

**Methodenentwicklung zur Berechnung der Materialeigenschaften von  
Aluminium Gusslegierungen in 3D für lokale Wärmebehandlungen mit  
optionaler Optimierung**

Eingereichte Doktorarbeit an der Montanuniversität Leoben  
zur Erhaltung des akademischen Doktorgrads  
Doktor der Montanistischen Wissenschaften (Dr. mont.)

verfasst von

**Tobias Holzmann**

Leoben, 2. März 2018



— Betreuer —

Univ. Prof. habil. Dr. rer. nat. Andreas Ludwig  
Lehrstuhl für Simulation und Modellierung Metallurgischer Prozesse  
Montanuniversität Leoben, Österreich



# Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfsmittel bedient habe. Alle in dieser Arbeit dargestellten Grafiken wurden selbst erstellt oder falls dies nicht zutrifft, mit entsprechender Referenz angegeben.

---

Leoben, den 2. März 2018

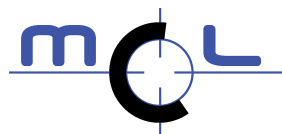


# Danksagungen

Der österreichischen Bundesregierung (insbesondere dem Bundesministerium für Verkehr, Innovation und Technologie und dem Bundesministerium für Wirtschaft, Familie und Jugend) vertreten durch die Österreichische Forschungsförderungsgesellschaft mbH (FFG), und den Ländern Steiermark und Tirol, vertreten durch die Steirische Wirtschaftsförderungsgesellschaft mbH (SFG) sowie die Standortagentur Tirol wird für die Förderung im Rahmen des COMET Förderprogramms herzlich gedankt.

Des Weiteren möchte ich mich bei meinem Betreuer, Prof. Dr. Andreas Ludwig, für die Unterstützung und den hilfreichen Diskussionen während der Dissertationszeit bedanken. Ferner möchte ich meinen Kollegen, Alexander Vakhrushev, einschließen, der für einen starken Austausch im Bereich der Numerik und Programmierung — betreffend der Open Source Toolbox OpenFOAM®— verantwortlich war. Zudem möchte ich mich bei Prof. Dr. Clemens Brand bedanken, der mir viele hilfreiche Anregungen im Bereich der Mathematik gegeben hat.

Abschließend danke ich den Mitarbeitern des Material Centers Leoben (LCS) für die Zusammenarbeit und Herrn Prof. Dr. Ernst Kozeschnik, vom Institut für Werkstoffwissenschaft und Werkstofftechnologie an der technischen Universität Wien, für die Unterstützung im Bereich der Materialberechnung und MatCalc.



Competence Centers for  
Excellent Technologies



*Gewidmet Andrea Jall  
Montanuniversität Leoben  
April 2014 - März 2018*





# Vorwort

Die vorliegende Dissertation beschreibt die während der Dissertation durchgeführten Arbeiten sehr detailliert und ausführlich. Trotzdem wird vorausgesetzt, dass der Leser über fundierte Kenntnisse im Themengebiet der numerischen Simulation verfügt. Hierzu zählen die Themengebiete der numerischen Mathematik, Thermodynamik, Physik und der Programmierung.

Wird aus Gründen der Übersichtlichkeit die Thematik nicht im Detail aufgezeigt, ist stets einschlägige Literatur zitiert, die der Leser für das Selbststudium heranziehen kann. Tritt der Fall ein, dass ein bekannter Sachverhalt in der Literatur unterschiedlich und/oder fehlerhaft beziehungsweise ohne Angaben der verwendeten Annahmen vorliegt, werden diese im Anhang durch eigenständige Herleitungen gegeben.

## Aufbau der Doktorarbeit

Da sich die vorliegende Dissertation mit mehreren Themengebieten beschäftigt, die für sich genommen geschlossen dargestellt werden können, werden diese einzeln behandelt. Das bedeutet, dass theoretische Grundlagen, diverse Beispiele und/oder Validierungsberechnungen innerhalb eines Kapitels behandelt werden. Damit wird sichergestellt, dass jedes Thema einzeln betrachtet werden kann und dient einer erhöhten Übersichtlichkeit. Im Gesonderten werden die nachfolgenden Bereiche in sich geschlossen diskutiert:

- Thermische Strukturanalyse mithilfe der Finiten Volumen Methode (FVM) unter Verwendung der Open Source Toolbox OpenFOAM<sup>®</sup>,
- Numerische Modellierung der Festkörperkinetik zur Bestimmung der Materialeigenschaften während einer Wärmebehandlung mithilfe der Software MatCalc,
- Möglichkeiten zur Optimierung diverser Problemstellungen mittels der Open Source Toolbox DAKOTA<sup>®</sup>.

## Visualisierungen, Grafiken und die Mathematik

Die in dieser Arbeit generierten Grafiken beinhalten durchweg als Dezimalseparator den Punkt. Eine Abänderung wird nicht durchgeführt, da die Übersichtlichkeit mit diesem Separator erhöht wird.

Ferner sind die mathematischen Formulierungen für den allgemeinen Ingenieur bestimmt. Eine mathematisch korrekte Behandlung einzelner Terme und Herleitungen wird nicht durchgeführt. Das bedeutet, dass keine Angaben über Dimensionsräume  $\mathcal{R}^n$  und ähnliches gegeben wird.



# Kurzfassung

Die vorliegende Arbeit befasst sich mit der lokalen Wärmebehandlung von Aluminiumgusslegierungen. Dies bedeutet, dass das Lösungsglühen vor dem Auslagern nur partiell, in einer bestimmten Region, stattfindet. Die inhomogene Temperaturverteilung ergibt thermische Spannungen, welche berücksichtigt werden müssen, sofern plastische Verformung vermieden werden soll. Zur numerischen Abbildung des Gesamtprozesses werden alle notwendigen Einzelschritte in einer Prozesskette zusammengefasst. Erweiternd wird die Prozesskette mit einem Optimierer gekoppelt, sodass diverse Problemstellungen automatisiert behandelt werden können und Möglichkeiten zur Optimierung gegeben sind. Die Struktur dieser Arbeit ist wie folgt.

Zu Beginn wird eine fundierte Literaturrecherche bezüglich der drei wichtigsten Themengebiete — thermische Spannungsberechnung, Aluminiumgusslegierungen und Optimierungsprobleme — gegeben. Anschließend werden Gründe präsentiert, weshalb die Spannungsberechnung unter Zuhilfenahme der Finiten Volumen Methode (FVM) erfolgt. Ferner wird die thermo-elastische Grundgleichung und die Energiegleichung hergeleitet und auf die Kopplungsbeziehung beider Formulierungen sowie auf die numerische Umsetzung eingegangen. Dabei werden Stabilisierungsaspekte diskutiert und aufgezeigt. Alle wichtigen numerischen Randbedingungen werden anschließend kurz wiedergegeben und Neuentwicklungen detailliert hergeleitet. Zur Prüfung der korrekten Implementierung der Gleichungen in die Open-Source-Software OpenFOAM<sup>®</sup> werden drei verschiedene Validierungsfälle betrachtet und explizit auf zwei verschiedene Konvergenzmethoden und deren Einfluss auf das Lösungsverhalten sowie der Genauigkeit eingegangen.

Subsequent wird ausführlich die Materialberechnung beschrieben und alle Eingabeparameter sowie die verwendeten Modelle aufgegriffen. Die Erklärung der Problematik einer lokalen Wärmebehandlung wird dazu verwendet alle Einzelschritte der Materialmodellierung zu beleuchten. Hierbei wird das Scheil-Gulliver Modell aufgezeigt, erläutert wie die Radienverteilungen einzelner Ausscheidungen nach dem Erstarren bestimmt werden und eine detaillierte Beschreibung der Kinetikmodellierung gegeben. Alle verwendeten numerischen Modelle und deren Parametersätze sowie die Phasen- und Matrixeigenschaften werden durch Literaturrecherchen abgeleitet und untermauert. Eine explizite Beleuchtung der Phasentransformationen wird gegeben und anschließend die entwickelten Kinetikskripte diskutiert sowie drei verschiedene Methoden zur Materialbestimmung in 3D gegeben.

Eine Vielzahl von Optimierungsalgorithmen wird anschließend präsentiert. Zur Darstellung der Arbeitsweise einiger Methoden werden einfache Beispielfunktionen herangezogen, die Ergebnisse ausführlich diskutiert und die Besonderheiten sowie Mängel dieser aufgezeigt.

Aufbauend darauf, werden alle drei Themenbereiche innerhalb einer Prozesskette verknüpft, wodurch die Untersuchung von lokale Wärmebehandlung von Aluminiumgusslegierungen ermöglicht wird. Die Anwendung dieser Prozesskette wird mithilfe zwei interessanter Beispiele veranschaulicht. Hierbei werden verschiedene Problemstellungen akribisch untersucht, die erhaltenen Ergebnisse aufgezeigt und interpretiert. Entsprechend wird die lokale Wärmebehandlung unter

Berücksichtigung thermischer Spannungen und der Bedingung, dass plastische Verformungen sowie das Aufschmelzen des Materials nicht eintreten, untersucht. Hinzu kommt die Analyse der Von Mises Spannung für zwei verschiedene Abkühlvorgänge (nach dem Lösungsglühen) sowie die Analyse einer örtlichen Wärmebehandlung mit subsequenter Vorhersage der Materialfestigkeit in 3D. Ferner wird eine Optimierung der Auslagerungsparameter (Dauer und Temperatur) für zwei Lösungsglühzeiten durchgeführt. Hierbei wird eine volumengewichtete integrale Materialfestigkeit — in einer definierten Region — als Objektfunktion herangezogen.

Abschließend wird die Flexibilität der neu entwickelten Prozesskette anhand eines Beispiels, dass die Modellierung des Zusammenschweißens zweier Platten (Vergütungsstahl) darstellt, präsentiert und abschließend die neu gewonnen wissenschaftlichen Erkenntnisse zusammengefasst sowie ein Ausblick gegeben.

# Abstract

The present thesis investigates into the topic of local heat treatments of aluminum alloys. In the following this means, that the annealing step is done only partially. Therefore, the non-homogeneous temperature distribution induces thermal stresses which have to be taken into account, if yielding of the material has to be avoided. For the numerical modeling of the whole process, all single steps are combined into a process chain framework. In addition, an optimizer is included in order to fully automatize the chain and enable the possibility of optimization. The structure of the thesis is as follows:

At the beginning a fundamental literature research of the three most important topics — thermal stress analysis, aluminum alloys and optimization strategies — is given. Subsequent, an introduction to the theoretical topics of the thermal-elastic stress calculation and energy equation is given while arguing that the finite volume method (FVM) is compatible to the well-known finite element method (FEM). During that, the coupled stress-energy system is discussed and three possibilities of numerical implementations are given. In addition, numerical stabilization aspects are argued. After that, the most important numerical boundary conditions are briefly described. Special ones are derived elaborately. To check the correct derivations and implementation of the numerical model, three different validation cases are analyzed and debated. Furthermore, two discriminate residual calculations will be discussed and a clear statement of the relaxation methods is presented.

Afterwards the whole modeling approach for the material is discussed profoundly while all input parameters and numerical models — that are used in the modeling approach — are given. Based on a clear statement of the problem which occur during a local heat treatment, the single modeling steps are delineated. Thus, the Scheil-Gulliver approach is explained whereat the estimation of the radius distribution of the phases, which occur after solidification, is explained. Subsequently, a fundamental description of the kinetic modeling approach is given while all numerical models, inclusive all phase and matrix properties, are announced by using a wide range of literature resources. To complete the theoretical part, three different approaches for the estimation of the yield strength in 3D are discussed and the advantages/disadvantages are pointed out.

Furthermore, a class of different kind of optimization algorithms are discussed and presented. In order to get familiar with some methods, simple object functions are used in order to describe and demonstrate the procedure of these algorithms. During the discussion the advantages and disadvantages of special algorithms are indicated.

Subsequently, the three different topics are combined in a new developed process chain framework while different examples are examined intensively. The first investigation is related to the local heat treatment without material calculation while the challenge of the non-uniform heat-up is discussed. Afterwards two different cooling methods are investigated namely self-quenching and water-quenching while the thermal stresses will be analyzed and discussed. Thereafter, the

material calculation is included in order to predict the yield strength of the material after the local heat treatment and subsequent artificial aging in 3D by using the new developed Random-Cell-Poisson method. At the end an optimization procedure is given to find the best process parameters for different local heat treatments, cooling methods and artificial aging parameters. For that purpose, a genetic algorithm is used in order to find the global maximum in the parameter space. Furthermore, the flexibility of the developed framework is presented by investigating into a steel welding example.

At the end the new scientific findings are summarized and an outlook is given.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Thermisch induzierte Spannungen . . . . .	2
1.2	Aluminium Legierungen . . . . .	3
1.3	Optimierung . . . . .	5
<b>2</b>	<b>Spannungsmodellierung mit der Finiten Volumen Methode</b>	<b>9</b>
2.1	Methoden zur Spannungsberechnung . . . . .	9
2.2	Gründe für die Implementierung in ein FVM Programm . . . . .	10
2.3	Herleitung der thermo-elastischen Gleichung . . . . .	11
2.4	Herleitung der Energiegleichung . . . . .	14
2.5	Berechnung der gekoppelten Gleichungen . . . . .	16
2.6	Numerische Optimierung der Duhamel-Neumann Gleichung . . . . .	19
2.7	Randbedingungen . . . . .	20
2.7.1	Randbedingungen für die Verschiebungsgleichung . . . . .	20
2.7.2	Randbedingung für die Temperaturgleichung . . . . .	21
2.8	Integralform der Gleichungen für die FVM . . . . .	23
2.9	Validierung und Ergebnisse . . . . .	24
2.9.1	Erster Validierungsfall - Grenzfall Würfel . . . . .	25
2.9.2	Zweiter Validierungsfall - Hohlrohr . . . . .	31
2.9.3	Dritter Validierungsfall - FVM vs FEM . . . . .	33
2.10	Numerik und Relaxationsmethoden . . . . .	36
<b>3</b>	<b>Modellierung der Materialeigenschaften</b>	<b>39</b>
3.1	Das Ziel und die einzelnen Modellierungsphasen . . . . .	39
3.2	Problemstellung . . . . .	41
3.3	Erstarrungsmodellierung . . . . .	42
3.3.1	Das Scheil-Gulliver Modell . . . . .	42
3.3.2	Legierungselemente und Phasen während der Erstarrung . . . . .	44
3.3.3	Erstarrungssequenz bei Aluminiumlegierungen . . . . .	45
3.4	Radius Verteilung der $\beta$ - und Si-Phase . . . . .	46
3.5	Kinetikmodellierung . . . . .	47
3.5.1	Numerische Modelle . . . . .	48
3.5.2	Phasen und Phasenumwandlungen in Aluminiumlegierungen . . . . .	51
3.5.3	Phaseneigenschaften . . . . .	52
3.5.4	Matrixeigenschaften . . . . .	53
3.5.5	Zusammenfassung der Phasen- und Matrixeigenschaften . . . . .	54
3.5.6	Phasentransformationsmodellierung . . . . .	55

3.6	MatCalc Skripte und Einstellungen . . . . .	58
3.6.1	Scheil Skript . . . . .	58
3.6.2	Kinetics_1 Skript . . . . .	59
3.6.3	Kinetics_2 Skript . . . . .	61
3.6.4	Kinetics_3 Skript . . . . .	61
3.7	Extrapolation des Field-Average Approach's auf 3D Konfigurationen . . . . .	62
3.7.1	Flächen-Gewichtetes Interpolationsverfahren . . . . .	62
3.7.2	Datentabellierung (Look-Up-Table) . . . . .	63
3.7.3	Random-Cell-Poisson Methode . . . . .	63
<b>4</b>	<b>Optimierungsalgorithmen</b>	<b>65</b>
4.1	Kategorisierung der Verfahren . . . . .	65
4.2	Algorithmen und Beispiele . . . . .	66
4.2.1	Multidimensionsanalyse . . . . .	67
4.2.2	Vektoranalyse . . . . .	68
4.2.3	Centered-Parameteranalyse . . . . .	68
4.2.4	List-Parameteranalyse . . . . .	68
4.2.5	Samplinganalyse . . . . .	69
4.3	Optimierungsverfahren . . . . .	69
4.3.1	Gradientenbasierte Verfahren . . . . .	69
4.3.2	Genetische Algorithmen . . . . .	70
4.4	Abschließend . . . . .	73
<b>5</b>	<b>Ergebnisse</b>	<b>75</b>
5.1	Prozesskette . . . . .	76
5.2	Lokale Wärmebehandlungen an zwei Beispielen . . . . .	77
5.2.1	Die Geometrien der Strukturbauteile . . . . .	78
5.2.2	Numerisches Netz, Randbedingungen und wichtige Hinweise . . . . .	79
5.2.3	Ergebnisse . . . . .	81
5.2.4	Zusammenfassend und Erkenntnisse . . . . .	86
5.3	Thermisch induzierte Spannungen beim Abkühlvorgang . . . . .	87
5.4	Lokale Wärmebehandlung mit Materialberechnung . . . . .	88
5.4.1	Selbstkühlung (Self-Quenching) . . . . .	89
5.4.2	Abschrecken in Wasser (Water-Quenching) . . . . .	92
5.5	Einfluss der lokalen Lösungsglühzeit auf die integrale Festigkeit . . . . .	93
5.5.1	Lokale Wärmebehandlung mit Selbstabkühlung . . . . .	94
5.5.2	Lokale Wärmebehandlung mit Abschrecken . . . . .	97
5.6	Optimierung der Auslagerungsparameter . . . . .	99
5.6.1	Ergebnisdiskussion . . . . .	99
5.6.2	Zusammenfassend . . . . .	102
5.7	Anwendungsbeispiel für Stahllegierungen (Schweißen) . . . . .	103
5.7.1	Numerisches Netz, Randbedingungen und wichtige Hinweise . . . . .	103
5.7.2	Lokale Wärmebehandlung (Schweißen) . . . . .	104
5.7.3	Streckgrenzwerte $R_{p,0.2}$ in 3D . . . . .	106
5.7.4	Abschließend . . . . .	106



<b>6 Erlangte Erkenntnisse und Aussicht</b>	<b>107</b>
<b>Anhang</b>	<b>123</b>
<b>A Thermische Spannungsmodellierung</b>	<b>125</b>
A.1 Herleitung der Temperaturerhaltungsgleichung . . . . .	125
A.2 Herleitung der Temperaturrandbedingung . . . . .	128
A.2.1 Wärmeströme und Bilanzierung . . . . .	128
A.2.2 Wärmefluss Bilanzierung (ohne Strahlung) . . . . .	130
A.2.3 Wärmefluss Bilanzierung (mit Strahlung) . . . . .	132
A.2.4 Quellcode . . . . .	132
A.3 Bestimmung des effektiven Wärmeübergangskoeffizienten . . . . .	134
A.4 Dynamischer LASER Spot . . . . .	137
A.4.1 Kreisbewegung . . . . .	137
A.4.2 Linear Bewegung . . . . .	138
A.5 Herleitung der Neumann Bedingung für den Verschiebungsvektor $\mathbf{D}$ . . . . .	139
A.6 Ergänzendes zu den Validierungsfälle . . . . .	141
A.6.1 Erster Validierungsfall - Grenzfall Würfel . . . . .	141
A.6.2 Zweiter Validierungsfall - Hohlrohr . . . . .	146
A.6.3 Dritter Validierungsfall - Biegebalken . . . . .	147
A.7 thermalElasticStressFoam C++ Code Beschreibung . . . . .	148
A.7.1 Der Strukturbaum . . . . .	148
A.7.2 Code Beschreibung . . . . .	148
<b>B Numerik</b>	<b>155</b>
B.1 Das lineares Gleichungssystem . . . . .	155
B.2 Der lineare Löser . . . . .	155
B.3 Lösungsalgorithmus zur Berechnung des linearen Gleichungssystems . . . . .	156
B.4 Konvergenzmethoden . . . . .	158
B.4.1 L1-Norm . . . . .	158
B.4.2 Relative Methode – Feldbetrachtung . . . . .	159
B.5 Feld- und Matrixrelaxation . . . . .	160
B.5.1 Feldrelaxation . . . . .	160
B.5.2 Matrixrelaxation . . . . .	161
B.5.3 Finaler Relaxationsfaktor . . . . .	164
B.6 Einfluss der Relaxations- und Konvergenzmethoden . . . . .	164
B.6.1 Feldrelaxation in transienten Berechnungen . . . . .	165
B.6.2 Matrixrelaxation in transienten Berechnungen . . . . .	167
B.6.3 Feld- und Matrixrelaxation in stationären Berechnungen . . . . .	170
B.7 Einfluss der inneren Schleife des linearen Löser auf die Konvergenzrate . . . . .	172
<b>C Vektor- und Tensoralgebra</b>	<b>175</b>
C.1 Das innere Produkt . . . . .	176
C.2 Das äußere Produkt . . . . .	176
C.3 Gradientenoperator . . . . .	177
C.4 Divergenzoperator . . . . .	177

C.5	Produktregel mit dem Divergenzoperator . . . . .	177
C.6	Spuroperator (Trace-Operator) . . . . .	178
C.7	Zusammenhang zwischen Spur- und Divergenzoperator . . . . .	178
C.8	Substantielle Ableitung (materielle Ableitung) . . . . .	178
C.9	Deviatorischer und hydrostatischer Anteil einer Matrix . . . . .	179
<b>D</b>	<b>Verfahren zur automatischen Netzerzeugung</b>	<b>181</b>
D.1	Das Lösungsgebiet . . . . .	181
D.2	Volumentypen . . . . .	181
D.3	Generierung der Netztypen . . . . .	182
D.3.1	Hexaedernetze . . . . .	182
D.3.2	Tetraedernetze . . . . .	183
D.3.3	Polygonnetze . . . . .	184
D.3.4	Hybride Netze . . . . .	184
<b>E</b>	<b>Ergänzendes zur Materialberechnung</b>	<b>187</b>
E.1	Bestimmung der Phasenverteilung . . . . .	187
E.1.1	Bestimmung der Siliziumverteilung . . . . .	187
E.1.2	Bestimmung der $\beta$ -Verteilung . . . . .	189
E.2	Ermittlung der Phasentransformationsradien . . . . .	195
E.3	Bestimmung der Komposition des reduzierten Systems . . . . .	204
E.4	Bestimmung der Abkühlkurve von AlSi7Mg0.3 nach dem Erstarren im Sandguss	206
E.5	cellListGeneration C++ Code Beschreibung . . . . .	207
E.5.1	Der Strukturbaum . . . . .	207
E.5.2	Code Beschreibung . . . . .	207
E.6	MatCalcPreparation C++ Code Beschreibung . . . . .	212
E.6.1	Der Strukturbaum . . . . .	212
E.6.2	Code Beschreibung . . . . .	212
E.7	YieldStrengthLaplacianFoam C++ Code Beschreibung . . . . .	214
E.7.1	Der Strukturbaum . . . . .	214
E.7.2	Code Beschreibung . . . . .	214
<b>F</b>	<b>Zusatzinformationen zu den Ergebnissen</b>	<b>217</b>
F.1	Numerische Netze des Motorengehäuses und der Dämpferbrücke . . . . .	218
F.2	Numerische Netze der Stahlplatten . . . . .	219
<b>G</b>	<b>Stoff- und Materialdaten</b>	<b>221</b>
G.1	Stoffwerte für Luft . . . . .	221
G.2	Materialwerte für AlSi7Mg0.3 . . . . .	222

# 1. Einleitung

Die vorliegende Arbeit diskutiert die Möglichkeit, eine lokale Wärmebehandlung an Aluminiumlegierungen zur räumlich lokalen Verbesserung des Materials einzusetzen. Diese Methodik weicht vom Standardverfahren ab, da im Allgemeinen die Wärmebehandlung mit dem gesamten Gussstück durchgeführt wird und nicht örtlich begrenzt ist. Die bis heute verwendeten und vorliegenden numerischen Methoden zur Beschreibung von Wärmebehandlungen können nur für Standardverfahren eingesetzt werden. Dies bedeutet, dass für lokale Betrachtungsweisen neue Methoden entwickelt werden müssen, die in dieser Arbeit präsentiert werden. Da die lokale Wärmebehandlung im gesamten Gussteil inhomogene Temperaturen hervorruft, entstehen Gradienten, die mechanische Spannungen induzieren — diese werden auch als thermische Spannungen bezeichnet. Demnach müssen bei räumlich begrenzten Wärmebehandlungen die thermischen Spannungen berücksichtigt werden, um zu determinieren, ob plastische Verformung eintritt oder nicht. Zusätzlich zur neuen Methodenentwicklung, die es ermöglicht lokale Wärmebehandlungen numerisch zu betrachten, wird ferner ein Optimierungstool hinzugefügt, um die örtlich begrenzte thermische Behandlung näher zu untersuchen und diverse Prozessparameter zu optimieren.

Die Entwicklung einer solchen Prozesskette, die das Herstellen des Aluminiumgussstücks, die anschließende lokale Wärmebehandlung mit Erfassung von thermisch induzierten Spannungen und Verformungen und der abschließenden Optimierung abbildet, bedarf das Verständnis verschiedener wissenschaftlicher Themengebiete wie beispielsweise der numerischen Mathematik, der Thermodynamik und vielen weiteren. Es ist dabei wichtig die derzeitigen Forschungsaktivitäten und den technischen Stand zu beleuchten, um mit den neuesten Methoden und Möglichkeiten die Problemstellung zu bearbeiten.

Die beschriebene Thematik dieser Arbeit kann wie folgt in drei Einzelkategorien aufgeteilt werden:

1. Thermisch induzierte Spannung und Verformung,
2. Aluminium Legierungen und deren numerische Modellierung,
3. Optimierungsansätze.

Diesbezüglich wird nachfolgend ein Überblick der einzelnen Bereiche gegeben. Es wird aufgezeigt in welchen Anwendungsbereichen thermisch induzierte Spannungen auftreten und welche Probleme diese verursachen. Des Weiteren werden die technologischen Eigenschaften von Aluminiumlegierungen erwähnt, welche Möglichkeiten zum Verbessern der Materialeigenschaften bestehen, diverse Forschungsaktivitäten aufgezeigt und ein Überblick über den Anwendungsbereich dieser Legierungen diskutiert. Abschließend wird ein kurzer Einblick in die Optimierung gegeben und beschrieben, in welchen Themenbereichen Optimierer eingesetzt werden.

## 1.1. Thermisch induzierte Spannungen

Thermische Spannungen werden durch Temperaturunterschiede im Material hervorgerufen und beruhen im Allgemeinen darauf, dass sich die Materie entweder ausdehnt oder zusammenzieht. Dies kann durch die Massenerhaltung und der Relation zwischen Masse, Volumen und Dichte erklärt werden. Wird angenommen, dass die Masse eines Materials nicht vernichtet oder in Energie umgewandelt werden kann, so beschreibt eine Dichteänderung eine reziproke proportionale Volumenänderung. Mit anderen Worten: wird die Dichte eines Stoffes größer, muss sich zwangsläufig das Volumen verkleinern und umgekehrt. Dieses Phänomen wird allgemein durch eine temperaturabhängige Dichte hervorgerufen.

Da es grundsätzlich bekannt ist, dass die letzte Instanz der Energieumwandlung stets die innere Energie darstellt, ist offensichtlich, dass in allen möglichen Ingenieurbereichen (und darüber hinaus) zwangsläufig mit Temperaturunterschieden im Material zu rechnen ist.

So ist beispielsweise im Bereich der Elektronik fast ausschließlich jedes Bauelement mit thermisch induzierten Spannungen belastet. Lienig und Brümmer (2014) geben in ihrem Buch eine Übersicht an, in der deutlich wird, dass durch thermische Gradienten Spannungen entstehen und damit die Lebensdauer der elektronischen Bauteile verringert wird. Dies ist auf das Ohmsche Gesetz zurückzuführen und ist abhängig von der Stromstärke und dem gegebenen Widerstand des Leiters. Diesbezüglich produzieren beispielsweise einzelne IC-Bausteine (*Integrated Circuits*), die mit sehr kleinen Strömen betrieben werden (im Bereich von  $\mu\text{A}$  bis  $\text{mA}$ ), wesentlich weniger innere Energie (Wärme) und dadurch thermisch induzierte Spannungen wie beispielsweise ein moderner Mehrkernprozessor, der bis zu einigen hundert Watt erzeugen kann. Gerade bei Hochleistungsprozessoren muss eine externe Kühlung angebracht werden, um die Wärmeenergie abzuführen. Entsprechend liegen hier zwangsläufig thermische Gradienten vor, die mechanische Spannungen induzieren. Gerade in der Leiterbahntechnologie können diese Belastungen die Ursache des Versagens des gesamten elektronischen Bauteils sein; vergleiche Jiang et al. (1997)

Auch im Bereich der Nanotechnologie sind thermische Spannungsprobleme vorzufinden. Werden beispielsweise einzelne Aluminiumatome auf eine Siliziumschicht aufgetragen, entstehen bei Temperaturgradienten thermische Spannungen, die eine schädigende Wirkung auf die Grenzschichtstruktur zur Folge haben. Eiper et al. (2004) gibt zu dieser Problematik eine Diskussion und vergleicht numerisch erhaltene Werte mit Messungen.

In der Metallurgie sind thermisch induzierte Spannungen fast in jedem Bereich anzutreffen. Sie treten beispielsweise beim Erstarren der flüssigen Metallschmelze oder beim Schweißen auf. Matsushita et al. (2017) zeigt für Bronzeabgüsse — die keinen Bleizusatz enthalten — auf, dass thermische Spannungen die Neigung zur Rissbildung im Abguss verstärken. Ähnliche Analysen für eine Kupfer-Zink Legierung wurden von Kasuya et al. (2016) durchgeführt. In Druck-Guss Formen, die vor allem für hohe Stückzahlen und komplexe Geometrien eingesetzt werden, kann die Lebensdauer der Form bezüglich thermischen Spannungen erheblich herab gesenkt werden, wenn das Design der Dauerform schlecht entworfen wird oder Prozessschritte wie dem Vorheizen der Form nicht oder ungenau durchgeführt werden; vergleiche Klobč und Tušek (2008).

Des Weiteren können thermische Spannungen in Legierungen aufgrund von Ausscheidungen auftreten. Dies ist dadurch begründet, dass Ausscheidungen einen anderen thermischen Ausdehnungskoeffizienten wie das Grundmaterial aufweisen können. Dadurch — auch bei gleichmäßiger Temperaturerhöhung oder -absenkung — entstehen Diskrepanzen bei der natürlichen Ausdehnung beider Stoffe, die zu Spannungen führen und damit die maximal mögliche Belastung ver-

ringern können; siehe dazu Dash et al. (2016).

Die Liste der Beispiele, die aufzeigen wie wichtig thermische Spannungen sind, ist unbegrenzt. Daher werden Simulationstools eingesetzt, um bereits bei der Designkonzeption mögliche Schwachstellen herauszufiltern und diesen entgegenzuwirken. Eine Vielzahl von industriellen Projekten, die thermisch induzierte Spannungen untersuchen, sind auf der Projektseite von Heiny und Dölle (2018) publiziert. Alle Themenbereiche, angefangen von elektronischen Bau- oder Maschinenteilen über Kugelhähne bis hin zu Zylinderköpfen und Haushaltsgeräten werden dort präsentiert und bestätigen das Vorhandensein thermisch induzierter Spannungen in allen Anwendungsbereichen des Ingenieurwesens.

## 1.2. Aluminium Legierungen

Eine Legierung besteht im Gegensatz zu Reinstoffen aus einem Hauptelement (hier Aluminium) das mit weiteren Elemente versetzt wird, um gezielt Materialeigenschaften zu beeinflussen. Die Elemente, die zur Modifikation des Aluminiums verwendet werden, sind hauptsächlich Silizium, Magnesium und Kupfer. Seltener wird Zinn, Lithium und Mangan verwendet. Allgemein können Aluminiumlegierungen in zwei Kategorien eingeteilt werden: den Gusslegierungen und den Knetlegierungen. Diese werden weiter zwischen aushärtbaren und nicht aushärtbaren Legierungen klassifiziert. Die Auswahl der Legierungselemente und deren Stoffmengenanteile bestimmen das Einsatzgebiet. Eine Übersicht diesbezüglich ist in Gudladt (2017) gegeben. Des Weiteren geben die Normen DIN EN 573-1:2005-02 (2005) und DIN EN 1706:2013-12 (2013) weitere Informationen bezüglich Bezeichnung und Zusammensetzung an.

Aluminiumlegierungen zeichnen sich vor allem durch ihre besonderen Eigenschaften aus. Wang und Davidson (2001); Möller et al. (2010) und Li et al. (2011) geben in ihren Publikationen die ausgezeichneten Gießeigenschaften, hohe thermische und elektrische Leitfähigkeit und eine starke Korrosionsbeständigkeit von Aluminiumlegierungen an. Zudem zeichnen sich diese Legierungen insbesondere durch die Eigenschaft der Aushärtbarkeit, das vor allem durch die Entstehung von Ausscheidungen erfolgt, aus. Dadurch ist es möglich die Streckgrenze  $R_{p,0.2}$  deutlich anzuheben und ermöglicht es ein exzellentes Verhältnis zwischen Festigkeit und Gewicht zu erzielen; vergleiche Falahati et al. (2010) und Borodianskiy et al. (2015). Myriounis et al. (2008) erklärt, dass durch weitere Modifikation die Verschleiß- und Dauerfestigkeit gezielt gesteigert werden können und Li et al. (2011) erwähnt die gute Schweißbarkeit. Daher werden Aluminiumlegierungen in vielen industriellen Bereichen wie beispielsweise im Fahrzeug- und Flugzeugbau eingesetzt; Wang und Davidson (2001); Myriounis et al. (2008); Li et al. (2011); Mohamed und Samuel (2012)

Die Aushärtbarkeit bei höheren Temperaturen (80 °C bis 220 °C) ist überwiegend durch magnesiumhaltige Ausscheidungen wie der stabilen  $\beta$ - oder  $\Theta$ -Phase beziehungsweise der metastabilen  $\beta''$ -Phase bedingt. Dies ist in der Literatur anerkannt und wird von vielen Autoren wie Apelian et al. (1989); Dutta und Allen (1991); Falahati et al. (2010) oder Ryen et al. (2015) bestätigt. Weitere Phasen, die während einer Wärmebehandlung auftreten, werden an geeigneter Stelle eingehender diskutiert.

Zur Festigkeitssteigerung mittels einer Kalt- oder Warmauslagerung wird der sogenannte Lösungsglüh-Prozess vorangestellt. Hier wird das Werkstück in einem Ofen bei Temperaturen nahe der Schmelztemperatur für eine bestimmte Zeit gehalten — grundsätzlich bei Temperaturen zwischen 520 °C und 540 °C für 2 h bis 6 h — und anschließend in einem Wasser- oder

Ölbad abgeschreckt. Das Lösungsglühen dient dazu, i) Seigerungseffekte die während der Erstarrung auftreten zu beseitigen (Erzeugung einer homogenen Verteilung der Legierungsatome im gesamten Gussteil durch Diffusionsvorgänge); ii) dem Aufbrechen und Abrunden der nadelartigen Siliziumphase im Eutektikum; iii) dem Auflösen bereits entstandener Ausscheidungen, die während dem Abkühlprozess in der Gussform oder aufgrund von Lagerungszeiten der Gussteile auftreten können und iv) das Generieren eines übersättigten Mischkristalls; vergleiche Colley (2011). Das Abschrecken ist wichtig, um den übersättigten Mischkristall beizubehalten. Dieser Zustand bedeutet nicht anderes, als dass das Aluminium bei höheren Temperaturen die Fähigkeit besitzt eine größere Konzentration von beispielsweise Magnesium oder Kupfer zu lösen als bei Raumtemperatur. Durch das schnelle Abschrecken wird die Entmischung verhindert, da der Diffusionsvorgang eingefroren wird. Zusätzlich entstehen die sogenannten Freistellen im Aluminiumgitter, die essenziell die Diffusionsvorgänge im Material verstärken; vergleiche Fazeli und Sinclair (2008).

Beim Kaltauslagern wird das Gusstück für einige Tage bis Wochen bei Raumtemperatur gelagert. Durch die Übersättigung des Mischkristalls und der Freistellen im Aluminiumgitter, formen sich durch Diffusionsvorgänge die sogenannten Cluster, welche eine Festigkeitssteigerung zur Folge haben. Starink et al. (2012) gibt einen Zeitraum von einigen Tagen für das Erreichen der Maximalverfestigung an, wobei diese sich über Jahre hinweg nicht weiter verändert. Je nach Legierung bewegen sich — aufgrund der Struktur der Cluster-Ausscheidungen — die maximal erzielbaren Werte für die Streckgrenze zwischen 135 MPa bis 150 MPa; vergleiche Colley (2011) und Starink et al. (2012).

Wird das Gusstück bei höheren Temperaturen ausgelagert, können quantitativ größere Werte der Streckgrenze  $R_{p,0.2}$  erzielt werden, da sich die Cluster-Phase aufgrund der erhöhten Kinetik (bezüglich der höheren Temperatur) in andere Phasen umwandelt und diese einen größeren Verfestigungsbeitrag erzielen. Colley (2011) erreicht bei experimentellen Untersuchungen mit einer T6 Wärmebehandlung und der AlSi7Mg0.3 Legierung Werte bis 256 MPa. Die Werte der Streckgrenze  $R_{p,0.2}$  sind stark von der verwendeten Legierung und der durchgeführten Wärmebehandlung abhängig und werden durch den Zeitpunkt der Beendigung der Wärmebehandlung maßgeblich bestimmt. Klassifiziert wird hier zwischen Unteralterung, Peak-Alterung und Überalterung die in DIN EN 1706:2013-12 (2013) mit den entsprechenden Bezeichnungen angegeben sind.

Wie bereits beschrieben, werden festigkeitssteigernde Maßnahmen durch Wärmebehandlungen erzielt, sofern die Legierung dies erlaubt (unterschiedliche Temperaturlöslichkeit der Legierungselemente in Aluminium). Ist dies nicht gegeben, können über das Kalt- oder Warmumformen (Schmieden) der Legierung Gitterversetzungen erzeugt werden, die eine Festigkeitssteigerung bewirken. Eine entsprechende Studie diesbezüglich ist von Kavosi et al. (2014) veröffentlicht worden.

Eine andere Möglichkeit, die Eigenschaften der Legierung zu verändern, kann durch die Modifikation der Mikrostruktur bewerkstelligt werden. Hier wird vor allem die Struktur des Eutektikums, das unter anderem von Hajkowski et al. (2012); Kuzmanov et al. (2014); Jakesch (2015) oder Dybowski et al. (2015) beschrieben wird, abgeändert. Ermöglicht wird dies durch die Zugabe von Elementen wie Strontium oder Antimon; vergleiche die Werke von Prasada Roa et al. (2008) und Timpel et al. (2012). Diese zwei Elemente wirken wachstumshemmend auf die Siliziumphase, wodurch die Größe der einzelnen Partikel merklich verringert wird und es da-

durch zu einer Verfeinerung der Mikrostruktur kommt. Zusätzlich können Additive der Schmelze hinzugeführt werden, die einen höheren Schmelzpunkt als die der Ausgangslegierung besitzen. Prasada Roa et al. (2008) gibt hier den Kornfeinungszusatz Al–Ti–C an. In anderen Studien, wie beispielsweise der von Aguirre-De la Torre et al. (2014), wird der Feinungszusatz namens Al–5Ti–B verwendet. Da diese Feststoffe ihren Aggregatzustand in der Schmelze nicht ändern, wirken diese als Nukleationsstellen während des Erstarrens und beeinflussen dadurch die Mikrostruktur. Zusätzlich ist es möglich keramische Nanopartikel als erweiternde Nukleationsstellen in die Legierung einzubringen. Ein sich darauf beziehende Studie ist von Borodianskiy et al. (2015) verfasst worden.

Die Mikrostruktur wird außerdem stark durch die Zusammensetzung der Legierung selbst beeinflusst. Salleh et al. (2015) beschreibt, dass die Erhöhung der Magnesiumkonzentration zu einer Verfeinerung der Aluminiumdendriten und der eutektischen Siliziumphase führt und signifikanten Einfluss auf die Verfestigung während einer T6 Wärmebehandlung hat. Ferner wird aufgezeigt, dass bei gewissen Konzentrationen der Legierungselemente andere Phasen ausgebildet werden können, die Einfluss auf die Eigenschaften der Legierung nehmen. Ähnliche Untersuchungen werden von Caceres et al. (1998) diskutiert. Mackay und Gruzleski (1998) verweist darauf, dass bei höherer Eisenkonzentration stets ein Teil des Magnesiums in die sogenannten eisenhaltigen Phasen wie beispielsweise der AlMgFeSi-Phase ( $\text{Al}_8\text{Mg}_3\text{FeSi}_6$ ) eingebaut wird und nicht mehr für die verfestigenden Phasen beim Auslagern zur Verfügung steht.

Zwei weitere Aspekte, welche die Mikrostruktur von Aluminiumlegierungen beeinflussen sind die Gießtemperatur und die verwendete Abgusstechnik. Wang et al. (2004) diskutiert die Einflussnahme der Abgusstemperatur und zeigt mithilfe von experimentellen Untersuchungen die Abhängigkeit der Korngröße zur Gießtemperatur auf. Eine Studie bezüglich der Abgusstechnik und der daraus resultierenden Mikrostruktur wurde von Raji (2010) verfasst.

Eine Übersicht der oben genannten Aussagen ist in Apelian et al. (1989) zusammengeführt. Hier wird detailliert auf die Einflussnahme der einzelnen Prozessschritte und deren Parameter wie beispielsweise dem Lösungsglühen (Zeit und Temperatur), dem Abschrecken (Kühlrate), dem Auslagern (Zeit und Temperatur) oder der Legierungszusammensetzung (Variation der Magnesiumkonzentration) eingegangen. Weitere Details und Einblicke in Aluminiumlegierungen können aus dem Buch von Mohamed und Samuel (2012) in Kapitel 4 eingesehen werden.

### 1.3. Optimierung

Der Begriff der Optimierung tritt in vielen Themenbereichen auf. So kann beispielsweise das Profil eines Tragflügels optimiert werden, um den Auftriebsbeiwert zu erhöhen und damit Energiekosten zu sparen.

Entsprechend bedeutet die Optimierung im mathematischen Sinne, funktionale Zusammenhänge zu erkennen beziehungsweise zu erfassen (Objektfunktion) und daraus die Extrempunkte abzuleiten oder zu bestimmen — Minima oder Maxima. Im Allgemeinen werden variable Parameter mit dieser Objektfunktion verknüpft. Als Objektfunktion kann die Streckgrenze  $R_{p,0.2}$  des Materials herangezogen werden. Die variablen Parameter stellen hierbei gegebenenfalls die Temperatur und Haltezeit beim Auslagern dar.

Ein Optimierer ist somit ein mathematisches Konstrukt, das die veränderbaren Eingangsparameter mit einer oder mehreren Antwortfunktionen verknüpft und mithilfe entsprechender For-

mulierungen und Algorithmen versucht ein Optimum zu finden. Die Methoden, die grundsätzlich verwendet werden, erstrecken sich von einfachen Sampling- über Gradienten basierten bis hin zu genetischen Verfahren. Zudem gibt es eine Vielzahl von verschiedenen Kombinationen wie vergleichsweise die Surrogate-Modelle, die Sampling-Methoden mit genetischen Verfahren kombinieren.

Sampling Methoden werden eingesetzt, wenn nicht bekannt ist, wie sich die Antwortfunktion mit einer Änderung der Eingabeparameter verhält. Sie werden verwendet, um Statistiken zu ermitteln und eine Vorstellung zwischen Eingabe- und Ausgabewert zu erhalten. Eine Optimierung selbst findet nicht statt. Bekannte Sampling Methoden sind die *Random-Sampling* oder *Lattice-Hypercube-Sampling* Methode (LHS). Beide generieren individuelle Punkte, welche sich im Gültigkeitsbereich, der vom Benutzer angegeben wird, bewegen. Eine Auflistung verschiedener Sampling-Methoden geben Shields und Zhang (2015) in ihrer Veröffentlichung an und beschreiben die mathematischen Zusammenhänge dieser. Olsson et al. (2003) gibt die Wichtigkeit dieser Methode im Gegensatz zum Standard *Monte Carlo Sampling* Verfahren (MCS) an und verweist auf die Einsparung von Rechnerressourcen um mehr als 50 %. Es ist anzumerken, dass jede Sampling-Methode seine Berechtigung hat. Beispielsweise wird die MCS für die Wahrscheinlichkeitsberechnung der geostationären Position von Satelliten verwendet, wie Sun et al. (2017) in seinem Artikel aufzeigt.

Gradienten basierte Verfahren werden für die Optimierung, dem Suchen eines Minimums oder Maximums, eingesetzt. Diese Verfahren finden mit wenig Rechenaufwand die korrespondierenden Extrempunkte, sind aber — für sich genommen — nur auf lokale Gebiete beschränkt. Das bedeutet, dass der Algorithmus ein Extremum findet, welcher jedoch nicht das Globale/Absolute darstellen muss. Eines der populärsten Gradientenverfahren ist das Schema nach *Fletcher-Reeves* (FR); vergleiche Fletcher und Reeves (1964). Agrawal und Silakari (2014) wenden dieses Verfahren im Bereich der Ornithologie an. Hierbei wird das FR-Verfahren erweitert, um das Finden des globalen Extremums zu ermöglichen. Ähnlich wie Agrawal und Silakari (2014) verwendet auch Zhang et al. (2006) einen modifizierten FR-Algorithmus zur Suche von globalen Extrempunkten. Die Bestimmung der Gradienten nach dem FR-Verfahren wird in vielen Bereichen angewendet und wird oft mit anderen Methoden verknüpft; siehe Bhunia et al. (2015). Für die Verwendung von Gradienten basierten Verfahren muss die Antwortfunktion kontinuierlich und differenzierbar sein.

Genetische Algorithmen arbeiten mit Populationen und werden häufig für Problemstellungen mit mehreren abhängigen Antwortfunktionen verwendet; vergleiche Kahrom und Javadi (2012). Eine Population stellt dabei willkürliche Parameter dar, die durch Sampling-Methoden generiert werden; oftmals wird nur die erste Population mit einer Sampling-Methode erstellt. Ausgehend von dieser Population wird das beste Resultat gesucht. In der Nähe des besten Punktes wird anschließend eine neue Population erstellt und der Algorithmus beginnt erneut die zweite Generation auszuwerten. MacCormack et al. (2002) verwendete einen solchen Algorithmus, um die Separation der Luft während des Überströmens einer Platte zu minimieren, wodurch der Widerstandsbeiwert minimiert werden konnte. Als Antwort- beziehungsweise Objektfunktion wurde der Druck herangezogen. Eine ähnliche Studie diesbezüglich wurde von Ng und Leng (2002) veröffentlicht, wobei die Minimierung des Widerstandsbeiwerts bei gleichzeitiger Maximierung des Auftriebsbeiwerts im Fokus liegt. Kahrom und Javadi (2012) verwenden diesen Algorithmus zur Optimierung des Wärmeübergangskoeffizienten zwischen Fluid und Festkörper



während einer turbulenten Strömung. In der Literatur werden diese Algorithmen häufig in *Single Object Generic Algorithm* (SOGA) und *Multi Object Generic Algorithm* (MOGA) eingeteilt. Da genetische Problemstellungen, wie bereits zu Beginn erwähnt, bei der Beurteilung mehrerer Antwortfunktionen verwendet werden, stellt MOGA die populärste Methode dar. Es sollte hierbei erwähnt werden, dass die Auswahl der nächsten Populationen stets mit dem sogenannten *Pareto-Set* durchgeführt wird. Durch die Verwendung von Pareto-Sets ist sichergestellt, dass die neue Generation — im Bezug auf die Alte — stets eine Verbesserung darstellt (sofern dies gewünscht ist).

Werden die *Surrogate-Modelle* (SG) verwendet, wird mittels einer Sampling Methode die Antwortfunktion an den sogenannten *experimentellen* Punkten bestimmt und anschließend mit einem geeigneten mathematischen Verfahren durch diese eine Funktion approximiert. Die Funktionsauswahl kann hier frei gewählt werden. Eine der bekanntesten Approximationsfunktion ist die Gaussfunktion, die zusätzlich Informationen in Hinblick auf die statistische Ungewissheit enthält. Die SG Methoden werden stets dann verwendet, wenn die numerische Berechnung der Antwortfunktion mit rechenintensivem Aufwand erzeugt wird. Durch die Approximation der Funktion, die mithilfe der experimentellen Punkte erhalten wird, kann eine sehr einfache Optimierung durchgeführt werden. Die Laboratories geben weiter an, dass sich diese Methode vor allem für rauschende Antwortfunktionen anbietet, da das Störsignal aufgrund des Funktionfits herausgefiltert wird. Suwartadi et al. (2015) verwenden diese Methodik beispielsweise im Bereich des Bauingenieurwesens.

Wie zu sehen ist, sind Optimierungsalgorithmen vielseitig einsetzbar und können miteinander verknüpft werden. Werden eine Reihe unterschiedlicher Methoden zu einer Optimierungsmethode zusammengesetzt, wird von Hybridmethoden gesprochen.

Alle die oben genannten Methoden und weitere Algorithmen sind in der Open Source Toolbox DAKOTA<sup>®</sup> auswählbar. Adams et al. (2015) gibt hierzu eine detaillierte Übersicht der Möglichkeiten und Einsatzgebiete an. Weitere Informationen zur Verwendung von DAKOTA<sup>®</sup> ist auf der offiziellen Webseite unter <https://dakota.sandia.gov/> zu finden.



## 2. Spannungsmodellierung mit der Finiten Volumen Methode

Nachfolgendes Kapitel erläutert die numerische Modellierung der mechanischen Spannungen, im Besonderen der thermisch induzierten Spannungen im linear-elastischen Bereich. Zu Beginn wird auf zwei verschiedene Methoden eingegangen, die zur Lösung komplexer strukturmechanischer Probleme eingesetzt werden und erläutert, weshalb die Finite Volumen Methode (FVM) anstelle der Finiten Elemente Methode (FEM) in der vorliegenden Arbeit verwendet wird. Anschließend wird die partielle Differenzialgleichung des Hookeschen Gesetzes für den Verschiebungsvektor  $\mathbf{D}$  hergeleitet, alle Terme analysiert und beschrieben sowie eine numerische Optimierung und die Erweiterung dieser Grundgleichung für thermische Problemstellungen gegeben. Nach der Herleitung der Gleichungen wird auf die neu entwickelte Randbedingung eingegangen. Abschließend werden Validierungsfälle betrachtet, die zur Bestätigung der korrekten Implementierung der Grundgleichungen herangezogen werden und kurz und prägnant die Einflussnahme der Numerik diskutiert.

### 2.1. Methoden zur Spannungsberechnung

Strukturanalysen von Bauteilen, die unter Beaufschlagung von Lasten und Kräften Verformungen erfahren, werden im Allgemeinen mit der FEM behandelt. In der numerischen Strömungsmechanik hingegen wird bevorzugt die FVM eingesetzt. Beide Varianten lösen große lineare Gleichungssysteme, jedoch wird die Diskretisierung der partiellen Differenzialgleichung unterschiedlich behandelt.

Bei der FEM wird die Variationsrechnung mit Formfunktionen auf vordefinierte Elemente angewendet. Daraus resultiert eine einfache Erweiterung der numerischen Genauigkeit aufgrund des Diskretisierungsgrads; Terme höherer Ordnung können ohne großen Mehraufwand implementiert werden. Die numerisch erhaltene Matrix besteht oft aus einzelnen Blockmatrizen — beispielsweise für jeden Freiheitsgrad eine —, wobei jede Blockmatrix eine individuelle Struktur aufweisen, welche von den verwendeten Elementen abhängig ist. Hinzu kommt, dass der Typ der Differenzialgleichung für die erhaltene Matrixstruktur verantwortlich ist; Stichworte: *Strong formulation of the problem* → *Weak formulation of the problem*, Variationsrechnung und Minimierungsproblem. Weitere fundamentale Angaben zum Verfahren werden in den Veröffentlichungen von Sayas (2008) und Goering et al. (2010) erläutert.

In der FVM hingegen werden Integralgleichungen verwendet, wodurch die Genauigkeit ohne Mehraufwand im Allgemeinen der zweiten Ordnung entspricht. Die Diskretisierung der Integralgleichungen ergeben schwach besetzte diagonal dominante Matrizen und ist vom Typ der Differenzialgleichung abhängig (Integral). Gekoppelte Größen und nicht-lineare Terme werden in der FVM meistens separat (segregiert, entkoppelt) behandelt, wodurch iterative Lösungsverfahren eingesetzt werden müssen. Allerdings sind heutzutage auch gekoppelte lineare Löser für nicht-

lineare Problemstellungen einsetzbar. Eine fundierte Einführung in die FVM ist in den Werken von Jasak (1996); Ferziger und Perić (2008) sowie Moukalled und Mangani (2016) zu finden.

Aufgrund der verschiedenen mathematischen Behandlungen ergeben sich Vor- und Nachteile für bestimmte Problemstellungen. Eine interessante These die Jasak und Weller (1998) bezüglich den Anwendungsbereichen beider Methoden heranziehen, ist die nicht eindeutige Klassifizierung der Einsatzgebiete zwischen FEM und FVM. Das bedeutet, dass es nicht offensichtlich ist, wann welche Methode verwendet werden soll, auch wenn beispielsweise die FEM für Struktur- und die FVM für Strömungsanalysen — gegebenenfalls aufgrund des geschichtlichen Hintergrunds — bevorzugt eingesetzt werden. Gezielt wird darauf eingegangen, dass es nicht möglich ist, im Voraus eine klare Aussage zu treffen, ob das Lösen der Problemstellung mit der FEM oder FVM besser geeignet ist.

Zudem stellt sich die Frage, mit welchem Aufwand das Erstellen der numerischen Netze — die Diskretisierung des Kontinuums — verbunden ist. Können gegebenenfalls bei der FEM nur spezielle Elementtypen für definierte Problemstellungen verwendet werden, erlaubt die FVM durch ihren integralen Charakter jeden Volumentyp zur Diskretisierung heranzuziehen. Folglich wird ein wichtiger Freiheitsgrad bei der Netzerstellung gewonnen. Zusätzlich ist es möglich, das erstellte Netz für mehrere unterschiedliche Analysen zu verwenden. Die FVM erlaubt es daher, das gleiche numerische Netz ohne weitere Anpassung für die Berechnung des Temperatur- und Spannungsfeldes zu verwenden. Bei der FEM hingegen muss der Elementtyp entsprechend der Problemstellung angepasst oder gegebenenfalls ein neues Netz erzeugt werden.

