmen zu vermeiden.

In die kreisrunde, 4 mm breite Öffnung des Graphitstreifens wird das zu vermessende Objekt (in der Skizze blau) hineingeschoben. Bevor die RTA-Anlage verschlossen wird, wird ein Typ B Thermoelement (bestehend aus Platin-30% Rhodium und Platin-6% Rhodium) in die Bohrung am Ende des Testobjekts gesteckt, um die Endtemperatur zu

messen. Auch die Anfangstemperatur kann mit einem Thermoelement gemessen werden, für die es eine 1,1 mm breite Bohrung in dem Graphitstreifen gibt. Alternativ kann sie über ein Quarzglasfenster in der Kuppel der RTA-Anlage mit einem Pyrometer ermittelt werden.

Um zu gewährleisten, dass das Testobjekt gleichmäßig erhitzt wird, besitzt der Graphitstreifen die in Abbildung 6.1 gezeigte Form. Das Graphit umgibt das Messobjekt mit gleicher Dicke und erhitzt sich somit aufgrund des gleichen Widerstands an dieser Stelle auf die gleiche Temperatur.

Über den Messaufbau wird eine stabile Metallkuppel platziert, die seitlich mit vier Schrauben festgezogen wird, um anschließend die RTA-Anlage zu evakuieren.

Abbildung 6.2 zeigt ein Foto der messbereiten RTA-Anlage im Gesamten.



Abbildung 6.2: Die RTA-Anlage mit geschlossener Kuppel. In der Bildmitte ist das Quarzglasfenster zu sehen, durch das die Messung mit dem Pyrometer stattfindet.

7 Ergebnisse

In diesem Kapitel werden die Ergebnisse der Messungen mit denen der analytischen und numerischen Lösung verglichen.

7.1 Ergebnisse der Messungen

Es wurden Messungen zunächst lediglich mit einem Thermoelement und einem Pyrometer und anschließend mit zwei Thermoelementen durchgeführt, auf die im Folgenden weiter eingegangen wird.

7.1.1 Messung mit Pyrometer und Thermoelement

Mit Hilfe eines Pyrometers wird die Anfangstemperatur bzw. die Temperatur des Graphitstreifens gemessen. Für die Endtemperatur wurde ein Thermoelement vom Typ B verwendet, das für einen Temperaturbereich von $50\text{ }^{\circ}\text{C}$ bis $1810\text{ }^{\circ}\text{C}$ geeignet ist. Die Spannungswerte können über eine Tabelle einer entsprechenden Temperatur zugeordnet werden.

Als erstes wurde ein Wolframvollzylinder mit einer Länge von $32,7\text{ mm}$ und einem Außenradius von 2 mm vermessen. Dazu schaut man in das Pyrometer auf den Graphitstreifen

T_A^g [K]	T_A^k [K]	ΔT_A^k [K]	T_E^g [K]	ΔT_E^g [K]
1740	1790	12	1090	40
1870	1930	50	1140	40
2000	2070	50	1190	40
2100	2180	50	1250	40
2210	2290	50	1290	40

Tabelle 7.1: Werte für die Messung mit Pyrometer und Thermoelement. T_A^g ist die mit dem Pyrometer gemessene, T_A^k die korrigierte Anfangstemperatur und T_E^g die gemessene Endtemperatur.

in der RTA-Anlage. Über die Skala an dem Pyrometer kann ein Temperaturwert abgelesen werden. Dieser Wert muss allerdings noch korrigiert werden. Es muss berücksichtigt werden, dass Graphit einen Emissionskoeffizienten von $0,75$ besitzt und dass die Messung mit

dem Pyrometer durch Quarzglas stattfindet. In der beiliegenden Bedienungsanleitung des Pyrometers ist die Durchlässigkeit eines Quarzfensters mit 0,94 angegeben, wodurch sich durch Multiplikation ein Emissionskoeffizient von 0,7 ergibt. Abhängig von diesem Wert und vom gemessenen Temperaturbereich sind im Handbuch Korrekturwerte angegeben, die auf den gemessenen Wert addiert werden.

Das Pyrometer verfügt über mehrere Skalen für drei Temperaturbereiche: Low Range (700 °C bis 1300 °C), Mid Range (1000 °C bis 2000 °C) und High Range (1400 °C bis 3500 °C). Der Fehler des Pyrometers ist im Handbuch als $\pm 0,6\%$ der vollen Skala für den unteren und mittleren Bereich und $\pm 1,2\%$ für den hohen Bereich angegeben. Dementsprechend wurde der Fehler für Tabelle 7.1 berechnet [Irc84].

Zusätzlich konnte über ein Thermoelement, was sich zusätzlich in der RTA befand, die Umgebungstemperatur bestimmt werden, die während den Messungen bei 500 K lag.

Für das Thermoelement kann ein Messfehler von 30 K angenommen werden. Die Messwerte des Thermoelements hängen davon ab, wie das Thermoelement in der vorgesehenen Bohrung hineingesteckt worden ist. Daher können die Werte um diesen Betrag schwanken. Zusätzlich kommt der Fehler des Digitalmultimeter von 0,1 mV, was einer Temperaturspanne von 10 K entspricht [Vol04] [Gmb12]. Über Fehlerfortpflanzung ergibt sich ein Fehler von 40 K für die Messwerte des Thermoelements.

7.1.2 Messung mit zwei Thermoelementen

Um die Messwerte des Pyrometers mit denen des Thermoelementes zu vergleichen und somit einen Anhaltspunkt für die Größe des Fehlers der Messwerte zu bekommen, wird zusätzlich eine Messreihe erstellt, bei der die Anfangstemperatur sowohl mit einem Thermoelement vom Typ B als auch mit dem Pyrometer gemessen wurde. Die Endtemperatur wurde mit einem Typ K Thermoelement (Nickel-Chrom und Nickel-Aluminium) gemessen.