## **Abschließend**

Die Unterschiede der beiden Methoden liegen hauptsächlich im verwendeten Gleichungstyp. In der FVM werden hauptsächlich Erhaltungsgleichungen für die Betrachtung einer Größe herangezogen. In der FEM jedoch wird das Variationsprinzip herangezogen wodurch die Erhaltungsgleichungen (Strong Formulation of the Problem) in eine Gleichung überführt, die die Weak Formulation of the Problem repräsentiert; vergleiche Goering et al. (2010).

## **2.2. Gründe für die Implementierung in ein FVM Programm**

In der vorliegenden Arbeit wird die thermo-elastische Spannungsanalyse für die FVM hergeleitet und im Open-Source-Programm OpenFOAM<sup>®</sup> implementiert; vergleiche Weller et al. (1998) und Jasak et al. (2007). Zwei bereits genannte Gründe, die für die FVM sprechen, sind die geringen Computerkosten für das Lösen der diagonal dominanten Matrizen und die Verwendung beliebiger Volumentypen. Eine weitere Motivation ist die bereits erfolgreiche Umsetzung der linear-elastischen Strukturanalyse mittels der FVM. Jasak und Weller (1998) berechneten bereits Lastfälle in 2D und 3D. Die betrachteten Problemstellungen waren stationären (statische Spannungsfälle) und transienten Ursprungs; eine detaillierte Analyse von Schallwellenausbreitungen im Material (Diesel Injektionshalter). Cardiff (2012) analysierte Hüftgelenke im linear-elastischen Bereich und entwickelte Methoden zur Berechnung von Kontaktinterferenzen mehrerer Festkörper. Die Verwendung der FVM für die Strukturanalyse ist besonders dann von Vorteil, wenn Fluid-Solid-Interaktion (FSI) auftreten. Beispiele wären hier die Deformation von Windturbinenrädern aufgrund des umströmenden Fluids oder die Verformung von Flugzeugtragflügeln durch die Wechselwirkung mit dem Fluid. Dieses Themengebiet wurde von Greenshields

und Weller (2005) bearbeitet und erfolgreich umgesetzt. Weitere Gründe, die für die Verwendung der FVM im Bereich der thermo-elastischen Strukturanalyse sprechen, sind folgende bereits erfolgreich umgesetzte Themenbereiche:

- Thermische Spannungsanalyse in einem Zylinderkopf; Demirdžić und Muzaferija (1995),
- Strukturanalyse mit großen Verformungen; Tuković und Jasak (2007),
- Block-Coupled Lösungsalgorithmen; Cardiff et al. (2016).

Es steht außer Frage, dass die Strukturanalyse im linear-elastischen Bereich auf beiden Seiten ihre Argumente aufweist. Gerade für sehr große mechanisch induzierte Verformungen kann die FEM aufgrund ihrer Diskretisierung und Transformationsregeln numerisch punkten, da bei großen Deformationen die Möglichkeit einer extremen Topologieänderung der Volumina entstehen kann, die beispielsweise in der FVM größere numerische Probleme mit sich bringt.

Ein weiterer Aspekt, der bislang außer Acht gelassen worden ist, jedoch stark für die Implementierung in OpenFOAM<sup>®</sup> beiträgt, liegt in der Verwendung der Objekt-Orientierten Programmierung (OOP), die vor allem die Modularität von Programmen fördert und damit die Weiterentwicklung, Wartung und Flexibilität enorm steigert.

### 2.3. Herleitung der thermo-elastischen Gleichung

Der Ausgangspunkt zur Herleitung der Grundgleichung für die thermo-elastische Strukturanalyse kann vielseitig gewählt werden. Im vorliegenden Fall wird als Ausgangspunkt die Impulserhaltungsgleichung in ihrer allgemeinen Form verwendet, wie sie beispielsweise in Bird et al. (1960), Wilcox (1994) oder in Ferziger und Perić (2008) zu finden ist. Es wird angemerkt, dass alle nachfolgenden Vektoroperationen im Anhang C auf Seite 175 zusammengefasst sind.

Die Ausgangsgleichung lautet:

$$\underbrace{\frac{\partial \rho \mathbf{U}}{\partial t}}_{\text{zeitliche Änderung}} + \underbrace{\nabla \bullet (\rho \mathbf{U} \otimes \mathbf{U})}_{\text{konvektiver Fluss}} = \underbrace{\nabla \bullet \boldsymbol{\sigma}}_{\text{diffusiver Fluss}} + \underbrace{S_{\mathbf{U}}}_{\text{Quellterm}}. \quad (2.1)$$

Dabei entspricht  $\rho$  der Dichte des Materials,  $\mathbf{U}$  dem Geschwindigkeitsvektor mit den Komponenten in  $u_x$ ,  $u_y$  und  $u_z$ ,  $t$  der Zeit,  $\boldsymbol{\sigma}$  dem Cauchy Spannungstensor und  $S_{\mathbf{U}}$  anderen Volumenkräften wie beispielsweise der Gravitationskraft, die im Nachfolgenden vernachlässigt werden. Das Symbol  $\nabla$  definiert den Nabla-Operator und enthält die örtlichen Ableitungen in  $x$ ,  $y$  und  $z$ . Das Konstrukt  $\nabla \bullet$  beschreibt den Divergenzoperator; vergleiche Anhang C.

Anders als in der Strömungsmechanik, bei der die Geschwindigkeit des Fluids gesucht wird, muss bei der Strukturmechanik die Verschiebung bestimmt werden. Der Geschwindigkeitsvektor  $\mathbf{U}$  wird daher als die zeitliche Änderung der Verschiebung ausgedrückt:  $\mathbf{U} = \dot{\mathbf{D}}$ . Des Weiteren wird angenommen, dass die Verformungen sehr klein sind (linear-elastischer Bereich), wodurch der konvektive Term in Gleichung (2.1) vernachlässigt werden kann. Daraus resultiert die Gleichung für den Verschiebungsvektor  $\mathbf{D}$ :

$$\frac{\partial \rho \partial \mathbf{D}}{\partial t^2} = \nabla \bullet \boldsymbol{\sigma}. \quad (2.2)$$

Der Cauchy Spannungstensor  $\boldsymbol{\sigma}$  wird durch das *Hookesche Gesetz* ausgedrückt, welches analog zum viskosen Spannungstensor für newtonsche Fluide formuliert wird. Gurtin et al. (2010) fasst dies prägnant mit allen mathematischen Beweisen zusammen. Der resultierende Spannungstensor  $\boldsymbol{\sigma}$  ist definiert als:

$$\boldsymbol{\sigma} = 2\mu\mathbf{D}^S + \lambda \operatorname{tr}(\mathbf{D}^S)\mathbf{I} , \quad (2.3)$$

wobei  $\mathbf{I}$  den Einheitstensor und  $\operatorname{tr}()$  den Trace-Operator angibt.  $\mathbf{D}^S$  wird auf nächster Seite definiert.

Das Hookesche Gesetz enthält die Größen  $\lambda$  und  $\mu$ , die unter dem Namen der *Lamé Koeffizienten* bekannt sind und durch Materialeigenschaften beschrieben werden. Die Definition diese Größen lauten; vergleiche Boley und Weiner (1985); Jasak und Weller (1998) sowie Cardiff (2012):

$$\mu = \frac{E}{2(1 + \nu)} , \quad (2.4)$$

$$\lambda = \begin{cases} \frac{\nu E}{(1+\nu)(1-\nu)} & \text{ebener Spannungszustand} \\ \frac{\nu E}{(1+\nu)(1-2\nu)} & \text{ebener Verformungszustand und 3D .} \end{cases} \quad (2.5)$$

Zusätzlich enthält die Gleichung (2.3) den Verschiebungstensor  $\mathbf{D}^S$ , auch bekannt unter der englischen Bezeichnung des Strain-Tensors. Die formale Definition der Gleichung des Verschiebungstensors  $\mathbf{D}^S$  ist äquivalent zum Strain-Rate-Tensor — Ferziger und Perić (2008) bezeichnet diesen auch als Deformationstensor — in der Impulserhaltungsgleichung; allerdings wird im vorliegenden Fall der Verschiebungs- und nicht der Geschwindigkeitsvektor verwendet:

$$\mathbf{D}^S = \frac{1}{2} \left[ \nabla \otimes \mathbf{D} + (\nabla \otimes \mathbf{D})^T \right] . \quad (2.6)$$

Das mathematische Konstrukt  $\nabla \otimes \mathbf{D}$  beschreibt das dyadische Produkt zweier Vektoren — hier den Nabla Vektor  $\nabla$  und den Verschiebungsvektor  $\mathbf{D}$  und wird ferner als *äußeres Produkt* bezeichnet; vergleiche Anhang C. Das hochgestellt  $^T$  beschreibt das Transponieren des Klammersausdrucks und ist auf die Matrix-Algebra zurückzuführen.

Die Formulierungen (2.2) bis (2.6) bilden damit ein geschlossenes Gleichungssystem, das zur Berechnung der Verschiebungen im Material aufgrund äußerer Einwirkungen, wie beispielsweise Zug- und/oder Druckbelastungen, verwendet werden kann. Die resultierende, zu lösende Gleichung lautet:

$$\frac{\partial \rho \partial \mathbf{D}}{\partial t^2} = \underbrace{\nabla \bullet [\mu(\nabla \otimes \mathbf{D})]}_{\text{implizit}} + \underbrace{\nabla \bullet [\mu(\nabla \otimes \mathbf{D})^T] + \nabla \bullet [\lambda(\nabla \bullet \mathbf{D})\mathbf{I}]}_{\text{explizit}} , \quad (2.7)$$

wobei folgende Beziehung für den dritten Term auf der rechten Seite Anwendung findet:

$$\operatorname{tr}(\mathbf{D}^S) = (\nabla \bullet \mathbf{D}) . \quad (2.8)$$

Weitere Informationen sind in Anhang C auf Seite 175 gegeben. Die Bezeichnung *explizit* und *implizit* der einzelnen Terme bezieht sich auf die numerische Lösungsprozedur. Explizite Terme fließen in den Quellterm  $\mathbf{b}$  und implizite Terme in die Koeffizientenmatrix  $\mathbf{A}$  hinein; vergleiche Anhang B.1 auf Seite 155.

Die Einschränkung des Anwendungsbereichs der Gleichung (2.7) ist durch das Hookesche Gesetz gegeben, wodurch diese Erhaltungsgleichung nur für den *linear-elastischen* Verformungsbereich Gültigkeit besitzt. Bei der Betrachtung der Gleichung (2.7) fällt weiter auf, dass keine Quellterme enthalten sind. Entsprechend können sich Verschiebungen und die daraus resultierenden Spannungen nur durch äußere Kräfte, die mithilfe von Randbedingungen definiert werden, ergeben. Dabei können sowohl Kräfte als auch Verschiebungen als Ursache wirken; die Betrachtung und Herleitung der Randbedingungen wird später gesondert behandelt.

Die Erweiterung der Gleichung (2.2) beziehungsweise (2.7) auf thermo-elastische Problemstellungen wird durch das Hinzufügen eines Quellterms, der aufgrund von Volumenänderung Verschiebungen induziert, geschaffen. Die Generierung der Volumenänderung basiert auf dem linearen Ausdehnungskoeffizienten  $\alpha$  und beschreibt die Kontraktion beziehungsweise Expansion des Volumens aufgrund thermischer Änderung. Die daraus resultierende Gleichung wird als *Duhamel-Neumann* Formulierung des Hookeschen Gesetzes bezeichnet. Der korrespondierende Quellterm ist wie folgt definiert und ist sowohl in Boley und Weiner (1985) als auch von Demirdžić und Muzaferija (1995) gegeben:

$$S_{\mathbf{D}} = \nabla \bullet [-(3\lambda + 2\mu)\alpha(T - T_0)\mathbf{I}] . \quad (2.9)$$

Wie bereits erwähnt, bezeichnet  $\alpha$  den linearen Ausdehnungskoeffizienten und  $T$  die Temperatur in Kelvin oder Grad Celsius. Für den Quellterm wird eine Referenztemperatur  $T_0$  hinzugefügt, bei der angenommen wird, dass keine thermischen Spannungen vorliegen.

Wird der neue Quellterm zur Hookeschen Gleichung hinzugefügt, wird die Duhamel-Neumann Formulierung erhalten. Diese lautet:

$$\frac{\partial \rho \partial \mathbf{D}}{\partial t^2} = \underbrace{\nabla \bullet [\mu(\nabla \otimes \mathbf{D})]}_{\text{implizit}} + \underbrace{\nabla \bullet [\mu(\nabla \otimes \mathbf{D})^T]}_{\text{explizit}} + \underbrace{\nabla \bullet [\lambda(\nabla \bullet \mathbf{D})\mathbf{I}]}_{\text{explizit}} - \underbrace{\nabla \bullet \{[(3\lambda + 2\mu)\alpha(T - T_0)\mathbf{I}]\}}_{\text{explizit}} , \quad (2.10)$$

und gibt Aufschluss über die Kopplung zum Temperaturfeld (letzter Term). Dies ermöglicht die Berechnung von Verschiebungen aufgrund thermischer Einwirkung ausgehend von der Referenztemperatur  $T_0$ . Zur Lösung der Duhamel-Neumann Gleichung wird kausal bedingt ein korrespondierendes Temperaturfeld benötigt — die Erhaltungsgleichung der Energie wird im nachfolgenden Kapitel hergeleitet. Des Weiteren wird im späteren Verlauf dieser Arbeit eine Analyse und Charakterisierung der Gleichung (2.10) getätigt und aufgezeigt, welche Methoden verwendet werden können, um die Temperatur-Verschiebung-Kopplung numerisch zu behandeln.

Abschließend wird darauf hingewiesen, dass die Duhamel-Neumann Formulierung für iterative Löser bislang ungeeignet ist, da die Diagonaldominanz der Matrix aufgrund des schwachen impliziten Anteils gering ist. Eine Verbesserung der Numerik wird später aufgezeigt.

## Von Mises Spannung

Liegt das Ergebnis des Verschiebungsvektors  $\mathbf{D}$  nach dem Lösen der Duhamel-Neumann Gleichung vor, ist es möglich, über Gleichung (2.3), den Spannungstensor  $\boldsymbol{\sigma}$  zu bestimmen. Anschließend können die bekannten äquivalenten Von Mises Spannungen und Dehnraten  $\sigma_{\text{Mises}}$

und  $\epsilon_{\text{Mises}}$  über folgende Korrelationen berechnet werden:

$$\sigma_{\text{Mises}} = \sqrt{\frac{3}{2} |\text{dev}(\boldsymbol{\sigma})|^2}, \quad (2.11)$$

$$\epsilon_{\text{Mises}} = \sqrt{\frac{2}{3} |\text{dev}(\mathbf{D}^S)|^2}. \quad (2.12)$$

Der Operator  $\text{dev}()$  bezeichnet den deviatorischen Teil des Tensors  $\mathbf{D}^S$  und ist in Anhang C definiert.

Die Von Mises Spannungen und Dehnraten werden generell im Ingenieurbereich zur Beurteilung des Lastfalls herangezogen. Begründet ist dies darin, dass die Von Mises Werte Skalare repräsentieren, die mit Materialwerten vergleichbar sind. Mithilfe der Spannungs- und Verschiebungstensoren ist es *nicht möglich* eine Gegenüberstellung verschiedener Lastfälle durchzuführen.

## 2.4. Herleitung der Energiegleichung

Zur Berechnung thermisch induzierter Spannungen, die in beliebigen Materialien durch Temperaturdifferenzen oder -änderungen hervorgerufen werden, ist es notwendig, den zeitlichen Verlauf des Temperaturfeldes zu kennen. Die Herleitung der Erhaltungsgleichung der Temperatur wird unter Verwendung der Enthalpieerhaltungsgleichung, welche in Bird et al. (1960); Schwarze (2013) oder Moukalled und Mangani (2016) beschrieben ist, durchgeführt. Der Grund der expliziten Herleitung basiert auf den oftmals unzureichenden und unvollständigen Angaben in diverser Literatur.

Der Erhaltungssatz für die spezifische Enthalpie lautet:

$$\underbrace{\frac{\partial \rho h}{\partial t}}_{\text{zeitliche Änderung}} + \underbrace{\nabla \bullet (\rho \mathbf{U} h)}_{\text{konvektiver Fluss}} = \underbrace{-\nabla \bullet \mathbf{q}}_{\text{diffusiver Fluss}} + \underbrace{S_h}_{\text{Quellterm}}, \quad (2.13)$$

wobei  $\mathbf{q}$  den Wärmeflussvektor beschreibt und über das *Fourier Gesetz* definiert ist als:

$$\mathbf{q} = -\lambda_{\text{th}} \nabla T. \quad (2.14)$$

$\lambda_{\text{th}}$  repräsentiert die Wärmeleitfähigkeit des Materials. Um die Erhaltungsgleichung der Enthalpie — Formulierung (2.13) — in eine Gleichung für die Temperatur zu überführen, wird die Definition zwischen Enthalpie und Temperatur herangezogen. Diese lautet:

$$h(T) = \int_{T_0}^T c_p(T) dT. \quad (2.15)$$

Unter Verwendung der letzten drei genannten Gleichungen kann die Erhaltungsgleichung für die Temperatur hergeleitet werden. Es folgt:

$$c_p \frac{\partial \rho T}{\partial t} = \nabla \bullet (\lambda_{\text{th}} \nabla T) + S_T. \quad (2.16)$$

Eine ausführliche Herleitung mit allen Zwischenschritten ist in Anhang A.1 auf Seite 125 ge-



geben und korreliert mit den Aufzeichnungen von Ferziger und Perić (2008) sowie Moukalled und Mangani (2016). Es soll angemerkt werden, dass der konvektive Term aufgrund der kleinen Verformungen vernachlässigt werden kann. Bei Verformungen die nicht im linear-elastischen Bereich erfolgen, muss dieser berücksichtigt werden; Stichwort *mesh fluxes*. Die Dichte  $\rho$  und die Wärmekapazität  $c_p$  werden als temperaturabhängig betrachtet. Diesbezüglich ist es erwähnenswert, dass keine Ergebnisänderungen eintritt, wenn die Dichte vor das Differenzial der Zeitableitung gezogen wird.

Der letzte Term in Gleichung (2.16) repräsentiert Temperatur bezogene Quellterme. Boley und Weiner (1985) beschreiben einen möglichen Quellterm basierend auf den Verschiebungstensor wie folgt:

$$S_T = (3\lambda + 2\mu)\alpha T_0 \operatorname{tr}(\dot{\mathbf{D}}^S) . \quad (2.17)$$

Der letzte Ausdruck auf der rechten Seite wird als Scherratentensor beziehungsweise im Englischen als Strain-Rate Tensor bezeichnet und stellt die zeitliche Ableitung von Gleichung (2.6) dar. Es folgt:

$$\dot{\mathbf{D}}^S = \frac{1}{2} \left[ \nabla \otimes \left( \frac{\partial \mathbf{D}}{\partial t} \right) + \left\{ \nabla \otimes \left( \frac{\partial \mathbf{D}}{\partial t} \right) \right\}^T \right] . \quad (2.18)$$

Damit liegt eine beidseitige Kopplungsbeziehung zwischen der Energie- (Temperatur-) und Duhamel-Neumann Gleichung vor. Das bedeutet, dass Temperaturänderungen Verschiebungen und entsprechend Spannungen hervorrufen und zugleich Deformationen eine Änderung der inneren Energie und damit eine Erhöhung oder Verringerung der Temperatur darstellen; ein Beispiel für letztere Kopplung ist die Verformung von Stahl beim Schmieden.

Boley und Weiner (1985) beweisen, dass Verschiebungen im linear-elastischen Bereich keinen Einfluss auf die Temperatur des Materials haben und geben an, dass der Quellterm  $S_T$  vernachlässigt werden darf; diese Annahme gilt für die meisten Eisen- und Aluminiumlegierungen.

Der Beweis wird mithilfe einer Entdimensionierung der Temperaturgleichung herangetragen, wodurch folgende Bedingung, erhalten wird:

$$\frac{\operatorname{tr}(\dot{\mathbf{D}}^S)}{3\alpha \dot{T}} \ll \left( \frac{\lambda + \frac{2}{3}\mu}{\lambda + 2\mu} \right) \frac{1}{\delta} . \quad (2.19)$$

$\delta$  stellt eine dimensionslose Zahl mit der Definition:

$$\delta = \frac{(3\lambda + 2\mu)^2 \alpha^2 T_0}{\rho^2 c_v \nu_e^2} , \quad (2.20)$$

dar. Ferner repräsentiert  $c_v$  die Wärmekapazität bei konstantem Volumen und  $\nu_e$  die Ausbreitungsgeschwindigkeit der *Dehnungswelle* in einem elastischen Medium. Es gilt:

$$\nu_e = \sqrt{\frac{(\lambda + 2\mu)}{\rho}} . \quad (2.21)$$

Boley und Weiner (1985) leiten daraus zwei Bedingungen ab. Zum einen muss gelten, dass  $\delta \ll 1$  ist und zum anderen muss Gleichung (2.19) stets Gültigkeit besitzen. Für Aluminiumlegierungen beträgt der Wert von  $\delta \approx 0,029$ , wodurch sich folgende Charakteristik für die

Formulierung (2.19) ergibt:

$$\frac{\text{tr}(\dot{\mathbf{D}}^S)}{3\alpha\dot{T}} \ll 20. \quad (2.22)$$

Dies kann wie folgt interpretiert werden: Für thermische Beanspruchungen müssen die Verschiebungen den Temperaturänderungen Folge leisten. Aufgrund dessen dürfen keine markanten Vibrationen oder verzögerten Verschiebungen eintreten. Mit anderen Worten bedeutet dies, dass der erwärmte Körper die Möglichkeit besitzen muss, sich frei auszudehnen. Kann dies nicht vorausgesetzt werden, kann der Fall eintreten, dass die oben genannten Kriterien nicht erfüllt werden, wodurch der Quellterm  $S_T$  nicht vernachlässigt werden darf. Für lokale Wärmebehandlungen, die in dieser Arbeit untersucht werden, besitzt Gleichung (2.19) stets Gültigkeit.

Die erste Bedingung ist durch das Material selbst charakterisiert. Basierend auf der Verwendung einer Aluminiumlegierung folgt,  $\delta \ll 1$ . Des Weiteren wird Gleichung (2.22) erfüllt, da sich die Geometrien, welche in dieser Arbeit untersucht werden, frei ausdehnen können; alle Oberflächen werden als spannungsfrei betrachtet. Vibrationen im Allgemeinen können bei lokalen Wärmebehandlungen ausgeschlossen werden.

## 2.5. Berechnung der gekoppelten Gleichungen

Das Lösen beider Erhaltungsgleichungen mithilfe der FVM erfordert, sofern beide Kopplungsterme verwendet werden, einen iterativen Lösungsansatz. Da beide Gleichungen keine weiteren Quellterme enthalten, ist es nur durch Randbedingungen möglich, die Systeme zu verändern.

Boley und Weiner (1985) verweisen beim Lösen der gekoppelten Gleichungen (2.10) und (2.16) auf drei mögliche Optionen, die zur Kopplung des Verschiebungsfeldes  $\mathbf{D}$  und Temperaturfeldes  $T$  verwendet werden können. Die ausgewählte Lösungsvariante bestimmt das Einsatzgebiet und den iterativen Prozess. Boley und Weiner (1985) geben in ihrem Buch folgende Aufteilung an:

- Voll-gekoppeltes System,
- Halb-gekoppeltes System,
- Halb-gekoppeltes quasi-stationäres System.

Die Nomenklatur von voll-gekoppelt, halb-gekoppelt und halb-gekoppelt quasi-stationär ist von Boley und Weiner (1985) übernommen und stellt die Relation zwischen der Temperatur- und Duhamel-Neumann Gleichung — bezogen auf die Quellterme — dar

Jedes dieser Systeme hat spezielle Merkmale, die Auswirkungen auf den Lösungsalgorithmus haben. Zudem wird im letzten Fall die Charakteristik der Grundgleichung verändert, welche die Methode des iterativen Löser und der erhaltenen Ergebnisse beeinflusst. Im Folgenden werden alle drei Methoden näher beschrieben und die Umsetzung in einer FVM grafisch aufgezeigt.

### Voll-gekoppeltes System

Wird ein voll-gekoppeltes System betrachtet, bedeutet dies, dass jedes der beiden Gleichungssysteme vom anderen abhängt. Dies ist bedingt durch die Quellterme in den Gleichungen — siehe Gleichung (2.9) und (2.17). Mit anderen Worten bedeutet dies, dass Temperaturunterschiede Verschiebungen hervorrufen und diese wiederum eine Quelle von Temperaturänderungen darstellen. Das numerische Lösen eines solch gekoppelten Systems ist in Abbildung 2.1a dargestellt.

Voll-gekoppelte Systeme werden eingesetzt, wenn die Verformungen quantitativ große Werte annehmen und diese einen merklichen Einfluss auf die Energiegleichung besitzen. Anders ausgedrückt bedeutet das, dass der quantitative Wert des Quellterms  $S_T$  nicht vernachlässigt werden kann. Es ist erwähnenswert zu wiederholen, dass die Duhamel-Neumann Gleichung ihre Gültigkeit für kleine Verformungen besitzt. Folglich entfällt der Quellterm  $S_T$ . Werden größere Verschiebungen außerhalb des linear-elastischen Bereichs erwartet, müssen weitere Aspekte, wie dem plastischen Verformen, beachtet werden. Weitere Informationen diesbezüglich sind in Cardiff (2012) oder Tuković und Jasak (2007) zu finden.

Da Gleichung (2.10) bezüglich der Zeitableitung einen elliptischen Charakter besitzt, können sich Informationen in alle Raumrichtungen innerhalb des Lösungsgebietes ausbreiten; vergleiche Anderson (1995) oder Ferziger und Perić (2008). Daher gibt es entsprechende Kriterien die zur Stabilität und Genauigkeit eingehalten werden müssen, wie beispielsweise die Einhaltung der Courant-Zahl; hier wird die Schallgeschwindigkeit des Festkörpers als Referenzgeschwindigkeit herangezogen. Detaillierte Einblicke bezüglich Schallwellenausbreitung in Festkörpern geben Jasak und Weller (1998) in ihrer Publikation.

Es soll angemerkt werden, dass die Berechnung der Schallwellenausbreitung in dieser Arbeit lediglich im dritten Validierungsfall beachtet wird, da hier eine transiente Problemstellung betrachtet wird.

### **Halb-gekoppeltes System**

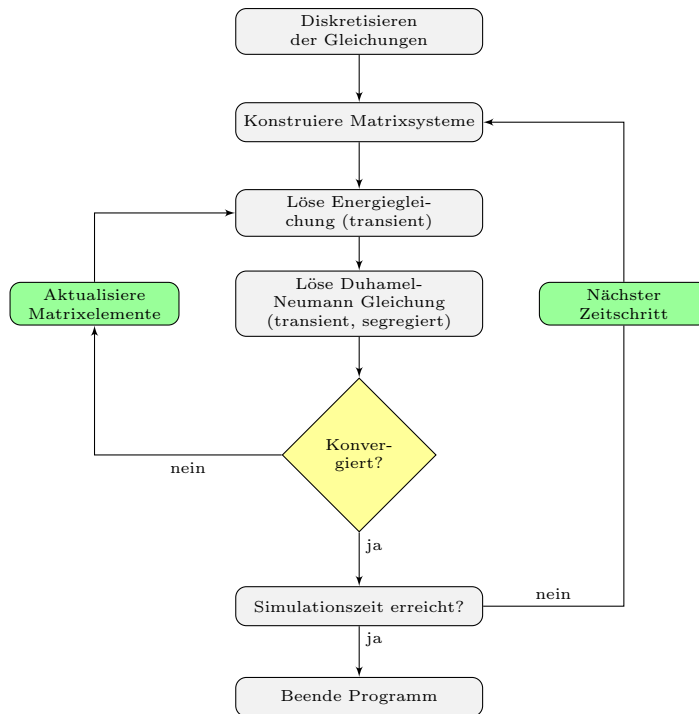
Halb-gekoppelte Systeme reduzieren die Kopplungsbeziehung. Wie bereits erwähnt, kann bei kleinen Verformungen davon ausgegangen werden, dass diese keinen nennenswerten Einfluss auf die Temperatur des Materials aufweisen. Daraus resultiert, dass der Quellterm in der Erhaltungsgleichung der Temperatur vernachlässigt werden kann. Dabei wird die Rückkopplung der Duhamel-Neumann Gleichung (2.7) auf die Temperaturgleichung (2.16) aufgehoben, wodurch der Aufwand zum Lösen dieser Gleichung vereinfacht wird. Die numerische Betrachtung dieses Problems ist in Abbildung 2.1b dargestellt.

Die Entkopplung verringert die Berechnungszeit, da jedes Gleichungssystem und das korrelierende Matrixsystem einzeln betrachtet werden kann. Hinzu kommt, dass der Quellterm (2.9) in jedem Zeitschritt konstant gehalten wird.

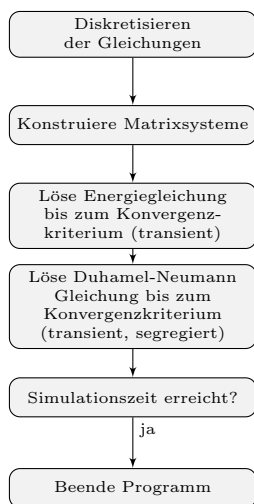
### **Halb-gekoppeltes quasi-stationäres System**

Eine weitere Vereinfachung des Gleichungssystems kann dadurch gewonnen werden, wenn Gleichung (2.10) stationär betrachtet wird. Dabei entfällt der instationäre Term und der Typ der Gleichung ändert sich. Die Berechnung der Temperaturentwicklung wird weiter transient betrachtet, wobei für die Analyse der Spannungsgleichung die Annahme getroffen wird, dass sich das Temperaturfeld in einem *bereits stationären Zustand befindet*. Daher werden die Verschiebungen ebenfalls *stationär* betrachtet. Aufgrund der Tatsache, dass das Temperaturfeld instationär und die Verschiebung stationär analysiert wird, wird dieses System als *quasi-stationär* bezeichnet. Die numerische Behandlung ist in Abbildung 2.1c gegeben und wird in der vorliegenden Arbeit eingesetzt.

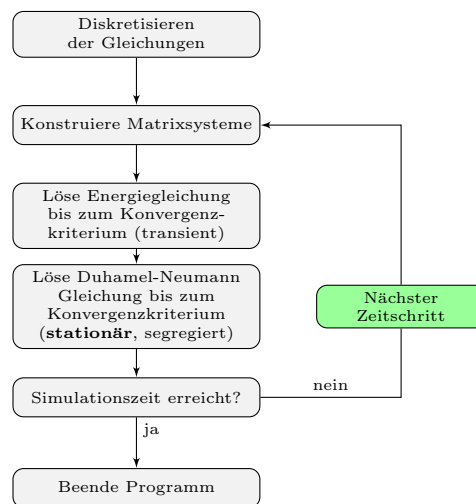
Die Möglichkeit, das System quasi-stationär zu betrachten, ist nur dann gegeben, wenn der instationäre Term keinen quantitativ-signifikanten Einfluss auf das Ergebnis hat. Boley und Weiner (1985) geben in ihrem Buch eine mathematische Betrachtung von drei verschiedenen



(a)



(b)



(c)

Abbildung 2.1.: Numerischer Lösungsalgorithmus zur Berechnung eines (a) *voll-gekoppelten*, (b) *halb-gekoppelten* und (c) *halb-gekoppelten quasi-stationärem* Systems unter Verwendung der FVM.

Legierungen an. Dabei wird explizit auf Aluminiumlegierungen eingegangen und bewiesen, dass die Annahme der Quasistationaritat gegeben ist. Der Verwendung eines halb-gekoppelten quasi-stationaren Losungsalgorithmus ist durch Gleichung (2.19) bis (2.22) begrundet.

## Abschließend

Da in der vorliegenden Arbeit die Spannungen und die sich daraus resultierenden Verformungen im linear-elastischen Bereich bewegen, kann die Kopplung zwischen Energie- und Verschiebungsgleichung als *Halb-gekoppelt* angesehen werden. Aufgrund der mathematischen Betrachtung, die durch Boley und Weiner (1985) gegeben wurde, kann weiter eine Quasistationaritat angenommen werden. Entsprechend wird das Modell, welches in Abbildung 2.1c dargestellt ist, verwendet.

## 2.6. Numerische Optimierung der Duhamel-Neumann Gleichung

Wie bereits wahrend der Herleitung der Duhamel-Neumann Gleichung (2.10) beschrieben wird, ist diese Formulierung stark explizit. Fur Losungsalgorithmen, die lineare Gleichungssysteme im Bereich der FVM berechnen, verbessert sich die Konvergenzrate, wenn die impliziten Terme uberwiegen. Daher wird zur Verbesserung der numerischen Eigenschaft, die eine Stabilisierung und Optimierung der Konvergenzrate des iterativen Prozesses bewirkt, ein mathematischer Trick angewendet, um die Diagonaldominanz der zu losenden Matrix zu erhohen.

Die Idee der numerischen Stabilisierung beruht darauf, mogliche *implizite* Anteile aus den *expliziten* Termen zu extrahieren, wodurch sich die Diagonaldominanz der Matrix verbessert und sich dementsprechend die Dominanz der expliziten Terme verringert.

Jasak und Weller (1998) sowie Cardiff (2012) fugen zur Duhamel-Neumann Gleichung folgenden Term hinzu:

$$\nabla \bullet [(\lambda + \mu)(\nabla \otimes \mathbf{D}) - (\lambda + \mu)(\nabla \otimes \mathbf{D})] = 0 . \quad (2.23)$$

Aufgrund der Tatsache, dass das gleichzeitige addieren und subtrahieren eines Terms die Aquivalenz der Gleichung nicht verandert, ist diese Methodik konsistent. Jedoch wird dadurch erreicht, dass eine Neuformulierung der Grundgleichung erreicht wird, wodurch der implizite Anteil erhohet und simultan der explizite Anteil verringert wird. Die optimierte Grundgleichung zur Bestimmung der Verschiebungen lautet:

$$\begin{aligned} \frac{\partial \rho \partial \mathbf{D}}{\partial t^2} = & \underbrace{\nabla \bullet (2\mu + \lambda)(\nabla \otimes \mathbf{D})}_{\text{implizit}} + \underbrace{\nabla \bullet \mu(\nabla \otimes \mathbf{D})^T + \nabla \bullet \lambda(\nabla \bullet \mathbf{D})\mathbf{I}}_{\text{explizit}} \\ & - \underbrace{\nabla \bullet (\lambda + \mu)(\nabla \otimes \mathbf{D}) - \nabla \bullet [(3\lambda + 2\mu)\alpha(T - T_0)\mathbf{I}]}_{\text{explizit}} . \quad (2.24) \end{aligned}$$

Es ist deutlich ersichtlich, dass der implizite Anteil der oben stehenden Formulierung bezogen auf Gleichung (2.10) deutlich angestiegen ist und zugleich der explizite Anteil durch den vierten Term auf der rechten Seite verringert wird.

Weitere Autoren, die die vorgestellte numerische Optimierung in ihren Forschungstatigkeiten einsetzen, sind beispielsweise Tukovic und Jasak (2007) als auch Cardiff et al. (2016).

## 2.7. Randbedingungen

Für die numerische Betrachtung eines diskretisierten Kontinuums müssen an den Grenzflächen Randbedingungen für das Lösungssystem angegeben werden. Im Allgemeinen wird zwischen drei verschiedenen Randbedingungen unterschieden. Wird ein fester Wert auf der Grenzfläche angegeben, ist von einer *Dirichlet*-Randbedingung die Rede. Wird hingegen die erste Ableitung, also der Gradient angegeben, ist dies unter der *Neumann*-Bedingung bekannt. Die Klassifizierung der dritten Art wird unter dem Begriff *Cauchy* oder *Robinson* Randbedingung geführt und stellt eine Mischung — Linearkombination — der beiden zuvor genannten dar; vergleiche Gustafson und Takehisa (1998).

Nachfolgend werden die essenziellen Randbedingungen, welche für die Berechnung der Verschiebungen verwendet werden, diskutiert. Hierzu zählt auch die in dieser Arbeit neu entworfene und programmierte Randbedingung namens `laserConvectionBC` für die Energieerhaltung, welche eine lokale Energiequelle modellieren kann, während freie Konvektion und Strahlung berücksichtigt werden. Die ausführliche Herleitung dieser Randbedingung ist in Anhang A.2 auf Seite 128 gegeben.

### 2.7.1. Randbedingungen für die Verschiebungsgleichung

Für die Berechnung des Verschiebungsvektors gibt es generell vier essenzielle Randbedingungen; vergleiche Jasak und Weller (1998) und Cardiff (2012). Diese sind:

- Symmetrieebenen
- Vordefinierte Verschiebungen
- Spannungsfreie Oberflächen
- Vordefinierte Kraft- oder Druckbelastungen

Zur Vereinfachung des numerischen Problems können im Allgemeinen die Symmetrieeigenschaften vom Design ausgenutzt werden. Dies suggeriert, dass nur ein Teil der Form numerisch abzubilden ist und dadurch die Anzahl an linearen Gleichungen verringert werden kann, wodurch zum einen eine Vereinfachung des Problems und zum anderen eine Verringerung der Rechen-dauer erreicht wird. Zur Modellierung werden hierzu *Symmetrie* Randbedingungen verwendet. Die Symmetrie Randbedingung stellt eine Kombination aus Dirichlet und Neumann Bedingung dar. Das bedeutet, dass die Normalkomponente des Vektors  $\mathbf{D}$  auf null gesetzt (Dirichlet) wird — jeglicher Fluss durch diese Fläche wird verboten — und die beiden anderen Komponenten mit einer Nullgradientenbedingung modelliert werden.

Als weitere Option können definierte Verschiebungen auf eine Fläche vorgeschrieben werden. Diese werden direkt als Zahlenwerte in Form einer Dirichlet Randbedingung vorgegeben. Die Lösung des Verschiebungsvektors auf diesen Flächen wird daher dem Lösungssystem explizit mitgeteilt.

Die letzten zwei Bedingungen, die entweder eine spannungsfreie oder unter Belastung stehende Oberfläche darstellen, werden mit einer Neumann Randbedingung behandelt. Ein wichtiger Punkt hierbei ist, dass die definierten Spannungen in Verschiebungen transformiert werden müssen. Dies wird nachfolgend hergeleitet.

Für spannungsfreie oder unter Last stehende Oberflächen gilt:

$$\mathbf{S}_S = \boldsymbol{\sigma} \bullet \mathbf{n} , \quad (2.25)$$

wobei  $\mathbf{S}_S$  Zug- oder Druckkräfte beschreibt. Es ist offensichtlich, dass Zugkräfte in beliebigen Richtungen angreifen können, jedoch eine Druckkraft stets normal zur Oberfläche wirkt. Mithilfe der Formulierung (2.7) kann Gleichung (2.25) für den Verschiebungsvektor  $\mathbf{D}$  transformiert werden. Hierzu wird Gleichung (2.3) und (2.6) in (2.25) eingesetzt, sowie die im Kapitel 2.6 auf Seite 19 beschriebene Stabilisierung hinzugezogen. Wird die gesamte Gleichung bezüglich dem Gradienten aufgelöst, folgt für die Randbedingung des Verschiebungsvektors  $\mathbf{D}$ :

$$\frac{\nabla_{\perp} \mathbf{D}}{\Delta} = \frac{(\mathbf{F} - p\mathbf{n}) - \mu(\nabla \otimes \mathbf{D})^T \bullet \mathbf{n} - \lambda \operatorname{tr}(\nabla \otimes \mathbf{D}) \bullet \mathbf{n} + (\mu + \lambda)(\nabla \otimes \mathbf{D}) \bullet \mathbf{n}}{2\mu + \lambda} + \frac{(3\lambda + 2\mu)\alpha(T - T_0)\mathbf{I} \bullet \mathbf{n}}{2\mu + \lambda} . \quad (2.26)$$

Die ausführliche Herleitung und Teile der in OpenFOAM<sup>®</sup> implementierten Version sind in Anhang A.5 auf 139 zu finden. Es soll angemerkt werden, dass  $\frac{\nabla_{\perp} \mathbf{D}}{\Delta}$  den Gradienten normal zur Fläche in OpenFOAM<sup>®</sup> darstellt. Der Term  $\mathbf{S}_S$  ist mit folgender Definition ersetzt worden:

$$\mathbf{S}_S = (\mathbf{F} - p\mathbf{n}) , \quad (2.27)$$

wobei  $\mathbf{F}$  einen Vektor darstellt (Zugkräfte in Newton) und  $p$  eine Druckkraft in Pascal.

Anzumerken ist, dass in allen OpenFOAM<sup>®</sup> implementierten Randbedingungen die nicht-Orthogonalität auf den Lösungsrandern unberücksichtigt bleibt. Der Grund und der Beweis dafür ist in der Dissertation von Jasak (1996) gegeben und wird durch die Steigerung der Konservativität begründet, da für die Korrektur der Flächennormalenvektor in zwei Teile — einen Orthogonalen und nicht-Orthogonalen — aufgeteilt wird und der nicht-orthogonale Term als expliziter Zusatzterm vorliegt. Im Bereich der *Strukturmechanik* mit der FVM ist diese Korrektur von großem Interesse, da sie die Konvergenzrate des linearen Löser verbessert; vergleiche Cardiff (2012). Aus diesem Grund werden alle oben genannten Randbedingungen modifiziert, um die nicht-Orthogonalität für die Verschiebungsgleichung zu berücksichtigen. Eine Beschreibung der Durchführung ist in Jasak (1996) gegeben. Informationen zur Programmierung kann aus dem Foam Extend (2016) Projekt entnommen werden.

### 2.7.2. Randbedingung für die Temperaturgleichung

Die Behandlung der numerischen Grenzflächen für die Erhaltungsgleichung der Temperatur kann wie zuvor in die Kategorie der Dirichlet, Neumann und Robinson Bedingungen eingeteilt werden.

Für die Modellierung des lokalen Energieeintrags wird eine spezielle Robinson Randbedingung entwickelt, die sowohl den Energieeintrag mittels einem LASER als auch freie Konvektion und Wärmestrahlung abbilden kann. Die Quellcodedokumentation von OpenFOAM<sup>®</sup>, die von der Foundation (2018) zur Verfügung gestellt wird oder lokal einsehbar ist, gibt Aufschluss über die herzuleitende Gleichungsstruktur. Alle Randbedingungen können diesbezüglich durch eine

Linearkombination aus Dirichlet und Neumann abgebildet werden. Es folgt:

$$\phi_f = \zeta \phi_f^0 + (1 - \zeta) \left[ \phi_c + \frac{\nabla_{\perp} \phi}{\Delta} \right]. \quad (2.28)$$

Gleichung (2.28) beschreibt die Berechnung der Randwerte  $\phi_f$ , wobei  $\phi$  eine allgemeine Größe darstellt. Der Subskript steht dabei für die örtliche Position und bezeichnet das betrachtete *face*. Die Größe  $\zeta$  stellt einen linearen Übergangsfaktor dar und bestimmt den Charakter beziehungsweise die Gewichtung der Randbedingungen bezüglich Dirichlet und Neumann.

Für  $\zeta = 1$  entfällt der zweite Term auf der rechten Seite. Das resultierende Ergebnis ist eine Dirichlet Bedingung, die einen Festwert  $\phi_f^0$  auf jeder Randfläche  $\phi_f$  vorgibt. Die hochgestellte Null verweist auf eine vom Anwender explizite Vorgabe.

Im Fall  $\zeta = 0$  verschwindet der erste Term auf der rechten Seite in Gleichung (2.28). Der verbleibende Ausdruck repräsentiert die Gradientenbedingung. Der Wert für  $\phi_f$  kann dementsprechend mithilfe des Wertes im Volumenzentrum  $\phi_c$  und dem zusätzlichen Beitrag des Normalengradienten berechnet werden. Ein Beispiel hierzu ist eine Wärmezugabe oder -abgabe die über einen Gradienten beschrieben wird oder die Beschreibung einer adiabaten Wand (Nullgradienten Bedingung), wodurch der Wert der Fläche  $\phi_f$  mit dem des Zellwert  $\phi_c$  übereinstimmt, da  $\frac{\nabla_{\perp} \phi}{\Delta} = 0$ .

Bewegt sich der Linearfaktor  $\zeta$  zwischen  $0 < \zeta < 1$ , liegt eine Linearkombination aus Dirichlet und Neumann Bedingung vor und repräsentiert die Robinson Bedingung. Damit können komplexe Formulierungen für die Randbedingung hergeleitet und modelliert werden.

Zur Modellierung einer lokalen Wärmebehandlung mittels einem LASER und einer gleichzeitigen Wärmeabgabe an die umgebende Luft durch Konvektion und Strahlung, müssen die Wärmeströme aller existierenden Phänomene ins Gleichgewicht gebracht werden. Die aufgestellte Grundgleichung muss danach entsprechend auf die Struktur von Gleichung (2.28) gebracht werden. Eine ausführliche Herleitung mit der Beschreibung aller Terme ist in Anhang A.2 auf Seite 128 gegeben und gibt weiteren Aufschluss über die oben genannten Ausführungen.

Durch die Herleitung identifizierten Terme lauten:

$$\zeta = \frac{1}{1 + \frac{\lambda_{th}}{\delta + \alpha_{eff}}}, \quad (2.29)$$

$$\phi_f^0 = T_{\infty}, \quad (2.30)$$

$$\frac{\nabla_{\perp} \phi}{\Delta} = \frac{\psi \delta Q \text{ PDF}(x, y, \sigma_x, \sigma_y, \rho_L)}{\lambda_{th}}, \quad (2.31)$$

und werden für die Beschreibung der Robinson Randbedingung verwendet.

Zusätzlich wird die Möglichkeit gegeben, eine zeitliche Manipulation der Werte  $x$  und  $y$ , welche in der PDF() Funktion zu finden sind, vorzunehmen, um eine örtlich bewegbare Energiequelle zu modellieren. Es ist erwähnenswert, dass die LASER Bewegung die Zeitschrittweite des Löser beeinflusst. Dies ist in der Natur der Implementierung begründet und basiert auf der Tatsache, dass sich die Energiequelle zeitlich an diskreten Positionen in der numerischen Domäne befindet und dort für den entsprechenden Zeitschritt integriert wird. Wird beispielsweise eine Energiequelle in  $x$ -Richtung bewegt und die numerische Untersuchung mit drei verschiedene Zeitschritten  $\Delta t_1, \Delta t_2$  und  $\Delta t_3$  durchgeführt, wobei  $\Delta t_1 < \Delta t_2 < \Delta t_3$ , *fühlt* das Material verschiedene



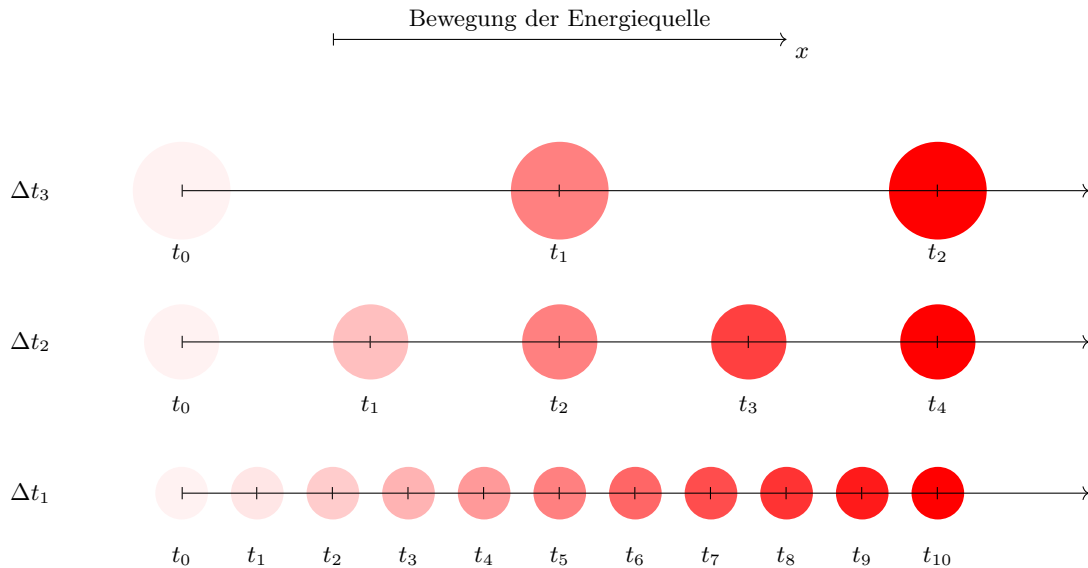


Abbildung 2.2.: Einfluss der Zeitschrittweite auf die Bewegung einer Energiequelle. Die Farbin-  
tensitäten spiegeln den Zeitpunkt der Energieeintragung wieder; Je kleiner der  
Zeitschritt gewählt wird, desto genauer ist die Auflösung der Bewegung und der  
entsprechende Energieeintrag.

Energieintensitäten an unterschiedlichen Positionen. Abbildung 2.2 gibt diesbezüglich weiteren  
Aufschluss. Folglich muss der Zeitschritt so gewählt werden, dass eine korrekte Abbildung des  
Energieeintrags erfolgt.

Alle weiteren Details zur Herleitung und Umsetzung der Bewegungsabläufe für die Energie-  
quelle befindet sich in Anhang A.4 auf Seite 137. Es ist weiter erwähnenswert, dass beliebig viele  
Energiequellen mit unterschiedlichen Einstellungen simultan vorgegeben werden können.

## 2.8. Integralform der Gleichungen für die FVM

Für das Lösen der Gleichung (2.16) und (2.24) mithilfe der FVM müssen beide Formulierungen  
als Integrationsgleichungen vorliegen. Jeder einzelne Term wird mit definierten Schemen diskre-  
tisiert. Ein breites Spektrum über die Diskretisierung einzelner Terme und deren Besonderheiten  
wird von Jasak (1996) als auch von Ferziger und Perić (2008) erläutert. Zur Vervollständigung  
werden beide Gleichungen in Integralform gegeben.

Durch Anwendung des Gaußschen Integralsatzes können alle Volumenintegrale, in denen der  
Divergenz-Operator enthalten ist, in Oberflächenintegrale überführt werden; vergleiche Ferziger  
und Perić (2008). Dadurch reduziert sich der Grad des Tensors um Eins (Volumenbetrachtung  
wird zur Oberflächenbetrachtung). Analog dazu kommt bei der Herleitung der FEM dieser In-  
tegralsatz in der Form des *Green's Theorem* zum Einsatz.

Zudem wird der Kopplungsterm in der Erhaltungsgleichung der Temperatur vernachlässigt,  
da die halb-gekoppelte quasi-stationäre Betrachtung verwendet wird. Daraus ergibt sich die  
integrale Form der Temperaturgleichung zu:

$$\int_V c_p \rho \frac{\partial T}{\partial t} dV = \oint_S \lambda \nabla T \cdot \mathbf{n} dS, \quad (2.32)$$

und die korrespondierende Gleichung für die Verschiebung zu:

$$\int_V \frac{\partial \rho \partial \mathbf{D}}{\partial t^2} dV = \oint_S (2\mu + \lambda)(\nabla \otimes \mathbf{D}) \bullet \mathbf{n} dS + \oint_S \mu(\nabla \otimes \mathbf{D})^T \bullet \mathbf{n} dS + \oint_S \lambda \operatorname{tr}(\mathbf{D}) \mathbf{I} \bullet \mathbf{n} dS - \oint_S (\lambda + \mu)(\nabla \otimes \mathbf{D}) \bullet \mathbf{n} dS - \oint_S (3\lambda + 2\mu)\alpha(T - T_0) \mathbf{I} \bullet \mathbf{n} dS . \quad (2.33)$$

Es ist erwähnenswert, dass die Diskretisierung der zweiten Zeitableitung der Verschiebungsgleichung in OpenFOAM<sup>®</sup> nur mit dem Euler Verfahren durchgeführt werden kann. Demnach ist die Genauigkeit dieses Terms stets erster Ordnung. Aufgrund der Tatsache, dass die lokale Wärmebehandlung als quasi-stationäres System betrachtet wird, kann und hierbei keine Zeitableitung erforderlich ist, wird hierauf nicht weiter eingegangen. Allerdings ist erwähnenswert, dass die Zeitableitung in den Validierungsfällen — da transiente Analysen — beachtet werden muss.

Nachdem die einzelnen Terme diskretisiert wurden, wird ein System linearer Gleichungen erhalten, das wie folgt, repräsentiert wird:

$$\mathbf{A} \mathbf{x} = \mathbf{b} . \quad (2.34)$$

Dabei steht  $\mathbf{A}$  für die Koeffizientenmatrix,  $\mathbf{x}$  für den Lösungsvektor und  $\mathbf{b}$  für den Vektor, welcher die Quellterme enthält (explizite Anteile). Weitere Details zum linearen Gleichungssystem  $\mathbf{A} \mathbf{x} = \mathbf{b}$  ist in Anhang B auf Seite 155 gegeben.

## 2.9. Validierung und Ergebnisse

Für die Sicherstellung der korrekten Berechnung beider Gleichungen werden zwei Problemstellungen vorgestellt, für die jeweils eine analytische Lösung zugrunde liegt. Der Vergleich wird hauptsächlich quantitativ durchgeführt. Im Anschluss an die beiden Validierungsfälle wird ein weiterer — imaginärer — Fall betrachtet, bei dem ein Vergleich zwischen FEM und FVM erfolgt.

Für die numerische Betrachtung werden Materialeigenschaften der Aluminiumlegierung Al-Si7Mg0.3 benötigt. Da im weiteren Verlauf der Arbeit gezielt thermo-elastische Problemstellungen diskutiert werden, sind die temperaturabhängigen Eigenschaften der Legierung mithilfe von Polynomansätze beschrieben. Die benötigten Materialwerte sind die Dichte  $\rho$  in  $\text{kgm}^{-3}$ , die Wärmekapazität  $c_p$  in  $\text{Jkg}^{-1}\text{K}^{-1}$  und die Wärmeleitfähigkeit  $\lambda_{\text{th}}$  in  $\text{Wm}^{-1}\text{K}^{-1}$ . Das allgemeingültige Polynom, welches verwendet wird, ist gegeben durch:

$$\phi(T) = ET^4 + DT^3 + CT^2 + BT + A . \quad (2.35)$$

Die Variable  $\phi(T)$  stellt eine der drei oben genannten Größen dar. Ferner repräsentieren die Vorfaktoren A bis E die Polynomkoeffizienten, welche in Tabelle 2.1 für alle Größen angegeben sind. Die Einheiten der Vorfaktoren A bis E entsprechend stets denen der gesuchten Größe. Es ist ersichtlich, dass die Faktoren B bis E mit der Einheit *pro Kelvin* respektive den Exponenten der Temperatur erweitert werden müssen. Die Bestimmung der Faktoren für die Aluminiumlegierung erfolgt wie bei der Berechnung des temperaturabhängigen Wärmeübergangskoeffizienten, siehe Anhang A.3, mithilfe der Methode der kleinsten Fehlerquadrate. Die Rohdaten, welche aus Flender (2015) entnommen sind und für das Fitting-Prozedere verwendet werden, sind in Anhang

G auf Seite 221 gelistet.

	A	B	C	D	E
Dichte $\rho$	2725,02000	-0,20010	$-2,96265e^{-5}$	–	–
Wärmekapazität $c_p$	916,17500	-0,95102	$4,69057e^{-3}$	$-7,85369e^{-6}$	$5,17389e^{-9}$
Wärmeleitfähigkeit $\lambda_{th}$	143,67200	0,05862	$-1,49814e^{-4}$	$1,42484e^{-7}$	–

Tabelle 2.1.: Polynomkoeffizienten zur Modellierung der temperaturabhängigen Materialeigenschaften für die verwendete Aluminiumlegierung (AlSi7Mg0.3). Koeffizienten mit dem Eintrag – sind mit null beziffert.

### Kurzbeschreibung Fall 1

Der erste Validierungsfall betrachtet ein dreidimensionales Spannungsproblem ohne thermische Einflüsse. Analysiert wird ein unendlich großer Würfel, bei dem im Schwerpunkt eine Kugel herausgeschnitten ist. Auf zwei sich gegenüberliegenden Seiten wirken Zugkräfte, die Verformungen und damit Spannungen im Würfel erzeugen. Die Modellerstellung wird durch die Ausnutzung der Symmetrieeigenschaften auf ein Achtel reduziert und ist endlich begrenzt. Es werden zusätzlich vier verschiedene Volumina Typen untersucht.

### Kurzbeschreibung Fall 2

Die zweite Problemstellung betrachtet thermische Spannungen die in einer Rohrleitung auftreten (Hohlrohr). Die Innen- und Außenseite des Rohres sind zu Beginn spannungsfrei (homogene Temperatur der Rohrwand) jedoch mit unterschiedlichen Temperaturen beaufschlagt. Aufgrund der thermischen Leitfähigkeit erfolgt ein Energieaustausch und damit eine Erhöhung der Temperatur der Rohrwand. Das Resultat sind thermische Ausdehnungen, die thermische Spannungen induzieren.

### Kurzbeschreibung Fall 3

Im dritten Analysefall werden die zwei Methoden der FEM und FVM gegenübergestellt und deren Genauigkeit für transiente und stationäre Berechnungen verglichen. Diskutiert wird ein einfacher Biegebalken, der sprunghaft mit einer Last beaufschlagt wird. Aufgrund des fehlenden Dissipationsterms stellt sich eine periodisch nicht-endende Schwingung ein.

#### 2.9.1. Erster Validierungsfall - Grenzfall Würfel

Wie bereits in der Kurzbeschreibung ersichtlich ist, betrachtet das nachfolgende Beispiel den Grenzfall (Limes) eines unendlich großen Würfels, bei dem im Massenschwerpunkt eine Kugel herausgeschnitten ist. Des Weiteren werden zwei sich gegenüberliegende Seiten mit einer Zugspannung beaufschlagt, wohingegen alle anderen Oberflächen als spannungsfrei betrachtet werden. Timoshenko und Goodier (1951) als auch Boley und Weiner (1985) geben in ihren Werken die Herleitung der analytischen Lösung für den Grenzfall wieder, die es ermöglicht die Spannungskomponente  $\sigma_{ii}$  im Block, die radial zum Zentrum der Kugel  $M$  und senkrecht zur Ebene der Zugspannungen steht, zu bestimmen; Abbildung 2.3 verdeutlicht die im letzten Satz wiedergegebenen Aussagen. Im vorliegenden Beispiel, wirkt die Zugspannung normal zur  $z - y$  Ebene. Die Berechnung der analytischen Lösung für  $\sigma_{zz}$  und  $\sigma_{yy}$  ist damit gegeben. Es soll

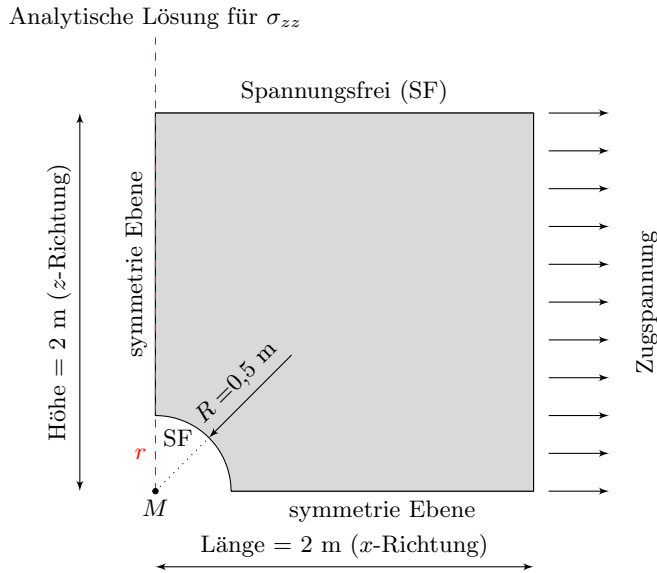


Abbildung 2.3.: 2D Darstellung des Validierungssetups. Die Breite in  $y$ -Richtung beträgt 2 m. Die Zugbeanspruchung entspricht 10000 Pa.

angemerkt werden, dass beide Größen aufgrund der Symmetrieeigenschaften identisch sind. Die analytische Grundgleichung, bei der alle Kräfte im Gleichgewicht sind, lautet:

$$\sigma_{zz} = F \left[ 1 + \frac{(4 - 5\nu)R^3}{2(7 - 5\nu)r^3} + \frac{9R^5}{2(7 - 5\nu)r^5} \right]. \quad (2.36)$$

Die Größe  $F$  stellt die Zugspannung,  $R$  den Kugelradius,  $r$  die radiale Entfernung vom Kugelmittelpunkt  $M$  und  $\nu$  die Querkontraktionszahl des Materials dar.

Für die numerische Betrachtung wird angenommen, dass das Material aus der AlSi7Mg0.3 Legierung besteht die eine Querkontraktionszahl von  $\nu = 0,333$ , eine Dichte von  $\rho = 2675 \text{ kgm}^{-3}$  und ein Elastizitätsmodul von  $E = 68 \text{ GPa}$  aufweist. Die Dichte  $\rho$  stellt bei der Betrachtung von stationären Problemen keine Einflussgröße dar, da der Trägheitsterm (Zeitableitung) bei stationären Problemen in der Duhamel-Neumann Formulierung gleich null ist.

### Die geometrischen Maße

Für die numerische Modellierung ist es nicht möglich, einen unendlich großen Würfel zu erstellen und diskretisieren. Daher wird die Berechnungsdomäne auf eine Blockgröße von 2 m beschränkt. Dieses Maß resultiert aus einer einfachen Analyse, die durch Verwendung von Gleichung (2.36) beschrieben werden kann. Werden die Spannungen in der Ebene die durch den Kugelmittelpunkt  $M$  und senkrecht zur Wirklinie der Zugspannungen analysiert — rote Strichlinie in Abbildung 2.3 —, kann folgendes festgehalten werden:

- Wird die Spannung in der Ebene für den Grenzfall  $r \rightarrow \infty$  betrachtet, muss die Spannung an diesen Positionen gegen den Wert von  $F$  streben. Da die Symmetriebedingung keine Verschiebung in  $x$ -Richtung zulässt, ist dies vergleichbar mit einem Zugstab, der auf der Unterseite festgehalten und auf der Oberseite belastet wird. Die Belastung an der Unterseite des Zugstabes entspricht genau derjenigen, die vorgegeben wird mit der Annahme, dass keine Gestaltänderungen (Verformung) des Stabes auftreten, wie dies beispielsweise beim Zugversuch der Fall ist.

Die Betrachtung von Gleichung (2.36) bestätigt dies, da unter der Annahme von  $r \rightarrow \infty$ , der zweite und dritte Term in der Klammer verschwindet und die Spannung normal zur Ebene gleich der Zugspannung entspricht:  $\sigma_{zz} = F$ .

- Wird hingegen der zweite Grenzfall betrachtet, wobei  $r \rightarrow 0$  strebt, muss aufgrund des Kugelausschnitts die Spannung ansteigen. Begründet ist dies durch die Änderung der Gestalt des Körpers (Ausschnitt der Kugel). Daraus resultiert eine Akkumulation der Verformungen und daraus die maximalen Spannungswerte die bei  $r = R$  vorzufinden sind.

Die Steigerung der Spannung kann erneut mithilfe der Gleichung (2.36) beschrieben werden. Für den Grenzfall  $r \rightarrow 0$  liegt eine Singularität vor. Diese wird allerdings nicht erreicht, da der Minimalwert von  $r = R$  beträgt. Es ist ersichtlich, dass für  $r \rightarrow R$  der zweite und dritte Term an Bedeutung gewinnen und einen Beitrag zur Zugspannung  $F$  liefern. Es ist erwähnenswert, dass beim Grenzfall  $r = R$  sich beide Größen aus den Termen herauskürzen und diesbezüglich der Maximalwert einzig und allein von der Querkontraktionszahl  $\nu$  abhängig ist; die Dichte des Materials spielt hier keine Rolle.

Aus den oben genannten Überlegungen kann schließlich geschlussfolgert werden, dass der Spannungsverlauf in der Nähe von  $r \rightarrow R$  einen rapiden Anstieg erfährt. Dies ist begründet durch die Potenzierung des Radialabstands in Gleichung (2.36). Dementsprechend ist ein extrem schneller Abfall der Spannungen in die entgegengesetzte Richtung zu erwarten; für  $r = 1$  m folgt  $\sigma_{zz} = 1,054F$  und für  $r = 2$  m folgt  $\sigma_{zz} = 1,004F$ . Daher folgt, dass bereits nach 2 m nur noch eine Restabweichung von weniger als fünf Promille zu erwarten ist. Dies ist der Beweis, dass eine Maximalgröße von 2 m in guter Korrelation zum unendlichen Würfel steht. Allerdings kann davon ausgegangen werden, dass mit der Restriktion bezüglich der Randbedingungen und den dadurch enthaltenen Abbildungsfehler eine Abweichung zur analytischen Lösung auftreten kann.

## Netztypen und Netzerstellung

Zur besseren Analyse der Robustheit und Konvergenzrate des FVM Löser werden vier verschiedene Netztypen untersucht, die häufig in der FVM Anwendung finden. Die verwendeten Netztypen sind *hexaedrisch*, *tetraedrisch*, *polyedrisch* und *hexadominant* Natur.

Das Erstellen der Netze wird mit unterschiedlichen Programmen durchgeführt. Hierbei wird für das Hexaeder- und Tetraedernetz die freien CAD Software Salome<sup>®</sup> verwendet. Das Polygonnetz wird unter Verwendung des Tetraedernetzes und der *Voronoi Dual* Methode durch das OpenFOAM<sup>®</sup> interne Programm `polyDualMesh` erzeugt. Eine Diskussion dieses Verfahrens ist im Buch von Berger et al. (2008) zu finden und im Anhang D.3 ausführlich an einem Beispiel beschrieben. Das hexadominante Netz, das eine Vielzahl unterschiedlicher Volumina Typen enthalten kann, wird mit dem Programm `snappyHexMesh` erstellt und ist im offiziellen Quellcode von OpenFOAM<sup>®</sup> zu finden.

Die Netze werden so erstellt, dass die einzelnen Volumengrößen der unterschiedlichen Netztypen möglichst identisch sind. Dies kann jedoch aufgrund der einzelnen Erstellungsstrategien nicht eingehalten werden. Daher wird explizit darauf geachtet, dass vor allem um den Kugelausschnitt eine ähnliche Anzahl an numerischen Zellen vorliegt. Hieraus kann abgeleitet werden, dass die Gesamtanzahl der Volumina stark variieren kann.

Weitere Details zur Netzerstellung, den Volumentypen und die Erstellungsmethodiken sind in Anhang D detailliert zusammengefasst.

Das Hexaedernetz ist aufgrund des Erzeugungsalgorithmus das am schwierigsten zu erstellende Netz. Bezüglich den Bedingungen, die in Anhang D.3.1 erwähnt werden, vergrößert oder verringert sich die Elementdichte in bestimmten Regionen.

Das Polygonnetz wird aus dem Tetraedernetz erstellt. Durch die Methodik, die in Anhang D.3.3 erklärt ist, ist offensichtlich, weshalb sich die Volumenanzahl um einen Faktor fünf reduziert. Der Vergleich der Elementdichte zwischen Tetraeder und hexadominantem Netz ist durch den Umstand erklärbar, dass ein Hexaederelement einen größeren Raum abdeckt, wenn gilt, dass alle Kantenlängen eines Volumens identisch sind. Alle Details hierzu sind in Tabelle 2.2 gegeben.

Die erstellten Netze für alle Typen sind in Anhang A.6.1 auf Seite 141 qualitativ dargestellt.

## Numerische Diskretisierung

Die numerische Diskretisierung der einzelnen Terme kann auf dem beigefügten Datenträger unter `Dokatorarbeit/thermalElasticStressFoam/` eingesehen werden. Es werden stets Diskretisierungsschemen 2. Ordnung angewendet. Lediglich die zweite Zeitableitung der Verschiebungsgleichung wird mit dem Euler Schema 1. Ordnung diskretisiert.

## Netzanalyse

Jeder Netztyp wird mit drei verschiedenen Volumenzahlen diskretisiert und die numerische Analyse gestartet, worauf sich die Zusammenfassung in Tabelle 2.2 und die nachfolgende Diskussion bezieht.

Die Konvergenzrate der vier Netztypen lässt zwei Schlussfolgerungen zu. Zum einen weisen die Tetraeder- und Hexaedernetze das Merkmal auf, dass für alle drei Diskretisierungsgrade annähernd die gleiche Anzahl an Iterationen zum Erreichen des Konvergenzkriteriums benötigt werden. Zum anderen verbessert sich die Konvergenzrate beim Polygon und hexadominanten Netz mit zunehmender Anzahl der numerischen Zellen. Letzteres kann dadurch begründet werden, dass beim Polygon- und hexadominanten Netz — numerisch betrachtet — qualitativ hochwertigere Volumina vorliegen, welche einen kleineren numerischen Fehler einführen. Des Weiteren stehen mehr Nachbarzellen für die Berechnung des Gradienten im Zellmittelpunkt zur Verfügung als bei den Tetraedernetzen — hier stehen allgemein nur vier Nachbarzellen zur Verfügung. Das reine Hexaedernetz beinhaltet stark verzerrte Zellen (Aspekt-Ratio und nicht-Orthogonalität), die sich negativ auf den linearen Löser auswirken; vergleiche Anhang A.6.1.

Werden die realen Berechnungszeiten verglichen, fällt eindeutig auf, dass mit steigender Anzahl der zu evaluierenden Volumina der Berechnungsaufwand größer wird. Dies ist dadurch begründet, dass die Matrix des zu lösenden Systems nicht linear zunimmt, sondern zur zweiten Potenz; die Matrix  $\mathbf{A}$  hat  $i \times i$  Einträge  $\rightarrow i^2$ . Eine Verdopplung der numerischen Zellen hat daher eine Vervierfachung der Matrixgröße zur Folge, wodurch die Berechnungsdauer ansteigt. Die Berechnungszeit ist jedoch nicht proportional zu  $i^2$  und ist vom linearen Löser abhängig sowie dem Speichermanagement.

Beim direkten Vergleich der realen Berechnungszeit der vier Netze ist ersichtlich, dass das Hexaedernetz mit Abstand das beste Netz darstellt, da die Berechnung am schnellsten durchgeführt wird. Das Polygonnetz ist im Vergleich zum hexadominantem Netz ähnlich. Allerdings ist bei der genaueren Analyse ersichtlich, dass das hexadominante Netz wesentlich besser abschneidet, wenn

	Hexaedernetz	Tetraedernetz	Polygonnetz	Hexadominantes Netz
	Anzahl Volumina			
Grobe Vernetzung	1575	30843	6621	9183
Mittlere Vernetzung	7872	231789	44563	68336
Feine Vernetzung	63744	1852886	334699	526995
	Iterationen bis zur Konvergenz			
Grobe Vernetzung	221	151	116	141
Mittlere Vernetzung	248	150	101	118
Feine Vernetzung	232	154	87	95
	Reale Berechnungszeit			
Grobe Vernetzung	2 s	20 s	6 s	5 s
Mittlere Vernetzung	9 s	191 s	48 s	44 s
Feine Vernetzung	86 s	2335 s	363 s	328 s
	Berechnungszeit pro Volumenelement			
Grobe Vernetzung	1,26 ms	0,65 ms	0,91 ms	0,54 ms
Mittlere Vernetzung	1,14 ms	0,82 ms	1,07 ms	0,64 ms
Feine Vernetzung	1,35 ms	1,26 ms	1,09 ms	0,62 ms

Tabelle 2.2.: Anzahl der Volumina für jeden Netztyp bei drei unterschiedlichen Verfeinerungsgraden; Lokale Länge circa: Grob: 0,1 m, Mittel: 0,05 m, Fein: 0,025 m.

die Anzahl der Volumina in die Betrachtung miteinbezogen werden; vergleiche Berechnungszeit pro Volumenelement. Die Berechnungszeit der Tetraedernetze ist im Vergleich zu allen anderen überdurchschnittlich hoch. Da jedoch die Anzahl an Volumina stark variiert, ist dieser Vergleich unzureichend und gezielte Aussagen können nicht getroffen werden.

Werden hingegen die Berechnungszeiten pro Volumenelement betrachtet, können gezielte Angaben über die unterschiedlichen Netztypen durchgeführt werden. Es ist hierbei deutlich ersichtlich, dass die hexadominanten Netze am besten abschneiden und ist dem strukturieren Netz geschuldet (überwiegend numerisch optimale Zellen). Ferner ist ersichtlich, dass die Berechnungszeit ansteigt, wenn die Anzahl der Volumenelemente erhöht wird und kann auf die größeren Matrixsysteme zurückgeführt werden; beispielsweise dauert das Auffinden der Nachbarzellen länger.

Es soll angemerkt werden, dass die oben genannten Aussagen im vorliegenden Fall nicht eindeutig bestimmbar sind, da die Netzstudie nicht groß genug und der Validierungsfall sehr banal ist. Des Weiteren wird in der Realität stets auf Großrechnern/Clustern gerechnet, bei denen oftmals mehrere Aufgaben gleichzeitig abgearbeitet werden, sodass eine Verringerung der Leistung und eine Erhöhung der Rechenzeit die Folge ist. Die in Tabelle 2.2 aufgeführten Werte sind daher nur Richtwerte und werden auf jeder anderen Rechenmaschine anders ausfallen und hängen zudem mit den im Hintergrund laufenden Prozessen ab.

### Vergleich zwischen Analytik und Numerik

Zur Validierung des erstellten Löser wird die in Gleichung (2.36) gegebene Formulierung verwendet und mit den numerischen Daten quantitativ gegenübergestellt. Der Vergleich wird für alle vier Netztypen und jedem Diskretisierungsgrad durchgeführt und ist in Abbildung 2.4 gegeben.

Der erste Eindruck vermittelt akkurate quantitative Übereinstimmungen zwischen Analytik und Numerik für alle Netztypen und Diskretisierungsgrade. Die Unterschreitung des Wertes von 10.000 Pa, dass bezüglich Gleichung (2.36) nicht möglich ist, rührt daher, dass der funktionale

Zusammenhang für den Grenzwert des unendlich großen Körpers gilt. Die Berechnung erfolgt jedoch auf einem endlichen — nach 2 m endenden — Körper. Dadurch ergeben sich Einflüsse der spannungsfreien Randzonen, die das Ergebnis verfälschen.

Das Hexaeder und hexadominante Netz schneiden am besten ab. Gerade im Bereich der Randzone des Kugelausschnitts ( $r \approx 0,5$  m) werden sehr gute Übereinstimmungen erhalten. Das Tetraeder- und Polyedernetz ergeben hier Unstimmigkeiten, die beim Polyedertyp für alle drei Netzaufösungen eine zu hohe Spannung vorhersagen. Beim Tetraedernetz gibt es eine quantitative Unterbewertung der Spannungsberechnung für das *grobe* Netz.

Für alle Netztypen wird eine sehr gute Konvergenzrate erreicht. Die Stabilität des Algorithmus ist im vorliegenden Fall uneingeschränkt gültig, da keine Unterrelaxation benötigt wird. Eine qualitative Analyse ist in Anhang A.6.1 geben. Dieser wird hier nicht präsentiert, da keine signifikanten Unterschiede erkennbar sind.

Der hier dargestellte Validierungsfall beweist die korrekte Implementierung der Grundgleichung für den Verschiebungsvektor  $\mathbf{D}$ . Es wurde aufgezeigt, dass die FVM unter Verwendung verschiedener Netztypen das vorliegende Problem sehr akkurat lösen kann, der Lösungsalgorithmus stabil ist — keine Relaxation nötig — und eine gute Konvergenzrate besitzt.

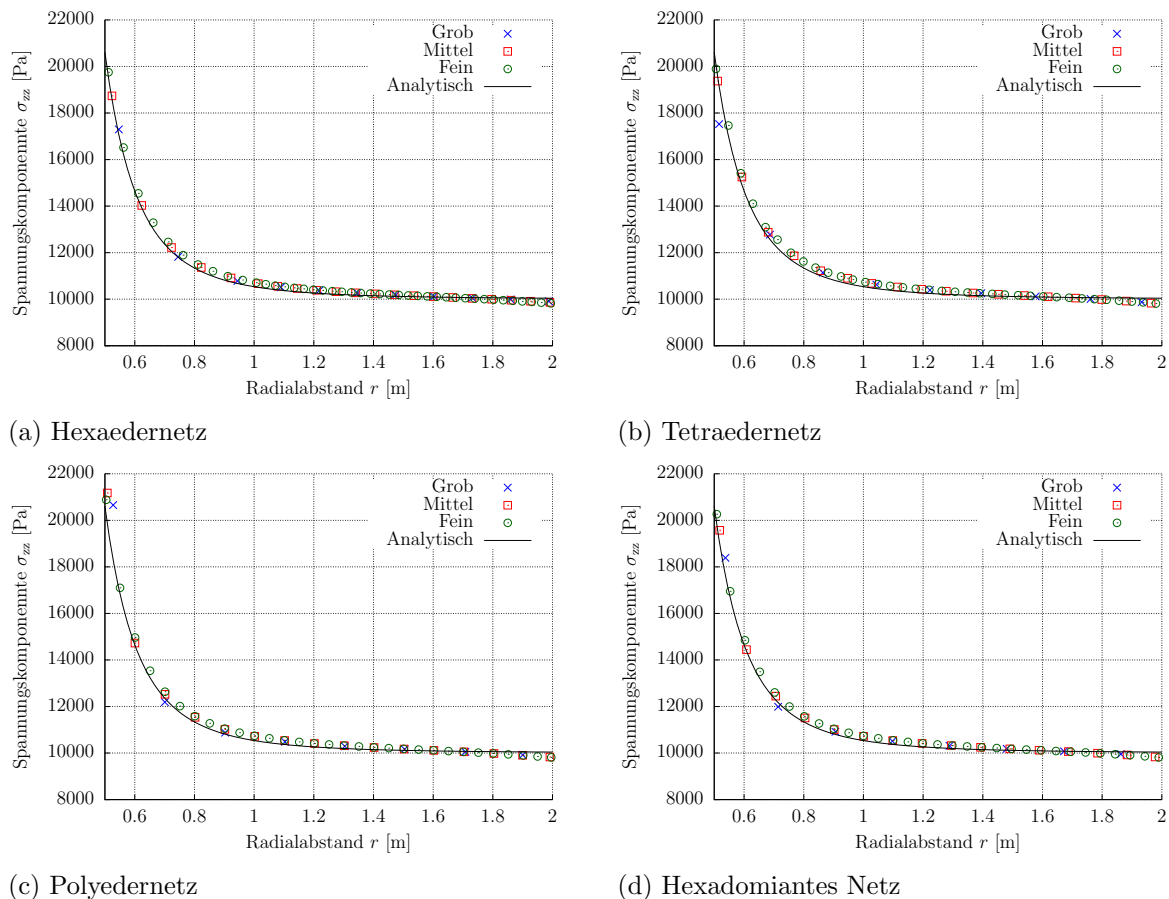


Abbildung 2.4.: Vergleich zwischen analytischer und numerischer Lösung für die vier verschiedenen Netztypen mit zusätzlicher Netzabhängigkeitsstudie.



### 2.9.2. Zweiter Validierungsfall - Hohlrohr

Zur Überprüfung der korrekten Implementierung der Energieerhaltungsgleichung und die durch Temperaturunterschiede hervorgerufenen Verschiebungen respektive Thermospannungen, wird ein Hohlrohr betrachtet, dass auf der Innen- und Außenseite festgelegte Temperaturwerte aufweist (100 °C und 0 °C). Zum Zeitpunkt  $t = 0$  s besitzt die Rohrwand eine einheitliche Temperatur von 0 °C. Dieser Zustand wird als spannungsfrei definiert. Mit voranschreiten der Zeit wird aufgrund des Fourier Gesetzes das Material erwärmt. Die resultierenden Volumenänderungen rufen thermische Spannungen hervor. Für das Hohlrohr können analytische Lösungen für beide Größen — Temperatur und Spannung — zur Berechnung der Profile genutzt werden. Diese dienen nachfolgend als quantitative Verifikation der numerisch erhaltenen Werte.

Die Grundgleichung für die örtlichen Temperaturwerte sind in Timoshenko und Goodier (1951); Boley und Weiner (1985) sowie in Gnielinski et al. (2005) gegeben und lautet:

$$T(r) = (T_i - T_a) \ln \left( \frac{r_a}{r_i} \right) \ln \left( \frac{r_a}{r} \right) . \quad (2.37)$$

Es ist direkt ersichtlich, dass keine materialspezifischen Daten einfließen. Entsprechend stellt sich für jedes Material die gleiche Temperaturverteilung im stationären Zustand ein; allerdings verändert sich die durch das Material transportierte Energiemenge. Diesbezüglich ist das Temperaturprofil durch die Differenz der Temperaturen auf der Innen- ( $T_i$ ) und Außenseite ( $T_a$ ) und den geometrischen Maßen (Innenradius  $r_i$  und Außenradius  $r_a$ ) definiert. Die zu evaluierende Position ist durch den Radialabstand  $r$ , ausgehend vom Rohrmittelpunkt  $M$ , bestimmt.

Des Weiteren geben Timoshenko und Goodier (1951) als auch Boley und Weiner (1985) die Formulierungen für die Thermospannungen an, die auch im Foam Extend (2016) Projekt zu finden sind. Die Herleitungen der Gleichungen liegen in Polarkoordinaten vor, sodass zu Vergleichszwecken mit den numerisch erhaltenen Werten eine Koordinatentransformation der analytischen Werte notwendig ist. Die Bestimmung der Radialspannungen  $\sigma_R$  erfolgt durch nachfolgende Gleichung:

$$\sigma_R(r) = \frac{\alpha E (T_i - T_a)}{2(1 - \nu) \ln(r_a r_i^{-1})} \left[ -\ln \left( \frac{r_a}{r} \right) - \frac{r_i^2}{r_a^2 - r_i^2} \left( 1 - \frac{r_a^2}{r^2} \right) \ln \left( \frac{r_a}{r_i} \right) \right] . \quad (2.38)$$

Analog dazu werden die Spannungen tangential (in  $\theta$ -Richtung) wie folgt bestimmt:

$$\sigma_\theta(r) = \frac{\alpha E (T_i - T_a)}{2(1 - \nu) \ln(r_a r_i^{-1})} \left[ 1 - \ln \left( \frac{r_a}{r} \right) - \frac{r_i^2}{r_a^2 - r_i^2} \left( 1 + \frac{r_a^2}{r^2 - \ln r_a r_i^{-1}} \right) \right] . \quad (2.39)$$

Die Spannungen in  $z$ -Richtungen sind gegeben durch folgenden Ausdruck:

$$\sigma_z(r) = [0, 3\sigma_R(r) + \sigma_\theta(r)] - E\alpha T(r) . \quad (2.40)$$

Mittels der transponierten Transformationsmatrix  $\mathcal{T}(\theta)$  und einem temporären Spannungstensor  $\sigma_{\text{tmp}}$ , die gegeben sind durch:

$$\mathcal{T}(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}^T , \quad \sigma_{\text{tmp}}(r) = \begin{bmatrix} \sigma_R(r) & 0 & 0 \\ \sigma_\theta(r) & 0 & 0 \\ \sigma_z(r) & 0 & 0 \end{bmatrix} . \quad (2.41)$$

kann der analytische Spannungstensor  $\sigma_{\text{analytisch}}(\theta, r)$  in kartesischen Koordinaten durch das innere Produkt aus beiden Tensoren erzeugt werden:  $\sigma_{\text{analytisch}}(x, y, z) = \mathcal{T}(\theta) \bullet \sigma_{\text{tmp}}(r)$ .

Die Berechnung der analytischen Werte erfolgt durch ein C++ Programm. Für alle Volumina wird sowohl die analytische Energiegleichung gelöst, als auch die Berechnung der Spannung (in Polarkoordinaten) und der anschließenden Transformation durchgeführt.

### Geometrische Maße und Netzerstellung

Die numerische Modellierung erfolgt in 2D. Zudem wird zur Vereinfachung die Symmetrieeigenschaften des Hohlrohres ausgenutzt. Dadurch kann das abzubildende Modell auf ein Viertel reduziert werden. Die Skizze des Rohres mit allen Maßen, Rand- und Startbedingungen sind in Abbildung 2.5 gegeben.

Diskretisiert wird das Modell mit einem Hexaedernetz, da dieses zum einen für die Numerik am besten geeignet ist und es sich im 2D Fall des Hohlrohres anbietet. Es wird angemerkt, dass durch die Krümmung der Außenlinien des Hohlzylinders die Würfel-Volumina ein Nicht-Orthogonales Gitter bilden. Eine visuelle Betrachtung des Gitters ist in Anhang A.6.2 auf Seite 146 gegeben.

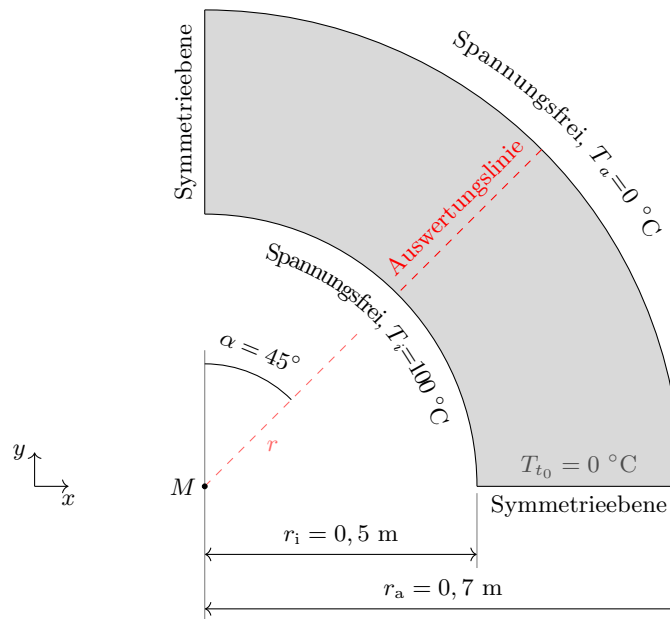


Abbildung 2.5.: Darstellung des Hohlrohres mit allen Maßen unter Berücksichtigung der Symmetrie Eigenschaften zur Validierung der thermischen Spannungsrechnung. Die Gegenüberstellung der analytischen und numerischen Ergebnisse wird an der Auswertungslinie durchgeführt.

### Vergleich zwischen Analytik und Numerik

Das quantitative Ergebnis zwischen der analytischen und numerischen Lösung, welche an der Auswertungslinie durchgeführt wird, ist in Abbildung 2.6 dargestellt. Eindeutig erkennbar sind die sehr guten quantitativen Übereinstimmungen zwischen Simulation und Analytik, sowohl für die Temperatur- als auch für die Spannungsberechnung. Die visuelle Darstellung der Größen ist in Anhang A.6.2 auf Seite 146 gegeben und wird hier nicht weiter verfolgt.

Es soll angemerkt werden, dass der Einfluss der Volumentypen und Netzfeinheit auf die quantitativen Ergebnisse, Rechenzeit und den Konvergenzverlauf nicht berücksichtigt wird, da dies bereits im vorherigen Validierungsfall diskutiert wurde.

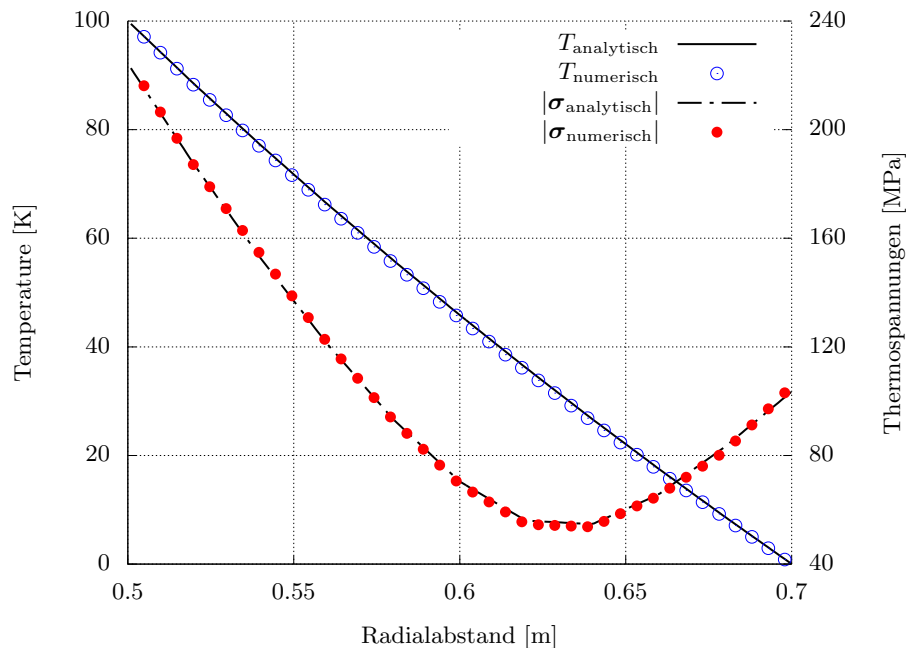


Abbildung 2.6.: Gegenüberstellung der analytischen und numerischen Ergebnisse für die Temperatur und Spannung an der Auswertungslinie; vergleiche Abbildung 2.5 für weitere Informationen.

### 2.9.3. Dritter Validierungsfall - FVM vs FEM

Die Gegenüberstellung der FVM und FEM wird an einem einfachen 2D Biegebalken diskutiert. Der Balken wird mit einer Drucklast von  $p = 10000 \text{ Pa}$  am Ende sprunghaft belastet; der Druck wirkt stets normal zur Oberfläche. Dies führt dazu, dass sich der Balken nach unten hin krümmt. Aufgrund der Belastungsart und der Tatsache dass keine Dissipationseffekte in der numerischen Berechnung berücksichtigt werden, muss bezüglich der Verformungsenergie und des linear-elastischen Verhaltens der Balken in die Ausgangsposition zurückfedert. Diese Bewegungsform ist periodisch und klingt in der numerischen Betrachtung nicht ab, da keine dissipativen Effekte berücksichtigt werden. Zur Veranschaulichung kann das Gedankenbeispiel eines Kunststofflineals herangezogen werden. Wird ein Ende des Lineals auf einem Tisch fixiert, wobei das andere (freischwingende) Ende sprunghaft mit einem Gewicht belastet wird, schwingt dieses einige Zyklen auf und ab, wobei sich die Amplitude der Schwingung stets verringert, bis sich der stationäre Zustand einstellt. Dies bedeutet, dass die Krafteinwirkung des Gewichts mit der Verschiebung im Gleichgewicht ist. Die Verringerung der Amplitude kann auf Dissipationseffekte im Material zurückgeführt werden, die die Bewegungsenergie in Wärme umwandeln. Skizze 2.7 verdeutlicht den Aufbau des Validierungsfalls und das Gedankenbeispiel.

### Geometrische Maße und Netzinformationen

Der geometrische Aufbau und die Eckdaten sind in Abbildung 2.7 gegeben. Der Vergleich zwischen FVM und FEM wird — wie bereits erwähnt — an einem 2D Beispiel durchgeführt. Die

Erweiterung in der  $z$ -Ebene beträgt 0,001 m. Das Kontinuum wird mithilfe von Hexaederzellen diskretisiert. Jede Zelle hat ein Volumen von  $1e^{-9} \text{ m}^3$  und stellt einen idealen Würfel dar. Entsprechend ist die Kantenlänge in  $x$ ,  $y$  und  $z$  Richtung mit 0,001 m definiert. Daraus resultiert eine Diskretisierung von  $10 \times 300$  Volumenelementen in der  $y$  und  $x$  Ordinate.

In Abbildung 2.7 sind zwei farblich hervorgehobene Punkte dargestellt. An diesen örtlichen Positionen werden nach der numerischen Berechnung die quantitativen Daten extrahiert, die anschließend zum Vergleich zwischen FVM und FEM herangezogen werden. Dabei ist die Position so zu verstehen, dass jeweils der Zellwert des ersten Volumenelements links oben und rechts unten für die Auswertung betrachtet wird. Im roten Punkt ( $P_1$ ) werden die Von Mises Spannungen und im blauen ( $P_2$ ) die Auslenkungen in  $y$ -Richtung gegenübergestellt.

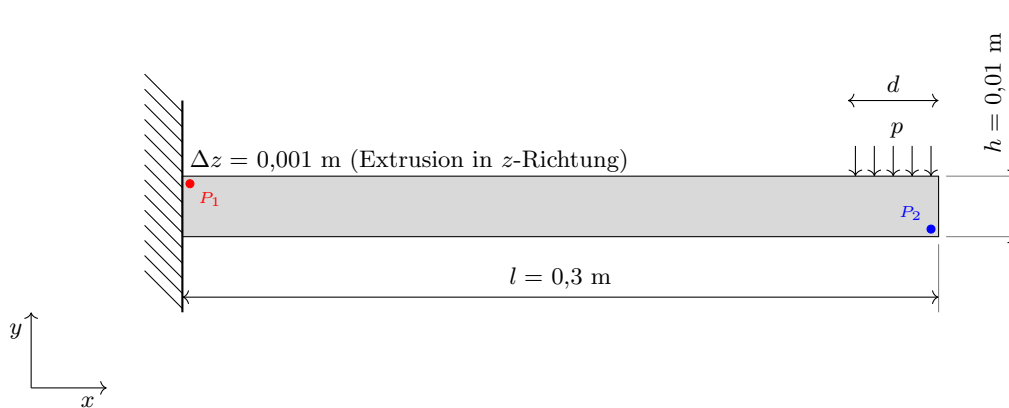


Abbildung 2.7.: Skizze des Biegebalkens mit den geometrischen Maßen. Die Druckkraft  $p$  beträgt 10.000 Pa und wird auf der Länge von  $d = 0,01 \text{ m}$  definiert. Die zwei dargestellten Punkte sind die Auswertungspositionen für die Von Mises Spannung ( $P_1$  ●) und der Verschiebung ( $P_2$  ●).

### Modellierungsparameter FVM und FEM

Die Materialeigenschaften werden aus dem ersten Validierungsfall entnommen und für beide Methoden verwendet. In der FEM wird die Justierung des Zeitschritts automatisch gesetzt und basiert auf der Änderung der Energie im System. Für die FVM ist eine automatische Zeitschrittauswahl ohne größeren Aufwand nicht möglich, sodass drei unterschiedliche Zeitschritte verwendet werden. Diese sind  $\Delta t_1 = 1e^{-4} \text{ s}$ ,  $\Delta t_2 = 1e^{-6} \text{ s}$  und  $\Delta t_3 = 1e^{-7} \text{ s}$ . Die Auswahl der drei Zeitschrittweiten kann über die Courant Zahl (Co) bestimmt werden. Eine Formulierung zur Berechnung der Courant Zahl und der entsprechenden Zeitschrittweite ist im Artikel von Jasak und Weller (1998) geben. Die sich ergebenden Courant Zahlen lauten  $Co = 50$ ,  $Co = 0,5$  und  $Co = 0,05$ .

### Quantitativer Vergleich zwischen der FVM und FEM

Nachfolgend wird der Vergleich zwischen den beiden genannten Methoden quantitativ am oben genannten Beispiel aufgezeigt. Hierzu wird zuerst ein Vergleich der Von Mises Spannung am Punkt  $P_1$  analysiert. Anschließend werden die transienten Auslenkung am Punkt  $P_2$  in  $y$ -Richtung betrachtet und diskutiert. Ferner werden die Ergebnisse der stationären Berechnung gegenübergestellt.

## Transiente Ergebnisse

Die Gegenüberstellung der Von Mises Spannungen beider Methoden — FVM und FEM— ist in Abbildung 2.8a dargestellt. Analog dazu sind die quantitativen Auslenkungen in  $y$ -Richtung am Punkt  $P_2$  in Abbildung 2.8b gegeben.

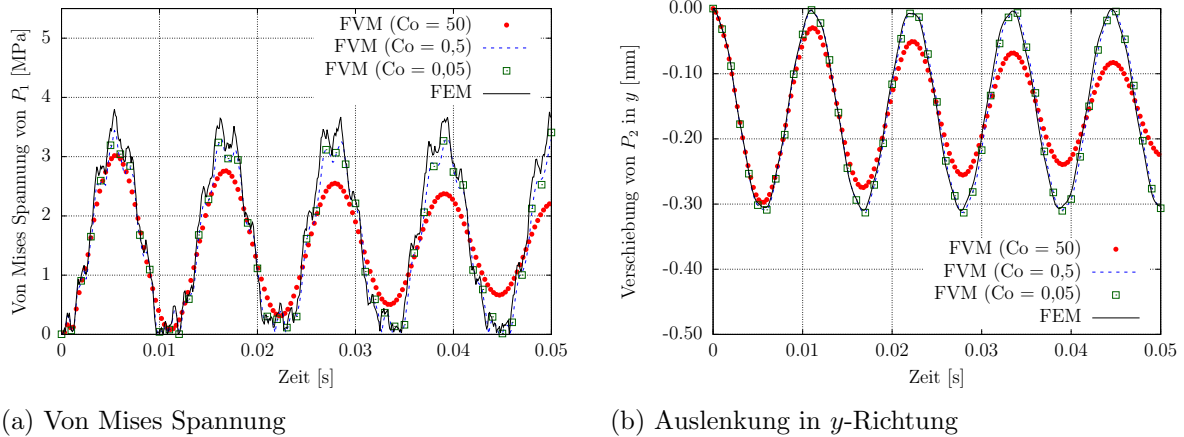


Abbildung 2.8.: Quantitativer Vergleich der Von Mises Spannungen und den Verschiebungen an den Auswertungspunkten ( $P_1$  und  $P_2$ ) zwischen der FVM und FEM.

Es ist erkennbar, dass beide Methoden gleiche Ergebnisse produzieren, sofern die Courant Zahl in der FVM kleiner Eins gehalten wird. Für größere Courant Zahlen wird die Amplitude — jedoch nicht die Frequenz — der Schwingung gedämpft. Dies kann auf die Zeitdiskretisierung der Duhamel-Neumann Gleichung (2.10) zurückgeführt werden. Da im vorliegenden Fall ein Euler Verfahren erster Ordnung verwendet wird und dies stark diffusiv ist, erfolgt eine Glättung der Gradienten, was zu einer Verringerung der Amplitude führt. Dieses Phänomen ist auf die Numerik zurückzuführen und korreliert mit den Aussagen von Jasak und Weller (1998). Zur Umgehung der numerischen Diffusion können Zeitableitungen höherer Ordnung verwendet werden. Da im folgenden Verlauf der Arbeit allerdings stets der stationäre Zustand, aufgrund der in Kapitel 2.5 erwähnten Annahmen, von Interesse ist und in OpenFOAM<sup>®</sup> die zweite Zeitableitung nur mit dem Euler Verfahren erster Ordnung diskretisiert werden kann, wird dies nicht näher untersucht.

Für Courant Zahlen kleiner Eins, werden identische Ergebnisse bezüglich der FEM und FVM erzielt. Wie bereits zu Beginn des Kapitels erwähnt, muss sich eine fortwährende Schwingung einstellen, sofern keine Modellierung der Dissipationseffekte durchgeführt wird. Für den kurzen zeitlichen Ausschnitt der in Abbildung 2.8b gegeben ist, kann diese Aussage als gültig angesehen werden. Analog dazu zeigen die von Mises Spannungen am Punkt  $P_1$  einen identischen periodischen Verlauf; Konvertierung der Verschiebungen in Spannungen und umgekehrt.

Die Fluktuationen, die in Abbildung 2.8a ersichtlich sind, können durch Schallwellenausbreitungen im Material erklärt und bezüglich der Charakteristik der Duhamel-Neumann Formulierung (elliptisch) hergeleitet werden. In elliptischen Gleichungssystemen können sich Informationen im ganzen Raum ausbreiten. Die Schallwellen werden an den Rändern der numerischen Domäne reflektiert, überlagern sich und ergeben diese Fluktuationen.

Es ist erwähnenswert, dass beide Methoden einen *fast* identischen Spannungsverlauf aufweisen. Entsprechend ist die Schallwellenausbreitung im Material gleicher Natur. Zudem soll erwähnt werden, dass in der FEM die Energie des Systems als Größe erhalten bleibt und daher keine

Dämpfung für transiente Berechnungen eintreten kann. In der FVM wird die Energie nicht betrachtet.

## Stationäre Ergebnisse

Als weiterer Vergleich wird die Lösung des stationären Zustandes zwischen der FVM und FEM gegenübergestellt. Da der stationäre Zustand ein Gleichgewicht zwischen einwirkender Kraft und Verformung darstellt, diese einzelne quantitative Werte repräsentieren, werden nachfolgend nur die Werte angegeben und diskutiert. Eine qualitative Betrachtung der Ergebnisse ist in Anhang A.6.3 gegeben.

Im Falle des stationären Zustandes, Gleichgewicht zwischen Druckkraft  $p$  und den Verschiebungstensor  $\mathbf{D}$  respektive dem Spannungenstensor  $\boldsymbol{\sigma}$ , wird eine maximale von Mises Spannung von  $\sigma_{\text{eq}} = 1,48$  MPa (FVM) und  $\sigma_{\text{eq}} = 1,63$  MPa (FEM) erhalten. Ein einfacher Vergleich dieser Zahlen lässt eine starke Abweichung vermuten. Allerdings ist die Deviation von circa 0,2 MPa im Ingenieurwesen vernachlässigbar klein. Ferner ist zu erwähnen, dass die numerische Diskretisierung des Kontinuums gegebenenfalls mit zu wenig Volumenelementen durchgeführt worden ist, dass das Ergebnis beeinflusst. Es ist davon auszugehen dass bei einer größeren Anzahl an Elementen, die Ergebnisse auf einen gemeinsamen Endwert laufen.

Entsprechend korrelieren die Ergebnisse der FVM und FEM für den stationären Zustand. Damit ist bewiesen, dass die FVM in der Strukturmechanik einsetzbar ist. Dies wird durch den Validierungsfall 1 und 2 untermauert.

## 2.10. Numerik und Relaxationsmethoden

Nachfolgend wird kurz auf den Einfluss der Relaxationsmethoden und den entsprechenden numerischen Auswirkungen eingegangen. Die ausführliche Beschreibung der untersuchten Fallbeispiele — mit allen Daten — ist in Anhang B.6 auf Seite 164 gegeben. Des Weiteren wird auf das Kapitel Numerik in Anhang B verwiesen.

Untersucht werden folgende Eigenschaften:

- Einfluss der Feldrelaxation für transiente und stationäre Berechnungen für die Größe  $\mathbf{D}$ ,
- Einfluss der Matrixrelaxation für transiente und stationäre Berechnungen für die Matrix der Größe  $\mathbf{D}$ ,
- Einfluss des finalen Relaxationsfaktors,
- Einfluss der Konvergenzgrenze und -methode (*initial* und *relative*).

Da dieses Kapitel deutlich in die Thematik der numerischen Mathematik eindringt, sind die oben erwähnten Untersuchungen detailliert im Anhang zu finden und werden hier komprimiert zusammengefasst.

### Feldrelaxation

Die Feldrelaxation limitiert den neuen Wert und kann zur Stabilisierung steifer Probleme verwendet werden. Der Begriff *Feld* bezieht sich auf alle numerischen Zellmittelpunktwerte der betrachteten Größe; hier dem Verschiebungsvektor  $\mathbf{D}$ .

Durch die Limitierung wird erreicht, dass sich der neue Iterationswert aus dem alten Iterationswert und der neuen Lösung zusammensetzt; vergleiche Anhang B.5.1 auf Seite 160. Dadurch kann verhindert werden, dass sich — quantitativ betrachtet — Größen zu stark verändern und sich numerische Fehler anfachen; mit anderen Worten: es wird versucht die Lösung des Feldes langsam voranschreiten zu lassen, um unphysikalische Ergebnisse zu vermeiden.

Die im Anhang durchgeführten Analysen zeigen auf, dass bei der Feldrelaxation vor allem der letzte äußere Iterationsschritt mit einem Relaxationsfaktor von Eins ausgeführt werden muss, sofern das Matrixsystem nicht ausreichend weit konvergiert ist. Dies trifft jedoch nur für *transiente* Problemstellungen zu. In stationären Problemstellungen kann die Feldrelaxation uneingeschränkt eingesetzt werden und liefert bei ausreichender Konvergenz einen akkuraten Endwert.

### **Matrixrelaxation**

Verglichen zur Feldrelaxation weist die Matrixrelaxation einen anderen Einflusscharakter auf das zu lösende Gleichungssystem auf. Dies kann einfach damit begründet werden, dass die Matrixrelaxation die Matrix  $\mathbf{A}$  und den Quellvektor  $\mathbf{b}$  manipuliert und als Hauptziel die Verbesserung der Diagonaldominanz der Matrix aufweist. Aufgrund der durchgeführten Tests, die in Anhang B.6 auf Seite 164 gegeben sind, wird klar hervorgehoben, dass die Relaxationsfaktoren einen signifikanten Einfluss auf die Lösungsprozedur haben. Vor allem die korrekte Verwendung des finalen Relaxationsfaktors ist essenziell zur richtigen Abbildung *transienter* Problemstellungen und muss stets Eins betragen, sofern das Matrixsystem nicht weit genug konvergiert ist. Für stationäre Problemstellungen ist bei der Verwendung der Matrixrelaxation darauf zu achten, dass ein strenges Konvergenzkriterium vorgegeben wird.

### **Finaler Relaxationsfaktor**

Der finale Relaxationsfaktor wird auf die letzte äußere Schleife angewendet; vergleiche Anhang B.3 auf Seite 156. Aus der mathematischen Sichtweise ist es eindeutig, dass der letzte Relaxationsfaktor stets eins betragen muss, da sonst der Zeitschritt nicht korrekt aufgelöst wird. Die in Anhang B auf Seite 155 gegebenen Beispiele verdeutlichen diese Aussage. Diese Aussage trifft sowohl für die Feld- als auch für die Matrixrelaxation zu.

### **Konvergenzgrenze und -methode**

Die Auswahl der Konvergenzmethoden (*initial* und *relative*) sowie die Einstellungsparameter des linearen Löser, sind im Anhang B.6 eindeutig geklärt. Die Verwendung der Konvergenzmethode hat keinen Einfluss auf die quantitativen Ergebnisse. Je nach verwendeter Konvergenzmethode muss allerdings eine entsprechend scharfe Konvergenzgrenze für den linearen Lösungsalgorithmus festgelegt werden. Es wird aufgezeigt das beispielsweise bei der Verwendung der *initial* Methode ein wesentlich strengeres Toleranzband eingehalten werden muss als, wenn die *relative* Methode verwendet wird.

Im Allgemeinen kann hieraus abgeleitet werden, dass das Lösungssystem — egal bei welcher Berechnung — stets genau genug berechnet werden muss, um akkurate Ergebnisse zu erhalten. Die Einstellungen des linearen Löser sind vom Anwendungsfall beziehungsweise von der Charakteristik der Gleichung abhängig.





## 3. Modellierung der Materialeigenschaften

Nachfolgend wird die numerische Modellierung der Materialeigenschaften für die Aluminiumlegierung AlSi7Mg0.3 im Detail erörtert. Dabei werden alle Aspekte, die essenziell für die Simulationen notwendig sind, diskutiert. Für die Modellierung der Materialeigenschaften der AlSi7Mg0.3 Legierung werden verschiedene wissenschaftliche Arbeiten herangezogen und an entsprechender Stelle angegeben.

Der Aufbau des nachfolgenden Kapitels ist wie folgt gegliedert:

- Zu Beginn wird kurz drauf eingegangen, welche Ziele in der Modellierung der Materialeigenschaften verfolgt werden und welche Herausforderungen und Schwierigkeiten bestehen.
- Anschließend wird die numerische Modellierung der Materialeigenschaften diskutiert. Hierbei wird zwischen der Erstarrungs- und Kinetikberechnung unterschieden und alle Aspekte, die für die Simulation notwendig sind, hervorgehoben sowie alle Eigenschaften der Phasen und der Matrix aufgezeigt.

Ferner wird aufgezeigt, wie die Radiusverteilungen zweier benötigten Phasen anhand von experimentellen Untersuchungen erzeugt werden.