$T_A^{T,B}$ [K]	$\Delta T_A^{T,B}$ [K]	T_A^P [K]	ΔT_A^P [K]	$T_E^{T,K}$ [K]	$\Delta T_E^{T,K}$ [K]
950	40	-	-	650	40
1300	40	1370	12	870	40
1510	40	1610	12	1010	40
1660	40	1790	50	1110	40

Tabelle 7.2: Mit dem Typ B Thermoelement und Pyrometer gemessene Anfangstemperaturen $T_A^{T,B}$ und T_A^P . Die Endtemperatur $T_E^{T,K}$ wurde mit einem Typ K Thermoelement bestimmt.

Bei dieser Messung konnten keine Werte für Temperaturen über 1700 K aufgenommen werden, da der Messbereich des Typ B Thermoelements, mit dem die Anfangstemperatur ermittelt wurde, bei 1700 K endet. Für eine Anfangstemperatur von 950 K wurde kein Wert mit dem Pyrometer gemessen, da dort die Umgebungstemperatur bei der Messung

zu dominant ist. Bei den anderen Werten kann man erkennen, dass das Thermoelement geringere Werte misst. Das Thermoelement führt durch seinen Kontakt mit dem Messobjekt selbst Wärme ab und misst dadurch eine geringere Temperatur als das Pyrometer. Weiterhin könnte es sein, dass bei der Evakuierung das Thermoelement nicht mehr richtig in der für die Messung vorgesehene Bohrung steckte, was ebenfalls zur Messung einer niedrigeren Temperatur führt.

7.2 Bestimmung des Emissionskoeffizienten

Bevor eine Simulation gestartet werden kann, die mit dem Experiment vergleichbare Ergebnisse liefert, muss zunächst die Oberfläche des Wolframzylinders untersucht werden. Da die technischen Möglichkeiten nicht zur Verfügung stehen, die Rauheit zu bestimmen, wurde ein Mikroskop verwendet, um diese abzuschätzen. Dies geschah über Fokussierung auf die Oberfläche und auf eine der Vertiefungen selbst (siehe Abbildung 7.2). Am Mikroskop konnten dann abgelesen werden, welche Distanz zwischen beiden Ebenen liegt, auf die jeweils fokussiert wurde.

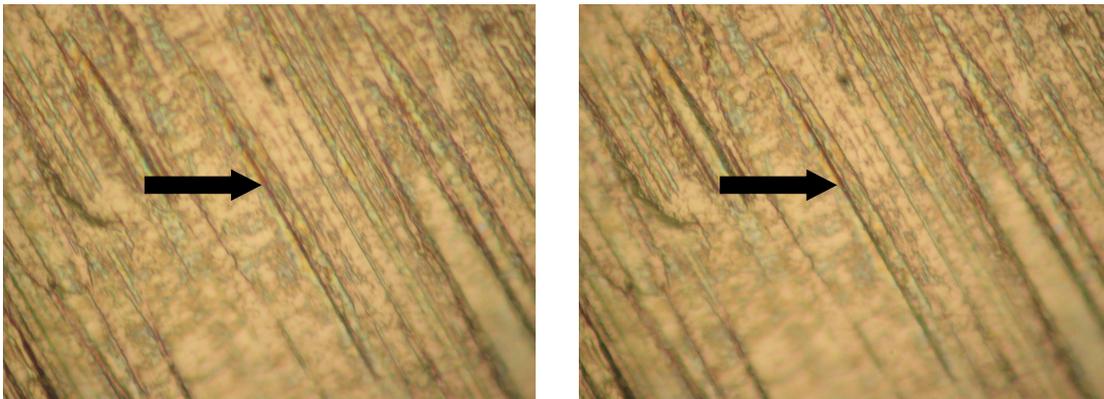


Abbildung 7.1: Fokussierung auf die Oberfläche (links) und auf den tiefsten Punkt des Profils. Die Pfeile weisen auf den Teil des Profils hin, bei dem die Unterschiede beider Fokussierungen deutlich werden.

Die Vermessung ergibt stellenweise eine Tiefe des Oberflächenprofils von bis zu $3\ \mu\text{m}$. Da Vertiefungen von weniger als $1\ \mu\text{m}$ durch Fokussierung vermessen werden können, wird ein Fehler von $0,5\ \mu\text{m}$ angenommen. Unter Verwendung von Abbildung 3.2 und der Tatsache, dass eine Oxidschicht bei Metallen den Emissionskoeffizienten erhöht, wird dieser auf $0,55$ geschätzt [Per73]. Es handelt sich hier um eine Abschätzung, da nicht bekannt ist, wie groß die effektive Größe der Vertiefungen ist.

7.3 Ergebnisse der Simulation und analytischen Lösung

Werden die gemessenen Anfangstemperaturen aus der Messung in das Programm und die analytische Lösung eingesetzt, erhält man die in Tabelle 7.3 gezeigten Endtemperaturen. Die Werte in der Tabelle wurden für eine Segmentzahl von 256 ($= 2^8$) berechnet. Schon für diese Anzahl von Segmenten ist gewährleistet, dass sich die Endtemperatur bei einer feineren Unterteilung kaum noch ändert (siehe Kapitel 5.3).

Um die analytische Lösung mit den Ergebnissen der Simulation zu vergleichen, wurde die Umgebungstemperatur T_U nicht berücksichtigt. Weiterhin ist in diesem Temperaturbereich der Einfluss von T_U vernachlässigbar.

Da die Anfangstemperatur sowohl nach oben als auch nach unten um den Fehler des Pyrometers schwanken kann, wurde für eine maximale, minimale und die tatsächliche gemessene Anfangstemperatur die Endtemperaturen berechnet. Die Unterschiede der drei Endtemperaturen T_E^S , $T_{E,\max}^S$ und $T_{E,\min}^S$ sind jedoch gering und liegen bei ungefähr 15 K. Die Werte T_E^A der analytischen Lösung liegen unter den gemessenen Werten. Bei steigender Anfangstemperatur wird die Differenz zwischen Anfangstemperatur und berechneter Endtemperatur immer größer. Die Ergebnisse der Simulation weisen bei $T_A = 1790$ K eine

T_A [K]	T_E^S [K]	$T_{E,\max}^S$ [K]	$T_{E,\min}^S$ [K]	T_E^A [K]	T_E^g [K]	ΔT_E^g [K]
1790	1258,10	1273,60	1241,86	968,70	1090	40
1930	1299,78	1313,37	1285,48	994,62	1140	40
2070	1336,58	1348,70	1323,91	1017,11	1190	40
2180	1362,78	1373,84	1351,21	1032,35	1250	40
2290	1386,56	1396,69	1376,01	1046,22	1290	40

Tabelle 7.3: Anfangstemperaturen und Endtemperaturen der analytischen und numerischen Lösung für einen Wolframvollzylinder. T_E^S ist die Endtemperatur der Simulation, T_E^A das Ergebnis der analytischen Lösung. T_E^g ist die gemessene Endtemperatur. $T_{E,\max}^S$ und $T_{E,\min}^S$ sind die maximale und minimale Endtemperatur aufgrund des Fehlers vom 50 K des Pyrometers.

Abweichung vom gemessenen Wert von ≈ 170 K. Für $T_A = 2290$ K beträgt die Differenz von Messwert nur noch 100 K. Mit dem Fehler für das Thermoelement von 40 K und unter Berücksichtigung der möglichen Schwankung der Endtemperatur ergibt dies eine Abweichung von 50 K vom gemessenen Wert. Der Wert liegt somit nicht mehr in den Fehlergrenzen.

Es muss jedoch bedacht werden, dass das Thermoelement mit einem Durchmesser von 1 mm direkt in einer Bohrung in der Stirnfläche des Zylinders steckt, dessen Durchmesser 4 mm beträgt. Zudem misst das Thermoelement einen zu geringen Wert, da es selbst Wärme abführt, was das Programm nicht berücksichtigt. Somit liegen die Messwerte unter den berechneten Werten der Simulation. Desweiteren ist der Emissionskoeffizient ε grob abgeschätzt, da bei der Vermessung des Profils nur kleine Teile der Oberfläche untersucht worden sind.

Zusätzlich sind sowohl für den Emissionskoeffizienten als auch für das Abführen der Wärme durch das Thermoelement keinerlei Fehler oder die Größenordnung dieser Effekte bekannt.

7.4 Messung und Simulation des vollständigen Ofens

Da für die Implantation ein Wolframofen der Form wie in Abbildung 2.1 verwendet wird, wird ein solcher vermessen und für 256 Segmente für sowohl den Zylinder als auch den Konus simuliert. Da hier die Vertiefungen auf der Oberfläche in der Größenordnung von $1\ \mu\text{m}$ liegen, wird ein Emissionskoeffizient von $\varepsilon = 0,2$ angenommen. Die Werte der Messung und der Simulation sind in Tabelle 7.4 dargestellt.

T_A^P [K]	ΔT_A^P [K]	T_E^S [K]	$T_{E,\max}^S$ [K]	$T_{E,\min}^S$ [K]	T_E^g [K]	ΔT_E^g [K]
1520	12	942,27	939,80	927,99	940	40
1650	12	966,64	968,75	966,28	960	40
1790	50	989,23	996,31	981,36	990	40
1920	50	1006,98	1013,57	1001,18	1000	40
2010	50	1018,62	1024,66	1012,28	1010	40
2150	50	1034,86	1040,18	1029,30	1020	40

Tabelle 7.4: Anfangstemperaturen und Endtemperaturen der Messung und der Simulation. T_A^P ist die mit dem Pyrometer gemessene Temperatur. T_E^S ist die Endtemperatur der Simulation und T_E^g ist die gemessene Endtemperatur. Auch hier wurde der Fehler der Anfangstemperatur berücksichtigt und es wurde eine minimale als auch maximale Endtemperatur berechnet.

Die Anfangstemperatur T_A wurde mit einem Pyrometer und die Endtemperatur T_E^g mit einem Typ B Thermoelement gemessen. Der zylindrische Teil steckte 13 mm im Graphitstreifen, wodurch ein Ofen mit $l_{\text{Zylinder}} = 50\ \text{mm}$, $l_{\text{Stumpf}} = 11\ \text{mm}$, $r_a = 2\ \text{mm}$, $r_i = 1,7\ \text{mm}$ simuliert werden muss. Aufgrund des Temperaturbereiches wurde die Umgebungstemperatur berücksichtigt und auf 500 K gesetzt.

Die Abweichung der simulierten Werte von den gemessenen Werten ist sehr gering. Mit dem Fehler von 50 K für das Pyrometer und 40 K für das Thermoelement liegen die gemessenen Werte sehr nah an den Werten der Simulation. Die Simulation liefert eine gute Vorhersage für Temperaturverteilung.

8 Fazit

Das Programm bietet eine gute Abschätzung für die Temperaturverteilung der jeweiligen simulierten Objekte. Die Simulationen für den Zylinder, den Zylinderstumpf und den vollständigen Ofen verhalten sich bei Variation verschiedener Größen gemäß den Erwartungen und physikalischen Gesetzen.