- Hinzu kommt die Beschreibung der Phasentransformation in MatCalc sowie eine Diskussion zur Erstellung der temperaturabhängigen Transformationsradien.
- Des Weiteren werden die entwickelten MatCalc Skripte vorgestellt und auf die wichtigsten Punkte eingegangen.
- Abschließend werden drei neu entwickelte und untersuchte Methoden zur Inter- und Extrapolation der Streckgrenze  $R_{p,0.2}$  in 3D vorgestellt, erläutert und diskutiert.

### 3.1. Das Ziel und die einzelnen Modellierungsphasen

Die Berechnung der Materialeigenschaften hat das Ziel, die Streckgrenze  $R_{p,0.2}$  der Aluminiumlegierung nach der Wärmebehandlung zu bestimmen. Mit anderen Worten müssen alle in Abbildung 3.1 dargestellten Prozessphasen modelliert werden. Da im allgemeinen Wärmebehandlungen von Aluminiumlegierungen mit dem gesamten Bauteil durchgeführt werden, in dieser Arbeit jedoch lokale Wärmebehandlungen zugrunde liegen, gestaltet sich die Materialberechnung umfangreicher und kann mit den Standardmethoden nicht durchgeführt werden. Dies ist begründet durch die Natur der in diesem Themenbereich angewendeten numerischen Annahmen. Die Materialberechnung basiert auf dem *Field-Average Approach* (FAA). Das bedeutet, dass alle Grundgleichungen nur für einen 0D Fall Gültigkeit besitzen. Entsprechend resultiert die Bedingung, dass die Wärmebehandlung mit dem gesamten Aluminiumbauteil durchgeführt werden muss und eine örtlich begrenzte Wärmebehandlung nicht abbildbar ist. Daraus können zwei Schlussfolgerungen für nicht-homogene Wärmebehandlungen getroffen werden. Zum einen

wird durch den lokal begrenzten Energieeintrag die Annahme der FAA nicht eingehalten und zum anderen muss das Setup für die Materialberechnung für *alle* möglichen Temperaturbereiche und -profile Gültigkeit aufweisen. Das Ziel ist es daher, eine Möglichkeit zu finden, die Materialberechnung für lokale Wärmebehandlungen beziehungsweise inhomogene Energieeinträge zu ermöglichen.

Die Modellierung der Materialeigenschaften kann in vier verschiedene Phasen eingeteilt werden; vergleiche Abbildung 3.1. Jede Phase stellt einen speziellen Schritt bei der Herstellung der Aluminiumlegierung dar und kann numerisch separat betrachtet werden.

In Phase 1 erfolgt die Erstarrung. Das in Abbildung 3.1 angegebene Temperaturprofil ist willkürlich und repräsentiert nicht die Realität, da bei Phasenumwandlungen oder eutektischer Erstarrung die Temperatur auf einem konstanten Wert verharrt.  $T_{\text{Liq.}}$  bezeichnet die *Liquidustemperatur*. Diese gibt an, ab welcher Temperatur die Erstarrung beginnt. Wird die *Solidustemperatur* ( $T_{\text{Sol.}}$ ) erreicht, liegt die eutektische Zusammensetzung vor, wodurch die Restschmelze im eutektischen Punkt erstarrt. Die Solidustemperatur wird solange aufrechterhalten, bis die gesamte Schmelze erstarrt ist. Erst danach, kühlt die Schmelze weiter ab. Es soll darauf hingewiesen werden, dass in Kapitel 3.3 auf Seite 42 nochmals detailliert zwischen der Solidustemperatur und der eutektischen Temperatur unterschieden wird; in der vorliegenden Aluminiumlegierung sind beide Werte identisch.

In Phase 2 kühlt die erstarrte Legierung beispielsweise in einer Sandform weiter ab, bis annähernd Raumtemperatur erreicht wird und das Ausformen stattfindet. Das Temperaturprofil von Phase 2 ist abhängig vom verwendeten Gießverfahren. In dieser Arbeit wird angenommen, dass das Aluminiumbauteil in einem Sandgussverfahren hergestellt wird.

Phase 3 wird als Homogenisierung bezeichnet. Hier wird das *gesamte* Bauteil bis knapp unter die Solidustemperatur erwärmt, für eine gewisse Zeit gehalten und subsequent abgeschreckt. Durch die hohen Temperaturen wandern die Atome der Legierung wie beispielsweise Silizium oder Magnesium und verteilen sich homogen im gesamten Bauteil; Seigerungseffekte werden weitestgehend beseitigt. Des Weiteren besitzt das reine Aluminiumgitter (Matrix) bei erhöhter Temperatur eine zunehmende Löslichkeit der Legierungselemente (Magnesium, Kupfer und so weiter). Durch ein rapides Abkühlen auf Raumtemperatur verringert sich diesbezüglich die Löslichkeit wieder, jedoch besteht keine Möglichkeit, dass die Atome an bevorzugte Positionen wandern, da die Diffusion aufgrund des Abschreckens unterbunden wird. Diese Konfiguration wird als übersättigter Mischkristall bezeichnet. Es soll angemerkt werden, dass alle Aluminiumlegierungen eine solcher Charakteristik aufweisen müssen, wenn ein Warmauslagern stattfinden soll.

Zum Schluss wird die Legierung mit dem übersättigten Mischkristall bei höheren Temperaturen (150 °C bis 220 °C) für mehrere Stunden bis Wochen ausgelagert. Während dieses Prozesses können die im Mischkristall gefangenen Atome diffundieren und bilden verschiedene Kristallstrukturen aus, welche die Materialeigenschaften der Legierung maßgeblich verändern.

Die Vier-Stufen-Berechnung welche in Abbildung 3.1 dargestellt ist, wird numerisch mit der kommerziellen Software MatCalc modelliert; vergleiche Kozeschnik (2017). Alle Einstellungen und Modelle werden nach dem Aufzeigen der Problemstellung erklärt und diskutiert.

### 3.2. Problemstellung

Die Materialberechnung, welche in MatCalc durchgeführt wird, basiert auf der FAA, die für örtliche Wärmebehandlungen nicht herangezogen werden kann. Zur Verdeutlichung dieses Problems wird eine lokale Wärmebehandlung an einem Strukturbauteil bei verschiedenen Zeiten dargestellt; vergleiche Darstellung 3.2. Aufgrund der hohen Wärmeleitfähigkeit von Aluminium

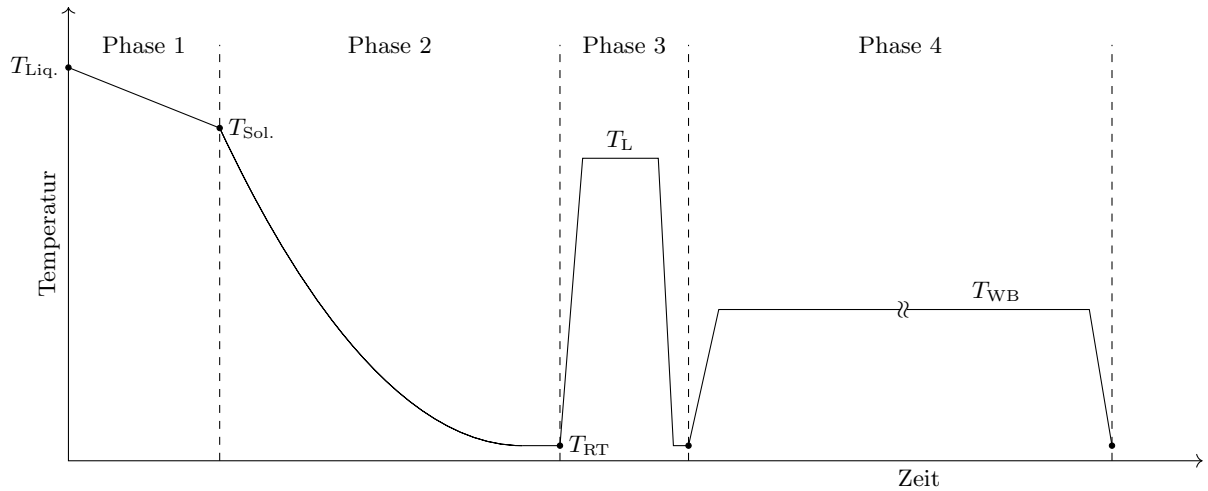
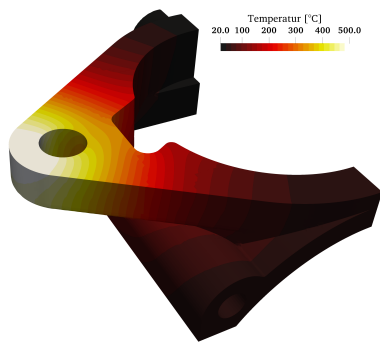
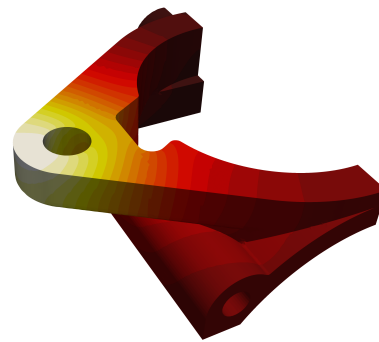


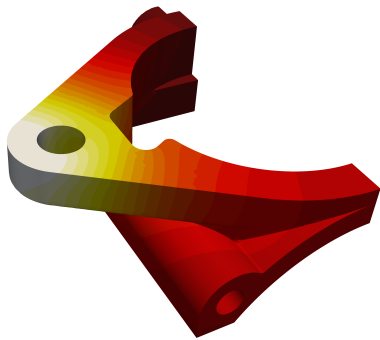
Abbildung 3.1.: Schematische Darstellung der in MatCalc durchgeführten Vier-Stufen-Berechnung zur Bestimmung der Materialeigenschaften nach der Wärmebehandlung.



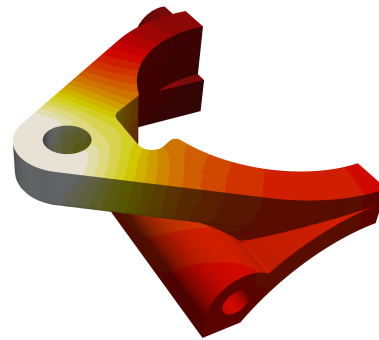
(a)  $t = 5$  s



(b)  $t = 10$  s



(c)  $t = 15$  s



(d)  $t = 20$  s

Abbildung 3.2.: Darstellung einer lokalen Wärmebehandlung eines Strukturbauteils aus Aluminium bei verschiedenen Zeiten.

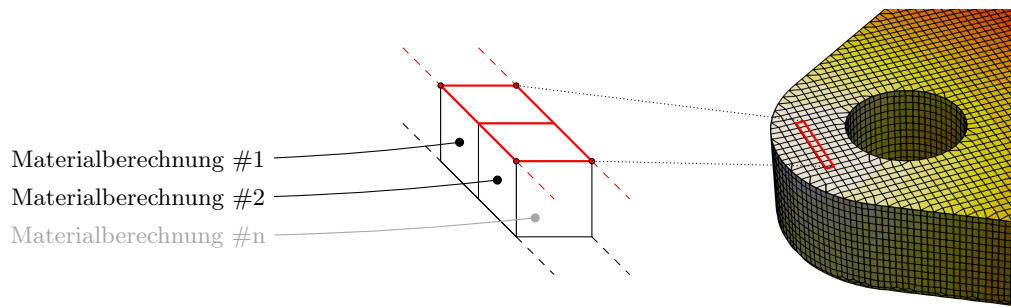


Abbildung 3.3.: Darstellung der Materialberechnung für jede einzelne numerische Zelle am Beispiel des strukturierten Aluminiumbauteils.

wird der lokale Energieeintrag im gesamten Bauteil unmittelbar verteilt. Es ist deutlich ersichtlich, dass zeitlich betrachtet, im gesamten Bauteil unterschiedliche Temperaturen vorherrschen. Diesbezüglich kann eine einzige Materialberechnung mittels der FAA nicht durchgeführt werden.

Jedoch kann unter einer weiteren Annahme die Modellierung über die FAA getätigt werden, und zwar genau dann, wenn jede numerische Zelle für sich als abgeschlossenes System betrachtet wird. Daraus folgt, dass für jedes numerische Volumen eine eigene Materialberechnung durchgeführt werden muss.

Abbildung 3.3 gibt weiteren Aufschluss über die oben genannten Äußerungen. Es ist offensichtlich, dass die Berechnung jeder numerischen Zelle stark davon abhängig ist, wie viele Computerressourcen die Materialberechnung in Anspruch nimmt (Rechenleistung und Zeit pro Zelle) und wie viele Volumina evaluiert werden müssen. Da im Allgemeinen eine Berechnung aller Zellen bezüglich Ressourcen nicht tragbar ist, wird eine neue Berechnungsmethodik entwickelt, die am Ende des Kapitels vorgestellt wird.

Nachfolgend wird davon ausgegangen, dass jede Zelle einzeln evaluiert wird. Aus dieser Annahme werden die vier Prozessphasen diskutiert und alle relevanten Themenbereiche und Modellierungsaufgaben erläutert.

### 3.3. Erstarrungsmodellierung

Die Modellierung der Erstarrung wird mithilfe des Scheil-Gulliver Modells durchgeführt, welches nachfolgend kurz erklärt wird. Es soll angemerkt werden, dass das Scheil-Gulliver Modell nur einen Bruchteil der Phänomene abbilden kann, die während der Erstarrung zugrunde liegen; vergleiche Dantzig und Rappaz (2009). Das Modell ist in MatCalc implementiert, wodurch keine eigenen Entwicklungen notwendig sind.

#### 3.3.1. Das Scheil-Gulliver Modell

Zur Bestimmung der Festigkeit der AlSi7Mg0.3 Legierung nach der Wärmebehandlung, ist es notwendig den Werkstoffzustand nach dem Abguss zu kennen. Numerisch wird dies mit dem Scheil-Gulliver Modell behandelt. Während des Erstarrens sinkt die Temperatur der Schmelze kontinuierlich. Durch Anwendung dieses Modells wird angenommen, dass die Diffusion im Festkörper Null und in der Flüssigkeit unendlich schnell ist; vergleiche Wu et al. (2013). Weitere Informationen zur Herleitung des Modells sind im Buch von Dantzig und Rappaz (2009) gegeben.

Abbildung 3.4 veranschaulicht die Segregation während der Erstarrung einer Aluminiumlegierung anhand des Phasendiagramms Al–Si sowie die unterschiedlichen Konzentrationen des Siliziums in den erstarrten Festkörpern (Bild rechts). Das Ergebnis der Scheil-Gulliver Berechnung ergibt daher eine gemittelte Phasenzusammensetzung.

### Begriffsdefinitionen der Liquidus-, Solidus- und eutektischen Temperatur

Die Begriffsdefinition zwischen Liquidus-, Solidus- und eutektischer Temperatur wird kurz gegeben. Wird ein Phasendiagramm betrachtet, beispielsweise Abbildung 3.4, und eine Legierung mit einer Ausgangszusammensetzung von  $c_0$  herangezogen, folgt: Wird vom Startpunkt (•) eine senkrechte Linie nach unten gezogen, so ist die Liquidustemperatur  $T_{liq.}$  der erste Schnittpunkt im Phasendiagramm (•). Wird die Linie bis zum nächsten Schnittpunkt im Phasendiagramm verfolgt, erreicht man die Solidustemperatur  $T_{sol.}$  (•). Im Allgemeinen reichert sich die Schmelze jedoch mit Legierungselementen an (die Erklärung hierzu erfolgt im nächsten Abschnitt), wodurch die Schmelzenkonzentration zum eutektischen Punkt wandert (•). Dieses Minimum wird als eutektische Temperatur  $T_{eut.}$  bezeichnet. Im Fall des Al–Si Phasendiagramms fallen die eutektische Temperatur  $T_{eut.}$  und Solidustemperatur  $T_{sol.}$  zusammen.

Es soll angemerkt werden, dass in der Literatur die Begriffsdefinitionen unterschiedlich dargestellt werden. Beispielsweise wird die eutektische Temperatur  $T_{eut.}$  von vielen Autoren als Solidustemperatur  $T_{sol.}$  bezeichnet.

### Modellerklärung

Ausgehend von einer flüssigen — homogenen — Aluminiumlegierung wird die Schmelze abgekühlt. Die Schmelzenzusammensetzung ändert sich bis zum Erreichen der Liquiduslinie nicht. Nachdem dieser Punkt (•) erreicht wird, bilden sich erste Aluminiumdendriten, die in der Literatur als  $\alpha$ -(Al) bezeichnet werden. Diese keimen bevorzugt an Verunreinigungen oder Partikel, die als Festkörper in der Schmelze vorliegen. Das Phasendiagramm gibt an diesem Punkt an, wie viel Silizium der Festkörper lösen kann. Es ist ersichtlich, dass der Punkt  $P_{F,1}$  (•) wesentlich weniger Silizium binden kann als in der Schmelze enthalten ist. Daraus resultiert, da mehr

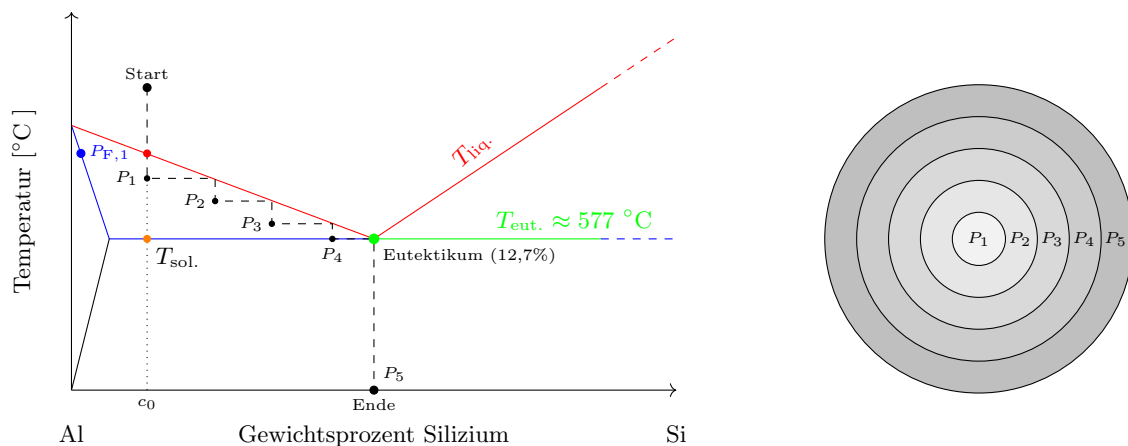


Abbildung 3.4.: Schematische Darstellung des Segregationphänomens das mit dem Scheil-Gulliver Modell bestimmt werden kann. Veranschaulicht am binären Al–Si System; links: Das Phasendiagramm; rechts: Das in MatCalc implementierte Scheil-Gulliver Modell ist als Schalenmodell abgebildet.

Aluminium als Silizium gebunden wird, eine Anreicherung der Schmelze mit Silizium, wodurch sich die Konzentration der verbleibenden Schmelze nach rechts bewegt.

Es ist offensichtlich, dass die neu entstehenden Festkörperanteile bis zum eutektischen Punkt (●) zunehmend mehr Silizium binden können, aber niemals soviel wie die Schmelzenzusammensetzung vorgibt; dieses Phänomen wird als Segregation bezeichnet. Wird der eutektische Punkt erreicht, erstarrt die verbleibende Schmelze rasch, wobei die Konzentration im Festkörper sich nicht weiter ändert. Das überschüssige Silizium wird daher als reine Phase im Eutektikum ausgebildet.

Das Scheil-Gulliver Modell beschreibt damit die Erstarrung der flüssigen Schmelze bis zur eutektischen Temperatur und berücksichtigt dabei das Phänomen der Segregation auf einfache Art und Weise.

Prinzipiell wäre eine Analyse der Legierungszusammensetzung nach der Erstarrung örtlich betrachtet — aufgrund der Segregation — unterschiedlich. Da MatCalc jedoch mit der FAA arbeitet, liegen nach der Scheil-Gulliver Berechnung gemittelte Phasenzusammensetzungen vor. Anschaulich wird ein arithmetisches Mittel der in Abbildung 3.4 (Bild rechts) dargestellten Schalen bestimmt.

### 3.3.2. Legierungselemente und Phasen während der Erstarrung

Das Scheil-Gulliver Modell ist anhand des Zwei-Phasensystems Al–Si erklärt worden. Für Multi-Komponenten Legierungen, wie dies beispielsweise bei der AlSi7Mg0.3 der Fall ist, treten während der Erstarrung nicht nur die  $\alpha$ -(Al) Dendriten und das reine eutektische Silizium auf, sondern es entstehen zudem weitere Phasen. Welche Phasen auftreten, hängt stark von der Zusammensetzung der Legierung ab. Da grundsätzlich Verunreinigungen in der Schmelze wie beispielsweise Eisen (Fe) vorliegen, bilden sich beispielsweise eisenhaltige Phasen aus. Die Norm DIN EN 1706:2013-12 (2013) gibt Aufschluss über die zulässigen Maximalwerte der einzelnen Legierungselemente und Verunreinigungen.

#### Eisen

Enthält die Legierung Eisen, bilden sich die sogenannten AlFeSi, AlFeMgSi oder AlFeMnSi Phasen aus, welche energetisch betrachtet sehr stabil sind (reaktionsträge). Einige Autoren bezeichnen diese als intermetallische Phasen; vergleiche Tan et al. (1996); Caceres et al. (1998) und Colley (2011). Aufgrund der Reaktionsträgheit folgt, dass sich diese Phasen während eines subsequentsen Lösungsglüh- und Auslagerungsprozess nicht weiter verändern und das System dadurch nicht beeinflusst wird. Paz (2003) bestätigt diese Aussage. Laut Tan et al. (1996) ist dies jedoch nur dann gegeben, wenn der Phasenanteil gering ist. Des Weiteren gibt Tan et al. (1996) an, dass Eisen die Materialeigenschaften negativ beeinflusst. Es ist erwähnenswert, dass Eisen als natürliche Verunreinigung in Aluminiumlegierungen auftritt.

#### Magnesium

Magnesium spielt vor allem bei der Auslagerung eine wichtige Rolle, da dieses Element mit Silizium und Aluminium metastabile Phasen bildet. Während der Erstarrung bildet Magnesium mit dem Partner Silizium die stabile  $\beta$ -Phase (Mg<sub>2</sub>Si); vergleiche Liu et al. (1999).

## Silizium

Silizium wird als Mitspieler für die meisten Phasen benötigt. Der hohe Anteil an Silizium in AlSi7Mg0.3 ist allerdings den verbesserten Abgusseigenschaften geschuldet. Silizium bildet sich während der Erstarrung hauptsächlich im Eutektikum, als reine Phase, aus. Die Tendenz, andere Atome zu lösen, geht gegen Null; vergleiche Hajkowski et al. (2012) und Jakesch (2015).

## Kupfer

Kupfer kann als weiteres Element zur Legierung hinzugefügt werden und bildet kupferhaltige Phasen, die — ähnlich zum Magnesium — vor allem während dem Auslagern verfestigende Ausscheidung bildet, welche die Materialeigenschaften verbessern; vergleiche Wang et al. (2007) und Mohamed und Samuel (2012).

## Titan

Das Hinzufügen von Titan, meist als Verbindungen wie  $Ti_2Al$  vorliegend, dient hauptsächlich zur Erhöhung der Keimstellen in der Schmelze. Grund hierfür ist der sehr hohe Schmelzpunkt. Die Titanverbindungen beeinflussen dadurch die Mikrostruktur während der Erstarrung; feineres Gefüge aufgrund erhöhter Keimstellen.

## Abschließend

Es gibt eine Vielzahl weiterer Legierungselemente die in Aluminiumlegierungen eingesetzt werden, diese sind aber nachfolgend nicht weiter aufgeführt. Es soll erwähnt werden, dass Silizium als einziges Element eine reine Phase bilden kann. Alle anderen Elemente gehen stets Verbindungen mit anderen Elementen ein. Des Weiteren können Phasen mehrere Konfigurationen (Anzahl an Atomen in der Verbindung) aufweisen. Dies ist bedingt durch die Legierungszusammensetzung.

### 3.3.3. Erstarrungssequenz bei Aluminiumlegierungen

Während der Erstarrung treten verschiedene Phasenumwandlungen auf. Für die AlSi7Mg0.3 Legierung ist die Phasensequenz, welche von Bäckerud et al. (1990) vorgestellt wurde, weit verbreitet und akzeptiert; vergleiche Tabelle 3.1. Diese Sequenz korreliert mit den Beobachtungen und Ergebnissen der DSC Messung von Mackay und Gruzleski (1998) sowie Wang und Davidson (2001).

Reaktionsnummer	Temperaturbereich [°C ]	Reaktion	
1	Schmelze	$\rightarrow$ Al Dendriten	611 – 615
2	Schmelze	$\rightarrow$ Al + Si	577
3a	Schmelze	$\rightarrow$ Al + Si + $Al_5FeSi$	575
3b	Schmelze + $Al_5FeSi$	$\rightarrow$ Al + Si + $Al_8FeMg_3Si_6$	567
4	Schmelze	$\rightarrow$ Al + Si + $Mg_2Si$	555
5	Schmelze	$\rightarrow$ Al + Si + $Mg_2Si$ + $Al_8FeMg_3Si_6$	550 – 554

Tabelle 3.1.: Akzeptierte Erstarrungssequenz von Al-Si-Mg Legierungen nach Bäckerud et al. (1990).

### 3.4. Radius Verteilung der $\beta$ - und Si-Phase

Nach der Scheil-Gulliver Berechnung liegt ein gemittelter Phasenanteil der erstarrten Phasen vor. Für die AlSi7Mg0.3 Legierung sind vor allem das reine Silizium im Eutektikum und die  $\beta$ -Phase entscheidend. Die Scheil-Gulliver Modellierung gibt jedoch keine Auskunft über die Größe der einzelnen Phasen beziehungsweise eine Phasenverteilung an (Radius und Anzahl der Partikel). Diese Verteilung wird benötigt, wenn nach der Scheil-Gulliver Berechnung eine Kinetikmodellierung folgt.

Die Erstellung der Verteilungsfunktionen für die AlSi7Mg0.3 Legierung wird nachfolgend kurz wiedergegeben. Eine detaillierte Diskussion der Bestimmung dieser Verteilungen sowie die Präsentation aller tabellarisierten Werte ist in Anhang E.1 auf Seite 187 gegeben.

Zur Herleitung der Größenverteilung für die  $\beta$ - und Siliziumphase werden die experimentellen Untersuchungen von Liu et al. (1999); Hajkowski et al. (2012) und Jakesch (2015) herangezogen.

Hierzu werden alle Partikel einer Phase, die in den Schlifffbildern vorzufinden sind, indexiert sowie deren Länge und Breite vermessen. Mit diesen Angaben wird anschließend ein Volumen bestimmt und daraus ein äquivalenter Kugelradius berechnet.

Die Bestimmung des Volumens erfolgt entweder unter der Annahme, dass die Phase als Nadel oder Platte vorliegt und ist dadurch begründet, dass die Schlifffbilder lediglich einen 2D Einblick geben. Die korrekte Struktur ist daher nicht ersichtlich. Aus der Literatur kann entnommen werden, dass das eutektische Silizium als nadelförmiges und die  $\beta$ -Phase als plattenförmiges Konstrukt vorliegt.

Die erhaltenen äquivalenten Kugelradien werden mithilfe eines Histogramms zusammengefasst und anschließend über eine Wahrscheinlichkeitsfunktion (PDF) approximiert. Verwendet wird die Gauß'sche und logarithmische Normalverteilung; vergleiche Abbildung 3.5. Die erstellte Verteilung wird anschließend in MatCalc importiert und für die subsequente Kinetikmodellierung verwendet; die Funktionsparameter, welche in MatCalc verwendet werden, sind in Tabelle 3.2 gegeben. Es wird darauf hingewiesen, dass die Darstellungen in 3.5 unterschiedliche Abszisseneinteilungen bezüglich den Radien aufweisen. Diese Umsetzung erlaubt es einen höheren Detaillierungsgrad zu erzeugen.

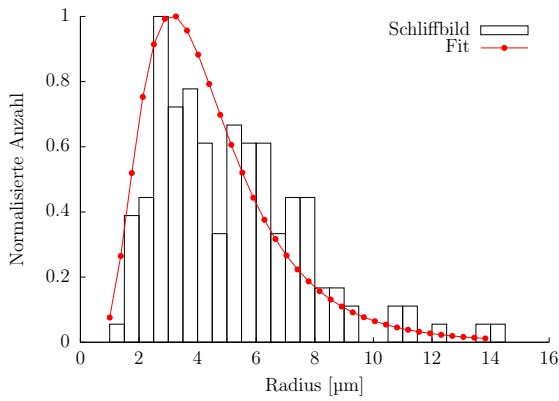
Die Abbildung der Phasenverteilung nach Weisz und Povoden-Karadeniz wird in dieser Arbeit nicht verwendet, begründet dadurch, dass diese Radienverteilung ohne Beweis vorliegt und nur eine grobe Annahme darstellt.

Es wird angemerkt, dass die hier erstellten Verteilungen eine Abschätzung darstellt, da vor allem die Erstellung des Histogramms maßgeblich durch die Bandbreite beeinflusst wird. Ferner können die approximierten Verteilungsfunktionen, vergleiche Abbildung 3.5, unterschiedlich gefittet werden. Des Weiteren ist die Bestimmung der Ausscheidungslänge und -breite anhand

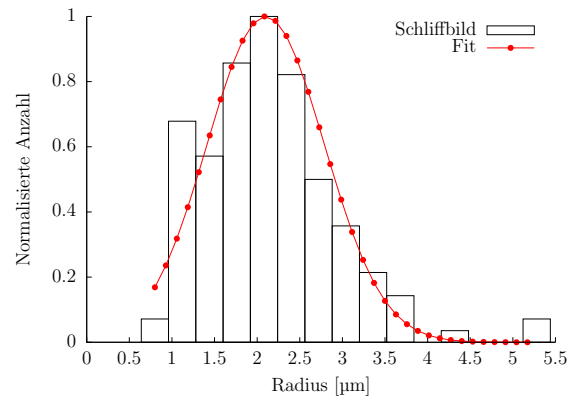
	Bezeichnung (Skript)	Funktionstyp	$r_{\min}$ [nm]	$r_{\max}$ [nm]	$r_{\text{mittel}}$ [nm]	Standardabweichung [-]
Weisz und Karadeniz	DEF	Log	5,0	15,0	10,0	0,02
Holzmann (Nadel)	HOL_N	Log-Normal	0,5	6,0	2,2	0,35
Holzmann (Platte)	HOL_P	Log-Normal	1,0	14,4	4,4	0,70

Tabelle 3.2.: Mögliche Optionen für die Verteilungsgenerierung der Siliziumphase. Angegeben sind die Funktionstypen, die verwendeten Radien und die Standardabweichung die zur Abbildung der Verteilung notwendig sind.

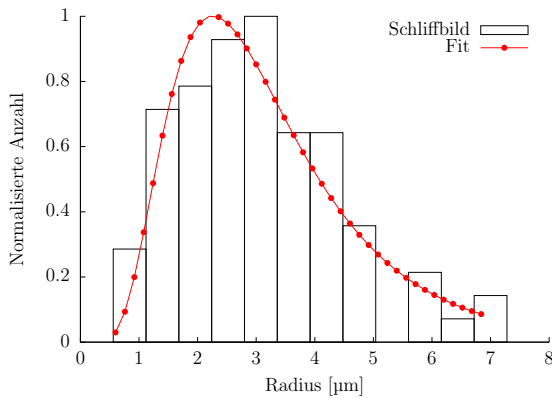




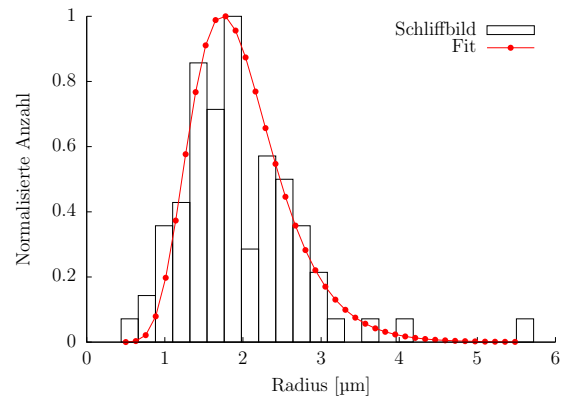
(a) Si-Phase, Annahme Platte



(b) Si-Phase, Annahme Nadel



(c)  $\beta$ -Phase, Annahme Platte



(d)  $\beta$ -Phase, Annahme Nadel

Abbildung 3.5.: Ermittelte Größenverteilung der Silizium- und  $\beta$ -Phase mit den Daten aus Hajkowski et al. (2012); Jakesch (2015) und Liu et al. (1999) nach dem Gießen..

von Schliffbilder mittels Bildbearbeitungsprogrammen ein weiterer Einflussfaktor, der bei der Gesamtbeurteilung hinsichtlich Genauigkeit der Verteilungen berücksichtigt werden muss. Diesbezüglich sind die hier vorgestellten Ergebnisse eine mögliche Variante.

### 3.5. Kinetikmodellierung

Nachdem die Phasenzusammensetzung nach der Erstarrung und die dazugehörigen Radienverteilungen bekannt sind, bleibt das abgegossene Bauteil weiter im Sand eingeformt und kühlt langsam auf Raumtemperatur herunter. Während dieser Prozessphase sowie den nachfolgenden — dem Lösungsglühen, Abschrecken und Auslagern — treten kinetische Aspekte auf, die das Material verändern. Dazu gehört die Nukleation neuer Phasen, deren Phasentransformation, das Phasenwachstum und ähnliches. Zur Vorhersage der im Material ablaufenden Ereignisse, werden Kinetikrechnungen durchgeführt. Alle relevanten Informationen diesbezüglich sind nachfolgend gegeben. Die Struktur dieses Kapitels ist wie folgt:

- Zu Beginn wird die numerische Modellierung der Kinetikberechnung aufgegriffen. Anders als im Kapitel der Thermospannungen, werden keine Gleichungen präsentiert, da mit der Software MatCalc gearbeitet wird und hierbei keine Eigenentwicklungen durchgeführt werden können (Blackbox). Daher wird der Leser auf Literaturquellen hingewiesen. Ferner werden die Einzelmodelle kurz erklärt.

- Anschließend wird auf die Phasenumwandlungssequenz eingegangen und nachfolgend die Eigenschaften der einzelnen Phasen und der Matrix diskutiert und mit Literatur belegt sowie in einer Tabelle zusammengetragen.
- Zum Schluss des Kapitels wird die Modellierung der Phasentransformation (Umwandlung) diskutiert, wobei die in dieser Arbeit neu entwickelte Methodik aufgezeigt wird.

### 3.5.1. Numerische Modelle

Die numerische Modellierung der kinetischen Aspekte wird mithilfe der kommerziellen Software MatCalc durchgeführt. Die Umsetzung der Berechnung erfolgt durch die Verwendung der bereits erwähnten FAA. Diesbezüglich können nur Informationen an einem Punkt — 0D Berechnung — bestimmt werden, wodurch die Annahme getroffen werden muss, dass das gesamte Bauteil stets identische Eigenschaften besitzt.

Nachfolgend werden die einzelnen mathematischen Schwerpunkte aufgegriffen, die für die Kinetikberechnung entscheidend sind.

#### Grundgleichung

Die Grundgleichungen basieren — ähnlich wie in der Verbrennungslehre — auf der Berechnung der freien Gibbs Energie, die wiederum mit vielen weiteren physikalischen Größen bestimmt wird, auf die hier nicht eingegangen wird. Die freie Gibbs Energie wird allgemein dazu verwendet die Nukleationsrate zu bestimmen. Des Weiteren sind Informationen bezüglich der Diffusion nötig, um die Kinetik wie beispielsweise das Wachstum von Ausscheidungen zu bestimmen. Die Modellierung dieser Größen ist sehr aufwendig und komplex und benötigt grundsätzlich alle Informationen über die Legierung (Zusammensetzung und Elemente), womit über CALPHAD anschließend alle weiteren Größen ermittelt werden. Alle Informationen dazu sind in Svoboda et al. (2004) und Kozeschnik et al. (2004) gegeben. Weiterführende Informationen — explizit die Interaktion mit der Software MatCalc — sind in Warczok (2017) zu finden.

#### Gitter-Freistellen (Vacancies)

MatCalc ermöglicht es Gitterfreistellen, welche beispielsweise während eines Abschreckvorgangs entstehen, zu berücksichtigen. Diese beteiligen sich maßgeblich an der Diffusion einzelner Atome, da diese über die Freistellen leichter wandern können. Zur Modellierung dieses Phänomens werden in MatCalc modifizierte Gleichungen verwendet; vergleiche die klassische Nukleationstheorie von Svoboda et al. (2004) mit der gegeben in Warczok (2017). Zur Modellierung müssen Formulierungen, die die Erzeugung und Vernichtung der Vacancies sowie das Besetzen dieser mit Atomen oder Ausscheidungen, vorgegeben werden. Alle Einzelheiten der numerischen Abbildung und die Wichtigkeit der Gitterfreistellen — vor allem beim Auslagerungen in Raumtemperatur — ist in Fischer et al. (2011, 2013) und Weisz et al. (2015) gegeben.

#### Shape-Faktoren

Diffusionsvorgänge sind stark von der Gestalt der Ausscheidungen beziehungsweise Phasen abhängig. In der klassischen Modellierung werden stets sphärische Annahme bezüglich Ausscheidungsgestalt getroffen. MatCalc ermöglicht es jedoch unter Verwendung der sogenannten

Shape-Faktoren die Grundgleichungen zu manipulieren und damit die Gestalt der Ausscheidungen zu berücksichtigen. Die Modifikation geht hauptsächlich in die Berechnung der freien Gibbs Energie und den Dissipationsraten ein. Das Shape-Faktor-Modell bietet dadurch die Möglichkeit, Ausscheidungen oder Phasen als Platte oder Stab abzubilden und führt dazu, dass beispielsweise die Wachstumsgeschwindigkeit verändert wird. Shape-Faktoren mit einem Wert  $> 1$  stellen nadel- und  $< 1$  plattenförmige Phasen dar. Es ist erwähnenswert, dass ein Wert von 1 keine Kugel repräsentiert. Dies geht aus der Natur der Gleichungen hervor und kann einfach bewiesen werden. Alle Details hierzu sind in Kozeschnik et al. (2006) gegeben.

### **Gitter-Fehlanpassung**

Ausscheidungen treten in kohärenten, teil-kohärenten und nicht-kohärenten Strukturen im Vergleich zum Aluminiumgitter auf. Untersuchungen zeigen allerdings, dass kohärenten Strukturen unter Abwesenheit des umgebenden Aluminiumgitters Abweichungen der Lattice-Parameter aufzeigen, wodurch natürliche Verspannungen entstehen und dadurch eine Änderung im Verhalten des Wachstums einhergeht. Des Weiteren führen diese Verspannungen zu einer Erhöhung der Zugfestigkeit und sind kritische Eingabeparameter, da das numerische System sehr sensibel auf diese reagiert. Die Modellentwicklung ist in Fischer et al. (2015) gegeben. Die Gitter-Fehlanpassung wird als volumetrischer Misfit bezeichnet.

### **Koagulation**

Einige Phasen koagulieren mit der Zeit, wie beispielsweise die Cluster-Phase, welche in Aluminiumlegierungen sehr entscheidend ist. MatCalc bietet hier verschiedene Modelle an, die in Warczok et al. (2012) diskutiert werden. In dieser Arbeit wird das Binder–Stauffer Modell verwendet, das explizit in Binder und Stauffer (1974) beschrieben ist.

### **Die Interface Energie**

Die Interface Energie ist eine fundamentale Größe in der Kinetikberechnung. Diese wird mit komplexen mathematischen Formulierungen und empirisch ermittelten Zahlenwerten bestimmt. Die Berechnung dieser Energie erfolgt in MatCalc mit der *Generalized Nearest-Neighbor-Broken-Bond* (GNNB) Approximation und setzt ein *Planar-Sharp* Interface voraus, das eine Allgemeingültigkeit besitzt. Die Artikel von Sonderegger und Kozeschnik (2009a,b) beschreiben und diskutieren dieses Modell. Werden diffuse Grenzflächen erwartet, kann eine Modifikation des GNNB verwendet werden; vergleiche Sonderegger und Kozeschnik (2010).

### **Materialfestigkeit**

Die Bestimmung der Materialfestigkeit aufgrund des Materialzustands (welche Phasen liegen vor sowie deren Ausscheidungsverteilung und Anzahl) ist eine weitere komplexe Modellierung, bei denen verschiedenen Mechanismen zur Festigkeit beitragen. Diese sind in MatCalc wie folgt definiert; aufgrund der Nomenklatur werden die englischen Fachbegriffe beibehalten:

- Intrinsic Strengthening (Grundfestigkeit der Matrix),
- Solution Strengthening (Steigerung durch gelöste Atome),

- Dislocation Strengthening (Steigerung durch Gitterversetzungen),
- Grain/Subgrain Boundary Strengthening (Steigerung durch Phasengrenzen),
- Precipitation Strengthening (Steigerung durch Ausscheidungen),
  - Dislocation Line Tension,
  - Interaction Between Dislocations and Precipitates,
  - Shearing mechanisms,
    - \* Coherency Effect,
    - \* Modulus,
    - \* Anti-Phase Boundary Effect,
      - Weak and Shearable Precipitates,
      - Stacking fault effect,
      - Interfacial effect,
    - \* Non-shearing mechanism.

Es ist deutlich ersichtlich, dass eine Vielzahl unterschiedlicher Mechanismen bei der numerischen Modellierung der Materialfestigkeit wirken. Die Auflistung der oben genannten Modelle und Diskussionen sind in den Veröffentlichungen von Ahmadi et al. (2014a,b) und Warczok (2017) gegeben.

Für die Cluster-Phase treffen die oben genannten Mechanismen nicht zu. MatCalc bietet hier die Möglichkeit das Cluster Modell von Starink et al. (2012) zu verwenden.

Allgemein kann für Al–Mg–Si Legierungen folgende Aussage getroffen werden: Je fein verteilter die Ausscheidungen in der Gitterstruktur vorliegen, desto höher ist ihr verfestigender Beitrag.

### **Theorien und Modelle explizit für Aluminiumlegierungen**

Die numerische Abbildung der einzelnen Phasen bedarf einer thermodynamischen Modellierung, bei der die freie Gibbs Energie berechnet werden kann. Eine Auflistung der Modelle, sowie deren Parameter sind in Povoden-Karadeniz et al. (2013) gegeben. Neuere Theorien im Bereich der Cluster-Bildung kann im Artikel von Svoboda et al. (2016) eingesehen werden.

Es soll vorab die Information gegeben werden, dass explizit die Cluster in Aluminiumlegierungen eine entscheidende Rolle für die Phasentransformationskette darstellt.

### **MatCalc Publikationen im Bereich der Aluminiumlegierungen**

Die Validierung und der Gültigkeitsbeweis der in MatCalc implementierten Modelle und Parameter sind in einigen Veröffentlichungen seitens dem Wiener Institut für Werkstoffwissenschaft und Werkstofftechnologie aufgeführt. Siehe hierzu Falahati et al. (2010, 2011); Pogatscher et al. (2014) und Weisz et al. (2015).

### 3.5.2. Phasen und Phasenumwandlungen in Aluminiumlegierungen

Während der Abkühlung auf Raumtemperatur, dem Lösungsglühen, Abschrecken und Auslagern, treten verschiedene Ausscheidung beziehungsweise Phasen im Material auf. Da jede Phase in MatCalc modelliert wird, ist es erforderlich alle während der Kinetikrechnung auftretende Ausscheidungen vor Beginn der Simulation zu definieren und alle nötigen Eingabeparameter zu setzen. Die in Aluminiumlegierungen auftretenden Phasen werden experimentell mittels der dynamischen Differenzkalorimetrie (DSC) bestimmt. Hierzu wird die zu untersuchende Legierung nach dem Abschrecken (übersättigter Mischkristall) langsam erwärmt. Die Aufheizrate beträgt lediglich einige Kelvin pro Minute;  $5 \text{ Kmin}^{-1}$  in Falahati et al. (2011). Während des Erwärmens werden die Temperaturen in der Legierung und einem Referenzblock (zumeist aus reinstem Aluminium) mit gemessen und beide gegenübergestellt. Da in der Legierung Phasen keimen, sich diese in andere umwandeln und endo- oder exotherm ablaufen, kann mithilfe der Temperaturdifferenz auf die Phasenumwandlungen im Material geschlossen werden. Für Al-Si-Mg Legierungen sind die typischen Ausscheidungen:

- Cluster-Phase (MgSi Verbindung mit einem Atomverhältnis von 1:1 — kann jedoch abweichen und zusätzlich Aluminium enthalten; vergleiche van Huis et al. (2007)),
- GP-Zonen (Agglomeration der Cluster-Phase mit Einbau von Aluminium),
- $\beta''$ -Phase ( $\text{Mg}_5\text{Si}_6$  Verbindung — frühe metastabile Phase, kann Aluminium enthalten),
- $\beta'$ -Phase (Mg Verbindung — frühe metastabile Phase, welche aus  $\beta''$  entsteht),
- $\beta$ -Phase ( $\text{Mg}_2\text{Si}$  Verbindung — stabile Phase),
- Silizium-Phase.

Die Phasentransformation während einer Wärmebehandlung folgt der hier dargestellten Aufzählung und ist von vielen Autoren bestätigt worden. Dabei wird die Phasentransformation, die die Umwandlung von einer Phase in die nächste beschreibt, stets mit dem lösungsgeglühten Zustand definiert und über das oben erwähnte DSC Verfahren bestimmt. Die Phasensequenz ist in Tabelle 3.3 gegeben. Es ist ersichtlich, dass die GP-Zone von verschiedenen Autoren unterschiedlich deklariert wird. Die U1-, U2- sowie  $\beta'$ -Phasen, die nach van Huis et al. (2006) und Marioara et al. (2006) klassifiziert werden, stellen andere Unterarten der  $\beta'$ -Phase dar.

Autor	Phasen Sequenz
Nach Marioara et al. (2006)	SSSS $\rightarrow$ atomic clusters $\rightarrow$ GP-zones (pre- $\beta''$ ) $\rightarrow \beta'' \rightarrow$ U1, U2, $\beta'$ , $\beta' \rightarrow \beta$ Si
Nach van Huis et al. (2006)	SSSS $\rightarrow$ GP-zones $\rightarrow$ pre- $\beta''$ $\rightarrow \beta'' \rightarrow$ U1, U2, $\beta'$ , $\beta' \rightarrow \beta$ Si
Nach Dutta und Allen (1991)	SSSS $\rightarrow$ GP-zones (I) $\rightarrow$ GP-zones (II) or $\beta''$ $\rightarrow \beta' \rightarrow \beta$
In dieser Arbeit betrachtet	SSSS $\rightarrow$ atomic clusters $\rightarrow \beta'' \rightarrow \beta' \rightarrow \beta$

Tabelle 3.3.: Postulierte Ausscheidungssequenz in Al-Si-Mg Legierungen von verschiedenen Autoren und der in dieser Arbeit verwendeten; SSSS bezeichnet dabei den lösungsgeglühten und in Wasser abgeschreckten Zustand und wird als *Super Saturated Solid Solution* benannt.

An dieser Stelle soll erwähnt werden, dass die in dieser Arbeit verwendete Phasentransformation der von Marioara et al. (2006) folgt, wobei nach Absprache mit Professor Ernst Kozeschnik die GP-Zonen vernachlässigt werden. Entsprechend wird die letzte Sequenz in Tabelle 3.4 für die Modellierung herangezogen.

### 3.5.3. Phaseneigenschaften

Nachfolgend wird auf signifikante Phaseneigenschaften eingegangen. Die Analyse der Literaturrecherche ist detailliert am Ende dieses Kapitels in Tabelle 3.4 zusammengefasst. Die Tabelle zeigt die Charakteristika und weitere Details verschiedener Autoren bezüglich der hier präsentierten Phasen auf. Für die spätere Modellierung wird diese Tabelle herangezogen und soll zudem einen Überblick über die Unsicherheit beziehungsweise Streuung diverser Größen darstellen sowie deren Diskrepanz in der Literatur aufzeigen.

#### Cluster-Phase

Die Cluster-Phase stellt die wichtigste Ausscheidungsphase dar. Vor allem beim Kaltauslagern (Raumtemperatur) ist diese Phase die festigkeitssteigernde. Die Clusterzusammensetzung stellt eine Verbindung aus Magnesium- und Siliziumatomen dar, die grundsätzlich im Verhältnis 1:1 auftreten. Des Weiteren kann diese Phase überall keimen und wachsen. Abweichungen des Verhältnisses sind möglich, wie van Huis et al. (2006) in seiner Auflistung zeigt. Sie ist kohärent zum Aluminiumgitter, metastabil und stellt den Anfang der Ausscheidungssequenz dar. Die Größe der Ausscheidungen liegt im Nanometerbereich.

#### $\beta''$ -Phase

Die frühe und metastabile  $\beta''$ -Phase ist die Ausscheidung, die den quantitativ größten Verfestigungsbeitrag in Al–Mg–Si Legierungen während des Warmauslagerns erzeugen. Dies liegt an ihrem kohärenten Charakter und dem quantitativ hohen volumetrischen Misfit. Das bedeutet, dass das Kristallgitter der Matrix stark verspannt wird, wodurch Gitterversetzungen (gleiten der einzelnen Atomschichten) erschwert werden. Hinzu kommt, dass diese Ausscheidungen sehr hart sind und sich bevorzugt an Gitterversetzungen ausbilden. Die Struktur dieser Phase ist mit der Formel  $\text{Mg}_5\text{Si}_6$  gegeben, kann jedoch variieren und Aluminium enthalten. Diese Ausscheidungen sind sehr klein und betragen nur wenige Nanometer. Die Gestalt der  $\beta''$ -Phase wird in der Literatur als nadelförmig beschrieben.

#### $\beta'$ -Phase

Wird das Warmauslagern lang genug durchgeführt, wandelt sich die  $\beta''$ -Phase in die  $\beta'$ -Phase um. Diese semi-kohärente Phase besitzt einen kleineren volumetrischen Misfit, ist metastabil und wie ihr Vorgänger an Gitterversetzungen zu finden. Diesbezüglich hat diese Phase einen beachtlichen Verfestigungsbeitrag zur Gesamtfestigkeit, ist aber im Vergleich zur  $\beta''$ -Phase geringer. Die Größe dieser Phase beträgt einige bis zehn Nanometer. Ihre Gestalt wird in der Literatur als stabförmige beziehungsweise nadelförmige Ausscheidung postuliert.

## $\beta$ -Phase

Bei zu langer Auslagerung transformiert sich die  $\beta''$ -Phase vollständig in  $\beta'$  und diese beginnt sich in die stabile  $\beta$ -Phase umzuwandeln; hier wird von Überalterung gesprochen. Diese Phase ist inkohärent und somit willkürlich im Aluminiumgitter angeordnet. Die Größe der Ausscheidungen beträgt einige zehn bis hundert Nanometer. Ferner ist die stabile  $\beta$ -Phase bereits nach der Erstarrung vorhanden, jedoch ist ihr Phasenanteil dort sehr gering. Die Form dieser Ausscheidung wird in der Literatur als stabförmig deklariert.

## Silizium

Reines Silizium ist als Hauptbestandteil im eutektischen Gefüge als nadelförmiges beziehungsweise plattenförmiges Konstrukt auffindbar. Während der Wärmebehandlung können sich erneut kleine Silizium Ausscheidungen bilden, die jedoch keinen signifikanten Beitrag zur Verfestigung beisteuern.

### 3.5.4. Matrixeigenschaften

Die Matrix der Aluminiumlegierungen, welche mit  $\alpha$ -(Al) bezeichnet wird, stellt das Aluminiumgitter dar. Zur Modellierung der Kinetikberechnung hat diese einen entscheidenden Einflussfaktor auf das Ergebnis. Beispielsweise legt die Matrix fest, wie viele Nukleationsstellen in der Matrix vorhanden sind, oder wie hoch die Versetzungsdichte ist. Darüber hinaus fließen hier das E-Modul und die Querkontraktionszahl mit ein. Die Korngröße des  $\alpha$ -(Al) beeinflusst weiter die Grundgleichungen und somit die Ergebnisse. Letztere Quantität ist vor allem durch Feinungszusätze, der Legierungszusammensetzung und/oder der Abkühlrate während des Erstarrens modifizierbar; vergleiche Wang et al. (2004); Jiang et al. (2012); Marzouk (2012); Borodianskiy et al. (2015) und Wlanis (2015).

### 3.5.5. Zusammenfassung der Phasen- und Matrixeigenschaften

Eine Gegenüberstellung der in der Literatur vorzufindenden Eigenschaften der oben genannten Phasen ist in Tabelle 3.4 zusammengetragen. Die GP-Zonen sind als Ergänzung hinzugefügt.