- Bei einer höheren Umgebungstemperatur ist der Wärmeverlust geringer.
- Wird die Außenfläche erhöht, führt auch dies zu einem stärkeren Abfall der Temperaturverteilung.
- Bei einem höheren Emissionskoeffizient und einer geringeren Querschnittsfläche wird mehr Wärme abgegeben.
- Die Länge des Testobjekts hat den erwarteten Einfluss.

Dennoch gibt es einige Faktoren, die beachtet werden müssen.

Zum einen ist eine genaue Kenntnis über die Oberflächenbeschaffenheit des Wolframs nötig, d.h. wie stark das Oberflächenprofil ausgeprägt ist. Im Rahmen dieser Arbeit wurde der Emissionskoeffizient abgeschätzt, wodurch dieser stärker von dem tatsächlichen Wert abweichen könnte.

Abgesehen davon muss beachtet werden, dass das Thermoelement selbst Wärme abführt und dadurch zu geringe Messwerte liefert. Gerade bei einem Messobjekt mit einem Durchmesser von 4 mm und 1 mm Durchmesser des Thermoelementes dürfte dieser Effekt nicht vernachlässigbar sein. Diesen zusätzlichen Wärmeabfluss berücksichtigt die Simulation nicht. Hinzu kommt der Messfehler der Thermoelemente von rund 40 K.

Desweiteren basiert das Programm auf einer Reihe von Annahmen, bei denen nicht bekannt ist, wie weit diese (z.B. Strahlungsgleichgewicht im Inneren des Zylinders, Vernachlässigung der Oxidationschicht) erfüllt werden.

Die Messwerte der Temperatur des Ofens stimmen mit denen der Simulation sehr gut überein. Zusätzlich bietet die Simulation einen guten Einblick in die Temperaturverteilung eines Wolframofens oder ähnlicher Objekte.

Abbildungsverzeichnis

1.1	Querschnitt durch den vorderen Teil der Ionenquelle	1
2.1	Querschnittsskizze des Wolframofens	3
3.1	Wärmeleitfähigkeit von Wolfram in Abhängigkeit von der Temperatur . .	6
3.2	Emissivität in Abhängigkeit von der Rauheit der Oberfläche	7
4.1	Skizze zur analytischen Betrachtung	10
4.2	Skizze zu Segmentierung des Zylinders	12
4.3	Wärmeströme in den Zylindersegmenten	13
4.4	Segmentierung des Zylinderstumpfes	15
5.1	Temperaturverteilung bei unterschiedlicher Querschnittsfläche und gleicher Mantelfläche	18
5.2	Temperaturverteilung bei unterschiedlicher Abstrahlfläche, aber gleichbleibender Querschnittsfläche	19
5.3	Temperaturverteilung bei unterschiedlicher Länge	20
5.4	Temperaturverteilung bei verschiedenen Umgebungstemperaturen	21
5.5	Simulierte Temperaturverteilungen bei unterschiedlichen Emissionskoeffizienten	22
5.6	Simulierte Temperaturverteilungen bei unterschiedlichen Anfangstemperaturen T_0	23
5.7	Iterationsverhalten der Simulation für vier und acht Segmente nach jeweils der ersten und fünften Iteration	24
5.8	Konvergenzverhalten des Programms	25
5.9	3D-Ansicht bei einer Anfangstemperatur von 2000 K für einen Wolframofen	26
6.1	Schematischer Versuchsaufbau	27
6.2	Messbereite RTA-Anlage	28
7.1	Vermessung des Oberflächenprofils durch Fokussierung	31

Tabellenverzeichnis

5.1	Endtemperaturen für die verschiedene Anzahl von Segmenten	25
7.1	Werte für die Messung mit Pyrometer und Thermoelement.	29
7.2	Messwerte für $T_A^{T,B}$, T_A^P und $T_E^{T,K}$	30
7.3	Anfangstemperaturen und Endtemperaturen der analytischen und numerischen Lösung für einen Wolframvollzylinder.	32
7.4	Anfangstemperaturen und Endtemperaturen der Messung und der Simulation für den Wolframofen	33

Literatur

- [ER12] Advanced Energy Technology Group Center for Energy Research. *Tungsten - Data and Correlations*. (aufgerufen am 20.08.2012). 2012. URL: aries.ucsd.edu/LIB/PROPS/PANOS/w.html.
- [Gmb12] Newport Electronics GmbH. *Thermoelement-Referenztablelle. TypB Referenztablelle*. Daimlerstraße 26 75392 Deckenpfronn, 2012. URL: www.omega.de.
- [IAE91] IAEA. *ITER Documentation Series No. 29. Blanket Shield Design and Material Database*. 1991.
- [Inc85] Frank P. Incropera et al. *Fundamentals of Heat and Mass Transfer*. 1985.
- [Irc84] Inc. Ircon. *Portable Disappearing-Filament Optical Pyrometer, IR-UF and IR-UC Models. Operations Manual*. Blankenburgerstr Straße 135, 13127 Berlin, 1984. URL: ircon.com.
- [Mei78] Meinhold et al. *Feld- und Potentialtheorie*. 3. Aufl. Deutsch Thun, 1978.
- [Per73] Robert H. Perry et al. *Chemical Engineers' Handbook*. 5. Aufl. McGraw-Hill, 1973.
- [Sko10] Goran Skoro. *Temperature correction of tungsten*. aufgerufen am 20.08.2012. 2010. URL: hepunx.rl.ac.uk/uknf/wp3/wiretest/Vibrometer/SurfaceRoughness.pdf.
- [Smi93] Ivica Smid et al. *Material and Design considerations for the Carbon Armored ITER Divertor*. 1993.
- [Stö00] Stöcker. *Taschenbuch der Physik*. 4. Aufl. Verlag Harri Deutsch, 2000.
- [Vol04] Voltcraft. *Bedienungsanleitung. Digitalmultimeter VC 820 und VC 840*. 92242 Hirschau, 2004.

Programmcode für den vollständigen Ofen

```
#include<stdio.h>

#include<math.h>
#include<stdlib.h>
#include <complex.h>

#ifndef M_PI
#define M_PI 3.14159265358979323846
#endif

/*-----
double emission: berechnet den Wert der Emssion in
Abhaengigkeit der Temperatur*/

double emission(double x){
    return 0.55;
}

/*-----
double leitfaehigkeit: setzt den Wert der Leitfaehigkeit in
Abhaengigkeit der Temperatur*/

double leitfaehigkeit(double x){
    return 129.694-0.0143124*x;
}

double factorstumpf(double schritt, double radiusaussen){
```

```
    return 2*(5.670373e-8)*schritt*schritt/((double) radiusaussen);
}

double funktionstrichstumpf(double x, double T, double schritt,
double radius, double mantel, double mittflaeche){

    return (mantel+radius*radius*M_PI)*5.67e-8*emission(x)*pow(x, 4)-
leitfaehigkeit(x)*mittflaeche*(T-x)/((double) schritt);

}

double nullstelleendestumpf(double T, double schritt, double radius,
double mantel, double mittflaeche){
    double epsilon=1e-10;
    double step=0.1;
    double l=0., r=l+step;
    double y=0.;

    while ( r-l > epsilon ){
        if ( funktionstrichstumpf(l, T, schritt, radius, mantel,
mittflaeche)*funktionstrichstumpf(r, T, schritt, radius, mantel,
mittflaeche) <= 0. ){
            y = (l+r)/2.;
            if ( funktionstrichstumpf(l, T, schritt, radius, mantel,
mittflaeche)*funktionstrichstumpf(y, T, schritt, radius, mantel,
mittflaeche) <= 0. ) r=y;
            else l=y;
        }
        else {
            r += step;
        }
    }

    return y;
}
```

```

/*-----
-----*/
double betrag: gibt den Betrag einer Zahl zurück*/

double betrag (double x) {
if (x > 0)
return x;
else
return -x;
}

double funktion(double x, double T, double factor){

    return factor*emission(x)*pow(x, 4)-(T-x);
}

double funktionstrich(double x, double T, double factor, double factor1){

    return (factor+factor1)*emission(x)*pow(x, 4)-(T-x);
}

double radius(double radiusinnen, double radiusaussen){
    double radiusinnenquadrat, radiusaussenquadrat;
    radiusinnenquadrat=pow(radiusinnen, 2);
    radiusaussenquadrat=pow(radiusaussen, 2);

return pow((radiusaussenquadrat-radiusinnenquadrat), 0.5);
}

double nullstelle(double T, double factor){
    double epsilon=1e-3;
    double step=0.1;
    double l=0., r=l+step;
    double y=0.;

    while ( r-l > epsilon ){
if ( funktion(l, T, factor)*funktion(r, T, factor) <= 0. ){
    y = (l+r)/2.;
    if ( funktion(l, T, factor)*funktion(y, T, factor) <= 0. ) r=y;
}
}
}