Phase	Bez. MatCalc	Formel	Struktur	Zustand	Nukleation	Form	SF
Cluster	CL_MGSI	MgSi	K <sup>1 2</sup>	MS <sup>1 3</sup>	B <sup>1</sup>	S <sup>1 4</sup>	–
GP-Zone	GP_MGSI	Mg <sub>1</sub> Si <sub>1</sub>	K <sup>3 5 6</sup>	MS <sup>1 3</sup>	B <sup>1 4</sup>	N <sup>4 7 8</sup> , S <sup>4 9</sup>	4
$\beta''$	AL_B_DP	Al <sub>2</sub> Mg <sub>5</sub> Si <sub>4</sub>	K <sup>1 11</sup> , SK <sup>3</sup>	MS <sup>12 13</sup>	B <sup>4</sup> , D <sup>1</sup>	S <sup>4</sup> , N <sup>10 14</sup>	8
$\beta'$	MGSLB_P	Mg <sub>1.8</sub> Si	SK <sup>3 5 11</sup>	MS <sup>1 3 15</sup>	D <sup>1 4</sup>	N <sup>3</sup> , S <sup>4 14</sup>	7
$\beta$	MG2SLB	Mg <sub>2</sub> Si	IK <sup>1 3 11</sup>	S <sup>3 5</sup>	D <sup>1 5</sup>	P <sup>2 8 11</sup> , W <sup>11</sup>	0,3
Silizium	SLDIAMOND_A	Si	IK <sup>3 5</sup>	S <sup>1 5</sup>	D <sup>1 5</sup>	P <sup>18</sup>	0,6

Phase	Größe (circa)	E-Modul	Poisson $\nu$	Fehlanpassung	
	[nm]	[GPa]	[-]	dL/L	dV/V
Cluster	1 <sup>2</sup> , 1–5 <sup>4</sup>	$\Delta_M$ 10 <sup>27</sup>	–	–	2–4*, 5 <sup>27</sup>
GP-Zone	2,5 <sup>10</sup> , 1–3 bis 2x2x20 <sup>11</sup>	–	–	–	1 <sup>5</sup> , 2,6–5,8 <sup>4</sup>
$\beta''$	4x4x50 <sup>10 12</sup> , ø4x15 bis ø6x50 <sup>2</sup>	67 <sup>12</sup>	0,333 <sup>12</sup>	–	3 <sup>5</sup> , 5 <sup>1</sup> , 9,8 <sup>4</sup>
$\beta'$	ø55x55 <sup>2</sup> , ø15x einige 100 <sup>11</sup>	–	–	2 <sup>15</sup>	4 <sup>5</sup> , 16 <sup>11</sup>
$\beta$	10000–20000 <sup>11</sup>	107 <sup>16</sup> , 120 <sup>27</sup>	0,167 <sup>16</sup>	–	6 <sup>5</sup>
Silizium	ø0,4x0,5–10 <sup>18</sup>	190 <sup>20</sup> , 105 <sup>22</sup> , 98–107 <sup>21</sup>	0,220 <sup>19</sup>	–	5 <sup>5</sup> , 19 <sup>23</sup>

Phase	Korngröße	E-Modul	Poisson $\nu$	Versetzungsdichte
	[ $\mu$ m]	[GPa]	[-]	x 10 <sup>11</sup> [m <sup>-2</sup> ]
Matrix	95–1000 <sup>17 18 24 25 26</sup> 27	40–70 <sup>10</sup>	0,33	0,9 – 90 <sup>26</sup> , 1 <sup>27</sup> , 100 <sup>28</sup> 500 <sup>5</sup>

Tabelle 3.4.: Zusammenfassende Eigenschaften einzelner Ausscheidungsphasen und der Matrix. Definitionen: Struktur: „K“ Kohärent; „SK“ Semi-Kohärent; „I“ In-Kohärent. Zustand: „MS“ Metastabil; „S“ Stabil. „SF“ Shape-Faktor bezogen auf Kozeschnik et al. (2006). Nukleation: „B“ Bulk; „V“ Versetzungen. Form: „S“ sphärisch; „P“ Platten ähnlich; „S“ Stab ähnlich; „N“ Nadel ähnlich; „W“ Würfel ähnlich.  $\Delta_M$  bezeichnet die Differenz zur Matrix (kein Absolutwert).

<sup>1</sup>Falahati et al. (2011)

<sup>4</sup>van Huis et al. (2007)

<sup>7</sup>Matsuda et al. (1998)

<sup>10</sup>Zandbergen et al. (1997)

<sup>13</sup>Wang und Davidson (2001)

<sup>16</sup>Ganeshan et al. (2009)

<sup>19</sup>Hopcroft et al. (2010)

<sup>22</sup>Skimin und Andreatch (2004)

<sup>25</sup>Borodianskiy et al. (2015)

<sup>28</sup>Kavosi et al. (2014)

<sup>2</sup>Dutta und Allen (1991)

<sup>5</sup>Falahati et al. (2010)

<sup>8</sup>Marioara et al. (2006)

<sup>11</sup>Vissers et al. (2007)

<sup>14</sup>Sagalowicz et al. (2010)

<sup>17</sup>Li et al. (2017)

<sup>20</sup>Petersen (1982)

<sup>23</sup>Hornbogen et al. (1993)

<sup>26</sup>Nes und Marthinsen (2002)

\*Dr. Piotr Warczok (MatCalc Engineering)

<sup>3</sup>Teichmann et al. (2013)

<sup>6</sup>Xiao et al. (2016)

<sup>9</sup>Apelian et al. (1989)

<sup>12</sup>Yu et al. (2010)

<sup>15</sup>Ehlers et al. (2014)

<sup>18</sup>Marzouk (2012)

<sup>21</sup>Borchi et al. (1994)

<sup>24</sup>Wang et al. (2004)

<sup>27</sup>Weisz et al. (2015)



### 3.5.6. Phasentransformationsmodellierung

Die Bestimmung der auftretenden Phasen während der Wärmebehandlung ist vor allem für die Vorhersage der Materialeigenschaften essenziell. Die akkurate Berechnung der Streckgrenze  $R_{p,0.2}$  in einem definierten Arbeitspunkt ist mit MatCalc ohne weiteres möglich; vergleiche Falahati et al. (2010). Dabei bezieht sich der Arbeitspunkt auf definierte Aufheiz- und Abkühlraten während des Lösungsglühens und dem Auslagern sowie der entsprechenden Haltezeit.

Zur Abbildung der Materialeigenschaften in einem fest definierten Arbeitspunkt können zwei verschiedene Ansätze verfolgt werden: zum einen ist es möglich die Phasentransformation generell zu ignorieren und das Fitten durch die Manipulation der in MatCalc möglichen Einstellungsparameter einzelner Phasen zu gewährleisten oder zum anderen kann das Radian basierende Transformationsmodell verwendet werden.

Ein auf physikalischen Grundlagen basierendes Modell ist bisher nicht postuliert und diesbezüglich in MatCalc nicht verfügbar.

#### Ignorieren der Phasentransformation

Werden die Phasentransformationen außer Acht gelassen und angenommen dass jede Phase willkürlich — ohne Abhängigkeit zu anderen Phasen — aufgrund der Thermodynamik keimen und wachsen, können mit den unzähligen Modifikationsparameter die MatCalc bietet, einzelne Arbeitspunkte angefahren und abgebildet werden. Diese Variante stellt aufgrund der Tatsache, dass Diffusionskoeffizienten mit Werten von über 40 verstärkt werden, die Gesamtmodellierung infrage. Des Weiteren werden unphysikalische Ergebnisse erhalten, sobald der Arbeitspunkt verlassen wird, wie beispielsweise das Erhöhen der Auslagerungstemperatur von 150 °C auf 180 °C. Es gibt eine Vielzahl weiterer Probleme, die auf die künstliche Erhöhung der Diffusion bestimmter Phasen zurückzuführen ist. Diesbezüglich wird diese Variante nicht weiter ausgeführt und in dieser Arbeit verfolgt.

#### Radian-Transformationsmodell

Das Radian-Transformationsmodell ermöglicht es eine Phasensequenz zu erstellen. Hierbei wird angegeben, welche Phase aus welcher hervortritt; es wird angemerkt, dass die Phasenabhängigkeit nicht realistisch sein muss. Das bedeutet, dass zum Beispiel die  $\beta$ -Phase zu einer Cluster-Phase oder einer kupferhaltigen  $\theta$ -Phase transformiert werden kann.

Die Umwandlung einer Phase in die nächste wird über Transformationsradien bestimmt. Das bedeutet, dass beispielsweise für die Umwandlung der Cluster-Phase in die  $\beta''$ -Phase zwei Radian angegeben werden müssen, die ein Intervall aufspannen, in dem die Transformation über eine lineare Wahrscheinlichkeitsfunktion gegeben ist; vergleiche Abbildung E.5. Hier ist zu beachten, dass das Modell nur die Transformation einer kleineren in eine größere Phasen — bezogen auf den Radius — modellieren kann. Mit anderen Worten: Die Umwandlung der Cluster-Phase in die  $\beta''$ -Phase ist möglich, jedoch kann der umgekehrte Prozess nicht durchgeführt werden.

Die Abbildung der Sequenz in dieser Arbeit folgt derjenigen aus Tabelle 3.3. Daraus folgt, dass die Cluster-Phase überall keimen darf, die  $\beta''$ -Phase aus der Cluster-Phase, die  $\beta'$  aus der  $\beta''$  und die  $\beta$  aus der  $\beta'$ -Phase hervortritt.

Die korrekte Phasensequenz während einer Wärmebehandlung kann damit abgebildet werden. Hierzu müssen die Transformationsradien so angepasst werden, sodass die numerischen

Ergebnisse mit den Messungen korrelieren.

Das Problem bei inhomogenen Wärmebehandlungen besteht allerdings darin, dass die Transformationsradien temperaturabhängig gestaltet werden müssen, um bei verschiedenen Temperaturen — jedoch gleichen MatCalc Einstellungen — die Realität korrekt abzubilden. Diesbezüglich ist es notwendig temperaturabhängige Transformationsradien zu erstellen. Diese Bestimmung dieser Radien wird nachfolgend prägnant beschrieben. Eine ausführliche Diskussion dieser Herleitung ist mit allen Verifikationen in Anhang E.2 auf Seite 195 gegeben und stellt eine signifikante Erweiterung des generellen Transformationsmodells dar.

### *Bestimmung der temperaturabhängigen Transformationsradien*

Die Transformationsradien jeder einzelnen Phase werden mit einer temperaturabhängigen Funktion approximiert. Zur Bestimmung dieser werden experimentelle Daten von Colley (2011) herangezogen und reproduziert. Hierzu werden die Programmskripte, die in Kapitel 3.6 vorgestellt werden, herangezogen.

Für jede Auslagerungstemperatur werden die Transformationsradien so angepasst, dass die experimentell erhaltenen Streckgrenzprofile  $R_{p,0.2}$  numerisch abgebildet werden. Es werden vier verschiedene Berechnungen durchgeführt. Dazu gehört sowohl das Auslagern bei Raumtemperatur 25 °C als auch bei höherer Temperatur; 150 °C, 180 °C und 200 °C. Für jede Auslagerungstemperatur bleiben alle in MatCalc gesetzten Einstellungen identisch. Lediglich die Transformationsradien werden justiert, um die experimentellen Datenverläufe von Colley (2011) nachzubilden.

Anschließend wird eine Analyse der erhaltenen Transformationsradien getätigt und eine Funktionsfindung durchgeführt, um die temperaturabhängigen Transformationsradien zu erhalten.

Die detaillierte Herleitung der temperaturabhängigen Transformationsradien mit allen Daten und Ergebnissen, ist in Anhang E.2 auf Seite 195 gegeben. Die identifizierten Gesetzmäßigkeiten lauten:

$$\text{Cluster} \longrightarrow \beta'' : r_{\text{trans},1} = AT^2 + BT + C , \quad (3.1)$$

$$\text{Cluster} \longrightarrow \beta'' : r_{\text{trans},2} = AT^2 + BT + C + \Delta_1 , \quad (3.2)$$

$$\beta'' \longrightarrow \beta' : r_{\text{trans},1} = AT^2 + BT + C + \delta_2 , \quad (3.3)$$

$$\beta'' \longrightarrow \beta' : r_{\text{trans},2} = AT^2 + BT + C + \delta_2 + \Delta_2 , \quad (3.4)$$

$$\beta' \longrightarrow \beta : r_{\text{trans},1} = AT^2 + BT + C + \delta_3 , \quad (3.5)$$

$$\beta' \longrightarrow \beta : r_{\text{trans},2} = AT^2 + BT + C + \delta_3 + \Delta_3 , \quad (3.6)$$

Die Koeffizienten A, B und C stellen die Polynomkoeffizienten mit den Werten:

$$A = -1,74959e^{-14} \quad [\text{m}^\circ\text{C}^{-2}] ,$$

$$B = -1,87199e^{-12} \quad [\text{m}^\circ\text{C}^{-1}] ,$$

$$C = 1,87756e^{-9} \quad [\text{m}] ,$$

dar. Die Größen  $\delta_i$  und  $\Delta_i$  sind Offset Parameter, welche die Grundgleichung (3.1) verschiebt.

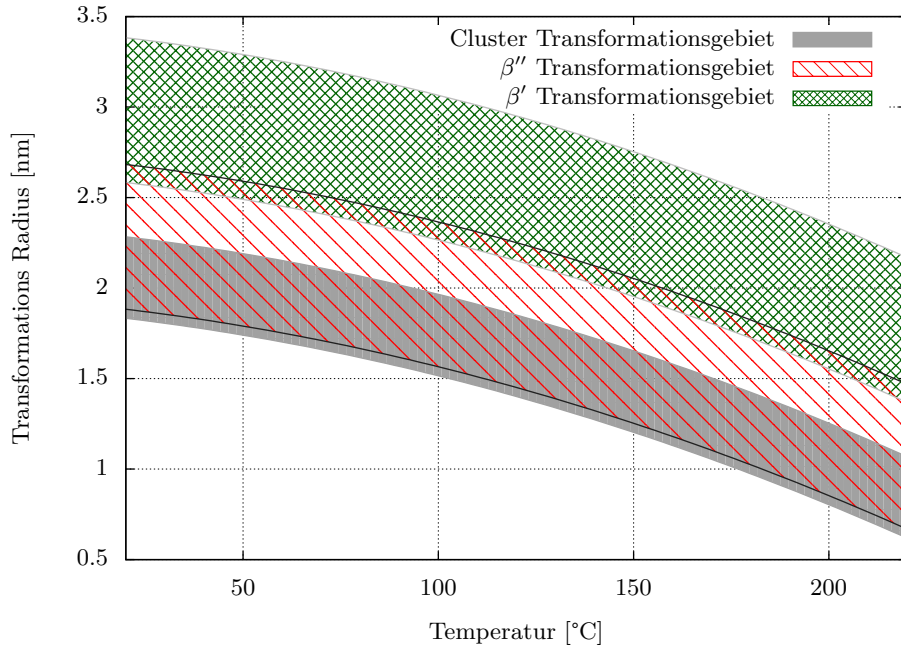


Abbildung 3.6.: Verwendete temperaturabhängige Transformationsradien zur Beschreibung der Phasensequenz. Die Legende bezieht sich auf die Umwandlung der angegebenen Phase in die nächste; das Cluster Transformationsgebiet gibt daher die Umwandlung der Cluster-Phase in die  $\beta''$ -Phase an.

Die identifizierten Werte lauten:

$$\delta_2 = 0,05e^{-9} \quad [\text{m}] , \quad \delta_3 = 0,70e^{-9} \quad [\text{m}] ,$$

$$\Delta_1 = 0,45e^{-9} \quad [\text{m}] , \quad \Delta_2 = 0,80e^{-9} \quad [\text{m}] , \quad \Delta_3 = 0,80e^{-9} \quad [\text{m}] .$$

Die visuelle Darstellung der Transformationsintervalle ist in Abbildung 3.6 gegeben und beschreibt die oben präsentierten Gleichungen. Es ist deutlich erkennbar, dass sich Umwandlungsgebiete überschneiden, wobei das Transformationsgebiet von  $\beta'' \rightarrow \beta'$  das Gebiet der Cluster  $\rightarrow \beta''$  Transformation fast komplett überdeckt. Daraus resultiert, dass Cluster,  $\beta''$  und  $\beta'$  in weiten Teilen koexistieren. Des Weiteren ist ersichtlich, dass große Cluster-Partikel (bezogen auf den Radius) kaum existieren, da diese rasch in die  $\beta''$ -Phase transformiert werden.

### Wichtige Anmerkung

Das vorgestellte Radien-Transformationsmodell ermöglicht es, unter Verwendung temperaturabhängiger Radien, lokale Wärmebehandlungen zu untersuchen. Werden beispielsweise feste Transformationsradien verwendet, dürfen sich Einstellungen für die Wärmebehandlung — Lösungsglühn und Auslagern — nicht verändern. Daher ist es zwingend erforderlich, temperaturabhängige Radien zu verwenden.

Des Weiteren soll angemerkt werden, dass die Bestimmung der Radiusfunktionen mithilfe der Auslagerungsanalyse von Colley (2011) durchgeführt wurde. Das Lösungsglühn wird hierbei nicht berücksichtigt. Es wird angenommen, dass die entwickelten Radienfunktionen auch für das Lösungsglühn Gültigkeit besitzt.

Dies ist dadurch begründet, dass es nicht möglich ist eine Reserve Phasensequenz in MatCalc zu modellieren. Mit anderen Worten, es besteht keine Möglichkeit beispielsweise die  $\beta''$ -Phase

in eine Cluster-Phase zu transformieren. Hinzu kommt, dass in der Literatur nicht postuliert wird, wie sich Phasen beim Lösungsglühen verhalten, da stets davon ausgegangen wird, dass sich während diesem Vorgang alle Phasen auflösen. Diese Voraussetzung trifft jedoch bei lokalen Wärmebehandlungen nicht zu.

## 3.6. MatCalc Skripte und Einstellungen

Die numerische Abbildung der in Kapitel 3.1 dargestellten Prozessphasen 1 bis 4 — vergleiche Abbildung 3.1 — wird analog mit vier neu entwickelten MatCalc Skripten umgesetzt, die nachfolgend diskutiert werden.

Alle Skriptdateien sind auf dem Datenträger unter `Doktorarbeit/MatCalc/Kinetics` abgelegt und können dort eingesehen werden. Die Korrelation der in Abbildung 3.1 dargestellten Sequenzen und den Skriptnamen lautet:

- Phase 1 → `Scheil.mcs`,
- Phase 2 → `Kinetics_1.mcs`,
- Phase 3 → `Kinetics_2.mcs`,
- Phase 4 → `Kinetics_3.mcs`.

Alle nachfolgenden Zahlenwerte sind aus Tabelle 3.4 entnommen oder stellen arithmetische Mittelwerte dieser dar. Sollten andere Werte verwendet werden, wird explizit die verwendete Literatur angegeben. Der Betrag der Formfaktoren ist angenommen, da in der Literatur keine Zahlenwerte postuliert werden. Jedoch wird die Gestalt bei der Auswahl berücksichtigt; stabförmig, nadelförmig und so weiter.

### 3.6.1. Scheil Skript

Das Scheil Skript berechnet mithilfe des Scheil-Gulliver Modells die Phasenanteile die aufgrund der Thermodynamik entstehenden Phasen und gibt diese an. Hierzu wird zu Beginn die Legierungszusammensetzung, welche aus Colley (2011) entnommen wird, eingestellt und alle auftretenden Phasen definiert. Die Elemente, die hier verwendet werden, sind Aluminium, Silizium, Magnesium, Eisen und Titan respektive mit einer Zusammensetzung von 7,42 gew.% Silizium, 0,3 gew.% Magnesium, 0,17 gew.% Eisen und 0,11 gew.% Titan.

In der Scheil-Gulliver Berechnung werden folgende Phasen für die Modellierung verwendet: `LIQUID`, `FCC_A1`, `SI_DIAMOND_A4`, `MG2SI_B`, `ALFESI_T6` und `PI_ALFEMGSI`. Sie repräsentieren das flüssige Nichteisenmetall, die Aluminiummatrix, reines Silizium, die  $\beta$ - sowie zwei eisenhaltigen Phasen. Die Auswahl der Phasen beruht auf einer Analyse, in der betrachtet wird, welche Phasen während des Erstarrens mit dieser Legierungszusammensetzung auftreten. Diese Untersuchung korreliert mit den Beobachtungen von Bäckerud et al. (1990) wird jedoch nicht präsentiert; vergleiche Kapitel 3.3.

Für die  $\beta$ - und Silizium-Phase wird anschließend — aufgrund des bekannten Phasenanteils — eine Ausscheidungsverteilung mit den in Kapitel 3.4 präsentierten Funktionen erstellt. Ferner wird die Typologie der Silizium-Phase als Platte und die der  $\beta$ -Phase als Nadeln betrachtet.

Die eisenhaltigen Phasen sind stabil und spielen in einer subsequenten Wärmebehandlung eine untergeordnete Rolle. Entsprechend werden beide Phasen aus dem System entfernt. Da

diese Aluminium, Eisen, Silizium und Magnesium binden, muss ein reduziertes System berechnet werden. Die neue Legierungszusammensetzung des reduzierten Systems lautet 7,34 gew.% Silizium, 0,274 gew.% Magnesium, 0,00 gew.% Eisen und 0,111 gew.% Titan. Die detaillierte Berechnung ist in Anhang E.3 auf Seite 204 gegeben. Es soll angemerkt werden, dass dieses Vorgehen beispielsweise von Falahati et al. (2011) nicht durchgeführt wird.

Abschließend werden die Ausscheidungsverteilungen und die neue Legierungszusammensetzung in verschiedene Dateien exportiert.

### 3.6.2. Kinetics\_1 Skript

Nachdem der Zustand des Materials nach der Erstarrung bekannt ist, wird mithilfe des Kinetics\_1 Skripts die weitere Materialbetrachtung durchgeführt. Die Abkühlung auf Raumtemperatur wird mit einem Temperaturprofil, welches mittels der Software MAGMASoft<sup>®</sup> bestimmt wird, modelliert; alle Einzelheiten hierzu sind in Anhang E.4 auf Seite 206 gegeben.

Zu Beginn werden erneut die möglichen Phasen, die während der Kinetikrechnung auftreten können, sowie die reduzierte Legierungszusammensetzung definiert. Anders als im Scheil Skript müssen diese Phasen als Ausscheidungsphasen definiert werden. Die ausgewählten Phasen lauten FCC\_A1, CL\_MGSI, AL\_B\_DP, MG\_SI\_B\_P, MG2SI\_B und SI\_DIAMOND\_A4. Die neu hinzugekommenen Phasen beschreiben die Cluster-,  $\beta''$ - und  $\beta'$ -Phase. Es ist anzumerken, dass die  $\beta''$ -Phase in MatCalc Aluminium bindet.

Alle Eingabeparameter werden aus Tabelle 3.4 entnommen. Standardeinstellungen die beim Laden der thermodynamischen und diffusiven Datenbanken in MatCalc verwendet werden, sind nicht explizit aufgeführt. Zum Einsatz kommen die Datenbanken aus der Beta Version 6.01.0003.

Des Weiteren werden alle Phasen mit 40 Radiusklassen definiert. Jede Klasse repräsentiert dabei eine entsprechende Anzahl von Ausscheidungen in einem gemittelten Radius.

Das Skript initialisiert zu Beginn alle Phasen und setzt die reduzierte Legierungszusammensetzung. Anschließend werden die Einstellungen an der Matrix und den Phasen vorgenommen sowie die Wärmebehandlung definiert; Temperaturkurve aus MAGMASoft<sup>®</sup>. Subsequent wird die Kinetikberechnung durchgeführt und der Zustand des Materials in einem MatCalc *State* eingefroren. Zuletzt werden alle Daten der Berechnung in eine Binärdatei geschrieben.

Die detaillierte Übersicht über die eingestellten Matrix- und Phaseneigenschaften sind den Skripten zu entnehmen. Die wichtigsten Aspekte werden jedoch nachfolgend gegeben.

#### Die Matrix

Die Matrixphase (FCC\_A1) wird mit einem temperaturabhängigen Elastizitätsmodul  $E(\vartheta) = (69220 - 40,1\vartheta)1e^6$  modelliert und wird aus den Voreinstellungen der MatCalc Datenbanken entnommen —  $\vartheta$  beschreibt die Temperatur in °C. Ferner wird die Querkontraktionszahl  $\nu$  auf 0,333 gesetzt. Die Versetzungsdichte im Aluminiumgitter wird mit  $5e^{12}$  [m<sup>3</sup>] angenommen und die Streckgrenze  $R_{p,0.2}$  der reinen Matrix mit einem Wert von 25 MPa definiert.

Für die Generierung und Vernichtung der Leerstellen wird das FSAK-Modell verwendet. Hinzu kommt, dass die Leerstellen durch Silizium- und Magnesiumatome sowie mit der Cluster-Phase besetzt und wieder freigegeben werden können. Dies ist unter der Nomenklatur des *Trapping* bekannt. Diesbezüglich sind Bindungsenthalpien notwendig, die respektive mit den Werten von 4370,38 Jmol<sup>-1</sup>, 2093,73 Jmol<sup>-1</sup> und 24000,00 Jmol<sup>-1</sup> gesetzt werden. Die Werte stellen Mittelwerte aus den experimentellen Untersuchungen von Özbilen und Flower (1989); Prabhu und

Howe (1990); Cheng et al. (1991); Wolverton (2007); Mantina et al. (2009); Lang et al. (2013) und Weisz et al. (2015) dar.

Des Weiteren wird die Matrixdiffusion mit einem Korrekturfaktor von 2,5 angepasst und die Korngröße auf den Wert von 296 nm abgeändert.

Die Berechnung der Materialfestigkeit wird mit den Standardeinstellungen durchgeführt, wobei die Ausscheidungsverfestigung mit dem erweiterten *Dislocation-Line-Tension* Modell und den Parametern des *innerCut-Off* ( $= 2 \cdot b$ ) und *outerCut-Off* ( $= 120$ ) eingestellt wird.  $b$  stellt den Burgers-Vektor dar und wird von MatCalc berechnet.

### Siliziumphase

Die Siliziumphase taucht nach der Erstarrung im Eutektikum auf und liegt als Ausscheidungsphase bereits vor. Die Verteilung wird nach der Scheil-Gulliver Berechnung erstellt. Da die Ausscheidungsverteilung des Siliziums im Eutektikum durch Schlibfbilder bei Raumtemperatur bestimmt worden ist, wird für die Abkühlung auf Raumtemperatur die Diffusion dieser Phase um den Faktor 0,001 verringert. Dies bewirkt, dass sich diese Phase während des Abkühlens nicht verändert. Da die Siliziumphase keinen Beitrag zur Materialverfestigung leistet, wird der Verfestigungsanteil dieser Phase ignoriert. Der Formfaktor wird auf 0,3 gesetzt. Die mechanischen Charakteristiken wie das E-Modul wird auf 107 GPa und die Querkontraktionszahl auf 0,22 gesetzt. Des Weiteren wird dieser Phase nicht erlaubt zu keimen.

Aufgrund der Tatsache, dass sich erneut reines Silizium als fein verteilte Ausscheidung bilden kann, wird eine zweite Siliziumphase mit den gleichen Einstellungen erstellt, wobei keine Ausscheidungsverteilung initialisiert und die Keimbildung an Versetzungen erlaubt wird.

### Cluster-Phase

Die Cluster-Phase wird mit einer volumetrischen Fehlanpassung von 0,03 eingestellt. Der verfestigende Beitrag wird mit dem Co-Cluster-Modell abgebildet, wobei die Bindungsenthalpie mit 24000,00 J festgelegt wird; vergleiche Cheng et al. (1991) sowie Starink et al. (2012). Da diese Phase sehr diffusiv ist, wird die Interface-Diffusions-Korrektur mit einer kritischen Temperatur von 600 K vorgegeben; dies stellt eine Erweiterung des GNNB dar. Um die Agglomeration der Phase zu modellieren, wird das Binder-Staufer Modell mit einem Vorfaktor von 1, einem Expansionskoeffizienten von  $-4/3$  und einem Alfa Wert von 1 vordefiniert; diese Werte werden durch die Nachbildung der Ergebnisse von Weisz et al. (2015) bestimmt. Des Weiteren wird die Diffusion durch die Gitterleerstellen erlaubt. Ein weiterer Parameter der in MatCalc als *Site-Efficiency* (SE) vorliegt und Einfluss auf das Nukleationsmodell nimmt, wird auf einen Wert von 800 gesetzt. Zuletzt wird der Keimungsort im gesamten Volumen definiert.

### $\beta''$ -Phase

Aufgrund der Tatsache, dass die Phasenumwandlung in MatCalc nur beim Auslagern eintritt, wird der  $\beta''$ -Phase das Keimen an Versetzungen erlaubt. Der volumetrische Misfitfaktor beträgt 0,052. Die Verfestigung wird mit dem Multi-Klassen-Strengthening-Modell bestimmt. Der Formfaktor wird auf 8 gesetzt und die SE wie bei den Clustern auf 800 erhöht. Die zusätzliche Diffusionserhöhung durch Vacancies wird aktiviert und das E-Modul auf 67 GPa angepasst.

### $\beta'$ -Phase

Ähnlich zur  $\beta''$ -Phase, wird der  $\beta'$ -Phase das Keimen an Versetzungen erlaubt. Der volumetrische Misfit wird auf 0,050 und der Formfaktor auf 7 gesetzt. Erneut wird die Diffusionserhöhung aufgrund von Gitterfreistellen, erlaubt. Wie zuvor wird das Multi-Klassen-Strengthening-Modell verwendet und die SE auf 800 erhöht.

### $\beta$ -Phase

Die stabile  $\beta$ -Phase liegt, wie die eutektische Silizium-Phase, nach dem Erstarren vor und wird aus dieser initialisiert. Der Formfaktor beträgt 0,3 und die Diffusion der Phase wird um den Faktor 0,1 verringert, sodass eine annähernd gleiche Ausscheidungsverteilung am Ende der Kinetikrechnung erhalten wird. Diese Phase darf während des Abkühlens auf Raumtemperatur an Versetzungen keimen und erfährt eine Diffusionserhöhung aufgrund der Leerstellen. Die volumetrische Fehlanpassung beträgt 0,03 und die SE wird wie bei allen anderen Phasen auf den Wert von 800 gesetzt.

### 3.6.3. Kinetics\_2 Skript

Das Kinetic\_2 Skript übernimmt den Berechnungsstatus der vorhergegangenen Simulation, indem die Binärdatei geladen wird. Durch diese Operation liegen alle Einstellungen, die im Kinetics\_1 Skript getätigt worden sind, erneut vor. Das Kinetics\_2 Skript wird für das Lösungsglühn herangezogen. Die Phaseneinstellungen bleiben identisch mit Ausnahme von:

- Die SE aller Phasen sowie die Diffusionsreduzierungen der  $\beta$ - und Silizium-Phase werden auf 1 zurückgesetzt,
- Die Möglichkeit das die  $\beta''$ -,  $\beta'$ - und  $\beta$ -Phase an Versetzungen keimen, wird deaktiviert, das Radien-Transformationsmodell ausgewählt und mit den entsprechenden temperaturabhängigen Polynomen definiert,
- Die kritische Temperatur für das Interface-Diffusionsmodell wird auf 650 K herabgesetzt.

Das Lösungsglühn wird mit einem Temperaturprofil, welches aus einer Datei gelesen wird, durchgeführt. Wie im Kinetics\_1 Skript wird der Materialzustand nach der Kinetikrechnung in einem *State* gespeichert und in eine Binärdatei geschrieben.

### 3.6.4. Kinetics\_3 Skript

Beim Auslagern wird der *State* aus der vorherigen Berechnung geladen und keine weiteren Änderungen durchgeführt. Die einzige Ergänzung besteht in der Erstellung des Temperaturprofils für die Auslagerung. Am Ende der Berechnung wird die aktuell vorliegende Streckgrenze  $R_{p,0.2}$  des Materials ausgegeben und kann für weitere Zwecke verwendet werden.

### 3.7. Extrapolation des Field-Average Approach's auf 3D Konfigurationen

Wie eingangs in Kapitel 3.2 erwähnt, kann die FAA Modellierung nur für 0D Problemstellungen angewendet werden. Da aufgrund des inhomogenen Energieeintrags diese Annahme nicht getroffen werden kann, werden drei Möglichkeiten, welche intensiv untersucht worden sind, diskutiert und deren Einsatzbereich aufgezeigt.

#### 3.7.1. Flächen-Gewichtetes Interpolationsverfahren

Eine Möglichkeit der Anwendung des FAA für inhomogene oder lokale Wärmebehandlungen in 3D stellt das Flächen-Gewichtete Interpolationsverfahren (FGI) dar. Diese Methode wird zu Projektbeginn verwendet und erlaubt es eine einfache Approximation der Materialeigenschaften in 3D mithilfe einzelner Punkte durchzuführen. Dieses Verfahren wird in den Publikationen von Ludwig und Holzmann (2016a) sowie Ludwig und Holzmann (2016b) angewendet. Die mathematischen Einzelheiten sind nachfolgend gegeben.

Sind die Materialeigenschaften an den Eckpunkten eines Rechtecks bekannt, ist es möglich, mithilfe der FGI die Materialeigenschaften für jeden beliebigen Punkt innerhalb dieses Rechtecks zu bestimmen. Abbildung 3.7 zeigt schematisch vier bekannte Eckpunkte auf, an denen die Werte von  $\phi_1, \phi_2, \phi_3$  und  $\phi_4$  bekannt sind. Der gesuchte Wert  $\phi^*$ , der über die Position  $\chi$  und  $\xi$  bestimmt ist, kann wie folgt ermittelt werden:

$$\phi^* = \frac{\phi_1 A_3 + \phi_2 A_4 + \phi_3 A_1 + \phi_4 A_2}{A_G} . \tag{3.7}$$

Die Flächen  $A_1$  bis  $A_4$  können unter Zuhilfenahme von Abbildung 3.7 hergeleitet werden und lauten:

$$A_G = (\chi_4 - \chi_1)(\xi_2 - \xi_1) , \tag{3.8}$$

$$A_1 = (\chi^* - \chi_1)(\xi^* - \xi_1) , \tag{3.9}$$

$$A_2 = (\chi^* - \chi_2)(\xi_2 - \xi^*) , \tag{3.10}$$

$$A_3 = (\chi_3 - \chi^*)(\xi_3 - \xi^*) , \tag{3.11}$$

$$A_4 = (\chi_3 - \chi^*)(\xi^* - \xi_4) , \tag{3.12}$$

Die Werte  $\xi_i$  und  $\chi_i$  stellen die Koordinaten an den entsprechenden Punkten (1, 2, 3 und 4) dar. Die Verwendung dieser beiden Symbole — anstelle von  $x$  und  $y$  — ist darin begründet, dass es nicht zwingend erforderlich ist, das FGI im kartesischen Koordinatensystem anzuwenden.

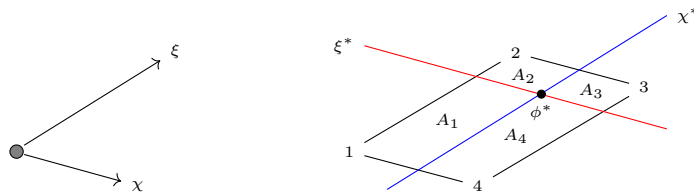


Abbildung 3.7.: Schematische Darstellung des Flächen-Gewichteten Interpolationsverfahrens im  $\chi$ - $\xi$  Koordinatensystem.



## Limitierung

Das Verfahren ist einfach, robust und leicht in einer numerischen Software zu implementieren. Allerdings ist aufgrund der starken Nichtlinearität der Materialeigenschaften — basierend auf dem inhomogenen Temperaturfeld — diese Methodik sehr ungenau. Abhilfe, zur Erhöhung der Genauigkeit, schafft die Einführung weiterer Stützpunkte. Hinzu kommt die Limitierung auf einfache Geometrien wie beispielsweise einer rotationssymmetrischen Platte. Das Verfahren ist auf 2D Fälle beschränkt. Für komplexe Bauteile kann dieses Verfahren nicht eingesetzt werden.

### 3.7.2. Datentabellierung (Look-Up-Table)

Die zweite Methodik, die akribisch untersucht wird, ist die Erstellung sogenannter Look-Up-Tables; die Nomenklatur stammt aus der Verbrennungslehre. Zur Prüfung dieser Methodik werden beliebige Temperaturprofile erzeugt, die anschließend in der MatCalc Berechnung verwendet werden, um damit die zeitlichen Verläufe der Streckgrenze  $R_{p,0.2}$  zu bestimmen. Anschließend wird versucht einen mathematischen Zusammenhang zwischen den Streckgrenzprofilen  $R_{p,0.2}(t)$ , den Temperaturprofilen  $T(t)$  und dessen Integral  $\int_{t_0}^t T dt$  zu finden, um diese subsequent zu tabellarisieren; vergleiche die Theorie zur Flammenflächenmodellierung mithilfe des stationären Flammenflächenmodells.

Der Tatsache geschuldet, dass keine eindeutige Zuweisung der Werte  $R_{p,0.2}(t)$  mithilfe von  $T(t)$  und  $\int_{t_0}^t T dt$  möglich ist, kann diese Methodik nicht angewendet werden. Die hochkomplexe Nichtlinearität, die die Materialberechnung aufweist, kann hierüber nicht wiedergegeben werden.

Es sollte jedoch der Versuch zur Umsetzung dieser Methode erwähnt werden.

### 3.7.3. Random-Cell-Poisson Methode

Zur Bestimmung der Materialeigenschaften in 3D wird als dritte Möglichkeit die Random-Cell-Poisson Methode (RCPM) vorgestellt, die während des Projekts entwickelt wurde und in Holzmann et al. (2018) vorgestellt wird.

Da die Berechnung von  $R_{p,0.2}$  über die FAA bestimmt wird, kann die Überlegung getroffen werden, jede numerische Zelle einzeln zu betrachten; vergleiche Kapitel 3.2 Abbildung 3.3 auf Seite 42. Da komplexe Geometrien eine hohe Dichte numerischer Zellen aufweisen, kann die Berechnung jeder diskreten Zelle nicht umgesetzt werden. Diesbezüglich wird abgeleitet, dass je weniger Volumina in der FAA betrachtet werden, desto ungenauer die Materialbestimmung in 3D ist, jedoch der rechentechnische Aufwand verringert wird.

Die RCPM basiert darauf, dass mit einer willkürlich getroffenen Anzahl an Volumina die Materialeigenschaft im gesamten Bauteil bestimmt werden kann. Da nur ein Teil der numerischen Zellen durch eine FAA betrachtet wird, fehlen die Information der Streckgrenze  $R_{p,0.2}$  in den restlichen Zellen. Zur Bestimmung dieser unbekanntenen Werte wird eine Poisson-Gleichung gelöst, wodurch die Verteilung in 3D erhalten wird.

Die Umsetzung dieser Methode ist wie folgt:

- Erstelle ein numerisches Netz, welches die Geometrie abbildet,
- Selektiere eine Anzahl  $k$  von Zellen, die für die FAA verwendet werden,
- Starte die lokale Wärmebehandlung und zeichne alle Temperaturprofile der  $k$  ausgewählten Zellen auf,

- Konvertiere alle Temperaturprofile in ein MatCalc lesbares Format,
- Führe die Materialberechnung (Phase 1 bis 4) mithilfe der FAA durch (MatCalc),
- Extrahiere die Streckgrenzwerte  $R_{p,0.2}$  nach der Materialberechnung für jede Position  $k$
- Löse eine Poisson-Gleichung  $\mathbf{Ax} = \mathbf{b}$ , wobei im Quellvektor  $\mathbf{b}$  an den Zellpositionen  $k$  der Wert der berechneten Streckgrenze  $R_{p,0.2}$  steht.

Die Verwendung der Poisson-Gleichung ist dadurch begründet, dass diese Funktion die Informationen der Quellterme — quantitative Werte der Streckgrenze  $R_{p,0.2}$  an den zuvor erstellten Punkten — durch Diffusion im gesamten Bauteil verteilt. Ferner wird die Interaktion der Quellterme untereinander berücksichtigt. Das Vorgehen korreliert mit der Bestimmung einer Temperaturverteilung in einem Bauteil, das explizite Quellterme aufweist (Poisson-Gleichung).

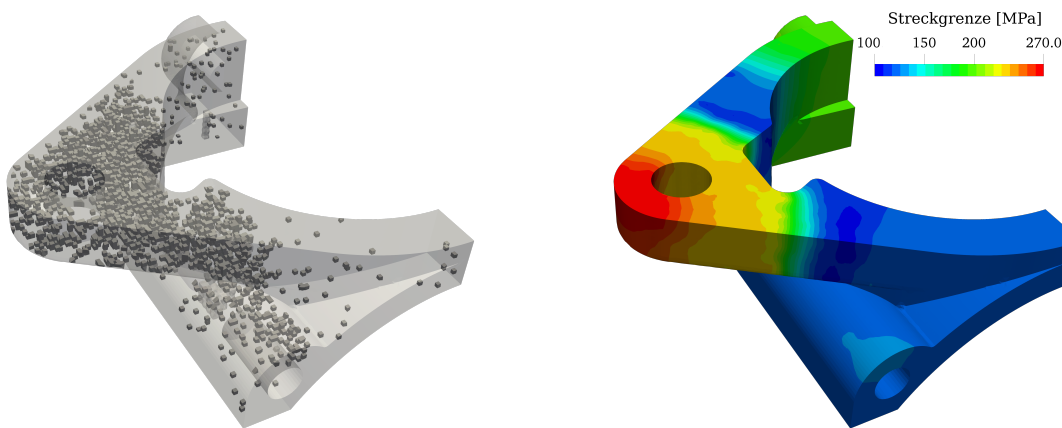
Die Random-Cell Generation wird mit dem neu entwickelten Programm `cellListGeneration` durchgeführt. Alle Einstellungsmöglichkeiten und Codeerklärungen sind in Anhang E.5 gegeben.

Ferner müssen die erzeugten Temperaturprofile aller Zellen in ein MatCalc lesbares Format konvertiert werden. Dies wird unter Zuhilfenahme des neu entwickelten Programm namens `MatCalcPreparation` getätigt.

Das Lösen der Poisson-Gleichung wird mit der neu programmierten Applikation namens `yieldStrengthLaplacianFoam` gelöst. Die Diskrepanz zwischen Poisson und Laplace rührt daher, dass das Gleichungssystem  $\mathbf{Ax} = \mathbf{b}$  in ein System der Form  $\mathbf{Ax} = 0$  überführt wird, in dem der Beitrag des Quellvektors direkt in die Matrix eingeht und programmiertechnisch begründet ist.

Alle Programme sind in C++ mit der `namespace FOAM` entwickelt und umgesetzt worden und stehen auf dem Datenträger unter `Doktorarbeit/Programming` zur Verfügung.

Zur Verdeutlichung der RCPM ist in Abbildung 3.8 eine willkürliche Random-Cell Verteilung und eine Berechnung der Materialfestigkeit  $R_{p,0.2}$  in 3D gegeben.



(a) Random-Cell Verteilung

(b)  $R_{p,0.2}$  in 3D

Abbildung 3.8.: Darstellung einer Random-Cell Verteilung und der zum Schluss erhaltenen Materialeigenschaften in 3D nach einer lokalen Wärmebehandlung mit anschließendem Warmauslagern; beide Bilder sind bereits in Holzmann et al. (2018) veröffentlicht.

## 4. Optimierungsverfahren

Folgendes Kapitel diskutiert unterschiedliche Verfahren und Algorithmen die in der freien Software DAKOTA<sup>®</sup> enthalten sind. Dabei werden eigene Funktionen analysiert und die verschiedenen Methoden angewendet sowie aufgezeigt wie diese arbeiten. Die Verwendung von DAKOTA<sup>®</sup> erfordert das Generieren einer Interface-Schnittstelle, welche im weiteren Verlauf nicht betrachtet wird. Die in diesem Abschnitt erbrachten Eigenleistungen bestehen in der Erstellung der Schnittstelle und den Skripten zur Berechnung der Funktionen.

### 4.1. Kategorisierung der Verfahren

Optimierungsverfahren werden stets dann verwendet, wenn lokale oder globale Minima oder Maxima nicht analytisch bestimmt werden können. Dies ist immer dann der Fall, wenn die zu evaluierende Funktion  $f$  — auch als Objektfunktion bezeichnet — nicht vorliegt. Optimierungsverfahren versuchen mittels unterschiedlichen Verfahren das Minimum oder Maximum jener Funktion zu finden, indem die Funktionsparameter verändert werden und der Funktionswert  $f$  bestimmt wird. Je nach Algorithmus werden die Parameter mehrere Male variiert, sodass ein Satz von Parameter-Antwortfunktion vorliegt. Diese Daten werden anschließend analysiert und ein neuer Satz von Parametern verwendet, um die Funktion  $f$  erneut zu berechnen. Es werden zwischen Dimensionsanalyse-Verfahren und Optimierer unterschieden. Beispiele für beide werden nachfolgend gegeben.

#### Dimensionsanalyse-Verfahren

Die Verwendung von Dimensionsanalyse-Verfahren ermöglicht es einen Parameterraum zu evaluieren, um somit die entsprechende Antwortfunktion zu erhalten.

Je nach verwendetem Algorithmus wird der Parameterraum entsprechend diskretisiert. Wird beispielsweise eine Multidimensionsanalyse der Funktion  $f(x, y) = x^3 + xy - 3y^2$  durchgeführt, werden die Parameter  $x$  und  $y$  in definierten Grenzen unter Verwendung einer gegebenen Anzahl von Stützstellen äquidistant eingeteilt und die Funktion  $f(x, y)$  für jedes Parameterpaar  $x, y$  berechnet. Diese Variante ist nützlich, wenn die Berechnung der Funktion  $f(x, y)$  schnell durchführbar ist. Die erhaltenen Ergebnisse können — abhängig von der Parameteranzahl — visualisiert und damit eine qualitative und quantitative Aussage bezüglich der Parameterabhängigkeit getroffen werden. Bekannte Verfahren in diesem Bereich sind die

- Multidimensionsanalyse,
- Vektoranalyse,
- Centered-Parameteranalyse,
- List-Parameteranalyse,

- Samplinganalyse.

Die Verfahren sind in Adams et al. (2015) näher beschrieben.

## Optimierungs-Verfahren

Optimierungsverfahren werden dann eingesetzt, wenn die Funktion  $f$  rechenintensiv zu ermitteln ist und ein Minimum oder Maximum der Funktion gesucht wird. Hierzu gibt es eine Vielzahl von Algorithmen, die diverse Ideen verfolgen und für unterschiedliche Problemstellungen entwickelt worden sind. Beispielsweise sind die Gradientenverfahren kostengünstig, wenn ein Minimum oder Maximum gefunden werden soll. Da die Funktion  $f$  nicht bekannt ist, enthalten Gradientenverfahren in ihrer allgemeinen Form die Schwachstelle, dass nur das Minimum oder Maximum nahe dem Startwert gefunden wird. Bei komplexen funktionalen Zusammenhängen wird hierbei oftmals nur der lokale, jedoch nicht der globale Extrempunkt gefunden. Des Weiteren werden diese Verfahren numerisch aufwendiger, wenn sich die Anzahl der abhängigen Funktionsparameter erhöht.

Für komplexere Problemstellungen werden Algorithmen verwendet, die mit Populationen oder Fitting-Algorithmen arbeiten. Bekannte Verfahren im Bereich der Optimierung sind:

- Klassische Gradientenverfahren,
- Methoden für Ungewisse Parameterstudien,
- Chaotische Polynomexpansion,
- Genetische Algorithmen; beispielsweise Surrogate-Modelle.

## 4.2. Algorithmen und Beispiele

Nachfolgend werden die Dimensionsanalyseverfahren anhand verschiedener Funktionen  $f_n$  beschrieben und diskutiert. Die oben aufgeführten Verfahren werden anschließend aufgezeigt und die Besonderheiten gegeben.

Die verwendeten Formulierungen lauten:

$$f_1(x) = x^2, \tag{4.1}$$

$$f_2(x) = x^4 - x^3 - 4x^2 + 4x, \tag{4.2}$$

$$f_3(x) = -x^2 - 2,5x + 3, \tag{4.3}$$

$$f_4(x) = 0,7x^3 + 3x^2 - 5x - 4, \tag{4.4}$$

$$g(x) = 0,4f_2(x) + 0,5f_3(x) + 0,1f_4(x). \tag{4.5}$$

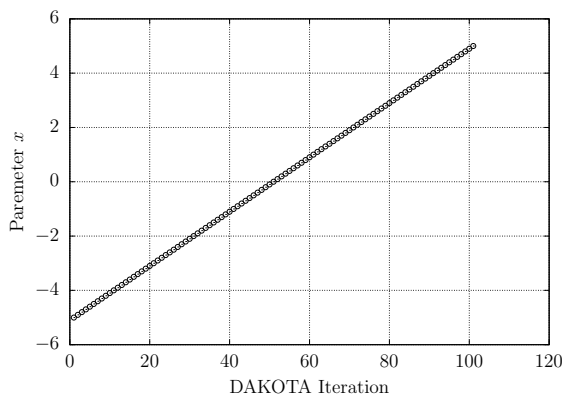
Alle präsentierten Beispiele sind auf dem Datenträger unter **Doktorarbeit/Dakota** zu finden. Die Berechnung, Auswertung und Grafikerstellung erfolgt automatisiert.

### 4.2.1. Multidimensionsanalyse

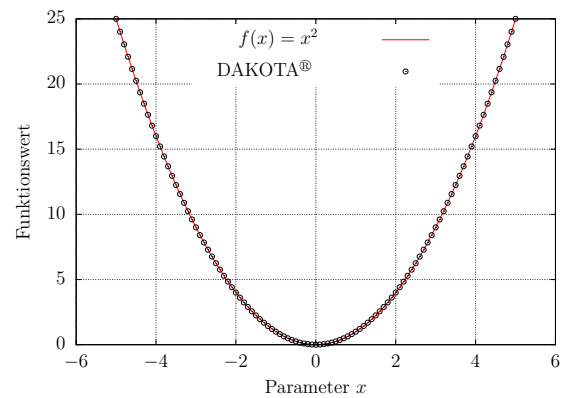
Die Multidimensionsanalyse der Funktion  $f_1(x)$ ,  $f_2(x)$  und  $g(x)$ , ist in Abbildung 4.1 gegeben.  $g(x)$  beschreibt dabei eine gewichtete Summenfunktion aus  $f_2(x)$ ,  $f_3(x)$  und  $f_4(x)$ . Die Durchführung der Dimensionsanalyse wird stets mit den gleichen Einstellungen durchgeführt.

Anhand der Grafik 4.1 ist ersichtlich, dass die Multidimensionsanalyse die genauen Werte der Funktionen an den diskreten Stellen berechnet. Dadurch ermöglicht diese Vorgehensweise, die Einflussnahme der Parameter auf den Funktionswert zu beurteilen. Bei jeder Iteration wird der neu zu evaluierende Funktionswert  $f$  an der Stelle  $x$ , welcher äquidistant im vordefinierten Raum beschrieben ist, berechnet. Die Genauigkeit der Approximation der wahren Funktion ist abhängig von der Gesamtanzahl der Stützstellen.

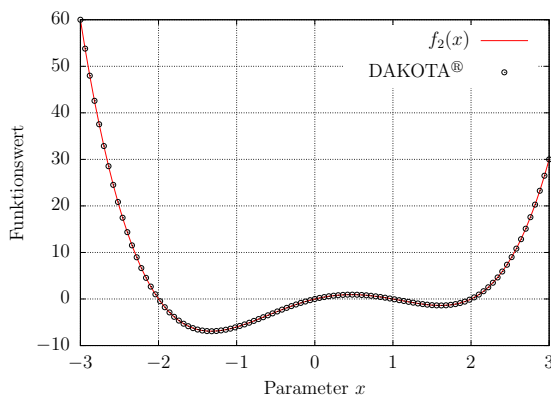
Es soll angemerkt werden, dass diese Beispiele einfach gehalten sind, sodass die Arbeitsweise des Algorithmus aufgezeigt werden kann. Sind Funktionen von  $m$  Parametern abhängig, erzeugt die Multidimensionsanalyse eine  $m$  dimensionale Funktionsabhängigkeit. Für  $m = 2$  wird eine 2D Oberfläche im 3D Raum erhalten. Für  $m > 2$  ist eine einfache visuelle Darstellung der Abhängigkeit des Funktionswertes von den Parametern nicht mehr gegeben.



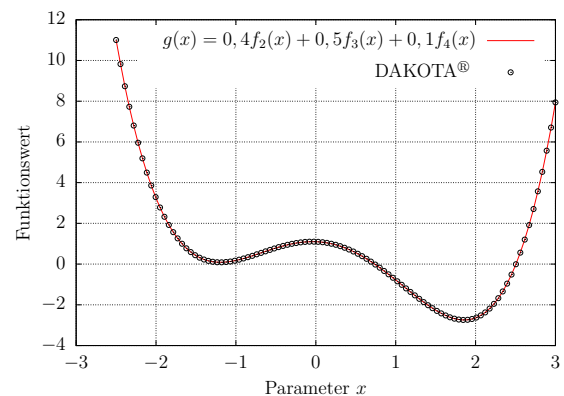
(a) Parameteränderung von  $x$



(b) Funktionsanalyse  $f_1(x)$



(c) Funktionsanalyse  $f_2(x)$



(d) Funktionsanalyse  $g(x)$

Abbildung 4.1.: Multidimensionsanalyse verschiedener Funktionen. Durchführung mithilfe der Software DAKOTA<sup>®</sup>.

### 4.2.2. Vektoranalyse

Die Vektoranalyse ist ähnlich zur Multidimensionsanalyse. Der Unterschied besteht darin, dass bei der Multidimensionsanalyse der Parameterraum  $x$  äquidistant eingeteilt wird und bei der Vektoranalyse eine Einteilung (als Vektorarray) vorgegeben werden muss. Die Dichte der diskreten Punkte kann auf diese Weise — durch den Anwender — an lokalen Positionen erhöht werden, um beispielsweise interessierende Regionen mit mehr Stützstellen zu versehen.

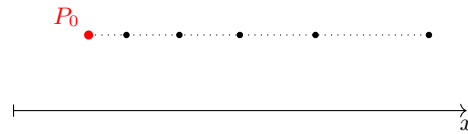


Abbildung 4.2.: Schematische Darstellung der Vektoranalyse für eine eindimensionale Funktion.  $P_0$  stellt den Startpunkt dar.

### 4.2.3. Centered-Parameteranalyse

Die Centered-Parameteranalyse kommt zumeist bei Funktionen zum Einsatz, die eine zweidimensionale Abhängigkeit aufweisen —  $f(x, y)$  — kann aber auch für willkürliche Dimensionsränge angewendet werden. Ausgehend von einem gegebenen Punkt  $P_0$ , evaluiert dieser Algorithmus die Punkte entlang der Parameterachsen. Der Benutzer gibt hierbei die Anzahl der zu evaluierenden Schritte und deren Schrittweite vor.

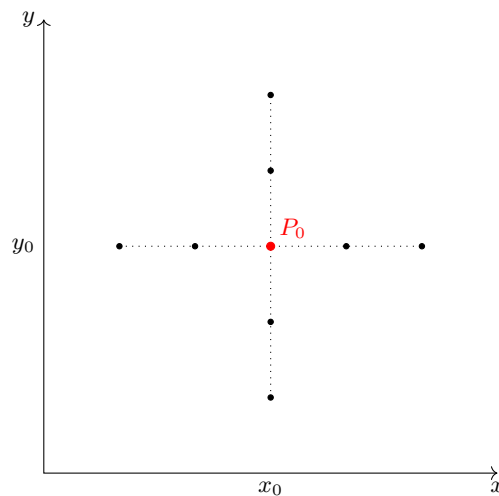


Abbildung 4.3.: Schematische Darstellung der Centered-Parameteranalyse für eine zweidimensionale Funktion. Ersichtlich ist der Initialpunkt  $P_0(x_0, y_0)$ , von dem eine Analyse entlang den Ordinatenachsen getätigt wird.