```

```
        else l=y;
    }
    else {
        r += step;
    }
}

return y;
}

double radiusenger(double neigungswinkel, double radiusaussen, double z){

return -tan(2*M_PI/((double) 360)*neigungswinkel)*z+radiusaussen;
}

double nullstelleende(double T, double factor, double factor1){
    double epsilon=1e-3;
    double step=0.1;
    double l=0., r=l+step;
    double y=0.;
    while ( r-l > epsilon ){
        if ( funktionstrich(l, T, factor, factor1)*
funktionstrich(r, T, factor, factor1) <= 0. ){
            y = (l+r)/2.;
            if ( funktionstrich(l, T, factor, factor1)*
funktionstrich(y, T, factor, factor1) <= 0. ) r=y;
            else l=y;
            /*printf("%.13f \n", y);*/
        }
        else {
            r += step;
        }
    }
return y;
}

double factor(double schritt, double radiusaussen, double radiusaequi,
double leit){
```

```
return 2*(5.670373e-8)*schritt*schritt*radiusaussen/((double)
radiusaequi*radiusaequi*leit);
}
```

```
void rechnungzylinder(double radiusinnen, double radiusaussen, float T0,
double laengezylinder, double laengestumpf, int segmente,
double neigungswinkel, int n){
    double epsilon=1e-10;
    int a;
    int i;
    int j;
    int k;
    double f, fst;
    double pruef=0;
    int iterations=1;
    int schleife1, schleife2;
    int b1=0, b2=0;
    int segmente_stumpf=segmente;
    int segmente_zylinder=segmente;
    int uebersegmente=segmente*2;
    double radiusaequi=radius(radiusinnen, radiusaussen);
    double schrittzyl, schritt;
    int iterationgesamt=0;
    double **Tzyl, **Tstumpf;
    double *Tnewzyl;
    double *Tnewstumpf;
    double *Tnewgesamt;
    double *mantel;
    double *mittquerflaeche;
    double *r;

    Tzyl=(double **)malloc(uebersegmente*sizeof(double));
    for(i=0;i<=segmente;i++){
        Tzyl[i]=(double *)malloc(uebersegmente*sizeof(double));
    }
}
```

```
Tstumpf=(double **)malloc(uebersegmente*sizeof(double));
    for(i=0;i<=segmente;i++){
        Tstumpf[i]=(double *)malloc(uebersegmente*sizeof(double));
    }

Tnewstumpf=(double *)malloc(2*uebersegmente*sizeof(double));
Tnewzyl=(double *)malloc(2*uebersegmente*sizeof(double));
Tnewgesamt=(double *)malloc(2*uebersegmente*sizeof(double));
mantel=(double *)malloc(2*uebersegmente*sizeof(double));
r=(double *)malloc(2*uebersegmente*sizeof(double));
mittquerflaeche=(double *)malloc(2*uebersegmente*sizeof(double));

for(schleife1=0; schleife1<=uebersegmente;schleife1++){
    Tzyl[0][schleife1]=T0;
}

for(segmente_zylinder=4;segmente_zylinder<=segmente;){
    segmente_stumpf=segmente_zylinder;

    if(segmente_zylinder==4){
        for(a=2, schleife1=b1;a<=segmente_zylinder;){
            int max=a;
            f=factor(laengezylinder/((double) a), radiusaussen,
radiusaequi, leitfaehigkeit(T0));
            schrittzyl=laengezylinder/((double) a);
            if(a==2){
                Tzyl[1][0]=nullstelle(Tzyl[0][0], f);
                Tzyl[2][0]=nullstelle(Tzyl[1][0], f);
            }
            for(i=1;i<=max*0.5;i++){
                int p=i*2;
                Tzyl[p][schleife1+1]=Tzyl[i][schleife1];
            }
            for(i=1;i<=max;i++){
                if(i%2==1){
                    Tzyl[i][schleife1+1]=(Tzyl[i+1][schleife1+1]+
Tzyl[i-1][schleife1+1])*0.5;
                }
                else;
            }
            schleife1++;
        }
    }
}
```

```

        a*=2;
    }

}

if(segmente_zylinder>4){
    for(a=segmente_zylinder, schleife1=b1;a<=segmente_zylinder;
){
        int max=a;

        schrittzyl=laengezylinder/((double) a);
        for(i=1;i<=max*0.5;i++){
            int p=i*2;
            Tzyl[p][schleife1+1]=Tzyl[i][schleife1];
        }
        for(i=1;i<=max;i++){
            if(i%2==1){
                Tzyl[i][schleife1+1]=(Tzyl[i+1][schleife1+1]+
Tzyl[i-1][schleife1+1])*0.5;
            }
            else;
        }
        schleife1++;

        a*=2;
    }
}

for(schleife2=0;schleife2<=n;schleife2++){
    Tstumpf[0][schleife2]=Tzyl[segmente_zylinder][schleife1];
}

if(segmente_stumpf==4){
    for(a=2, schleife2=b2;a<=segmente_stumpf;){
        int max=a;
        fst=factorstumpf(laengestumpf/((double) a),
radiusaussen)/((double) leitfaehigkeit(T0));
        schritt=laengestumpf/((double) a);
        if(a==2){
            Tstumpf[1][0]=nullstelle(Tstumpf[0][0], fst);
            Tstumpf[2][0]=nullstelle(Tstumpf[1][0], fst);
        }
    }
}

```

```
        for(i=1;i<=max*0.5;i++){
            int p=i*2;
            Tstumpf[p][schleife2+1]=Tstumpf[i][schleife2];
        }
        for(i=1;i<=max;i++){
            if(i%2==1){
                Tstumpf[i][schleife2+1]=(
Tstumpf[i+1][schleife2+1]+Tstumpf[i-1][schleife2+1])*0.5;
            }
            else;
        }
        schleife2++;
        a*=2;
    }
}

if(segmente_stumpf>4){
    for(a=segmente_stumpf, schleife2=b2;a<=segmente_stumpf;){
        int max=a;
        schritt=laengestumpf/((double) a);
        for(i=1;i<=max*0.5;i++){
            int p=i*2;
            Tstumpf[p][schleife2+1]=Tstumpf[i][schleife2];
        }
        for(i=1;i<=max;i++){
            if(i%2==1){
                Tstumpf[i][schleife2+1]=(
Tstumpf[i+1][schleife2+1]+Tstumpf[i-1][schleife2+1])*0.5;
            }
            else;
        }
        schleife2++;
        a*=2;
    }
}

b1=schleife1;
b2=schleife2;

for(i=0;i<=segmente_zylinder;i++){
    Tnewzyl[i]=Tzyl[i][schleife1];
}
for(i=0;i<=segmente_stumpf;i++){
```

```

        Tnewstumpf[i]=Tstumpf[i][schleife2];
    }
    for(i=0;i<=segmente_zyylinder;i++){
        Tnewgesamt[i]=Tnewzyl[i];
    }
    for(i=0;i<=segmente_stumpf;i++){
        Tnewgesamt[i+segmente_zyylinder]=Tnewstumpf[i];
    }

    /*-----
-----
    berechnet die entsprechenden Radien*/

    r[segmente_zyylinder]=radiusaussen;
    for(i=segmente_zyylinder;i<=segmente_zyylinder+segmente_stumpf+1;
i++){
        r[i+1]=radiusenger(neigungswinkel, r[i], schritt);
    }

    /*-----
-----
    berechnet die entsprechenden Mantelflächen*/

    for(i=segmente_zyylinder;i<=segmente_zyylinder+segmente_stumpf;i++
){
        double a=radiusenger(neigungswinkel, r[i], schritt) +
radiusenger(neigungswinkel, r[i+1], schritt);
        double b=pow(radiusenger(neigungswinkel, r[i], schritt)
- radiusenger(neigungswinkel, r[i+1], schritt), 2);
        mantel[i]=a*M_PI*pow((b+schritt*schritt), 0.5);
    }

    /*-----
-----
    berechnet die mittlere Querschnittsfläche*/

    for(i=segmente_zyylinder;i<=segmente_zyylinder+segmente_stumpf+1;
i++){
        mittquerflaeche[i]=M_PI*((r[i]+r[i+1])*0.5)*((r[i]+r[i+1])
*0.5);
    }