### 4.2.4. List-Parameteranalyse

Bei der List-Parameterstudie gibt der Anwender definierte Punkte vor, die der Algorithmus beim Starten automatisch durchläuft. Diese Variante ist simultan zur Vektoranalyse.

### 4.2.5. Samplinganalyse

Die Samplinganalyse stellt die generellste Form der Diskretisierung des Parameterraums dar. Ausgehend von einem Samplingalgorithmus, der vom Anwender ausgewählt werden muss — beispielsweise die **Random** oder **Hyper-Cube-Sampling** Methode —, wird der Parameterraum mit willkürlichen Punkten diskretisiert. Jeder Punkt wird subsequent evaluiert und das Ergebnis ausgegeben. Sampling-Verfahren werden generell bei genetischen Algorithmen eingesetzt, um zum Beispiel Population zu erstellen.

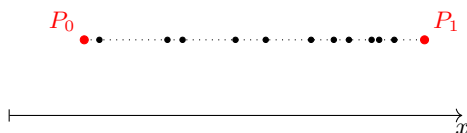


Abbildung 4.4.: Schematische Darstellung der Samplinganalyse für eine eindimensionale Funktion.  $P_0$  und  $P_1$  stellt jeweils den Start- und Endpunkt des Parameterraums dar.

## 4.3. Optimierungsverfahren

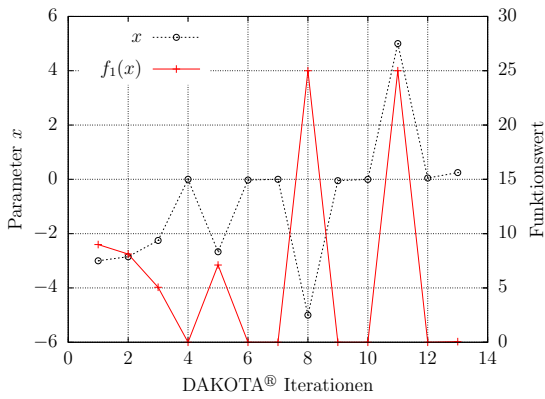
Nachfolgend werden die verschiedenen Optimierungsvarianten näher betrachtet. Auf die Methoden der ungewissen Parameterstudien und chaotischen Polynomexpansion wird nicht eingegangen. Zur Analyse der Funktionsweise der Algorithmen werden die gleichen Funktionen, wie im vorhergegangenen Kapitel, verwendet.

### 4.3.1. Gradientenbasierte Verfahren

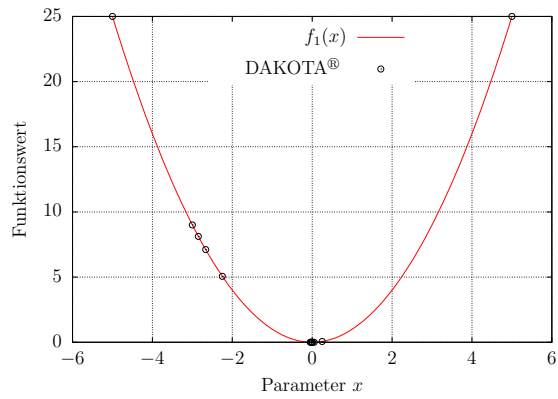
Gradientenbasierte Verfahren werden für Optimierungsproblemen verwendet. Im Folgenden wird das klassische Verfahren nach Fletcher und Reeves (1964) betrachtet. In eindimensionalen Problemstellungen, wird vom Startwert  $x_0$  ein entsprechender Betrag  $h$  addiert und subtrahiert, wodurch die Stützstellen  $x_{0,1}$  und  $x_{0,2}$  erzeugt werden. Mithilfe dieser drei Werte kann der numerische Gradient bestimmt werden. Über die Fletcher-Reeves (FR) Formulierung wird anschließend der neue zu evaluierende Punkt  $x_1$  gesucht und der Algorithmus beginnt von vorn.

Auf Funktionen, die nur ein Minimum oder Maximum aufweisen, können diese Algorithmen ohne weitere Modifikation eingesetzt werden; vergleiche Abbildung 4.5a und 4.5b. Es soll angemerkt werden, dass der Algorithmus den besten Wert am Ende der Optimierung ausgibt. Im vorliegenden Fall, die Evaluierung mit der Nummer 10;  $x_{10} = -4,3035e^{-8}$  und dem Funktionswert  $f_1(x_{10}) = 0$ .

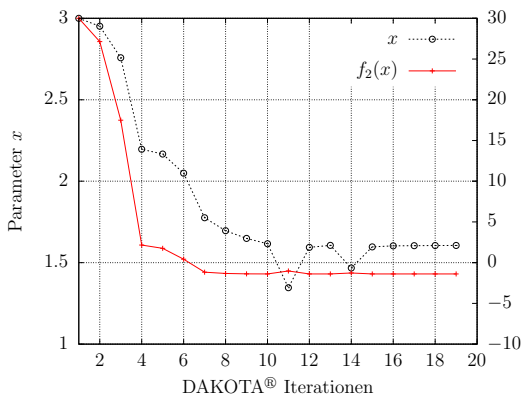
Wird hingegen eine Funktion herangezogen, die mehrere Minima oder Maxima aufweist, liefert das Gradientenverfahren stets den, vom Startpunkt, nächsten Extrempunkt. Zur Verdeutlichung wird Funktion  $f_2(x)$  herangezogen. Beginnend mit einem Startwert  $x_0 = 3$  wird das lokale Minimum auf der rechten Seite erhalten. Wird der Initialpunkt auf  $x_0 = -3$  abgeändert und die Optimierung erneut durchgeführt, wird das globale Minimum erhalten. Das globale Minimum, ausgehend von  $x_0 = -3$ , wird in der 16. Evaluierung bei  $x_{16} = -1,3258$  mit dem Funktionswert von  $f_2(x_{16}) = -6,9141$  und das lokale Minimum, beginnend bei  $x_0 = 3$ , in der 13ten Optimierungsschleife bei  $x_{13} = 1.6063$  und einem Minimalwert von  $f_2(x_{13}) = -1,3827$  gefunden.



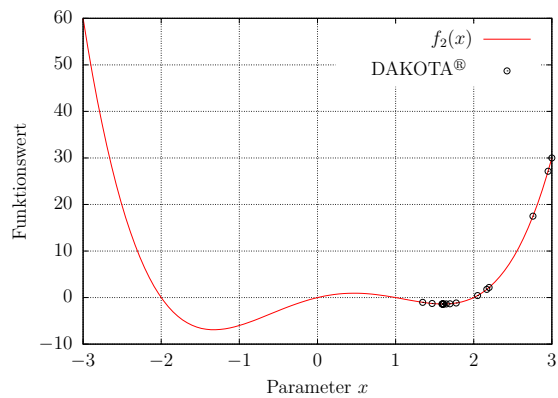
(a) Wertänderung von  $x$  und  $f_1(x)$



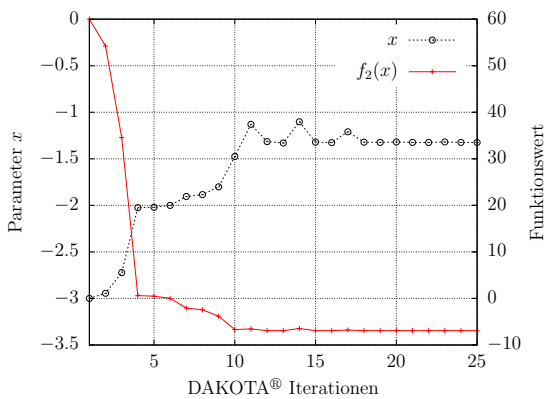
(b) Funktionswert  $f_1(x)$



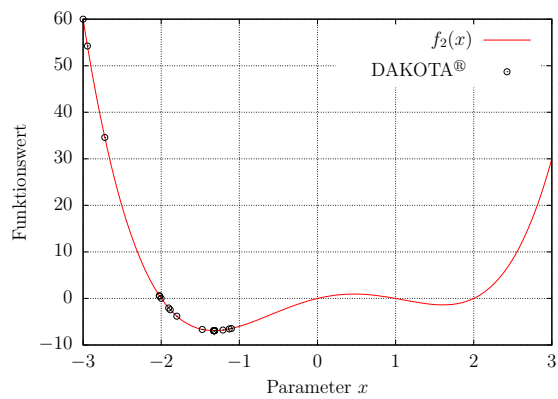
(c) Startpunkt  $x_0 = 3$



(d) Startpunkt  $x_0 = 3$



(e) Startpunkt  $x_0 = -3$



(f) Startpunkt  $x_0 = -3$

Abbildung 4.5.: Gradientenoptimierung an der Funktion  $f_1(x)$  (a), (b) und  $f_2(x)$  (c) bis (f). Die Funktion  $f_2(x)$  wird mit zwei verschiedenen Startpunkten optimiert. (c) und (d) korrespondieren zum Startpunkt  $x_0 = 3$ ; (e) und (f) gehören zum Startwert  $x_0 = -3$ .

### 4.3.2. Genetische Algorithmen

Die genetischen Algorithmen arbeiten zumeist mit Evolutionstechniken, die Cross-Over- und Mutationsmechanismen einsetzen, um die Problemstellung zu lösen. Zu Beginn wird eine Anfangspopulation im Parameterraum, die allgemein durch ein Samplingverfahren erzeugt wird, generiert. Jeder Punkt wird anschließend evaluiert. Der beste Funktionswert wird subsequent herangezogen, um die neue Population zu generieren; anschließend wiederholt sich der Algorithmus.



Daraus folgt, dass jede Paarung (Parameterkombination) eine Charakteristik des Problems darstellt, wobei jede Kombinationen als Chromosom bezeichnet wird. Die neue Population wird mithilfe der Cross-Over- und Mutationsmethode unter Verwendung der alten Population generiert.

Bei der Cross-Over-Methode werden die Chromosomen der vorherigen Population verwendet, um eine neue Paarung zu generieren und stellt damit eine Art *Vermischung* der Elternpaare dar.

Um zu verhindern, dass die Populationsevolution in ein lokales Minimum oder Maximum läuft, wird zusätzlich ein Mutationsmechanismus zum Algorithmus hinzugefügt. Dieser ändert Chromosomen willkürlich ab und kann dabei alle Werte im Parameterraum annehmen.

Welche Methode zur Erstellung der neuen Chromosomen verwendet wird, ist durch eine Wahrscheinlichkeitsfunktion, die durch einen prozentualen Eintrittswert definiert ist, gegeben.

Nachfolgend werden zwei genetische Algorithmen aufgezeigt. Es soll erwähnt werden, dass genetische Algorithmen vor allem dann zum Einsatz kommen, wenn die Antwortfunktion von einer Vielzahl von Parametern abhängig ist, Antwortfunktionen ein Rauschen aufweisen und/oder mehrere Extrempunkte erwartet werden.

### **Der Single-Objective-Generic-Algorithmus**

Der Single-Objective-Generic-Algorithmus (SOGA) wird dann verwendet, wenn nur *eine* Antwortfunktion optimiert werden muss. Liegen mehrere Objektfunktionen vor, wird mit einem MOGA gearbeitet (M = Multi). Der Algorithmus selbst kann mit einer Reihe von Parametern justiert werden, welche in Adams et al. (2015) gegeben sind. Allgemein wird beim Start der Methode, basierend auf einem ausgewählten Random-Generations-Algorithmus, die erste Population erstellt. Diese beinhaltet willkürliche Punkte, die anschließend evaluiert werden. Nachfolgend wird das beste Wertepaar (Chromosom) herangezogen, um in dieser Gegend eine neue Population mithilfe des Cross-Over- und Mutationsmechanismus zu generieren. Abschließend wiederholt sich der Algorithmus, bis die Abbruchkriterien erreicht werden; beispielsweise die maximale Anzahl an Populationen.

Wird der Algorithmus beispielsweise auf das Polynom 4ter Ordnung,  $f_2(x)$ , angewendet, werden die in Abbildung 4.6 dargestellten Ergebnisse erhalten. Aufgrund der Tatsache, dass genetische Algorithmen mit Wahrscheinlichkeitsfunktionen arbeiten, ist jede erneute Durchführung mit anderen Populationen verknüpft.

Die Darstellungen 4.6a und 4.6b geben die Änderung des Parameters  $x$  und der Funktion  $f_2(x)$ , bezogen auf die Evaluationen, an. Eine Evaluation repräsentiert dabei eine einzelne Berechnung. Es ist ersichtlich, dass eine Vielzahl von Chromosomen beibehalten wird, aber Abwandlungen davon auftauchen. Explizit in Abbildung 4.6a ist ersichtlich, dass der Algorithmus bis zur ungefähr 280sten Evaluation beide Minima berücksichtigt, aber aufgrund des deutlich kleineren Funktionswertes bei  $x \approx -2,2$  im weiteren Verlauf das lokale Minimum bei  $x \approx 2$  vernachlässigt.

Abbildung 4.6c zeigt die besten Funktionswerte jeder einzelnen Population auf, wobei bereits nach der elften Population, das globale Minimum gefunden wird. Die Berechnung 20 weiterer Populationen ist aufgrund des Abbruchkriteriums bedingt; es müssen mindestens 20 aufeinanderfolgende Populationen den gleichen Extremwert besitzen.

Abbildung 4.6d gibt einen Überblick über die berechneten Funktionswerte, bezogen auf die

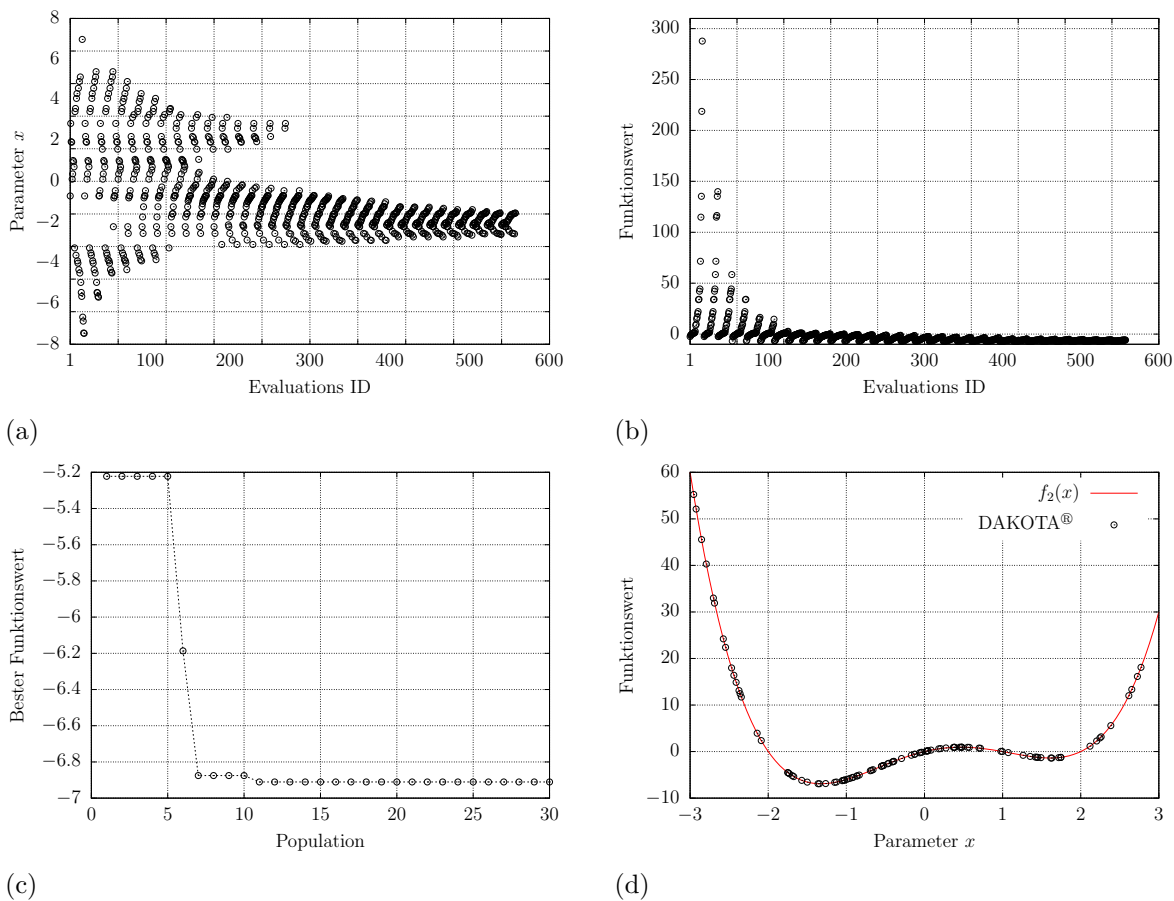


Abbildung 4.6.: SOGA Algorithmus aus der Familie der genetischen Methoden. (a) Variation des Parameters  $x$  für alle Evaluationen; (b) Variation des Funktionswertes für jede Evaluation; (c) Bester Funktionswert pro Population; (d) Realer Funktionsverlauf und diskrete Werte.

kontinuierliche Funktion  $f_2(x)$ .

Es ist deutlich ersichtlich, dass der SOGA Algorithmus das globale Minimum der Funktion  $f_2(x)$  findet. Allerdings ist auch darauf hinzuweisen, dass im Gegensatz zu den gradientenbasierten Verfahren wesentlich mehr Funktionsevaluierungen durchgeführt werden müssen. Im vorliegenden Fall benötigt SOGA circa 560 und der FR circa 16 Evaluationen.

### Genetischer Gauß Algorithmus

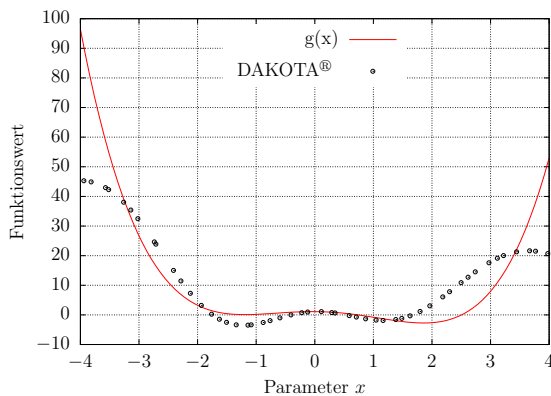
Eine andere Methode wie beispielsweise der genetische Gauß Algorithmus arbeitet mit einem komplexen Fitting-Prozedere. Hier wird versucht, mit einer komplexen Funktion (Trendfunktion) die tatsächliche Objektfunktion mit möglichst wenigen Stützstellen nachzubilden. Dies ist vor allem dann vorteilhaft, wenn die numerische Berechnung einer Stützstelle aufwendig ist. Diesbezüglich werden nur einige Stützstellen evaluiert und mit diesen Informationen die komplexe Funktion berechnet. Diese Funktion wird anschließend für die weitere Optimierung herangezogen, wodurch das Evaluieren neuer Stützstellen vermieden wird. Die Folge ist eine Reduzierung des numerischen Aufwands.

Die komplexe Trendfunktion besteht aus dem Kriging Emulator und einer Gauß'schen Fehlerreduktionsfunktion. Weitere Informationen diesbezüglich sind in Sacks et al. (1989); Santner et al. (2003); Rasmussen und Williams (2003) sowie Adams et al. (2017) gegeben.

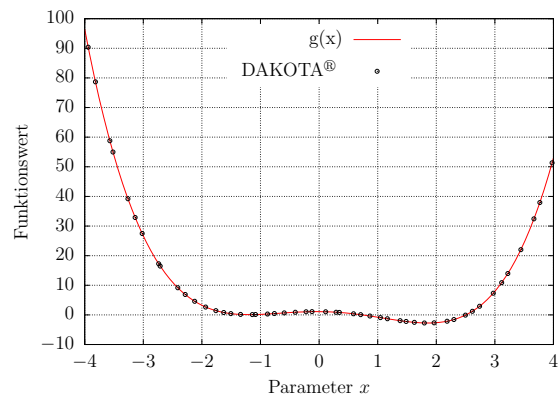
Wird die Funktion  $g(x)$  herangezogen, vergleiche Abbildung 4.1d, und die Optimierung mit einem genetischen Gauß Algorithmus durchgeführt, werden die Ergebnisse, welche in Abbildung 4.7 präsentiert sind, erhalten.

Es ist deutlich ersichtlich, dass die Trendfunktion bereits mit sieben Stützstellen — 21 Evaluation der Antwortfunktion — abbildbar ist. Dies ist der Grund weshalb sich diese Methode großer Popularität erfreut. Die Abbildung mit fünf Punkten hingegen reicht nicht aus, um die Funktion  $g(x)$  akkurat nachzubilden.

Werden fünf Stützstellen verwendet, berechnet der Algorithmus das Minimum bei  $x_{\text{opt}} = -1,1404$  mit einem Funktionswert  $g(x_{\text{opt}}) = -3,4252$ . Die genauere Lösung wird mit sieben diskreten Punkten erhalten und lautet  $x_{\text{opt}} = 1,7968$  wobei eine Antwortfunktion von  $g(x_{\text{opt}}) = -2,726$  bestimmt wird.



(a) Fünf Stützstellen



(b) Sieben Stützstellen

Abbildung 4.7.: Genetischer Gauß Algorithmus zur Approximation der Funktion  $g(x)$  mit mit fünf und sieben Stützstellen.

## 4.4. Abschließend

Wie aus den vorhergegangenen Darstellungen einzelner Analyse- und Optimierungsalgorithmen ersichtlich ist, gibt es eine Reihe von Methoden, die eingesetzt werden können, um lokale oder globale Extremwerte zu finden oder eine Parameterstudie durchzuführen. Die Auswahl des verwendeten Verfahrens ist problemabhängig und kann nicht pauschalisiert oder vorgeben werden. Eine Entscheidungshilfe kann die Antwortfunktion — falls diese annähernd bekannt ist — darstellen. Zudem ist die Berechnungszeit der numerischen Stützpunkte eine Quantität, die bei der Auswahl der Optimierer herangezogen werden kann/muss.



## 5. Ergebnisse

Folgendes Kapitel geht auf die lokale Wärmebehandlung und die Berechnung der Streckgrenze  $R_{p,0.2}$  in Strukturbauteilen ein. Zu Beginn wird die entwickelte Prozesskette näher gebracht und deren Funktionsweise erklärt.

Nachdem die Funktionsweise der neuen Prozesskette diskutiert wurde, werden thermische Spannungsberechnungen zweier Strukturbauteile vorgestellt und auf die dynamische Justierung des Energieeintrags, welcher über thermisch induzierte Spannung und/oder die Maximaltemperatur gesteuert wird, eingegangen und aufgezeigt, welche Erkenntnisse aus den Ergebnissen gewonnen werden. Ferner wird untersucht welchen Einfluss das Abschrecken — Selbstabkühlung oder Abschrecken in Wasser — auf die thermisch induzierte Maximalspannung besitzt.

Nach diesen Analysen wird eines der Strukturbauteile für eine Materialberechnung herangezogen. Hierbei wird zum einen die Streckgrenze  $R_{p,0.2}$  nach dem Lösungsglühen und zum anderen nach einem subsequenten Warmauslagern in 3D bestimmt. Des Weiteren wird der Einfluss des Abkühlens auf die Streckgrenze  $R_{p,0.2}$  nach dem Lösungsglühen diskutiert. Erneut wird die Selbstabkühlung und das Abschrecken in Wasser betrachtet.

Anschließend werden vier unterschiedliche lokale Wärmebehandlungen betrachtet (Dauer des Energieeintrags). Für jede einzelne Wärmebehandlung wird das Auslagern mit einem Dimensionsanalyseverfahren untersucht, um festzustellen welchen Einfluss die Auslagerungsparameter (Zeit und Temperatur) auf die Materialeigenschaften besitzen. Für die Bestimmung der Festigkeitswerte für die Multi-Dimensionsanalyse wird eine volumengewichtete integrale Festigkeit um die lokale Lösungsglühzone berechnet. Die genaue Definition dieser Region wird an gegebener Stelle gegeben. Hinzu kommt die Untersuchung beider zuvor erwähnten Abkühlvarianten, wodurch acht verschiedene Zustände für das Auslagern erzielt werden.

Die erhaltenen Informationen der Dimensionsanalysen werden abschließend dazu verwendet zwei Optimierungsberechnungen, welche mit einem genetischen Algorithmus durchgeführt werden, zu verifizieren.

Abschließend wird die Flexibilität der Prozesskette aufgezeigt. Hierbei wird auf die Modellierung des Zusammenschweißens zweier Platten (Vergütungsstahl 50CrMo4) kurz eingegangen und die Ergebnisse diskutiert.

## 5.1. Prozesskette

Die Modellierung der lokalen Wärmebehandlung in Kombination mit einer Materialberechnung und einem optionalen Optimierer, erfordert die Programmierung einer Prozesskette, die alle einzelnen Themengebiete, die für diese Betrachtung notwendig sind, verbindet. Hierzu werden die in Kapitel 2 bis 4 vorgestellten Arbeitsbereiche verknüpft. Die Umsetzung der Schnittstellen zwischen den einzelnen Programmen erfolgt durch die C++ Programmiersprache oder über Skripte, welche in der Linux Bourne Shell entwickelt worden sind; vergleiche Wolf (2013) und Ramey und Fox (2016). Die Beschreibung der einzelnen C++ Applikationen ist im Anhang A und E zu finden. Alle einzelnen Skripte, welche die Prozesskette steuern werden nicht aufgezeigt. Der interessierte Leser wird auf die Testfälle, die auf dem beigefügten Datenträger enthalten sind, verweisen.

Die Prozesskette wird so umgesetzt, sodass die Möglichkeit besteht diverse Modi zu verwenden und damit verschiedene Untersuchungen zu ermöglichen:

- Betrachtung simpler thermischer Problemstellungen unter Ausschluss der thermisch induzierten Spannungen; es wird lediglich die Energiegleichung (2.16) gelöst,
- Analyse thermischer Problemstellung mit Vorhersage der auftretenden Spannungen; hier wird ein halb-gekoppeltes System betrachtet — keine Kopplung der Verschiebungsgleichung zur Energiegleichung,
- Betrachtung lokaler Wärmebehandlungen mit Vorhersage der auftretenden Spannungen und Justierung des Energieeintrags aufgrund der Maximaltemperatur und/oder -spannung im Gesamtbauteil,
- Bestimmung der Materialeigenschaften nach einer lokalen Wärmebehandlung und/oder nachfolgendem Auslagern in 3D,
- Optionale Optimierung unter Verwendung einer kleinen oder großen Prozesskette.

Die oben genannten Modi können mit Abbildung 5.1 verdeutlicht werden. Vor allem die Bedeutung der kleinen und großen Prozesskettenoptimierung ist hier deutlich ersichtlich.

Der Ablauf der Modellierungssequenz ist wie folgt. Zu Beginn wird ein numerisches Netz erstellt; willkürliche FVM Netze können eingesetzt werden. Anschließend werden willkürliche numerische Zellen im diskreten Raum markiert und abgespeichert; vergleiche Anhang E.5. Subsequent wird das thermische Spannungsproblem gelöst. Dies beinhaltet — in dieser Arbeit — das lokale Lösungsglühn und Abkühlung; vergleiche Anhang A.7. Während dieser Modellierung werden die Temperaturprofile aller zuvor markierten Zellen aufgezeichnet und abgespeichert. Nach erfolgreicher Simulation werden alle Temperaturprofile der markierten Zellen in ein MatCalc lesbares Format konvertiert (Anhang E.6) und für jeden Punkt eine Materialberechnung durchgeführt (Kapitel 3.6). Abschließend werden die lokalen — an den willkürlichen Zellen — Materialinformationen für die 3D Extrapolation vorbereitet. Die Bestimmung der Streckgrenze in 3D erfolgt anschließend durch das Lösen einer Poisson-Gleichung; vergleiche Anhang E.7.

Zum Schluss wird eine volumengewichtete integrale Festigkeit an einer benutzerdefinierten Positionen berechnet; vergleiche Anhang E.7. Dieser Wert kann als Objektfunktion für eine Optimierungsaufgabe herangezogen werden.

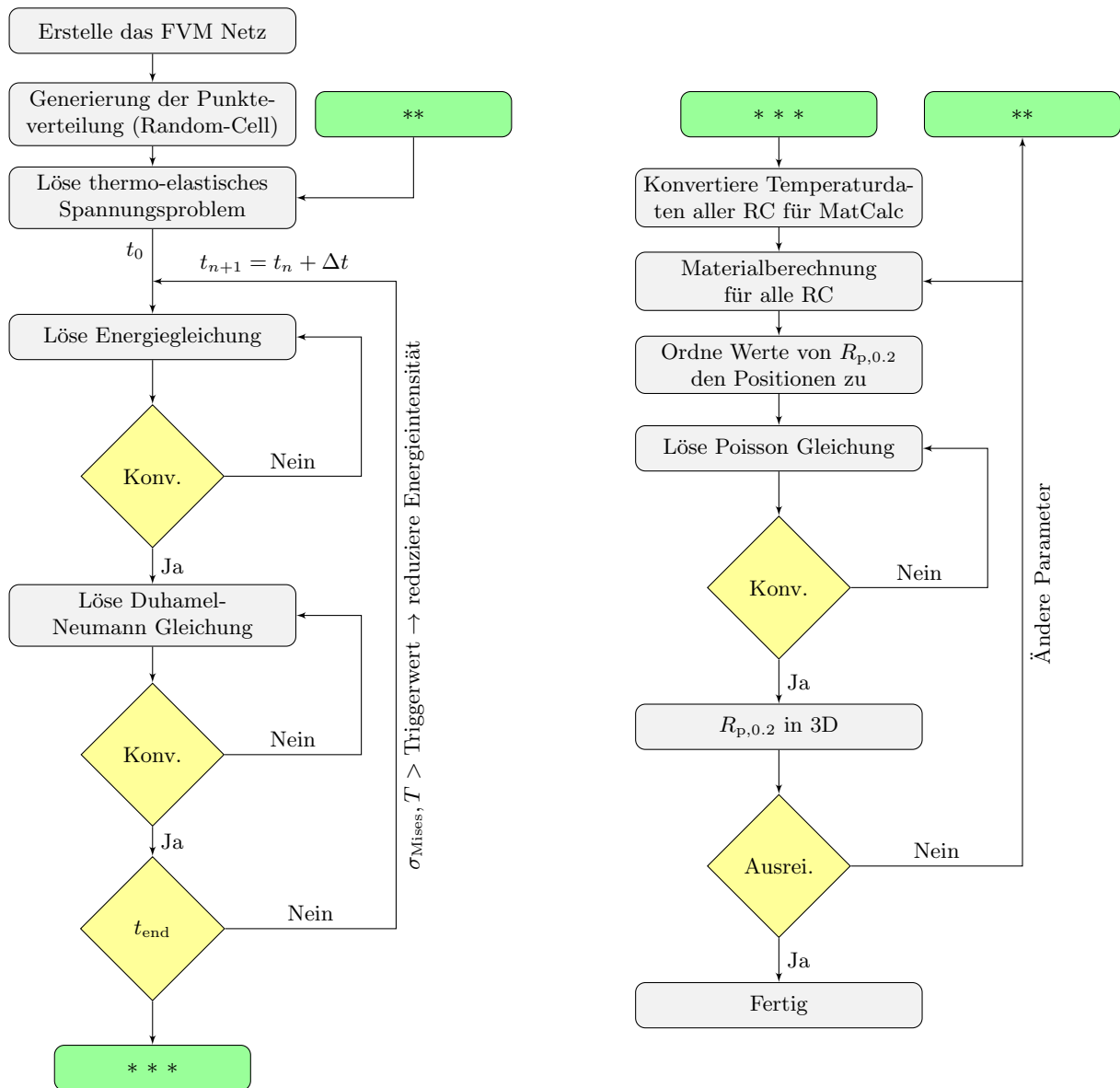


Abbildung 5.1.: Schematische Darstellung der entwickelten Prozesskettensequenz zur Berechnung der Materialeigenschaften in 3D für lokale beziehungsweise nicht-homogene Wärmebehandlungen unter Berücksichtigung thermischer Spannungen. *RC* steht für Random-Cell; *Ausrei.* für ausreichend; publiziert in Holzmann et al. (2018).

## 5.2. Lokale Wärmebehandlungen an zwei Beispielen

Nachfolgend wird an zwei Beispielgeometrien die lokale Wärmebehandlung mithilfe der Prozesskette modelliert und diskutiert. Hierbei wird aufgezeigt, wie Temperaturen das Spannungsfeld beeinflussen, welche Besonderheiten bei Aluminiumlegierungen vorherrschen und wie der Energieeintrag aufgrund von thermisch induzierten Spannungen und/oder der Temperatur reguliert wird.

Des Weiteren wird die lokale Wärmebehandlung in Phase 3 — vergleiche Abbildung 3.1 in Kapitel 3.1 — angewendet. Entsprechend wird ein lokales Lösungsglühen durchgeführt und wird subsequent für alle nachfolgenden Betrachtungen nicht verändert. Mit anderen Worten, die lokale Wärmebehandlung ist stets als örtliches Lösungsglühen anzusehen.

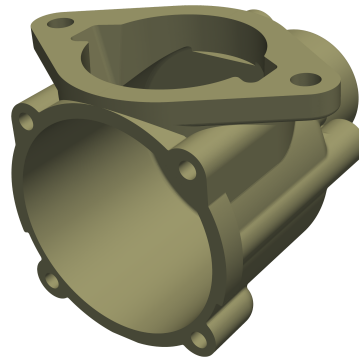
### 5.2.1. Die Geometrien der Strukturbauteile

Die zwei betrachteten Strukturbauteile sind in Abbildung 5.2 dargestellt. Grafik 5.2a und 5.2c zeigt eine komplexe Dämpferbrücke, die in RC Cars verbaut wird. RC steht hierbei für *Radio Controlled*. Die Brücke hat eine Profilstärke von fünf Millimeter und die Abmaße der Länge und Breite betragen circa 0,1 m x 0,1 m. Die zweite Geometrie stellt ein Motorgehäuse dar. Dieses hat eine Wandstärke von ebenfalls fünf Millimeter und ungefähre Abmessungen von circa 0,08 m x 0,08 m x 0,06 m (Länge, Breite und Höhe); die Größenangaben sollen nur ein Gefühl für das Design geben.

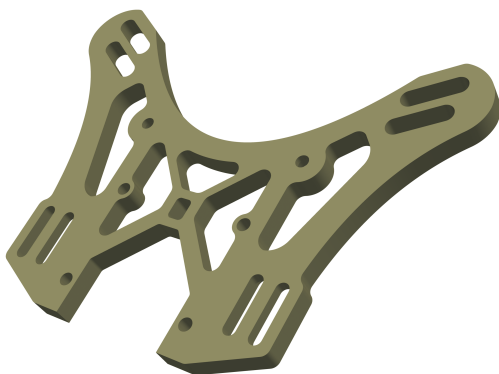
Beide Fallbeispiele sind auf dem Datenträger unter *Doktorarbeit/ResultCases/* zu finden. Die CAD Daten stammen von der Webseite <https://grabcad.com> und sind unter der GPL v2 veröffentlicht.



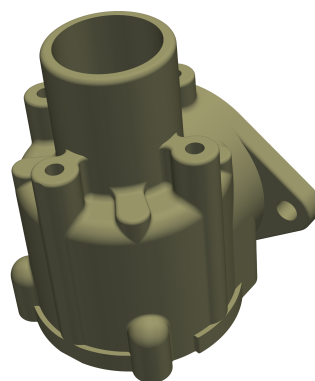
(a) Vorderseite



(b) Vorderseite



(c) Rückseite



(d) Rückseite

Abbildung 5.2.: Design der Dämpferbrücke und des Motorgehäuses die für die weitere Betrachtung herangezogen werden.



### 5.2.2. Numerisches Netz, Randbedingungen und wichtige Hinweise

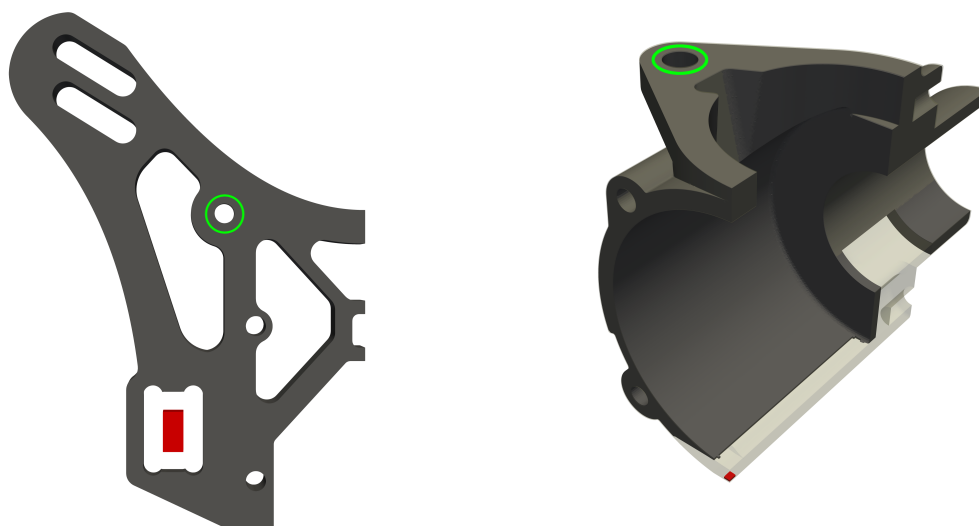
Die beiden Geometrien werden mit dem OpenFOAM® Vernetzer `snappyHexMesh` diskretisiert, wobei eine optimierte Version des Vernetzers verwendet wird; die Netze sind als Archiv auf dem Datenträger abgelegt.

In beiden Designs wird zur Minimierung des Berechnungsaufwands die Symmetrie ausgenutzt. Dies suggeriert, dass bei der Verwendung einer Energiequelle auf der Modellseite, diese auch auf der symmetrischen Gegenseite appliziert wird.

Auf allen Rändern — außer den Symmetrieebenen — wird die entwickelte `laserConvectionBC` verwendet. Daraus folgt, dass alle Oberflächen die freie Konvektion und Strahlung abbilden können, das mithilfe des temperaturabhängigen effektiven Wärmeübergangskoeffizient  $\alpha_{\text{eff}}$  modelliert wird. Der Energieeintrag erfolgt örtlich begrenzt unter Verwendung eines LASER's; Abbildung 5.3 gibt weitere Details diesbezüglich.

Für den Verschiebungsvektor werden alle Oberflächen als spannungsfrei betrachtet. Das Problem, das hierdurch entsteht, ist ein schlecht konditioniertes Matrixsystem, da nur Neumann Randbedingungen vorliegen und somit unendlich viele Lösungen zur Verfügung stehen — das gesamte Bauteil kann sich frei, in zwei Richtungen aufgrund der Symmetrieebene, bewegen. Um dem entgegenzuwirken, werden einige numerische Zellen explizit fixiert. Das bedeutet, dass die Zellwerte in der Matrix auf einen Nullvektor gesetzt werden; keine Verschiebung möglich.

Durch die Fixierung einiger Zellen ergeben sich an diesen und deren Umgebung sehr hohe Spannungen. Entsprechend wirkt sich dieser Umstand negativ auf das Steuerverhalten des LASER's aus, da die Maximalspannung im Bauteil zur Steuerung der Energieintensität herangezogen wird. Daher werden bei der Betrachtung der Maximalspannung nicht alle numerischen Zellen herangezogen; vergleiche die graue Zone in Abbildung 5.3. Mit anderen Worten, es werden einige numerische Zellen — um die Fixierung herum — weggeschnitten.



(a) Dämpferbrücke

(b) Motorgehäuse

Abbildung 5.3.: Erstellte Zellzonen. Die rote Zone repräsentiert alle numerischen Zellen die keine Verschiebung erfahren dürfen → der Verschiebungsvektor wird explizit in der Matrix auf einen Nullvektor gesetzt; die dunkelgraue Zone wird zur Bestimmung des maximalen Spannungswertes herangezogen. In grün dargestellt: LASER Einwirkzone.

Ferner wird die Energiequelle lotrecht auf die grün markierten Bohrungsdurchgänge, die in Darstellung 5.3 ersichtlich sind, appliziert. Die Modellierungsparameter für die Gauß'sche Normalverteilung (LASER) lauten; vergleiche Gleichung (A.20):

$$\sigma_x = 0,006 \text{ m} , \quad \sigma_y = 0,006 \text{ m} , \quad \rho_L = 0 [-] .$$

Das Zentrum der Gauß'schen Normalverteilung wird auf den Mittelpunkt der Bohrungen gelegt. Somit kann bereits abgeleitet werden, dass ein Großteil der Energie nicht eingetragen wird. Weitere Ausführungen diesbezüglich sind in Ludwig und Holzmann (2016b) gegeben.

Die Zeitschrittweite der Simulation beträgt  $\Delta t = 5e^{-4}$  s, wobei nur alle 0,05 s die Verschiebungsgleichung gelöst wird. In jedem Zeitschritt wird die Maximaltemperatur und/oder die maximale von Mises Spannung im Bauteil bestimmt und geprüft, ob benutzerdefinierte Triggerwerte überschritten werden. Trifft dies zu, wird die Intensität der Energiequelle reduziert. Die Justierung des Energieeintrags wird im Falle der Temperatur mit einem Polynom zweiter Ordnung und im Falle der von Mises Spannung mit einer Linearfunktion umgesetzt. Beide Formulierungen sind in Abbildung 5.4 schematisch dargestellt. Die verwendeten Triggerwerte lauten:

$$T_{\text{TrigA}} = 750 \text{ K} , \quad T_{\text{TrigE}} = 845 \text{ K} , \quad \sigma_{\text{Mises, TrigA}} = 60 \text{ MPa} , \quad \sigma_{\text{Mises, TrigE}} = 80 \text{ MPa} .$$

Die Kennzeichnung mit dem Subskript A und E deuten auf den Anfangs- und Endwert des Gültigkeitsbereichs der Funktionen hin; vergleiche Grafik 5.4. Ist der Maximalwert größer als der vorgegebene Endwert  $T_{\text{TrigE}}$  oder  $\sigma_{\text{MisesE}}$ , wird der Intensitätsfaktor  $\psi$  auf null gesetzt. Anzumerken ist, dass die Maximaltemperatur oder von Mises Spannung nicht ortsgebunden ist. Es werden stets die höchsten Werte aus der Geometrie herangezogen.

Des Weiteren wird ein *Sanftstart* des LASER's hinzugefügt. Das bedeutet, dass der Energieeintrag in einem vorgeschriebenen Zeitfenster von 0 % auf 100 % hochgefahren wird. Dieses Zeitfenster wird mit einem Wert von 0,2 s vorgegeben. Dadurch wird verhindert, dass zu Beginn der Simulation hohe Temperaturgradienten auftreten die entsprechend hohe thermische Spannungen erzeugen.

Die Überwachung der Triggerwerte wird — wie bereits angesprochen — zu jedem Zeitschritt getätigt, wobei stets der kleinste Wert der beiden Funktionen (Abbildung 5.4) für den Limi-

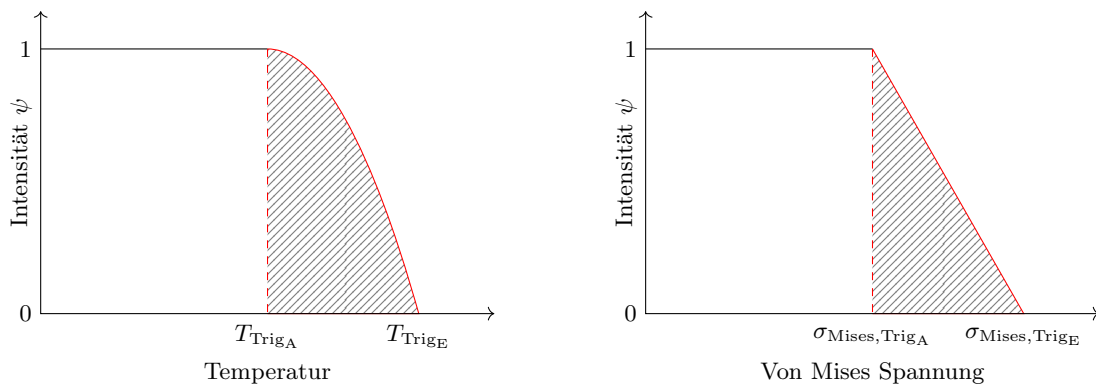


Abbildung 5.4.: Funktionspaarung zur Justierung der Energiequelle nach Erreichen des Triggerwertes (grau schraffierte Flächen). links: Temperaturfunktion; rechts: Spannungsfunktion.

tierungsfaktor  $\psi$  herangezogen wird. Dadurch wird den Bedingungen nachgegangen, dass kein Aufschmelzen des Materials sowie plastische Verformung eintritt. Sollte der Fall eintreten, dass der Maximalwerte der Temperatur die Solidustemperatur  $T_{\text{sol}}$  überschreitet, wird die Simulation abgebrochen.

Der betrachtete Zeitraum der lokalen Wärmebehandlung beträgt 60 s, wobei während dieser Zeit der LASER permanent eingeschaltet ist. Die Betrachtung des anschließenden Abkühlens und den daraus resultierenden Spannungs- und Temperaturverläufen wird im nächsten Kapitel diskutiert.

Es ist erwähnenswert, dass die hier vorgestellten Funktionen, welche zur Limitierung der Intensität verwendet werden, willkürliche Formen annehmen können.

### 5.2.3. Ergebnisse

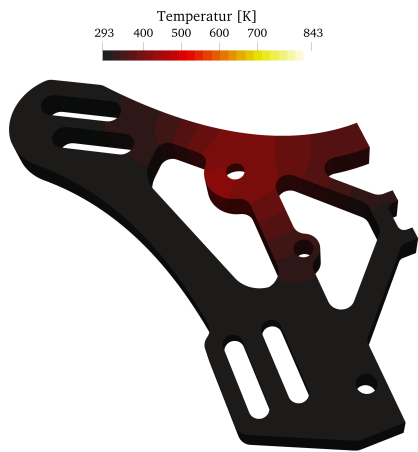
Die Temperaturprofile der lokalen Wärmebehandlung für beide Geometrien ist quantitativ in Abbildung 5.5 gegeben. Darstellung 5.6 gibt die von Mises Spannung in 3D wieder, wobei die Ausdehnung des Bauteils mit einem speziellen Post-Processing Filter vergrößert wird, um den Verzug der Geometrien zu visualisieren. Veranschaulicht sind die Momentaufnahmen für  $t = 2,5$  s,  $t = 10$  s und  $t = 40$  s. Hinzukommen die quantitativen Verläufe der Maximaltemperatur  $T_{\text{max}}$ , der maximalen von Mises Spannung  $\sigma_{\text{Mises,max}}$  und dem entsprechenden Intensitätsfaktor  $\psi$ , welche in Abbildung 5.7 gegeben sind.

### Temperatur- und Spannungsprofile

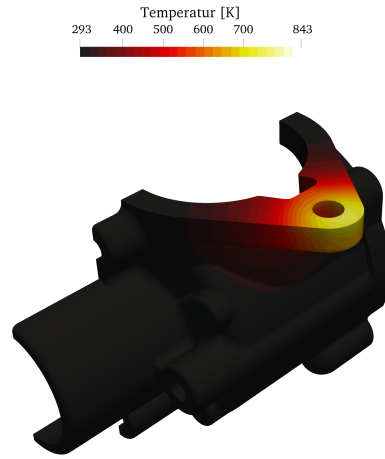
Aus den qualitativen Temperaturprofilen ist ersichtlich, dass eine rapide Energieverteilung innerhalb der Bauelemente stattfindet. Dies ist begründet durch die extrem hohe Wärmeleitfähigkeit von Aluminium. Wesentlich deutlicher ist dies am Motorgehäuse ersichtlich. Grund hierfür sind die höheren Energieeintragungen. Entsprechend kann abgeleitet werden, dass eine lokale Wärmebehandlung für Aluminiumlegierungen, wie bei der hier verwendeten AlSi7Mg0.3, problematisch sein kann, da zum einen durch die rapide Energieverteilung das gesamte Bauteil erwärmt wird und zum anderen das Halten hoher Temperaturen an lokalen Positionen durch die extreme Diffusion eine weitere Herausforderung darstellt; eine extreme Temperaturerhöhung, die lediglich örtliche begrenzt sein soll, ist damit ausgeschlossen. Der Gesamtenergieeintrag beider Modelle ist aufgrund der Modifizierung der Intensität in den dargestellten Fällen unterschiedlich.

Die von Mises Spannungen geben Aufschluss über die lokalen Belastungen der Bauteile. Aufgrund der Tatsache, dass sich das Material thermisch ausdehnen möchte, dies aber infolge der geometrischen Gestalt nur bedingt erlaubt ist (statisch steife Systeme), ergeben sich Belastungsakkumulationen. Insbesondere zeigt die Dämpferbrücke durch ihre geometrischen Gegebenheiten hohe Spannungskonzentrationen, welche weiter durch das sehr steife System und dem numerischen Fixieren der Halterung bedingt ist, auf; vergleiche Abbildung 5.3a. Entsprechend ist die freie Verformbarkeit der Brücke stark eingeschränkt, dass zu den hier dargestellten Spannungsspitzen — vor allem zwischen Fixierung und Symmetrieebene — führt. Anders hingegen kann sich die linke-obere Region der Dämpferbrücke relativ frei bewegen, sodass hier keine Extrema bezüglich der Spannung vorzufinden sind.

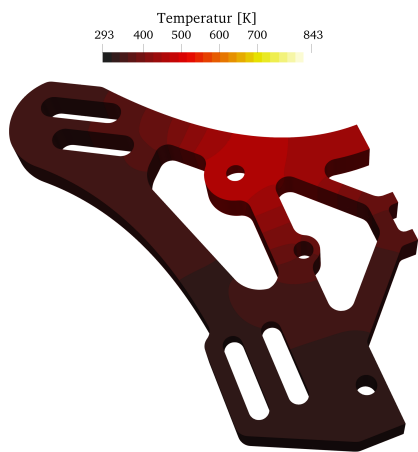
Diese Beobachtung kann im Falle des Motorgehäuses nicht bestätigt werden. Der Grund hierfür liegt vor allem an der anderen Handhabung der numerischen Fixierung. Da diese an der Symmetrieebene appliziert wird und nur einige Volumina ausgewählt werden, bleibt dem Gehäuse



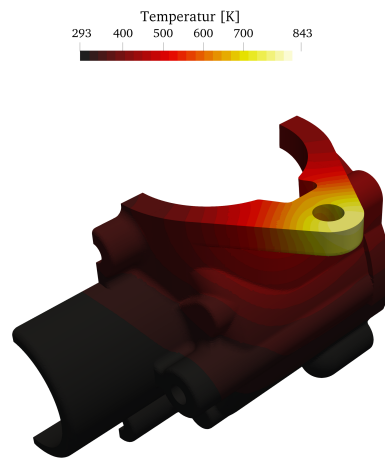
(a)  $t = 2,5$  s



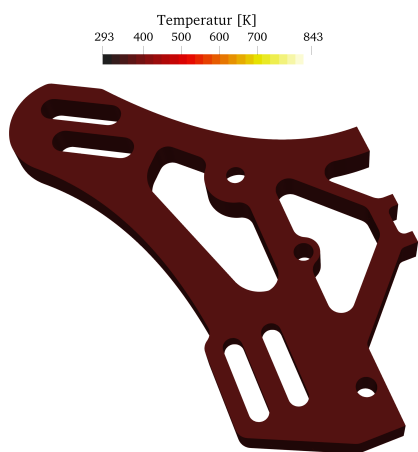
(b)  $t = 2,5$  s



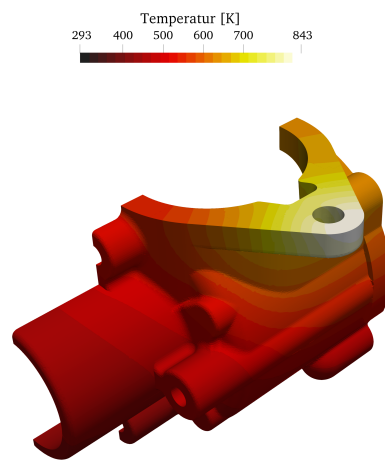
(c)  $t = 10$  s



(d)  $t = 10$  s



(e)  $t = 50$  s



(f)  $t = 50$  s

Abbildung 5.5.: Qualitative Ergebnisdarstellung der Temperaturprofile beider Geometrien für  $t = 2,5$  s,  $t = 10$  s und  $t = 50$  s; links: Dämpferbrücke; rechts: Motorgehäuse.