```

```

for(i=0;i<=segmente_zylinder+segmente_stumpf;i++){
    printf("%f \n", Tnewgesamt[i]);
}

/*-----
-----
for Schleife fuer die Iterationen*/
for(k=1;k<=iterations;k++){
    printf("%i \n", k);
    /*-----
-----

    for Schleife fuer das durchkorrigieren der Temperatur, */
    for(j=1;j<=segmente_zylinder;j++){
        /*-----
-----

                Sucht die Nullstellen in den entsprechenden
Bilanzgleichungen*/
                if(j<segmente_zylinder){
                    double step=100;
                    double l=0., r=l+step;
                    double u=0.;
                    while ( r-l > epsilon ){
                        if ( (pow((l),4)*factor(laengezylinder*
2/((double) a), radiusaussen, radiusaequi, leitfaehigkeit(l))*
emission(l) - Tnewgesamt[j-1]+2*l-Tnewgesamt[j+1])*(pow(r,4)*
factor(laengezylinder*2/((double) a), radiusaussen, radiusaequi,
leitfaehigkeit(r))*emission(r) - Tnewgesamt[j-1]+2*r-Tnewgesamt[j+1])
<= 0. ){
                            u = (l+r)/2.;
                            if ( (pow(l,4)*factor(laengezylinder*
2/((double) a), radiusaussen, radiusaequi, leitfaehigkeit(l))*
emission(l) - Tnewgesamt[j-1]+2*l- Tnewgesamt[j+1])*(pow(u,4)*
factor(laengezylinder*2/((double) a), radiusaussen, radiusaequi,
leitfaehigkeit(u))*emission(u) - Tnewgesamt[j-1]+2*u-Tnewgesamt[j+1])
<= 0. ) r=u;
                                else l=u;
                            }
                        else {
                            r += step;
                        }
                    }
                }
            }

```

```

    }
    Tnewgesamt[j]=u;

    }

    if(j==segmente_zylinder){
        double step=100;
        double l=0., r=l+step;
        double u=0.;
        while ( r-l > epsilon ){
            if ( (leitfaehigkeit(l)*M_PI*radiusaequi*
radiusaequi/((double) schrittzyl)*(Tnewgesamt[j-1]-l)-leitfaehigkeit(l)*
mittquerflaeche[j]/((double) schritt)*(1-Tnewgesamt[j+1])-5.67e-8*
emission(l)*mantel[j]*pow(l,4))*(leitfaehigkeit(r)*M_PI*radiusaequi*
radiusaequi/((double) schrittzyl)*(Tnewgesamt[j-1]-r)-
leitfaehigkeit(r)*mittquerflaeche[j]/((double) schritt)*(r-
Tnewgesamt[j+1])-5.67e-8*emission(r)*mantel[j]*pow(r,4)) <= 0. ){
                u = (l+r)/2.;
                if ( (leitfaehigkeit(l)*M_PI*radiusaequi*
radiusaequi/((double) schrittzyl)*(Tnewgesamt[j-1]-l)
-leitfaehigkeit(l)*mittquerflaeche[j]/((double) schritt)*(1-Tnewgesamt[j+1])-
5.67e-8*emission(l)*mantel[j]*pow(l,4))*(leitfaehigkeit(u)*M_PI*radiusaequi*
radiusaequi/((double) schrittzyl)*(Tnewgesamt[j-1]-u)-leitfaehigkeit(u)*
mittquerflaeche[j]/((double) schritt)*(u-Tnewgesamt[j+1])-5.67e-8*
emission(u)*mantel[j]*pow(u,4)) <= 0. ) r=u;
                    else l=u;
                }
            }
            else {
                r += step;
            }
        }
        Tnewgesamt[j]=u;

    }

    if(j>segmente_zylinder && j<segmente_stumpf+segmente_zylinder){
        double step=100;
        double l=0., r=l+step;
        double u=0.;
        while ( r-l > epsilon ){
            if ( (leitfaehigkeit(l)*mittquerflaeche[j]*
(Tnewgesamt[j-1]-l)/((double) schritt)- emission(l)*5.67e-8*pow(l, 4)*
mantel[j]-leitfaehigkeit(l)*mittquerflaeche[j+1]*

```

```

(1-Tnewgesamt[j+1])/((double) schritt))*(leitfaehigkeit(r)*
mittquerflaeche[j]*(Tnewgesamt[j-1]-r)/((double) schritt)- emission(r)*
5.67e-8*pow(r, 4)*mantel[j]-leitfaehigkeit(r)*mittquerflaeche[j+1]*
(r-Tnewgesamt[j+1])/((double) schritt)) <= 0. ){
    u = (1+r)/2.;
    if ( (leitfaehigkeit(l)*mittquerflaeche[j]*
(Tnewgesamt[j-1]-l)/((double) schritt)- emission(l)*5.67e-8*pow(l, 4)*
mantel[j]-leitfaehigkeit(l)*mittquerflaeche[j+1]*
(1-Tnewgesamt[j+1])/((double) schritt))*(leitfaehigkeit(u)*
mittquerflaeche[j]*(Tnewgesamt[j-1]-u)/((double) schritt)- emission(u)*
5.67e-8*pow(u, 4)*mantel[j]-leitfaehigkeit(u)*mittquerflaeche[j+1]*
(u-Tnewgesamt[j+1])/((double) schritt)) <= 0. ) r=u;
        else l=u;
        }
    else {
        r += step;
        }
    }
    Tnewgesamt[j]=u;

}

if(j==segmente_stumpf+segmente_zylinder){
    Tnewgesamt[j]=nullstelleendestumpf(
Tnewgesamt[j-1], schritt, r[j+1], mantel[j], mittquerflaeche[j]);
};

}
if(betrag(pruef-Tnewgesamt [segmente_zylinder+
segmente_stumpf])<1/((double) uebersegmente)){
    /*-----
-----
prueft das Abbruchkriterium*/
iterationgesamt+=k;
printf("Gesamtzahl der Iterationen: %i \n",
iterationgesamt);
break;
}
else {
pruef=Tnewgesamt [segmente_zylinder+segmente_stumpf];
iterations++;
};

```

```

for(i=0;i<=segmente_zyylinder+segmente_stumpf;i++){
    printf("T[%i] = %f \n", i, Tnewgesamt[i]);
}
}
for(i=0;i<=segmente_zyylinder;i++){
    Tzyl[i][schleife1]=Tnewgesamt[i];
}
for(i=0;i<=segmente_stumpf;i++){
    Tstumpf[i][schleife2]=Tnewgesamt[i+segmente_zyylinder];
}
/*-----
-----
print-Befehl*/
if(segmente_zyylinder==segmente){
    FILE *pFile8;
    pFile8 = fopen("zyylinder_iterationend_gesamt.txt",
"w");
    schrittzyl=laengezyylinder /((double)
segmente_zyylinder);
    float z=0;
    for(double winkel=0; winkel <=2*M_PI;){
        fprintf(pFile8, "%f \t %f \t %f \t %f \n", z,
radiusaussen*cos(winkel), radiusaussen*sin(winkel), Tnewgesamt[0]);
        winkel+=0.01;
    }
    fprintf(pFile8, "\n");
    for(i=1;i<=segmente_zyylinder;i++){
        z+=schrittzyl;
        for(double winkel=0; winkel <=2*M_PI;){
            fprintf(pFile8, "%f \t %f \t %f \t %f \n", z,
radiusaussen*cos(winkel), radiusaussen*sin(winkel), Tnewgesamt[i]);
            winkel+=0.01;
        }
        fprintf(pFile8, "\n");
    }
    fprintf(pFile8, "\n");
    schritt=laengestumpf /((double) segmente_stumpf);
    for(i=segmente_zyylinder+1;i<=segmente_zyylinder+
segmente_stumpf;i++){
        z+=schritt;
        for(double winkel=0; winkel <=2*M_PI;){
            fprintf(pFile8, "%f \t %f \t %f \t %f \n", z,
r[i-1]*cos(winkel), r[i-1]*sin(winkel), Tnewgesamt[i]);

```

```
        winkel+=0.01;
        }
        fprintf(pFile8, "\n");
        }
        fclose(pFile8);
        }
        segmente_zylinder*=2;
        for(i=0;i<=segmente_stumpf+segmente_zylinder*0.5;i++){

        printf("T[%i] = %f \n", i, Tnewgesamt[i]);
        }
    }
}

int main(){
    float T0;
    float laengestumpf, laengezylinder;
    int c=1, n;
    int segmente;
    float radiusinnen, radiusaussen, neigungswinkel;

    printf("-----
-----\n");
    /*-----
-----

    Parameter-Abfrage*/
    while(c==1){
        printf("Anfangstemperatur in K: ");
        scanf("%f", &T0);
        printf("Laenge des Stumpfes: ");
        scanf("%f", &laengestumpf);
        printf("Laenge des Zylinders: ");
        scanf("%f", &laengezylinder);
        printf("Innerer Radius: ");
        scanf("%f", &radiusinnen);
        printf("Aeusserer Radius: ");
        scanf("%f", &radiusaussen);
        printf("Anzahl der Halbierungen jeweils fuer Zylinder und Stumpf: ");
        scanf("%i", &n);
        printf("Neigungswinkel: ");
        scanf("%f", &neigungswinkel);
        segmente=pow(2, n);
```

```
    if(radiusaussen<radiusinnen){
        printf("Der eingegebene Aussenradius ist kleiner als der
Innenradius.\nBitte gueltige Werte eingeben.\n");
        c=1;
    }
    else{
        c=0;
    }
    }
    printf("%i \n", segmente);
    rechnungzylinder(radiusinnen, radiusaussen, T0, laengezylinder,
laengestumpf, segmente, neigungswinkel, n);

    printf("-----\n");
    return 0;
}
```


Danksagung

Ich möchte mich bei allen bedanken, die das Entstehen dieser Bachelorarbeit ermöglicht haben.

Als erstes möchte ich mich bei Herrn Priv. Doz. Dr. Vianden bedanken, der mit seiner sympathischen und ruhigen Art im Vorfeld das Interesse an der Arbeit geweckt hatte.

Bei Herrn Priv. Doz. Dr. Eversheim möchte ich mich für die Übernahme des Korreferats bedanken.

Natürlich gilt mein Dank auch der kompletten Arbeitsgruppe, die mich sehr nett aufgenommen hat: Thorsten Adolphs, Ishita Agarwal, Marius Arenz, Tim Flatten, Timo Heepenstrick, Patrick Keßler, Henning Kuhn, Juliana Marques Ramos, Jakob Penner und Michael Steffens.

Bei Marius, der immer mit Rat und Tat zur Seite stand, möchte ich mich für die hervorragende Betreuung bedanken.

Timo möchte ich für die Hilfe an der RTA-Anlage danken.

Natürlich gilt mein Dank auch Thorsten und Tim, mit denen es sehr lustig war.

Bei Conny Noll, Albert Dahl und Sigggi Hinderlich und der Zentralwerkstatt möchte ich mich für das Fertigen der Wolframöfen bedanken, da die Organisation der Herstellung der Öfen auch ein Teil meiner Aufgaben war.

Bei Bert Kann möchte ich mich für die Hilfe bei der Obflächenuntersuchung mit dem Mikroskop bedanken.

Zum Schluss möchte ich mich bei meinen Eltern und meinem Bruder bedanken, die mich seelisch und moralisch unterstützt haben.