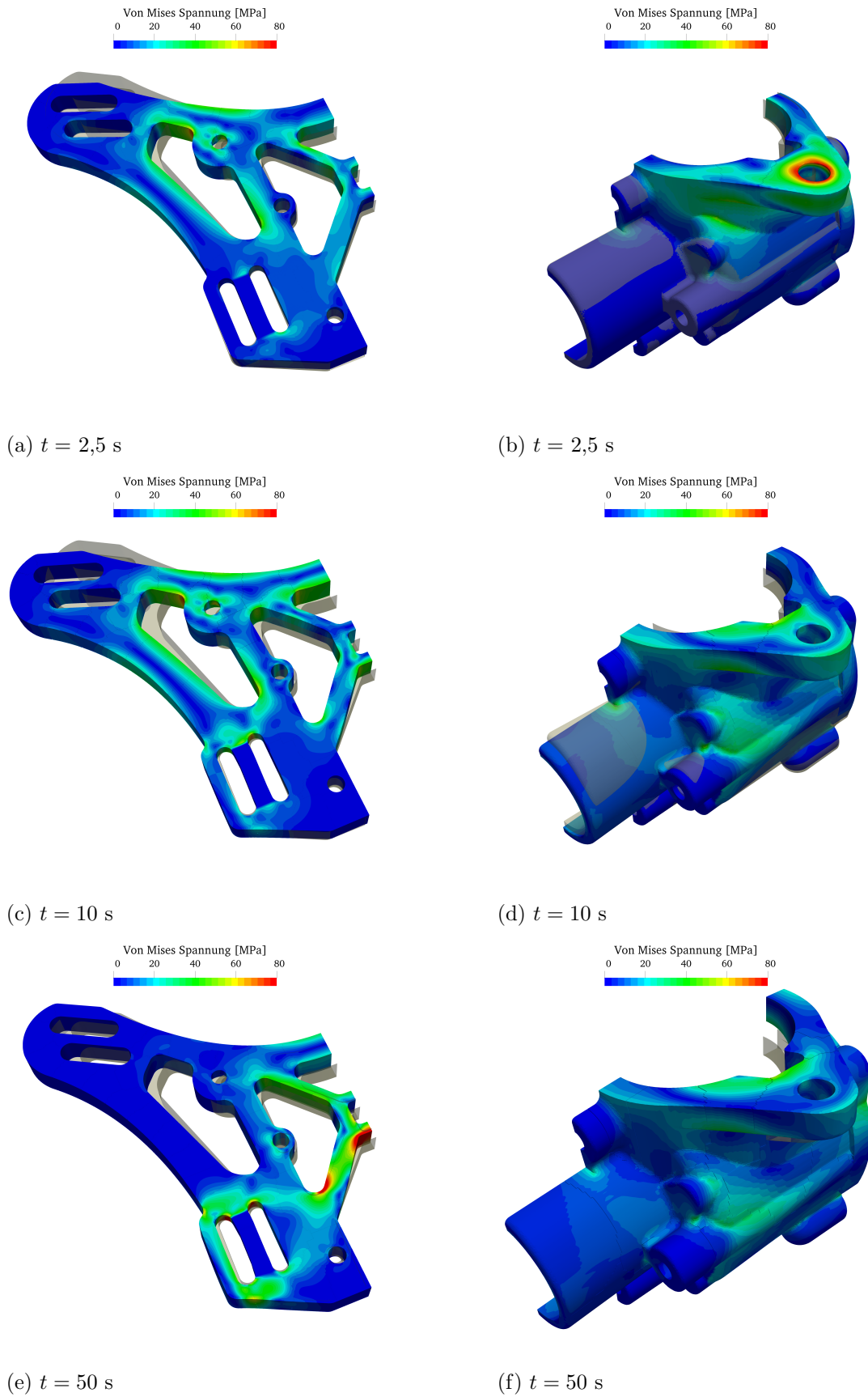


Abbildung 5.6.: Qualitative Ergebnisdarstellung der Von Mises Spannungen beider Geometrien für  $t = 2,5$  s,  $t = 10$  s und  $t = 50$  s. Die Verzerrung der Bauteile wird mit einem Faktor von 30 verstärkt. Des Weiteren ist die Originalform transparent zur besseren Beurteilung der Verschiebung eingefügt; links: Dämpferbrücke; rechts: Motorgehäuse.

während der Wärmebehandlung Spielraum sich auszudehnen und sich entsprechend den Verschiebungen zu verformen. Jedoch ist auch hier, aufgrund der Gestalt des Bauteils ersichtlich, dass vor allem Materialanhäufungen und/oder scharfe Kanten eine Quelle für Spannungsakkumulationen darstellen.

### **Energieeintrag Dämpferbrücke**

Der Energieeintrag kann unter Zuhilfenahme der Kenngrößen — maximale Temperatur und von Mises Spannung im Bauteil — erklärt werden; vergleiche Abbildung 5.7

Wird zu Beginn der Verlauf der Maximaltemperatur in der Dämpferbrücke analysiert, fällt auf, dass bis zum Zeitpunkt  $t \approx 0,6$  s die Temperatur ansteigt. Dies kann auf den Sanftstart zurückgeführt werden und ist in Abbildung 5.7e ersichtlich. Der Intensitätsfaktor  $\psi$  erhöht sich linear; respektive die Leistung des LASER's. Zugleich erhöht sich zwischen  $t = 0,1$  s und  $t = 0,6$  s die maximal auftretende Spannung, wodurch ab  $t = 0,6$  s der Triggerwert — bezogen auf die von Mises Spannung (blaue Kurve) — erreicht und eine Reduzierung des Energieeintrags eingeleitet wird. Der LASER wird entsprechend alle 0,05 s, aufgrund der Spannungsberechnung, neu justiert. Dies führt zu den Sägezahn ähnlichen Kurvenprofilen. Die Maximaltemperatur nimmt in dieser Zeit um ungefähr 90 K zu. Ab  $t \approx 10$  s tritt ein interessantes Phänomen auf, welches dazu führt, dass sich die Maximalspannungen weiter erhöhen, obwohl kein weiterer Energieeintrag stattfindet ( $\psi = 0$ ). Dieser Umstand ist bedingt durch die Komplexität des Designs und der hohen Wärmeleitfähigkeit des Materials. Die Wärmeenergie wird im Bauteil verteilt, wodurch sich steife — sich nicht bewegliche — Regionen erwärmen und sich ausdehnen. Solch eine Konfiguration weist stets, selbst bei kleinsten Ausdehnungen, quantitativ extrem hohe Spannungswerte auf; vergleiche das klassische Beispiel des beidseitig eingespannten Rohrs in der Festigkeitslehre. Infolgedessen werden in einzelnen Regionen Spannungswerte bis über 120 MPa erreicht. Dies kann vor allem beim Vergleich der qualitativen Temperatur- und Spannungsbilder bei  $t = 10$  s und  $t = 50$  s in Darstellung 5.5 und 5.6 bestätigt werden.

Es soll angemerkt werden, dass die Verformungseinschränkung durch das bewusste fixierten der numerischen Zellen verstärkt wird; vergleiche Abbildung 5.3b.

Entsprechen ist es nicht möglich hohe — lokale — Temperaturwerte am Energieeintragungsort zu generieren, wodurch eine Art Lösungsglühen nicht erzielt werden kann.

### **Energieeintrag Motorgehäuse**

Im Gegensatz dazu, ist der Energieeintrag im Motorgehäuse anders zu interpretieren. Zu Beginn ist erneut der lineare Anstieg, allerdings nur bis  $t = 0,1$  s, ersichtlich. Ab diesem Zeitpunkt sind die auftretenden Spannungen hoch genug, um die Energiequelle kurzzeitig auszuschalten, wodurch der Energieeintrag entfällt. Bei  $t = 0,15$  s sind die vorherrschenden Spannungen wieder klein genug, sodass der LASER erneut seinem linearen Sanftstartprofil folgt. Die subsequente Spannungsberechnung erzielt anschließend Werte bis über 200 MPa, wodurch ein pulsierender Energieeintrag zustande kommt; ersichtlich in allen drei Grafiken von Abbildung 5.7 rechts. Durch die starke Energiediffusion gleichen sich die Temperaturen im umliegenden Gebiet — ausgehend von der Energieeintragszone — an, wodurch die Spannungen sukzessiv zurückgehen. Ab  $t \approx 3$  s ist das System in einer Konfiguration, sodass der LASER kontinuierlich eingeschaltet bleibt, jedoch von den maximalen Von Mises Spannungen gesteuert wird. Die lineare Manipulation des Vorfaktors  $\psi$ , basierend auf der Maximalspannung, dauert bis  $t \approx 25$  s an, und wird

anschließend verlassen. Subsequent ist eine Justierung der Intensität, aufgrund auf der Maximaltemperatur, gegeben. Durch die weitere Temperaturangleichung im Bauteil verringern sich die Spannungen kontinuierlich bis unter 55 MPa.

Es soll angemerkt werden, dass die hohen Spannungen, welche zeitlich betrachtet nur sehr kurz vorherrschen, möglicherweise keine plastischen Verformungen bewirken können, da die Trägheit des Materials in diesem Fall mitberücksichtigt werden muss. Eine Analyse diesbezüglich muss mit erweiterten Spannungsgleichungen durchgeführt werden und wird nicht weiter betrachtet.

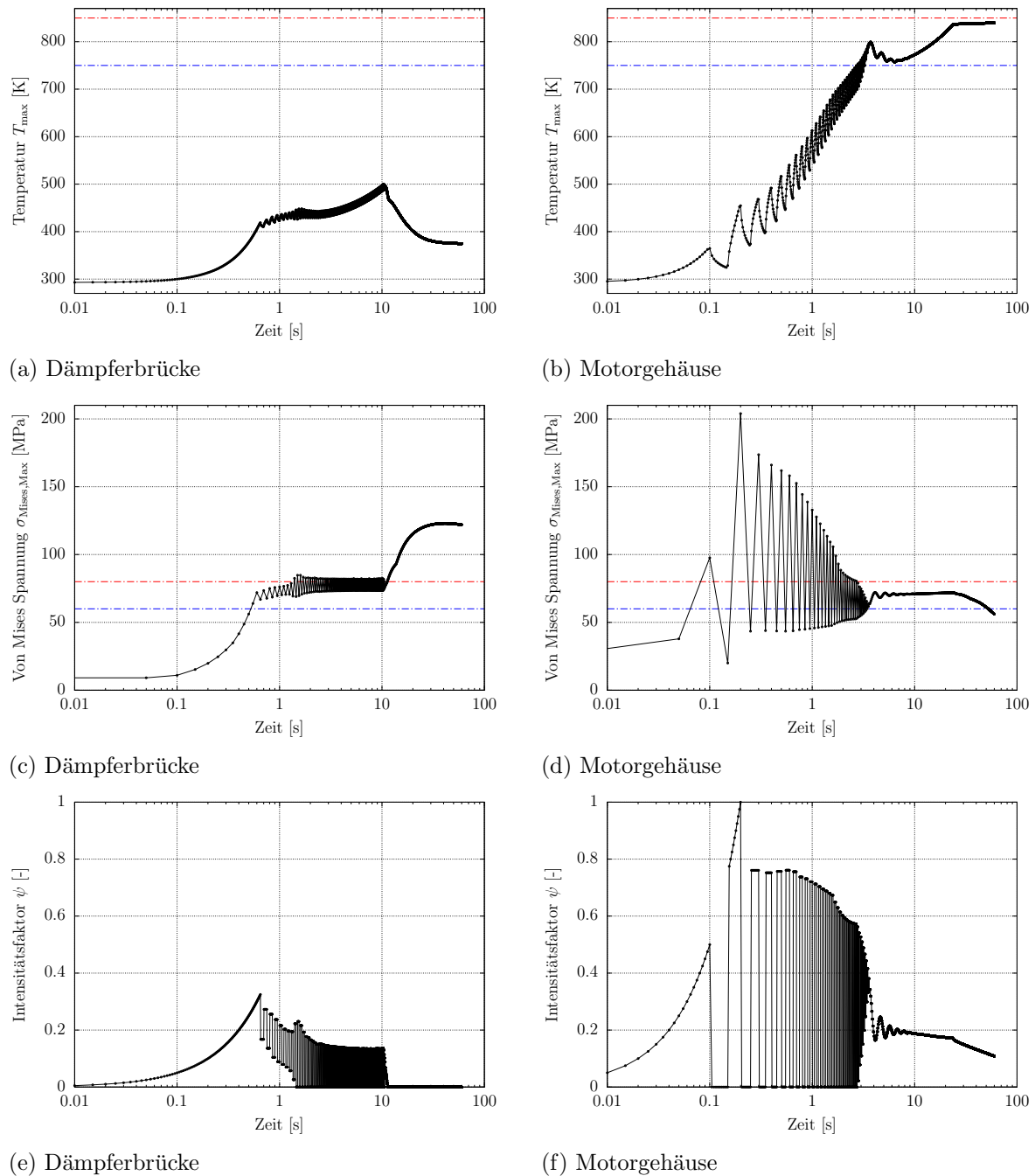


Abbildung 5.7.: Quantitative Verläufe der zeitlichen Maximalwerte der Temperatur und Von Mises Spannung in den Bauteilen während einer lokalen Wärmebehandlung (60 s — ohne Abkühlung) sowie der entsprechende Vorfaktor  $\psi$ ; links: Dämpferbrücke; rechts: Motorgehäuse. Die roten Linien geben jeweils die Schmelztemperatur und Fließgrenze wieder. Die blaue Kurve gibt jeweils den Triggerwert  $\text{Trig}_A$  an.

#### 5.2.4. Zusammenfassend und Erkenntnisse

Die vorgestellten Beispiele zeigen die Funktionsweise der Energiereduzierung, aufgrund der Anpassung des Intensitätsfaktors  $\psi$ , auf. Es wird nachgewiesen, dass sich die numerische Fixierung beziehungsweise ein statisch steifes Design nachteilig auf die Ausdehnung des Bauteils auswirkt und einen signifikanten Einfluss auf das Ergebnis hat — besonders auf die induzierten Spannungen. Des Weiteren werden die Einzelprofile der Maximaltemperatur und -spannung diskutiert und das Zusammenspiel zwischen den Limitierungsfunktionen (Abbildung 5.4) und dem Intensitätsfaktor  $\psi$  erläutert.

Gemäß diesen Untersuchungen wurde aufgezeigt, dass die entwickelte Prozesskette die numerische Modellierung von lokalen Wärmebehandlungen — unter Berücksichtigung der beiden Bedingungen: keine plastische Verformung und kein Aufschmelzen des Materials — bewerkstelligen kann.

Abschließend können die neu erlangten Erkenntnisse dafür eingesetzt werden, um beispielsweise eine Energiesteuerung für einen LASER zu generieren. Hier kann beispielsweise die Funktion des Intensitätsfaktors durch ein Kurvenprofil ersetzt werden, dass anschließend zur Steuerung des LASER's für experimentelle Untersuchungen herangezogen werden kann.

Ferner kann das Kurvenprofil für die numerische Modellierung verwendet werden. Da aufgezeigt wurde, dass die Spannungen lediglich zu Beginn eine entscheidende Rolle spielen — bezogen auf die Reduzierung des Energieeintrags —, kann durch die Vorgabe des Intensitätsfaktors  $\psi$  bis circa 20 s, die Spannungsrechnung komplett entfallen. Es muss jedoch dafür Sorge getragen werden, dass das verwendete Profil kein Fließen des Materials erlaubt. Dies wird dadurch erreicht, dass ein Profil für  $\psi$  im Zeitintervall  $0 \text{ s} < t < 20 \text{ s}$  erstellt wird und das lokale Lösungsglühen erneut mit einer Spannungsberechnung durchgeführt wird. Bleiben die Maximalspannungen unterhalb der Fließgrenze, kann das Profil verwendet werden. Der Vorteil der Verwendung eines Profils für den Intensitätsfaktor  $\psi$  ist in der Reduzierung der Berechnungszeit begründet; von

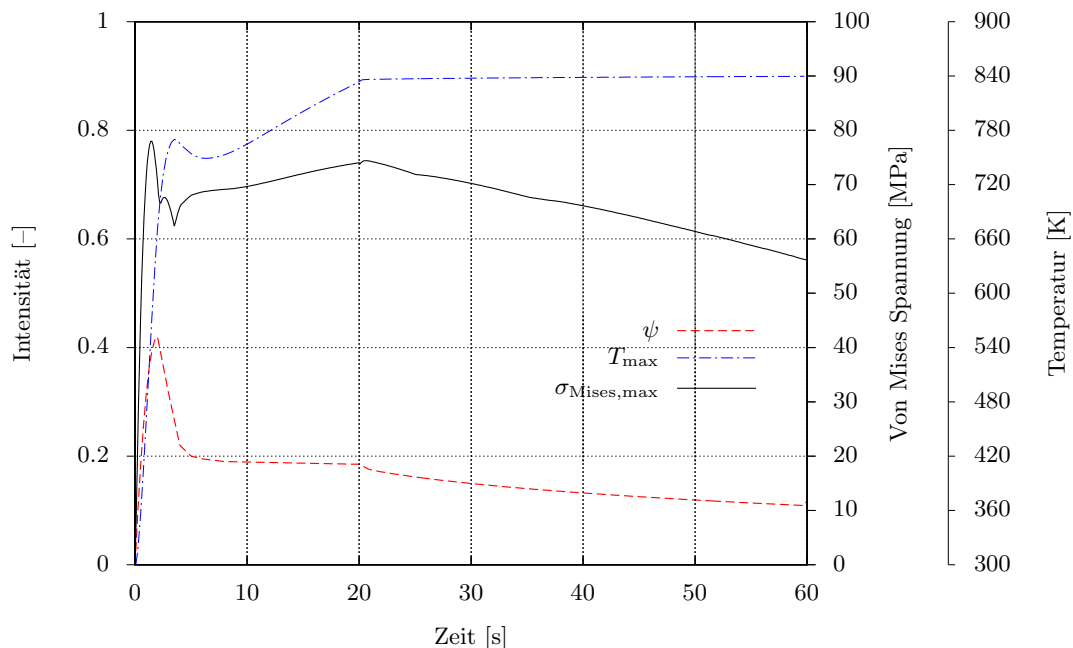


Abbildung 5.8.: Justierung des Energieeintrags unter Verwendung einer Funktion für den Intensitätsfaktor bis  $t = 20 \text{ s}$ .



Tagen auf Stunden.

In Abbildung 5.8 ist die Maximaltemperatur und -spannung erneut gegeben, wobei zwischen  $t = 0$  s und  $t = 20$  s eine Funktion für den Intensitätsfaktor  $\psi$  vorgegeben worden ist. Es ist deutlich ersichtlich, dass die maximale von Mises Spannung stets unterhalb von 80 MPa liegt und damit kein plastisches Verformen eintritt.

Ein solches Profil kann, wie bereits erwähnt, zur Steuerung eines LASER's in experimentellen Untersuchungen herangezogen werden. Die Generierung der Funktion ist in Anhang F vorgestellt.

### 5.3. Thermisch induzierte Spannungen beim Abkühlvorgang

Nachdem das Lösungsglühen abgeschlossen ist, wird im Allgemeinen das Bauteil in ein Wasserbad gelegt, um den lösungsgeglühten Zustand *einzufrieren*. Da die schnelle Temperaturreduzierung thermische Spannungen hervorruft, wird diese Problemstellung am Motorgehäuse weiter untersucht. Es werden zwei Abkühlarten — Selbstabkühlung und dem Abschrecken in Wasser — betrachtet.

Die lokale Wärmebehandlung wird erneut auf 60 s angesetzt. Nach dieser Zeit wird die Energiequelle ausgeschaltet. Anschließend kühlt das Motorgehäuse entweder durch freie Konvektion, Strahlung und Wärmeleitung (Selbstabkühlung) oder durch das Abschrecken in Wasser ab. Beim der Selbstabkühlung werden die Randbedingungen, bezogen auf das vorhergegangene Kapitel, für die Energiegleichung nicht verändert. Das bedeutet, dass die Abkühlung ausschließlich durch den temperaturabhängigen effektiven Wärmeübergangskoeffizienten  $\alpha_{\text{eff}}$  bedingt ist. Das Abschrecken in Wasser wird hingegen mit einem konstanten Wärmeübergangskoeffizienten  $\alpha_{\text{HTC}}$  von  $5.000 \text{ Wm}^{-2}\text{K}^{-1}$  modelliert und stellt eine Annahme dar; Leidenfrosteffekte und andere Phänomene werden hier nicht berücksichtigt.

Die Berechnung der Duhamel-Neumann Gleichung wird für die Selbstabkühlung alle 0,01 s und für das Abschrecken in Wasser im Abstand von 0,001 s gelöst. Der Grund hierfür ist die rasche Abkühlung, wodurch eine akkurate zeitliche Diskretisierung benötigt wird. Ferner wird lediglich ein kurzer Zeitabschnitt des Abkühlens betrachtet. Dieser beträgt nach dem Ausschalten der Energiequelle zehn Sekunden;  $60 \text{ s} \leq t \leq 70 \text{ s}$ .

Die Resultate der Untersuchung sind in Abbildung 5.9 gegeben; die Werte repräsentieren

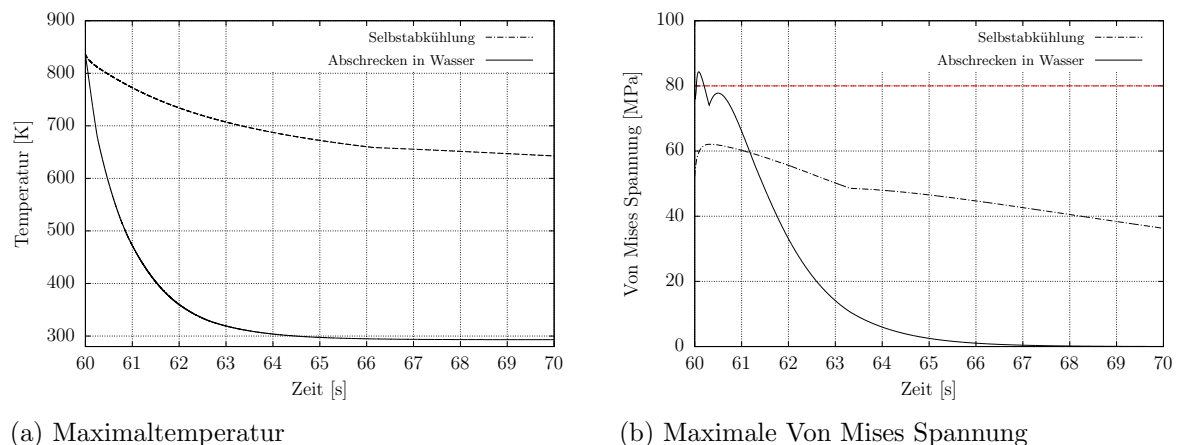


Abbildung 5.9.: Untersuchung der Von Mises Spannung bei zwei verschiedenen Abkühlvarianten des Motorgehäuses; Selbstabkühlung und Abschrecken in Wasser. Die rote Linie stellt die Grenze zwischen elastischer und plastischer Verformung dar.

jeweils die Maxima im gesamten Bauteil und sind daher, wie in allen vorangegangenen Untersuchungen, nicht ortsgebunden.

Aus Darstellung 5.9a sind die unterschiedlichen Abkühlraten ersichtlich. Die korrelierenden Von Mises Spannungen sind in Abbildung 5.9b gegeben. Aus diesen Ergebnissen kann geschlossen werden, dass bei der Selbstabkühlung keine plastische Verformung eintritt. Im Falle des Abschreckens in Wasser wird kurzzeitig die Streckgrenze  $R_{p,0.2}$  des Materials überschritten, so dass eine bleibende Verformung eintreten kann; die rote Linie zeigt die Grenze zur plastischen Verformung auf. Interessanterweise wird die Streckgrenze nur sehr kurz überschritten.

Da die Duhamel-Neumann Formulierungen für linear-elastische Bereiche ihre Gültigkeit besitzt und plastische Verformungen nicht berücksichtigt, muss zur Bestätigung der bleibenden Verformung ein numerisches Modell herangezogen werden, das plastische Deformationen berücksichtigt.

Ferner muss erwähnt werden, dass durch das Wegschneiden diverser Teile der Lösung, vergleiche Abbildung 5.3, gegebenenfalls einige Regionen mit extremer Spannungsakkumulation in der Betrachtung fehlen.

Hinzukommend ist die Streckgrenze  $R_{p,0.2}$  keine Konstante von 80 MPa. Durch das Lösungsglühen wird diese Quantität verändert, dass allerdings in der Prozesskette nicht betrachtet wird — zur Umsetzung muss ein voll-gekoppeltes System zwischen thermischer Spannungs- und Materialberechnung vorliegen.

## 5.4. Lokale Wärmebehandlung mit Materialberechnung

Nachfolgend wird das Motorgehäuse für die weitere Materialberechnung herangezogen. Die Wärmebehandlung erfolgt ohne Spannungsberechnung unter Verwendung der im vorangegangenen Kapitel erwähnten Funktion für den Intensitätsfaktor  $\psi$  bis  $t = 20$  s.

Für die Materialberechnung nach der Wärmebehandlung wird die Random-Cell-Poisson Methode verwendet. Es werden insgesamt 1300 willkürliche numerische Zellen für die Materialberechnung herangezogen, wobei 300 im gesamten Bauteil und weitere 1000, in einer Region um das Bohrloch, generiert werden; vergleiche Abbildung 3.8a. Dies stellt ein Kompromiss zwischen Genauigkeit und Berechnungsaufwand dar.

Die Dauer des Energieeintrags beträgt erneut 60 s. Anschließend wird der LASER ausgeschaltet und das Motorgehäuse entweder durch Selbstkühlung oder Abschrecken in Wasser auf Raumtemperatur heruntergekühlt. Das Selbstabkühlen wird für eine weitere Stunden numerisch modelliert und angenommen, dass das Bauteil anschließend in Wasser eingetaucht wird, um das gesamte Bauteil auf Raumtemperatur zu bringen. Diese Annahme wird eingeführt, um sicherzustellen, dass nach dem Abkühlen das Motorgehäuse stets auf Raumtemperatur abgekühlt ist.

Die an den willkürlichen Punkten (Random-Cell-Poisson) gespeicherten Temperaturverläufe, werden anschließend für die Materialmodellierung, explizit für das Lösungsglühen, herangezogen. Das subsequente Auslagern findet bei 120 °C für 10 h statt.

Ferner wird die integrale Festigkeit — volumengemittelt —, um das Bohrloch, bestimmt. Hierzu werden alle numerischen Zellen, die im Umkreis von 9 mm um die Bohrlochmittellinie vorzufinden sind, einbezogen. Abbildung 5.10 verdeutlicht dieses Gebiet. Ferner ist im Anhang E.7 auf Seite 215 die Berechnung der volumengewichteten integralen Festigkeit beschrieben.

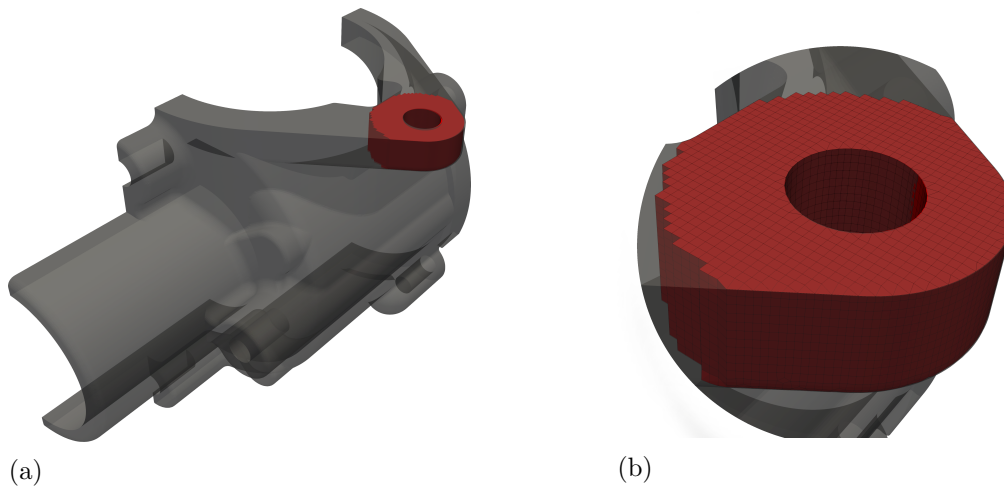


Abbildung 5.10.: Ausgewählte numerische Zellen für die Bestimmung der volumengewichteten integralen Festigkeit in der Nähe des Bohrlochs.

#### 5.4.1. Selbstkühlung (Self-Quenching)

Die Ergebnisse für die Selbstabkühlung nach einer lokalen Wärmebehandlung werden nachfolgend diskutiert. Die qualitativen Ergebnisse der subsequenten Abkühlung sind in Abbildung 5.11 für die Zeitpunkte  $t = 60$  s,  $t = 120$  s,  $t = 300$  s und  $t = 1800$  s gegeben.

Bei der Betrachtung kann eindeutig festgehalten werden, dass direkt, nachdem der LASER ausgeschaltet ist, die Maximaltemperaturen stark zurückgehen. Dies ist auf die thermische Leitfähigkeit des Materials zurückzuführen und wird als Self-Quenching bezeichnet. Nachdem sich die Energie homogen im Bauteil verteilt hat — bereits nach weniger als 120 s —, wird eine langsame Annäherung der Raumtemperatur erzielt; aufgrund von Konvektion und Strahlung.

Der quantitative Verlauf der gemittelten und maximalen Bauteiltemperatur verdeutlicht diese Aussage; vergleiche Abbildung 5.13. Durch die lokale Wärmebehandlung wird das Material am Ort des Energieeintrags stark erwärmt. Ab circa  $t = 20$  s sind die Maximaltemperaturen erreicht und der weitere Energieeintrag erhöht lediglich die Gesamttemperatur (Energie) des Motorgehäuses, die bis zum Abschalten des LASER's bei  $t = 60$  s zunimmt. Ab  $t > 60$  s erfolgt die Selbstabschreckung, wodurch die maximale Temperatur um circa 300 K rapide abfällt. Nachdem sich das Energieniveau im Bauteil angepasst hat, verschmelzen beide Graphen  $\rightarrow t \approx 200$  s. Anschließend ist die langsame Abkühlung aufgrund der Strahlung und Konvektion ersichtlich.

Die Berechnung der Materialeigenschaften nach dem lokalen Lösungsglühen und Abkühlen des Strukturbauteils, sowie nach der Auslagerung bei 120 °C und 10 h ist in Darstellung 5.12 gegeben. Die integrale Festigkeit um das Bohrloch beträgt nach dem Lösungsglühen und dem Auslagern respektive 80,3 MPa und 82,8 MPa. In den qualitativen Profilen (Abbildung 5.12) ist ersichtlich, dass durch das lokale Lösungsglühen die interessierte Region verbessert wird. Werden jedoch die Zahlenwerte verglichen kann festgehalten werden, dass diese — im vorliegenden Fall — marginal ist.

Es soll jedoch erwähnt werden, dass bei einer höheren Auslagerungszeit und/oder -temperatur, sich die Gesamtfestigkeit, aufgrund des Wachstums der Cluster-Phase, die sich weiter in die  $\beta''$ -Phase umwandelt, erhöht. Dies ist unabhängig von der lokalen Wärmebehandlung und auf das verwendete Radientransformationsmodell zurückzuführen.

Zur Bestätigung dieser Aussage wird daher die Simulation wiederholt, wobei die Auslage-

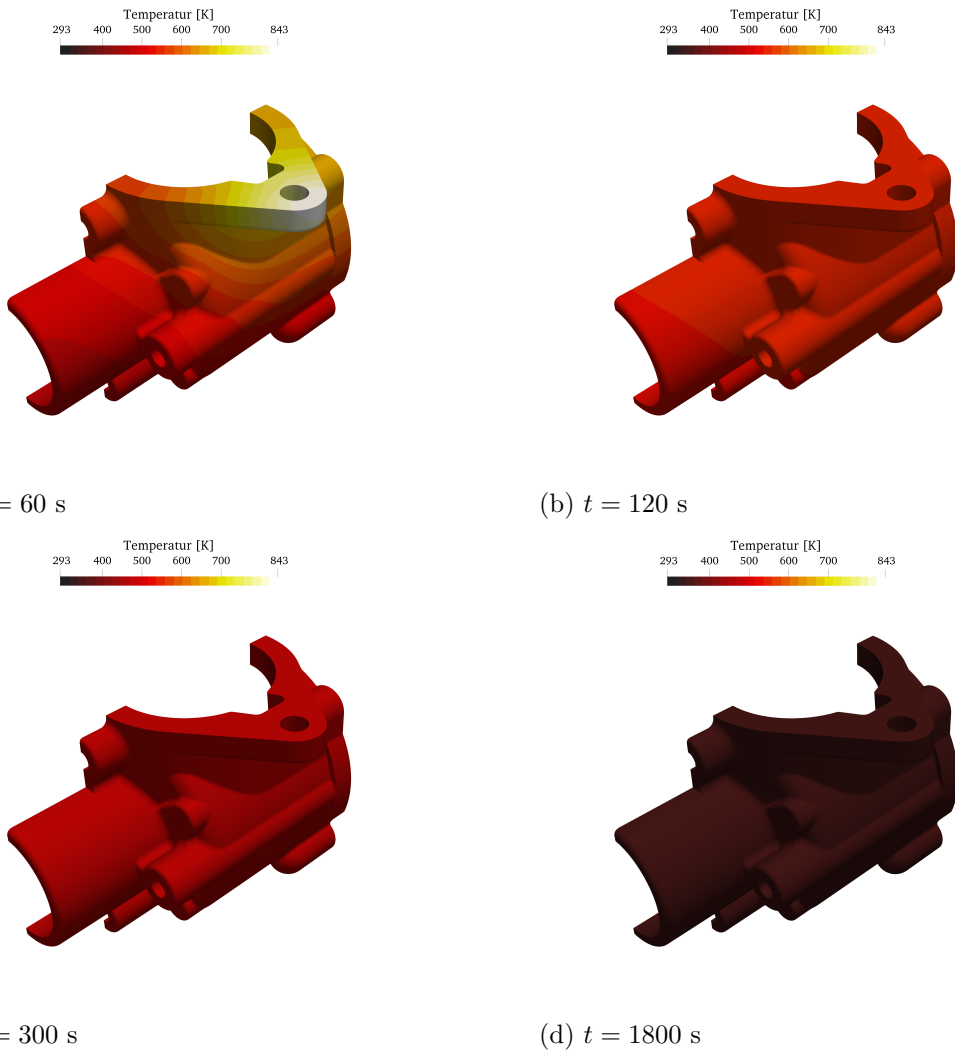


Abbildung 5.11.: Qualitative Ergebnisdarstellung der Temperaturprofile des lokal wärmebehandelten Motorgehäuses für  $t = 60$  s,  $t = 120$  s,  $t = 300$  s und  $t = 1800$  s unter der Annahme einer Selbstabkühlung.

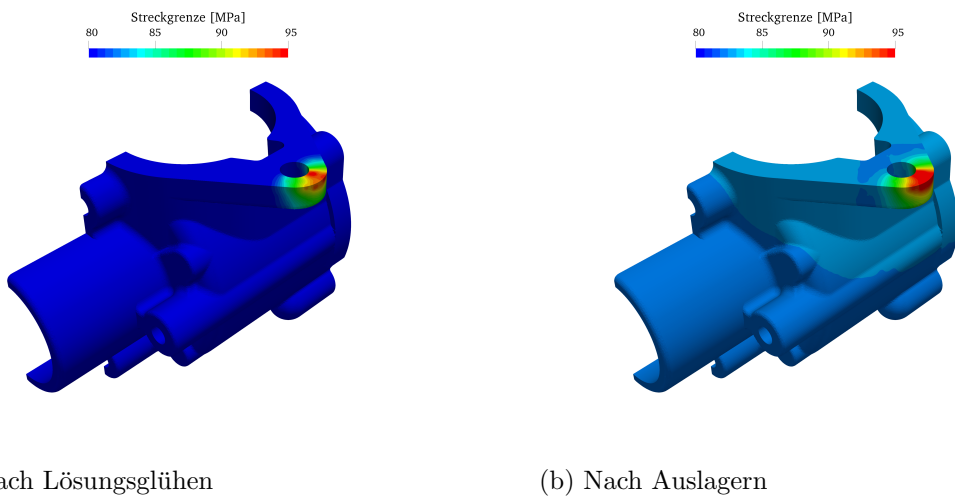


Abbildung 5.12.: Materialberechnung in 3D nach dem Lösungsglühen und dem anschließenden Auslagern bei  $120$  °C für 10 h; Annahme Selbstabkühlung.

rungszeit auf 20 h und die -temperatur auf 135 °C erhöht wird. Das qualitative Ergebnis dieser Berechnung ist in Abbildung 5.14 gegeben und untermauert die zuvor genannten Aussagen. Das lokale Lösungsglühen und dessen Einfluss auf die integrale Festigkeit wird im nachfolgenden Kapitel erneut herangezogen und intensiver untersucht.

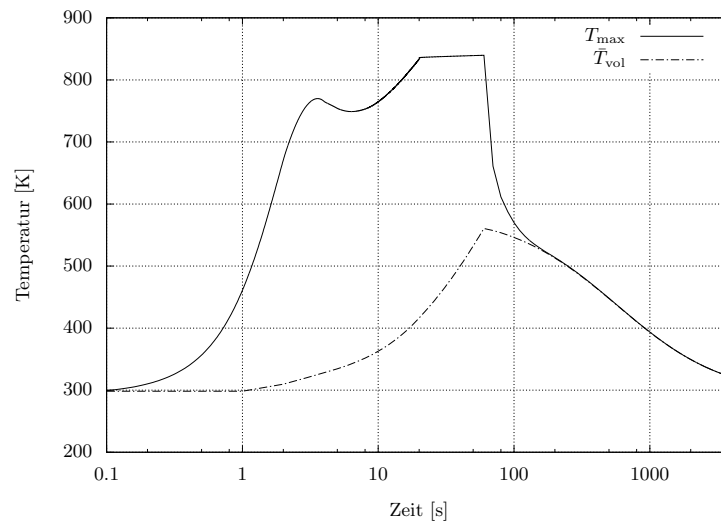


Abbildung 5.13.: Mittlere (volumengewichtet) und maximale Temperatur im Motorgehäuse während der lokalen Wärmebehandlung und subsequenter Selbstabkühlung.

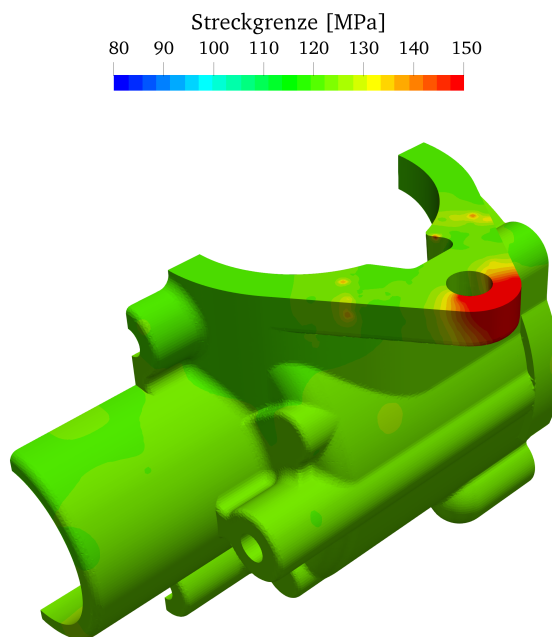


Abbildung 5.14.: Streckgrenze  $R_{p,0.2}$  in 3D der Aluminiumlegierung nach dem Auslagern bei 135 °C für 20 h bei vorangegangenen Lösungsglühen und Selbstabkühlung.

### 5.4.2. Abschrecken in Wasser (Water-Quenching)

Wird die gleiche Wärmebehandlung mit dem Abschrecken in Wasser durchgeführt, werden integralen Festigkeiten um das Bohrloch nach dem Lösungsglühen und dem Auslagern von 72,4 MPa und 135,1 MPa erhalten. Die wesentlich höhere integrale Festigkeit ist der rapiden Abkühlung geschuldet, wodurch ein übersättigter Mischkristall für die subsequente Auslagerung vorliegt. Aufgrund dessen, sind die kinetischen Vorgänge im Material, bezogen auf die Selbstabkühlung, unterschiedlich, wodurch diese Ergebnisse erzielt werden. Abbildung 5.15 zeigt ferner die mittlere und maximale Temperatur im Gehäuse auf. Es ist klar zu erkennen, dass nach dem Abschalten der Energiequelle, die Abkühlung des Bauteils auf Raumtemperatur in wenigen Sekunden abgeschlossen ist.

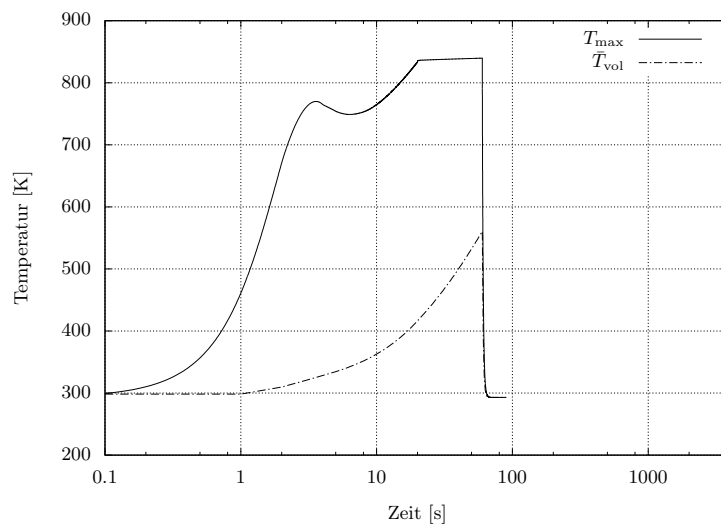


Abbildung 5.15.: Mittlere (volumengewichtet) und maximale Temperatur im Motorgehäuse während der lokalen Wärmebehandlung und subsequentem Abschrecken in Wasser.

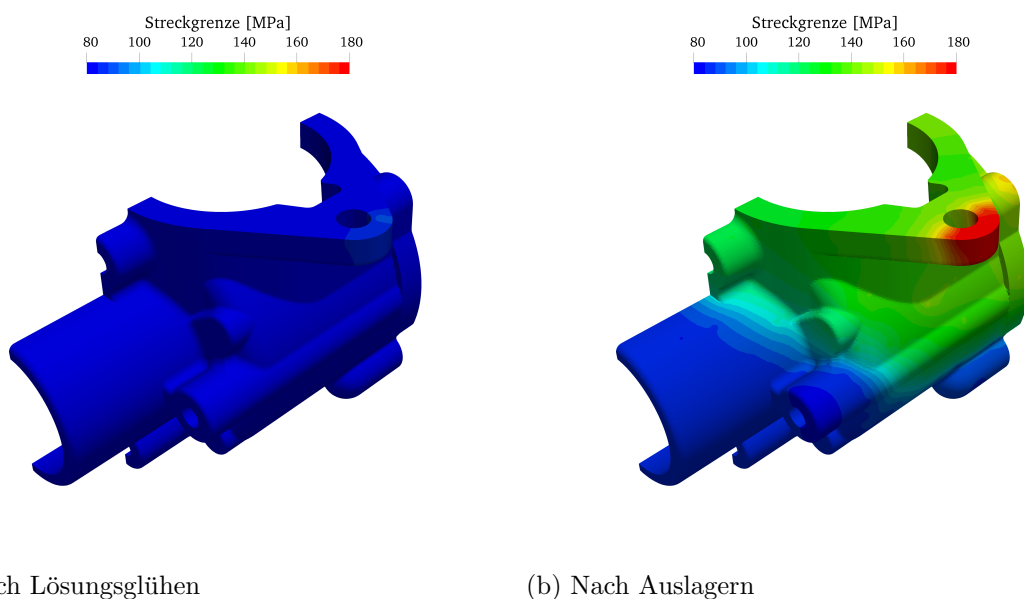


Abbildung 5.16.: Materialberechnung in 3D nach dem Lösungsglühen und dem anschließenden Auslagern für 10 h bei 120 °C; Annahme Abschrecken in Wasser.

## 5.5. Einfluss der lokalen Lösungsglühzeit auf die integrale Festigkeit

Nachfolgend wird die Prozesskette dazu verwendet, den Einfluss der lokalen Lösungsglühzeit auf die integrale Festigkeit um die Bohrung des Bauteils nach dem Auslagern zu untersuchen. Hierbei werden vier verschiedene lokale Lösungsglühzeiten —  $t_L = 30$  s,  $t_L = 60$  s,  $t_L = 120$  s und  $t_L = 300$  s — betrachtet und ferner der Einfluss der Abkühlung analysiert. Dabei wird erneut auf die bereits bekannten Abkühlvarianten, dem Selbstkühlen und dem Abschrecken in Wasser, eingegangen.  $t_L$  steht für die aktive LASER Zeit. Hinzukommend werden die Auslagerungsparameter variiert. Das bedeutet für die numerische Modellierung folgendes:

- Das lokale Lösungsglühen wird mit jeder Lösungsglühzeit  $t_L$  berechnet,
- Darüber hinaus werden für jede Lösungsglühzeit beide Abkühlvarianten betrachtet,
- Hinzu kommt, dass für jede Lösungsglühzeit und Abkühlvariante, eine Multidimensionalanalyse des Auslagerns durchgeführt wird,
- Für jeden zu evaluierenden Punkt in der Multidimensionalanalyse wird die volumengewichtete integrale Festigkeit in der Region um das Bohrloch bestimmt.

Die Modellierungsparameter für die Simulationen werden nicht geändert. Lediglich, wie bereits aus der Aufzählung ersichtlich, die Dauer des Energieeintrags, die Abschreckcharakteristik sowie die Parameter des nachfolgenden Auslagerns. Die Variation der Auslagerungsparameter erfolgt durch die Software DAKOTA<sup>®</sup>. Der Parameterraum für die Auslagerungszeit und -temperatur werden in acht äquidistante Abstände eingeteilt, wobei jeder diskrete Punkt evaluiert wird. Die Beschränkung der Parameterräume wird wie folgt gesetzt:

$$1800 \text{ s} \leq t_{\text{Ausl.}} \leq 360000 \text{ s} , \quad 50 \text{ °C} \leq T_{\text{Ausl.}} \leq 150 \text{ °C} ,$$

wobei der Subskript das Auslagern bezeichnet. Es soll angemerkt werden, dass die Spannungsberechnung nicht betrachtet wird. Für die Berechnung der integralen Festigkeiten wird erneut die Random-Cell-Poisson Methode verwendet. Aufgrund der Berechnungsdaten werden nunmehr 330 willkürliche Punkte betrachtet. Davon sind 300 in der Nähe des Bohrlochs und die weiteren 30 im gesamten Bauteil verteilt.

Tabelle 5.1 gibt einen zusammenfassenden Überblick über die numerischen Untersuchungen.

Lösungsglühen	Abkühlvariante	Auslagern
$t_L = 30$ s	Selbstabkühlung	Multidimensionalanalyse
$t_L = 30$ s	Abschrecken in Wasser	Multidimensionalanalyse
$t_L = 60$ s	Selbstabkühlung	Multidimensionalanalyse
$t_L = 60$ s	Abschrecken in Wasser	Multidimensionalanalyse
$t_L = 120$ s	Selbstabkühlung	Multidimensionalanalyse
$t_L = 120$ s	Abschrecken in Wasser	Multidimensionalanalyse
$t_L = 300$ s	Selbstabkühlung	Multidimensionalanalyse
$t_L = 300$ s	Abschrecken in Wasser	Multidimensionalanalyse

Tabelle 5.1.: Übersicht der durchgeführten numerischen Analysen.

### 5.5.1. Lokale Wärmebehandlung mit Selbstabkühlung

Die Berechnung der volumengewichteten integralen Festigkeit um das Bohrloch für die oben genannte Modellierungsanalyse wird wie in Abbildung 5.1 dargestellt, durchgeführt. Entsprechend kann abgeleitet werden, dass für jede Lösungsglühzeit zwei Abschreckvarianten durchgeführt werden. Dabei werden für jede Abkühlvariante 81 Parametersätze für das Auslagern evaluiert wobei für jeden Parametersatz 330 Kinetikberechnungen durchgeführt werden. Daraus ergeben sich 106.920 Materialberechnungen für die Selbstabkühlung und weiter 106.920 für das Abschrecken in Wasser.

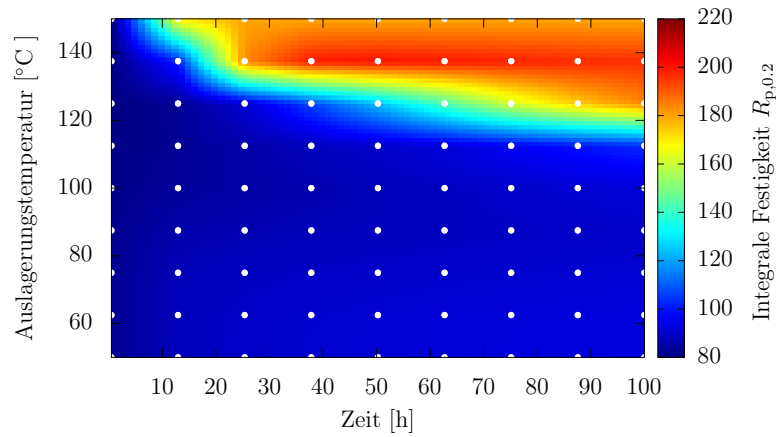
Zu Beginn werden die erhaltenen integralen Festigkeiten zur Begutachtung und Beurteilung des Parameterraums für die Selbstabkühlung diskutiert. Die Ergebnisse sind in Abbildung 5.17 gegeben. Die weißen Punkte stellen die diskreten Evaluierungspositionen der Multidimensionalanalyse dar. Mittels `gnuplot` werden diese noch weiter interpoliert, sodass die dargestellten Skalenverläufe erhalten werden.

Es ist eindeutig erkennbar, dass der maximal erreichbare quantitative Wert der integralen Festigkeit mit zunehmender Dauer der lokalen Wärmebehandlung zunimmt. Hinzu kommt die Verbreiterung des Parameterraums, in welchem höhere Festigkeitswerte erzielt werden können. Vor allem bei einer langen Auslagerungszeit und hohen Temperaturen, werden die Festigkeitswerte deutlich erhöht.

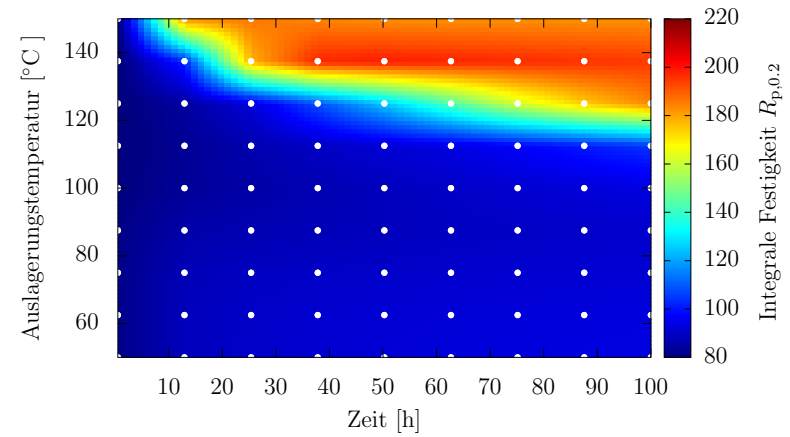
Ferner kann abgeleitet werden dass die optimale Auslagerungstemperatur zwischen 130 °C und 140 °C liegt und die Dauer des Auslagerns angehoben werden muss, wenn die Temperaturen niedriger sind.

Im Vergleich aller vier dargestellten Fälle ist Abbildung 5.17d der Beste, da hier die höchsten Festigkeitswerte erzielt werden. Die anderen drei Konturdarstellungen ähneln sich im Bezug auf die Maximalfestigkeit. Des Weiteren kann geschlussfolgert werden, dass eine kurze Auslagerungszeit und kleine Auslagerungstemperaturen keine nennenswerte Festigkeitsänderung hervorrufen; erst ab Auslagerungstemperatur größer wie  $T_{\text{Ausl}} > 110$  °C können merkbare Festigkeitssteigerungen erzielt werden.

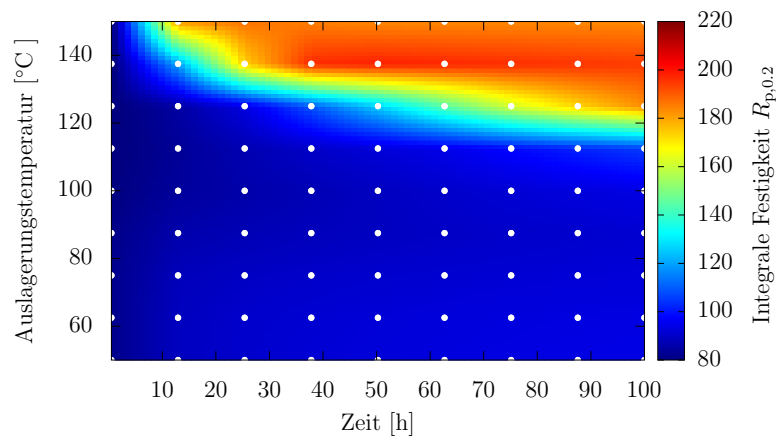




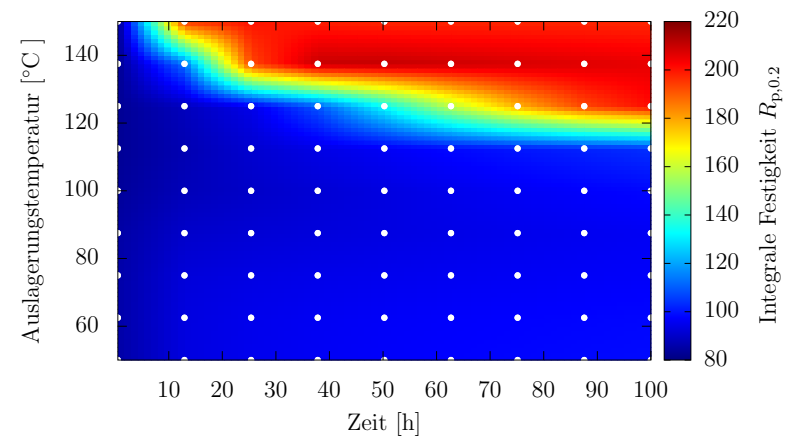
(a)  $t_L = 30$  s



(b)  $t_L = 60$  s



(c)  $t_L = 120$  s



(d)  $t_L = 300$  s

Abbildung 5.17.: Multidimensionalanalyse. Einfluss der Auslagerungszeit und -temperatur auf die volumengewichtete integrale Festigkeit der Bohrungsregion; bezogen auf vier verschiedene lokale Lösungsglühzeiten mit der Annahme der Selbstabkühlung.

## Zusammenfassend

Aufgrund der zuvor diskutierten Ergebnisse kann zusammengefasst werden, dass eine lokale Wärmebehandlung einen Einfluss auf die integrale Festigkeit aufzeigt. Zur Überprüfung inwieweit der inhomogene Energieeintrag auf die Festigkeit Einfluss nimmt, wird erneute eine Multidimensionalanalyse ohne Lösungsglühen durchgeführt. Das Ergebnis ist in Abbildung 5.18 gegeben.

Wird diese Darstellung mit denen aus 5.17 gegenübergestellt, kann eindeutig festgehalten werden, dass die lokale Wärmebehandlung einen positiven Einfluss auf die integrale Festigkeit um die Bohrung besitzt. Vor allem die Vergrößerung der Parameterauswahl für das Auslagern kann aufgezeigt werden.

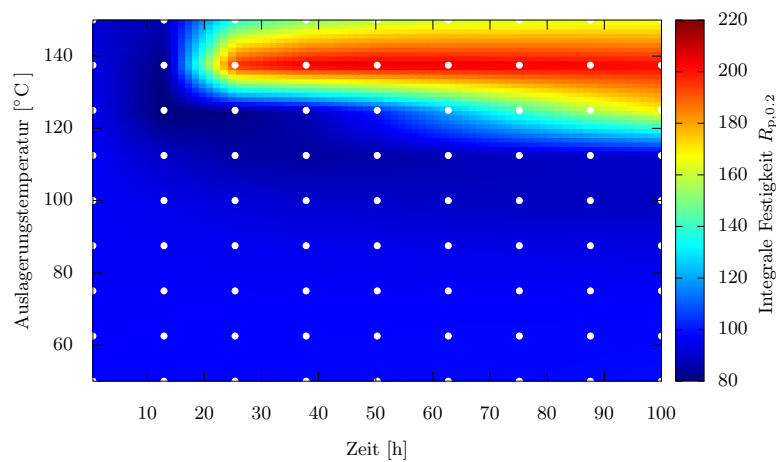


Abbildung 5.18.: Ergebnis der Multidimensionalanalyse (Auslagern) unter der Annahme, dass kein lokales Lösungsglühen durchgeführt wird.

### 5.5.2. Lokale Wärmebehandlung mit Abschrecken

Ergänzend zum vorherigen Kapitel werden die gleichen Simulationen durchgeführt, allerdings wird nach dem Abschalten des LASER's das Motorgehäuse in Wasser abgeschreckt und anschließend ausgelagert. Das Abschrecken wird erneut mit einem Wärmeübergangskoeffizienten  $\alpha_{\text{eff}} = 5000 \text{ Wm}^{-2}\text{K}^{-1}$  modelliert. Dadurch wird der Materialzustand während der lokalen Wärmebehandlung eingefroren, wodurch sich Unterschiede in den berechneten integralen Festigkeiten abzeichnen sollten.

Die Ergebnisse dieser Analysen sind in Abbildung 5.19 dargestellt. Es kann direkt abgeleitet werden, dass das Abschrecken einen signifikanten Einfluss auf die integralen Festigkeitswerte besitzt. Aus den Darstellungen ist ersichtlich:

- Die maximal erzielbaren Festigkeitswerte für kurze Glühzeiten sind wesentlich höher verglichen zur Selbstabkühlung,
- Fast im gesamten Parameterraum kann eine Erhöhung der volumengewichteten integralen Festigkeit erzielt werden. Lediglich bei Auslagerungszeiten kleiner wie 5 h sind keine Verbesserungen ersichtlich,
- Bei zunehmender Dauer des Lösungsglühens vergrößert sich der Parameterraum, indem integrale Festigkeiten größer 200 MPa erreicht werden,
- Die optimale Auslagerungstemperatur verschiebt sich zu niedrigeren Niveaus;  $100 \text{ °C} < T_{\text{Ausl.}} < 130 \text{ °C}$ ,
- Auslagerungstemperaturen von  $50 \text{ °C}$  erzielen bei entsprechender Dauer eine Festigkeitserhöhung; diese ist jedoch nur sehr gering.

Diesbezüglich kann bestätigt werden, dass das Einfrieren des lösungsgeglühten Zustands einen signifikanten Einfluss auf die Festigkeit des Materials bewirkt und ist durch den übersättigten Mischkristall und die entstehenden Gitterleerstellen bedingt.

#### Zusammenfassend

Die Ergebnisauswertung zeigt auf, dass das lokale Lösungsglühen in Kombination mit dem Abschrecken in Wasser einen vollkommen anderen Einflusscharakter auf die Festigkeitswerte besitzt. Es werden nicht nur die maximal erreichbaren Werte bei kürzerer lokaler Wärmebehandlung erhöht, sondern auch der Parameterraum, der Einfluss auf die Festigkeit besitzt, deutlich vergrößert.

Wie bei der Selbstabkühlung auch besitzt die lokale Wärmebehandlung einen Einflusscharakter auf die Festigkeit. Aus diesen und den vorherigen Untersuchungen kann gefolgert werden, dass die lokale Wärmebehandlung sowohl für das Selbstabkühlen als auch für das Abschrecken in Wasser einen positiven Einfluss auf die Festigkeit des Materials bewirkt.

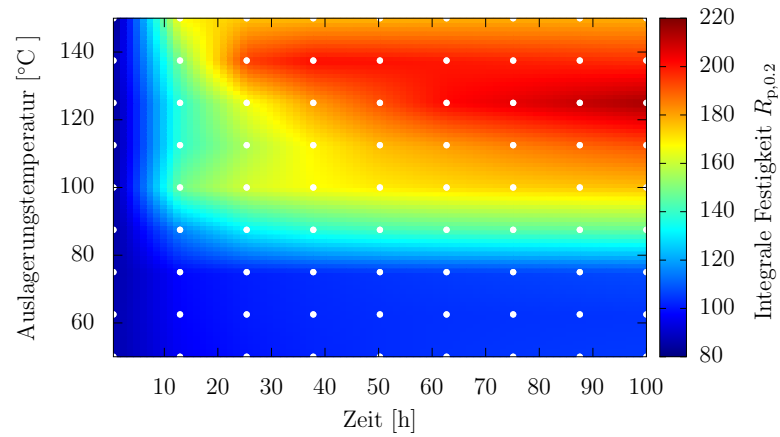
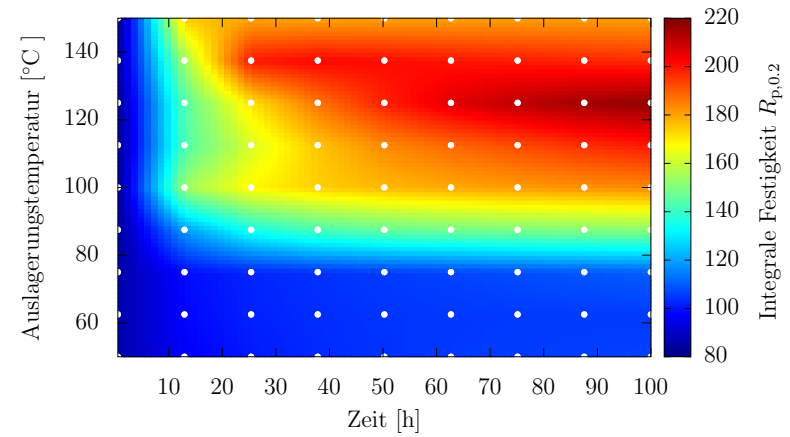
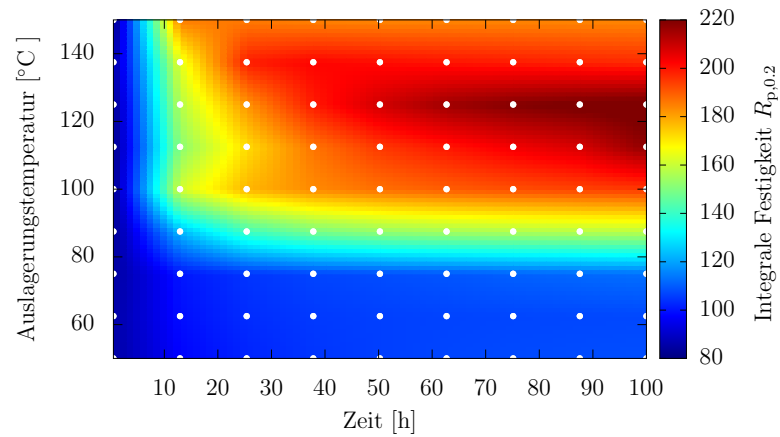
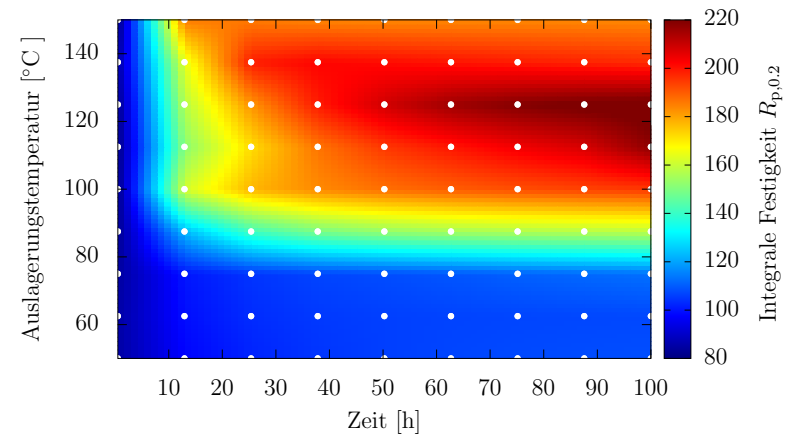
(a)  $t_L = 30$  s(b)  $t_L = 60$  s(c)  $t_L = 120$  s(d)  $t_L = 300$  s

Abbildung 5.19.: Multidimensionalanalyse. Einfluss der Auslagerungszeit und -temperatur auf die integrale Festigkeit; bezogen auf vier verschiedene lokale Lösungsglühzeiten und subsequentem Abschrecken in Wasser.

## 5.6. Optimierung der Auslagerungsparameter

Abschließend wird die Optimierung der volumengewichteten integralen Festigkeit um das Bohrloch nach dem lokalen Lösungsglühfen diskutiert. Es werden zwei der bereits durchgeführten Multidimensionalanalysen als Fallbeispiele herangezogen. Ausgewählt werden die Untersuchungen, bei denen die Dauer der lokalen Wärmebehandlung 120 s und 300 s betragen (Selbstabkühlung).

Die subsequente Auslagerungsoptimierung wird mit den gleichen Parameterraumlimitierungen, welche in Kapitel 5.5 gegeben sind, durchgeführt. Allerdings wird nachfolgend der SOGA-Algorithmus verwendet. Die Methodik wird so justiert, dass der Algorithmus nach maximal elf Populationen die Berechnung stoppt. Jede Population beinhaltet dabei zwölf Chromosomen (Parameter-Paare). Ferner wird das *Unique Random* Verfahren zur Erstellung der ersten Population ausgewählt; die Generierung der ersten zwölf willkürlich zu evaluierenden Punkte.

Die Konvergenzgrenze des SOGA-Algorithmus wird so justiert, sodass die Optimierung beendet wird, wenn die Abweichung der besten integralen Festigkeit von acht nacheinander folgenden Population kleiner als fünf Prozent beträgt. Der Algorithmus stoppt zudem, wenn die Maximalanzahl der Populationen erreicht ist.

### 5.6.1. Ergebnisdiskussion

Die Ergebnisse der Optimierung für die lokale Wärmebehandlung für  $t_L = 120$  s sind in Abbildung 5.20 gegeben. Gezeigt sind die Parametervariationen, Bild 5.20a und 5.20b, bezogen auf die einzelnen Evaluationen. Ferner ist die volumengewichtete integrale Festigkeit der Bohrlochregion in Abbildung 5.20c gegeben. Darstellung 5.20d zeigt den besten Festigkeitswert für jede analysierte Population.

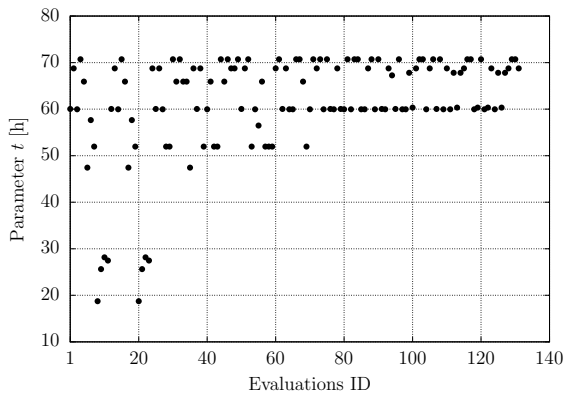
Werden die Parameterräume für die Temperatur und die Auslagerungszeit betrachtet, kann abgeleitet werden, dass sich der Optimierungsalgorithmus zu einem Maximum bewegt. Wertepaare die eine geringe volumengewichtete integrale Festigkeit liefern werden sukzessive ignoriert. Das bedeutet, dass die Auslagerungszeiten unterhalb 60 h nicht weiter beachtet werden und die Auslagerungstemperatur über 130 °C angehoben wird.

Während der optimale Bereich der Auslagerungstemperatur zwischen 130 °C und 140 °C prognostiziert wird, werden zwei Auslagerungszeiten festgehalten. Diese sind bei 60 h und 80 h zu finden. Die korrespondierenden integralen Festigkeitswerte, Abbildung 5.20c, beweisen, dass sich in diesen Intervallen die Materialeigenschaften kaum verändert.

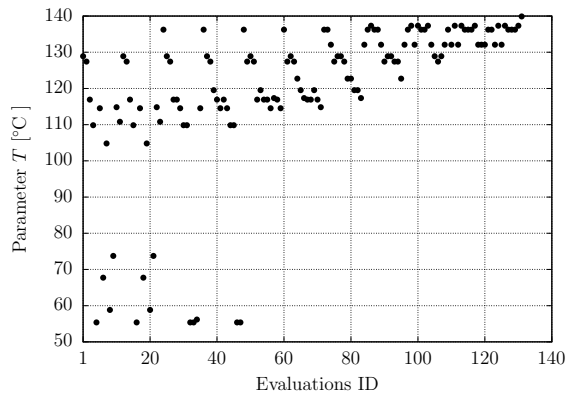
Die Darstellung der besten Ergebnisse pro Population zeigt auf, dass nach der dritten Population ein wesentlich bessere Paarung gefunden wird. Dadurch wird ein Festigkeitsanstieg von über 20 MPa erzielt. Nachfolgende Populationen pendeln um einen Wert von 197 MPa herum, wobei in der siebten Population erneut ein kleiner Anstieg beobachtet werden kann.

Unter Zuhilfenahme der Multidimensionalanalyse, Abbildung 5.17c, kann der Verlauf der Optimierung visuell dargestellt werden; vergleiche Abbildung 5.21a — die einzelnen Populationen sind als schwarze Punkte eingetragen. Da die meisten Populationen ähnliche Festigkeitswerte aufweisen, können lediglich drei Punkte unterschieden werden.

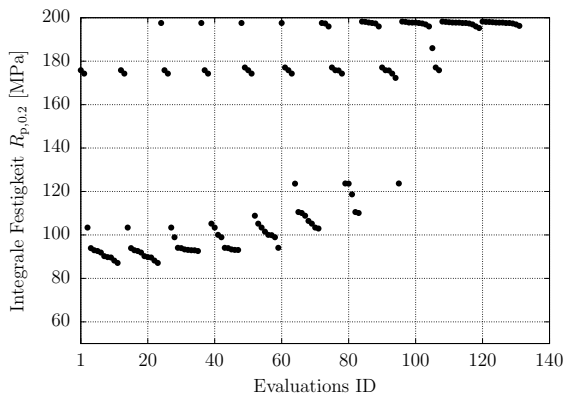
Es ist erwähnenswert, dass die Betrachtung von Abbildung 5.21a die Schlussfolgerung nahe bringt, dass der rechte Punkt — bei circa 72 h und 130 °C — falsch berechnet wurde, da anlässlich der Multi-Dimensionalanalyse an dieser Stelle eine integrale Festigkeit von ungefähr



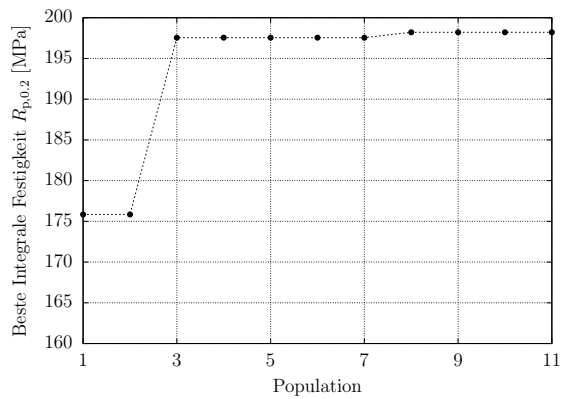
(a)



(b)



(c)



(d)

Abbildung 5.20.: Optimierung der Auslagerungsparameter mithilfe des genetischen Algorithmus SOGA. Lokale Wärmebehandlung  $t_L = 120$  s.

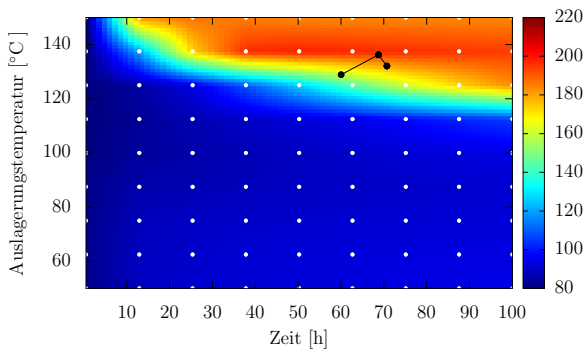
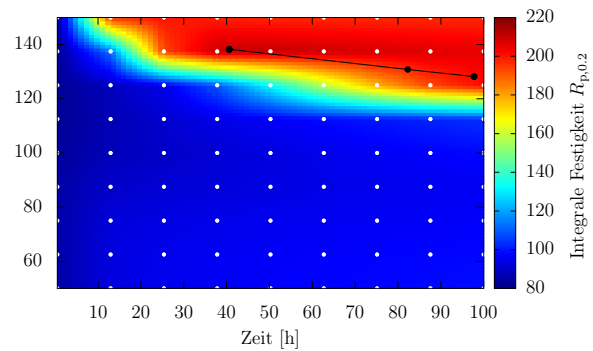
(a) Lokale Wärmebehandlung  $t_L = 120$  s(b) Lokale Wärmebehandlung  $t_L = 300$  s

Abbildung 5.21.: Visuelle Darstellung der Optimierungsverläufe; gezeigt sind die besten Resultate pro Population (schwarze Punkte).

170 MPa abgelesen werden kann. Dies ist dadurch begründet, dass die Multi-Dimensionsanalyse mit diskreten Punkten bestimmt und die Farbskala interpoliert wird. Es kann dementsprechend davon ausgegangen werden, dass hier große Gradienten vorhanden sind; bezogen auf den Anstieg und Abfall der volumengewichteten integralen Festigkeit und ist auf die Komplexität und nicht-Linearität der Materialberechnung zurückzuführen.

Nach der Optimierung wird die beste integrale Festigkeit bei 196,24 MPa mit den Parametern  $t_{\text{Ausl.}} = 254601 \text{ s}$  (70,7 h) und  $T_{\text{Ausl.}} = 132,09 \text{ °C}$  gefunden.

Ein ähnliches Ergebnis wird bei der Optimierung für den zweiten Fall erhalten ( $t_L = 300 \text{ s}$ ). Allerdings kann in diesem Optimierungsdurchgang festgestellt werden, dass der Algorithmus keine Eingrenzung der Auslagerungszeit durchführt. Lediglich die Zeiten unterhalb 25 h werden nach der 30. Evaluation nicht weiter betrachtet. Bei der Auslagerungstemperatur werden erneut die Werte zwischen  $130 \text{ °C}$  und  $140 \text{ °C}$  vorhergesagt, das mit der Multidimensionsanalyse korreliert.

Ferner wird bereits in der ersten Population ein sehr guter Treffer erzielt. Wodurch alle nachfolgenden Evaluationen kaum einen Anstieg der Festigkeit erzielen. In Population 4 und 5 sind kleine Anstiege ersichtlich.

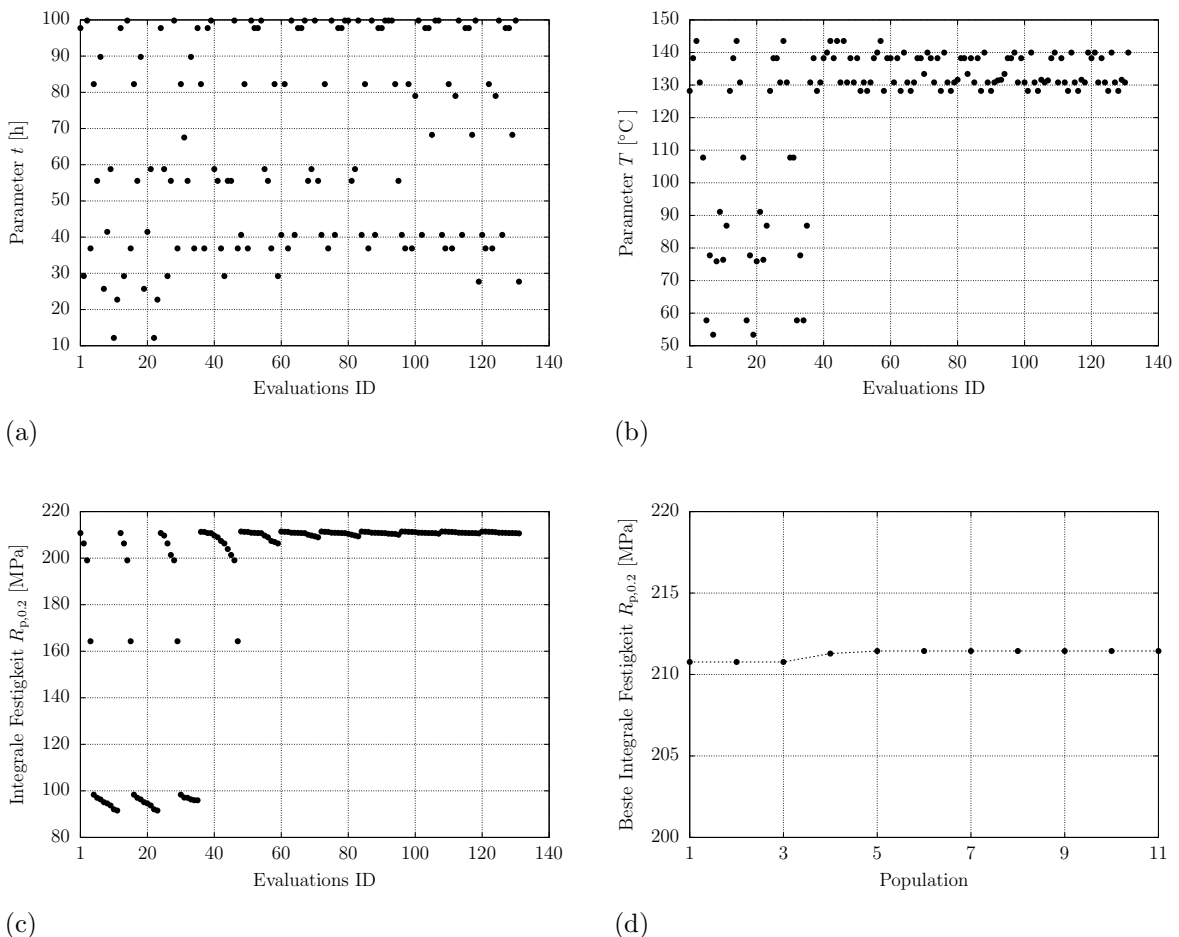


Abbildung 5.22.: Optimierung der Auslagerungsparameter mithilfe des genetischen Algorithmus SOGA. Lokale Wärmebehandlung  $t_L = 300 \text{ s}$ .

### 5.6.2. Zusammenfassend

Durch die Verwendung des genetischen SOGA-Algorithmus kann aufgezeigt werden, dass die Optimierung der zwei Auslagerungsparameter — Temperatur und Zeit — durchgeführt werden kann. Da der genetische Algorithmus die Maxima der integralen Festigkeit um das Bohrloch bestimmt stellt dieser Algorithmus eine wesentliche Verbesserung zur Multi-Dimensionsanalyse dar. Zum einen werden die optimalen Werte der Auslagerungstemperatur und -zeit erhalten und zum anderen ist die Berechnungszeit zwischen SOGA-Optimierung und Multidimensionsanalyse extrem unterschiedlich. Wird der Parameterraum mittels der Multidimensionsanalyse an den 81 diskreten Punkten berechnet, folgt eine Berechnungszeit von 84,3 h. Durch die Verwendung des genetischen SOGA-Algorithmus reduziert sich der Zeitaufwand um circa 73 h auf 11,4 h.



## 5.7. Anwendungsbeispiel für Stahllegierungen (Schweißen)

Abschließend wird ein Beispiel herangezogen, dass die Flexibilität der Prozesskette aufzeigt. Es wird angenommen, dass zwei Stahlplatten durch eine V-Naht verschweißt werden, wobei die Materialänderungen um die Schweißnaht vorhergesagt werden sollen. Das geometrische Modell ist in Abbildung 5.23 dargestellt. Die Umsetzung erfolgt mit dem Open Source CAD System Salome<sup>®</sup>. Die Plattendimensionen werden wie folgt definiert: Stärke 0,004 m, Länge 0,200 m und Breite 0,100 m.

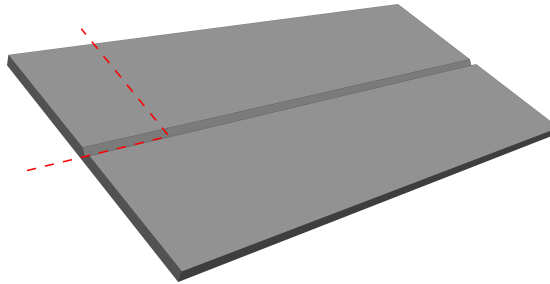


Abbildung 5.23.: Plattenkonfiguration die mit einer V-Naht verschweißt werden sollen. Die rot gestrichelte Linie repräsentiert den Ausschnitt des Netzes, dass in Abbildung F.2 gezeigt ist.

### 5.7.1. Numerisches Netz, Randbedingungen und wichtige Hinweise

Die Diskretisierung des Kontinuums wird mit dem Netzmodul von Salome<sup>®</sup> erzeugt. Das erzeugte Netz unterliegt dem Generierungsalgorithmus, der in Anhang D.3.2 auf 183 beschrieben ist, wodurch ein Tetraedernetz erhalten wird. Anschließend wird eine Netztransformation mittels der Applikation `polyDualMesh` durchgeführt, wodurch das Tetraedernetz in ein Polygonnetz überführt wird. Die Netzausschnitte sind im Anhang Anhang F.2 auf Seite 219 gegeben.

Der LASER wird linear entlang der Naht verfahren. Die LASER Leistung wird so angesetzt, dass die Temperaturen während der Simulation nahe an der Liquidustemperatur ( $T_{\text{sol.}} = 1713 \text{ K}$ ) des Vergütungsstahls (50CrMo4) liegen; das Aufschmelzen selber wird nicht betrachtet und kann mit der hier entwickelten Prozesskette nicht abgebildet werden. Die Dauer des Energieeintrags beträgt 5 s.

Die Materialberechnung wird mittels einem Kinetikskript, das vom Materials Center Leoben (MCL) zur Verfügung gestellt wurde, durchgeführt; in die Modellierungsdetails der Stahllegierung wird nicht eingegangen und angenommen, dass diese korrekt sind. Es soll jedoch darauf hingewiesen werden, dass das Kinetikskript für *drei* Arbeitspunkte (konstante Temperaturen) optimiert ist. Demzufolge kann davon ausgegangen werden, dass bei der Verwendung von zeitlich variierenden Temperaturprofilen falsche Streckgrenzwerte  $R_{p,0.2}$  prognostiziert werden, da beispielsweise keine temperaturabhängigen Transformationsradien für die Phasenumwandlung verwendet wird.

Das numerische Experiment wird ohne Spannungsberechnung durchgeführt. Das LASER-Profil wird verjüngt, wodurch folgende Werte des Gauß'schen Profils angepasst werden:

$$\sigma_x = 0,005 \text{ m} , \quad \sigma_y = 0,005 \text{ m} .$$

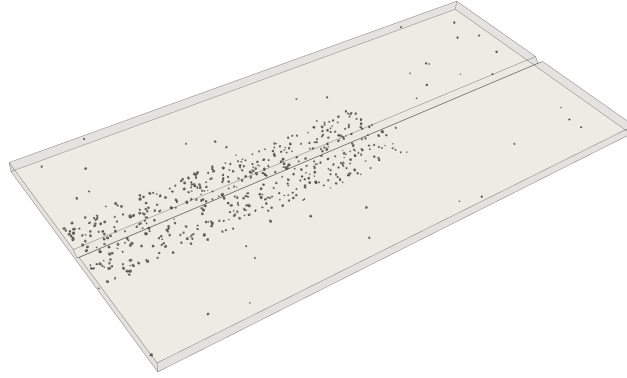


Abbildung 5.24.: Mögliche Random-Cell Verteilung, die mit der `cellListGeneration` Applikation erzeugt wird. Es ist eine klare Verdichtung der Punkte um die Schweißnaht ersichtlich.

Die Bewegung der Energiequelle startet nach 0,5 s und bewegt sich mit einer Geschwindigkeit von  $0,015 \text{ ms}^{-1}$  für fünf Sekunden. Anschließend wird der LASER ausgeschaltet. Entsprechend kühlt das Bauteil, aufgrund von Strahlung und Konvektion, von allen Seiten ab; `laserConvectionBC`.

Es werden insgesamt 550 Random-Cells für die Materialberechnung herangezogen. Davon werden 50 in der Gesamtgeometrie und weitere 500 in der Nähe der Schweißnaht erzeugt. Abbildung 5.24 visualisiert die genannten Zahlen.

Die für die Simulation benötigten Materialdaten werden als Konstant angenommen und betragen:

$$\rho = 7800 \text{ kgm}^{-3}, \quad c_p = 470 \text{ Jkg}^{-1}\text{K}^{-1}, \quad \lambda_{\text{th}} = 42 \text{ Wm}^{-1}\text{K}^{-1}.$$

Die Energiejustierung wird mit den Triggerwerten  $T_{\text{TrigA}} = 1300 \text{ K}$  und  $T_{\text{TrigE}} = 1750 \text{ K}$  umgesetzt — als Referenzwert:  $T_{\text{sol.}} = 1713 \text{ K}$ . Die Intensität wird in den ersten 0,5 s von 0 % auf 100 % angefahren (*Sanftstart*).

Es ist erwähnenswert, dass die in Anhang A.3 erstellte Polynomapproximation für den effektiven Wärmeübergangskoeffizienten  $\alpha_{\text{eff}}$  verwendet wird. Diese muss streng genommen für den vorliegenden Fall angepasst werden. Jedoch wird angenommen, dass das erhaltene Polynom für den vorliegenden Anwendungsfall Gültigkeit besitzt.

### 5.7.2. Lokale Wärmebehandlung (Schweißen)

Die quantitativen Ergebnisse der Temperaturverteilung sind in Abbildung 5.25 für sechs verschiedene Zeiten — 0,5 s, 2,5 s, 5,0 s, 10,0 s, 55,0 s und 120,0 s — dargestellt.

In den Profilen 5.25a, 5.25b und 5.25c kann die Bewegung des LASER's nachvollzogen werden. Es fällt auf, dass die lokal eingetragene Energie, verglichen mit der Aluminiumlegierung, örtlich länger verharrt und ist der geringen Wärmeleitfähigkeit geschuldet. Des Weiteren kann eine rasche Abkühlung der Platten nach dem Abschalten der Energiequelle bei der Betrachtung der Zeitschritte  $t = 5 \text{ s}$  und  $t = 10 \text{ s}$  festgestellt werden. Begründet ist dies durch das im effektiven Wärmeübergangskoeffizienten  $\alpha_{\text{eff}}$  implizit enthaltene Strahlungsgesetz, dass vor allem bei hohen Temperaturen die Wärmeabfuhr bestimmt. Anschließend kühlt die Platte weiter ab, wobei bei niedrigeren Temperaturen die Wärmeabgabe an die Umgebung stets kleiner wird und die Wärmeleitung dominiert. Daraus resultiert eine Art energetisches Stufenprofil, dass bei  $t = 120 \text{ s}$  ersichtlich ist. Subsequent gleicht sich das Energieniveau innerhalb der Platten an und kühlt bis auf Raumtemperatur herunter.

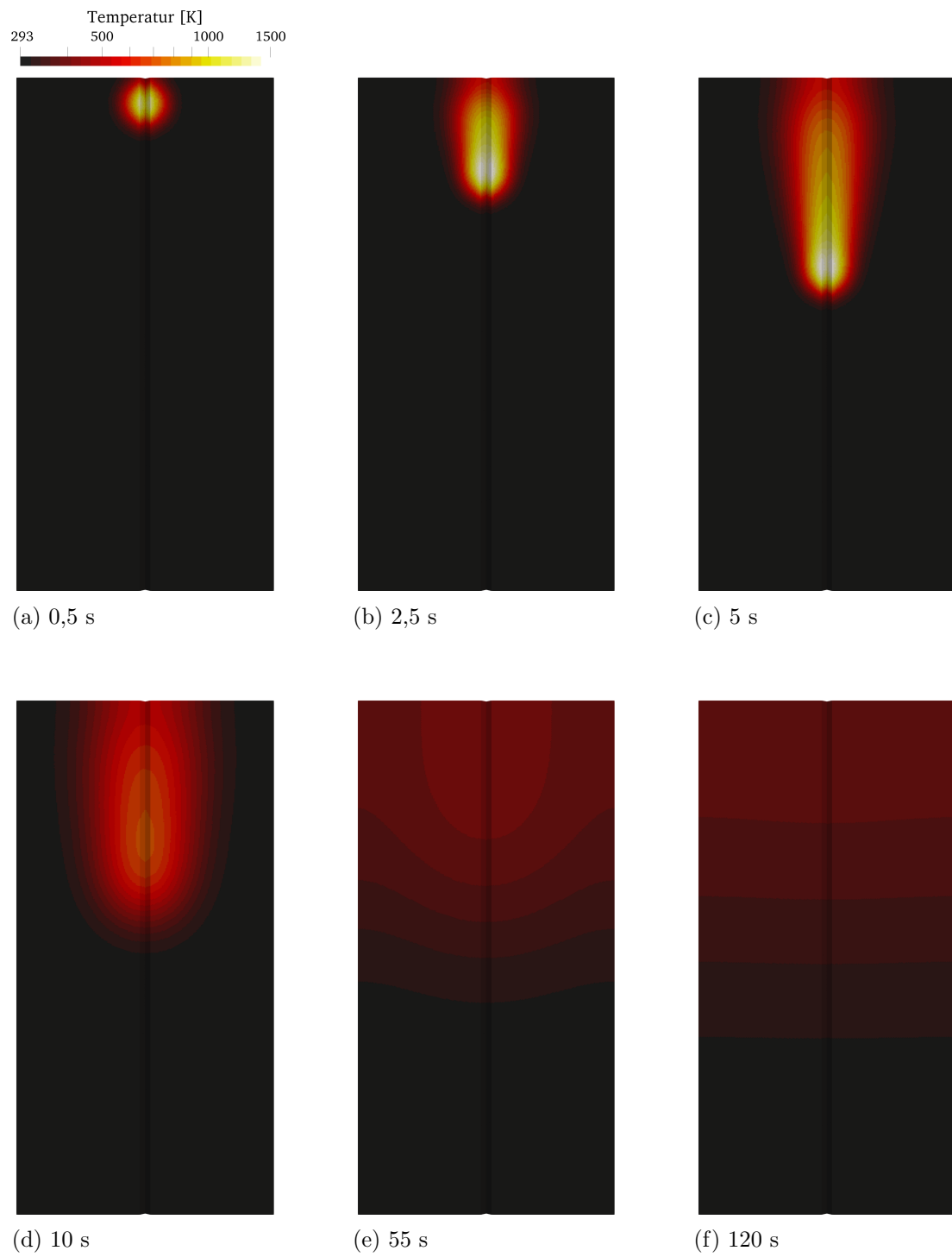


Abbildung 5.25.: Temperaturprofile während des Schweißens und Abkühlens bei verschiedenen Zeitpunkten.

### 5.7.3. Streckgrenzwerte $R_{p,0.2}$ in 3D

Abschließend werden die berechneten Materialwerte in 3D vorgestellt. Abbildung 5.26 gibt weiteren Aufschluss darüber. Es ist deutlich ersichtlich, dass sich durch die lokale Energieeinwirkung das Material um die Schweißnaht verschlechtert. Da die Aufschmelz- und Erstarrungsprozesse nicht abgebildet werden, wird vermutet, dass die Streckgrenzwerte in der V-Naht falsch prognostiziert werden; dies wird dadurch untermauert, dass im Regelfall ein Ausfall beziehungsweise der Bruch nicht an der Naht selbst, sondern immer in der Nähe dieser beobachtet wird; die Hauptgründe sind jedoch oftmals die Kerbwirkung, die nach dem Schweißen zugrunde liegen und nicht die Schwächung des Materials.

Die berechneten Werte der Streckgrenze in den Regionen, in denen keine Temperatureinwirkung stattgefunden hat —  $1500 \text{ MPa} \leq R_{p,0.2} \leq 2000 \text{ MPa}$  —, werden quantitativ zu hoch vorhergesagt. Diese Aussage wird durch die von Brnic et al. (2010) erwähnten Maximalwerte dieser Legierung von circa 1200 MPa bestätigt. Allgemeine kann jedoch keine Aussage bezüglich Validität der Ergebnisse getätigt werden.

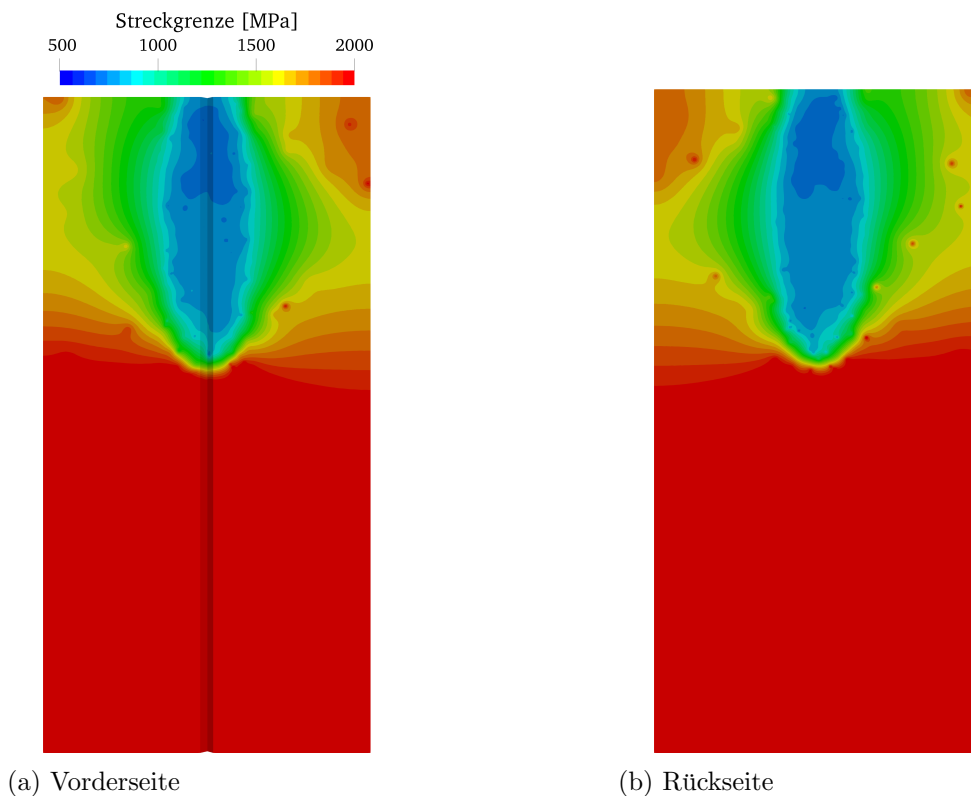


Abbildung 5.26.: Qualitatives Profil der Streckgrenze  $R_{p,0.2}$  des Vergütungsstahls nach dem Schweißen in 3D der Vorder- und Rückseite.

### 5.7.4. Abschließend

Es wurde aufgezeigt, dass die Prozesskette flexibel einsetzbar ist und realistische Materialwerte prognostizieren kann, sofern die Kinetikberechnung für willkürliche Temperaturprofile die Realität präsentiert. Des Weiteren wurde die Möglichkeit der Energiequellenbewegung aufgezeigt und die Flexibilität bezüglich dem numerischen Netz indirekt bestätigt; Verwendung eines Polyedernetzes. Aus diesen Gründen kann festgehalten werden, dass die entwickelte Prozesskette auf eine Vielzahl von Problemstellungen angepasst werden kann.

## 6. Erlangte Erkenntnisse und Aussicht

In der folgenden Arbeit wird in drei verschiedene Themenbereiche intensiv eingestiegen und eine automatisierte Prozesskette für lokale Wärmebehandlungen entwickelt. Die wichtigsten Ergebnisse und Erkenntnisse sind nachfolgend gegeben:

- Während der Validierung, die zur Prüfung der korrekt implementierten und hergeleiteten Duhamel-Neumann Gleichung unter Zuhilfenahme der FVM betrachtet wird, kann aufgezeigt werden, dass die Ergebnisse der FVM Berechnung quantitativ in exzellenter Übereinstimmung mit analytischen Lösungen stehen. Des Weiteren kann bestätigt werden, dass die FVM im Bereich der Spannungsberechnung identische Werte zu ihrem Konkurrenten der FEM erzielt, unter Berücksichtigung wichtiger numerischer Aspekte.
- Die numerische Vorhersage der Materialeigenschaften für lokale oder nicht-homogene Wärmebehandlungen für die AlSi7Mg0.3 Legierung wird mithilfe verschiedener Berechnungsskripte realisiert. Ferner werden Unsicherheiten verschiedener Phasenparameter aufgezeigt; vergleiche Tabelle 3.4. Die numerisch Modellierung der Materialeigenschaften ist maßgeblich abhängig durch die im Material vorliegenden Phasen. Es wird aufgezeigt, dass die Phasentransformation entscheidend für die korrekte Vorhersage der Materialeigenschaften ist. Da die in dieser Arbeit verwendete Software kein physikalisches und/oder thermodynamisches Modell für die Phasentransformation bereitstellt und auch in der Literatur keine Modelle diesbezüglich postuliert werden, wird das einfache Radien-Transformationsmodell herangezogen, wobei temperaturabhängige Transformationsradien hergeleitet und angewendet werden. Dieses Verfahren stellt einen Kompromiss zwischen Genauigkeit und Einsatzbereich dar und ist vor allem dann wichtig, wenn willkürliche Temperaturprofile zur Materialberechnung herangezogen werden.
- Die Kopplung der einzelnen Themenbereiche — thermische Spannungsmodellierung, Materialberechnung und Optimierung — erlaubt es die Prozesskette numerisch abzubilden. Die Durchführung der Modellierung wird mit der neue entwickelten Simulationskette durchgeführt. Diese beinhaltet alle Kommunikationsschnittstellen, welche zur Analyse von lokalen oder nicht-homogenen Wärmebehandlungen oder verwandten Prozessen, wie dem Schweißen, benötigt werden. Ferner wird die Prozesskettenmodellierung ausschließlich mit Open Source Programmen oder Eigenentwicklungen umgesetzt. Lediglich die Materialberechnung wird durch ein kommerzielles Programm modelliert.
- Die lokale Wärmebehandlung führt zu inhomogenen Materialeigenschaften, basierend auf unterschiedlichen kinetischen und thermodynamischen Vorgängen im Material. Die Vorhersage der Streckgrenze  $R_{p,0.2}$  in 3D, unter Verwendung der klassischen Modellierungstheorien — anwendbar auf 0D —, liefert die Methodenentwicklung, die als Random-Cell-Poisson Methode bezeichnet wird; vergleiche Kapitel 3.7.3.

- Durch die neue Random-Cell-Poisson Methode ist es möglich, integrale Festigkeiten überall im Bauteil zu bestimmen. Die Berechnungsvariation ist uneingeschränkt und erlaubt es — durch die Kenntnis des 3D Profils — integrale Festigkeiten an interessierenden Regionen zu bestimmen. Hierbei kann eine Vielzahl von Berechnungsvarianten herangezogen werden. In dieser Arbeit wird das volumengewichtete Integrationsverfahren herangezogen. Die integrale Festigkeit um die lokale Lösungsglühzone wird, beziehungsweise kann anschließend als Zielfunktion für einen Optimierer verwendet.
- Die Verwendung der Simulationskette zeigte auf, dass eine lokale Wärmebehandlung von Aluminiumlegierungen machbar aber auch Design abhängig ist, sofern plastische Verformung und das Aufschmelzen der Legierung verhindert werden sollen. Im Falle der Dämpferbrücke wird aufgezeigt, dass aufgrund der Struktur des Bauteils, sehr hohe Spannungen auftreten, die dazu führen, die Intensität der Energiequelle zu reduzieren, wodurch keine hohen lokalen Temperaturwerte erreicht werden können. Des Weiteren wird ersichtlich, dass lokale Wärmebehandlungen, die strikt örtlich begrenzt sein müssen, nicht durchführbar sind. Der Grund hierfür ist die hohe Wärmeleitfähigkeit von Aluminiumlegierungen.
- Während der Analyse des lokalen Lösungsglühens wird aufgezeigt, dass diese einen positiven Einfluss auf die Festigkeit des Materials besitzt. Die verschiedenen Lösungsglühzeiten zeigen weiter auf, dass die Dauer des inhomogenen Energieeintrags einen entscheidenden Einflussfaktor auf die nachfolgende Auslagerung besitzt. Je länger dieser andauert, desto höhere Materialfestigkeiten können erzielt werden. Auf wirtschaftliche und ökonomische Aspekte wurde in dieser nicht eingegangen. Aus den erhaltenen Ergebnissen ist jedoch ersichtlich, dass kurze Auslagerungszeiten (10 h bis 30 h) bei gleichzeitig hoher Auslagerungstemperatur ( $> 130\text{ °C}$ ) ähnliche Festigkeitswerte erzielt, wie wenn eine Auslagerungszeit von 60 h oder länger verwendet wird.
- Die Gegenüberstellung der Selbstabkühlung und dem Abschrecken in Wasser nach dem lokalen Lösungsglühen beweist, dass das Einfrieren des lösungsgeglühten Zustands einen erheblichen Einfluss auf die zu erreichenden Festigkeitswerte besitzt. Hinzu kommt dass die Auslagerungsparameter beim Abschrecken in Wasser wesentlich freier gewählt werden können, um hohe Festigkeitswerte zu erzielen.
- Durch die Analyse des Spannungsverlaufs im Motorgehäuse während der Selbstkühlung wird die Erkenntnis erlangt, dass keine plastische Verformung eintritt. Die maximale von Mises Spannung im Design ist stets kleiner als 80 MPa. Wird jedoch das Bauteil in Wasser abgeschreckt, werden kurzzeitig erhöhte Spannungswerte registriert. Hierbei kann es zu plastischer Verformung kommen, dass allerdings aufgrund der verwendeten Grundgleichung (Duhamel-Neumann) nicht vorhergesagt werden kann. Lediglich das Übertreten der Fließgrenze kann hierbei aufgezeigt werden.
- Ferner wird durch die Verwendung des genetischen SOGA-Algorithmus bestätigt, dass die Optimierungsmethode das Maximum der integralen Festigkeit um die Bohrung korrekt vorhersagt. Die numerische Berechnung kann — im Vergleich zur Multidimensionsanalyse — damit um circa 72 h von 83,3 h auf 11,4 h verkürzt werden.

- Die hohe Flexibilität der Prozesskettensimulation wird mit dem Beispiel des Schweißens zweier Platten aus Vergütungsstahl aufgezeigt. Die Vorhersage der Materialeigenschaften können mit generellen Beobachtungen in Einklang gebracht werden, allerdings ist hier auf die Unzulänglichkeit des Materialberechnungsskriptes hinzuweisen. Des Weiteren wird gezeigt, dass die entwickelte Randbedingung `laserConvectionBC` bewegte Energiequellen abbilden kann.

## Ausblick

Die während der Arbeit aufgetauchten Kernprobleme müssen in weiterführenden Untersuchungen geklärt werden, sodass Korrektheit der vorhergesagten Materialeigenschaften sichergestellt werden kann. Folgende Punkte werden als mögliche Themenbereiche für nachfolgende Untersuchungen vorgeschlagen:

- Die Phasentransformation muss numerisch, unter Verwendung neu entwickelter Modelle die auf physikalischen und/oder thermodynamischen Grundlagen aufbauen, abbildbar sein. Das hier verwendete Phasentransformationsmodell kann ohne weitere Untersuchungen weder als falsch noch richtig eingestuft werden.
- Das Auffinden eines globalen Maximums der integralen Festigkeit um das Bohrloch kann gegebenenfalls mit anderen genetischen oder hybriden Optimierungsalgorithmen schneller und effizienter gefunden werden — beispielsweise der genetische Gauß Algorithmus.
- Die Optimierung der Auslagerungsparameter unter Berücksichtigung wirtschaftlicher und/oder ökonomischer Aspekte. Hierbei müssen weitere Objektfunktionen aufgestellt und analysiert werden.
- Die lokale Wärmebehandlung kann möglicherweise an Bedeutung gewinnen, wenn das Bauteil teilweise in einem Fluid gekühlt wird. Die Analysemöglichkeit und Flexibilität der Prozesskette kann hierbei ausgenutzt werden.
- Die mögliche Verformung des Motorgehäuses beim Abschrecken in Wasser muss mit anderen numerischen Modellen beziehungsweise Gleichungen betrachtet werden. Dies ist dadurch begründet, da die Gültigkeit der Duhamel-Neumann Formulierung im linear-elastische Regime vorliegt. Daher kann das Eintreten einer plastischen Verformung nicht bestätigt werden.
- Weitere Untersuchungen mit anderen Strukturbauteilen die Praxis relevanter sind und gegebenenfalls andere Legierungen können für die Untersuchung von lokalen Wärmebehandlungen in Betracht gezogen werden.





# Literaturverzeichnis

- B. M. Adams, L. E. Bauman, W. J Bohnhoff, K. R. Dalbey, M. S. Ebeida, J. P. Eddy, M. S. Eldred, P. D. Hough, K. T. Hu, J. D. Jakeman, J. A. Stephens, L. P. Swiler, D. M. Vigil, und T. M. Wildey. *Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.0 User's Manual*. 2015. Updated November 2015 (Version 6.3).
- B. M. Adams, L. E. Bauman, W. J Bohnhoff, K. R. Dalbey, M. S. Ebeida, J. P. Eddy, M. S. Eldred, P. D. Hough, K. T. Hu, J. D. Jakeman, J. A. Stephens, L. P. Swiler, D. M. Vigil, und T. M. Wildey. *Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Theory Manual*. 2017.
- S. Agrawal und S. Silakari. Frpso: Fletcher–reeves based particle swarm optimization for multimodal function optimization. *Soft Computing*, 18:2227–2243, 2014.
- E. Aguirre-De la Torre, U. Afeltra, C. D. Gómez-Esparza, J. Camarillo-Cisneros, R. Pérez-Bustamante, und R. Martínez-Sánchez. Grain refiner effect on the microstructure and mechanical properties of the a356 automotive wheels. *Journal of Materials Engineering and Performance*, 23:581–587, 2014. doi: 10.1007/s11665-013-0596-x.
- M. Ahmadi, B. Sonderegger, E. Povoden-Karadeniz, A. Falahati, und E. Kozeschnik. Precipitate strengthening of non-spherical precipitates extend in  $\langle 100 \rangle$  or 100 direction in fcc crystals. *Materials Science and Engineering A*, 2014a.
- M.R. Ahmadi, E. Povoden-Karadeniz, K. I. Öksüz, A. Falahati, und E. Kozeschnik. A model for precipitation strengthening in multi-particle systems. *Computational Material Science*, 91: 173–186, 2014b. doi: 10.1016/j.commatsci.2014.04.025.
- J. Anderson. *Computational Fluid Dynamics - The Basics With Applications*. 1995.
- D. Apelian, S. Shivkumar, und G. Sigworth. Fundamental aspects of heat treatment of cast al-si-mg alloys. *AFS Transaction*, 97:727–742, 1989.
- L. Bäckerud, G. Chai, und J. Tamminen. *Solidification Characteristics of Aluminium Alloys*, volume 2. AFS/SKANAluminium, 1990.
- M. de Berger, O. Cheong, M. van Kreveld, und M. Overmars. *Computational Geometry*, volume 3. Springer, 2008.
- A. K. Bhunia, P. Pal, und S. Chattopadhyay. A hybrid of genetic algorithm and fletcher-reeves for bound constrained optimization problems. *Decision Science Letters*, 4:125–136, 2015.

- K. Binder und D. Stauffer. Theory for the slowing down of the relaxation and spinodal decomposition of binary mixtures. *Physical Review Letters*, 33(17):1006–1009, 1974. doi: 10.1103/PhysRevLett.33.1006.
- R. B. Bird, W. E. Stewart, und E. N. Lightfoot. *Transport Phenomena*. John Wiley & Sons, Madison, Wisconsin, June 1960.
- B. A. Boley und J. H. Weiner. *Theory of Thermal Stresses*. Robert E. Krieger Publishing Company, 1985.
- E. Borchi, M. Bruzzi, und U. Biggeri. Elastic moduli of polycrystalline diamond. In *Diamond, Diamond-like and Related materials*, 1994.
- K. Borodianskiy, V. Selivorstov, Y. Dotsenko, und M. Zinigrad. Effect of additions of ceramic nanoparticles and gas-dynamic treatment on al casting alloys. *Metals*, 5:2277–2288, 2015. doi: 10.3390/met5042277.
- J. Brnic, M. Canadija, G. Turkalj, und D. Lanc. 50crmo4 steel-determination of mechanical properties at lowered and elevated temperatures, creep behavior, and fracture toughness calculation. *Journal of Engineering Materials and Technology*, 132, 2010. doi: 10.1115/1.4000669.
- C. H. Caceres, C. J. Davidson, J. R. Griffiths, und Q. G. Wang. The effect of mg on the microstructure and mechanical behavior of al-si-mg casting alloys. *Metallurgical and Materials Transactions A*, 1998.
- P. Cardiff. *Development of the Finite Volume Method for Hip Joint Stress Analysis*. PhD thesis, National University of Ireland, Oktober 2012.
- P. Cardiff, Z. Tukovic, H. Jasak, und A. Ivankovic. A block-coupled finite volume method for linear elastic and unstructured meshes. *Computers and Structures*, 175:100–122, 2016.
- CASCADE. The open source integration platform for numerical simulation, June 2017. URL [www.salome-platform.org/](http://www.salome-platform.org/).
- Forum CFD-Online. An online forum center for computational fluid dynamics and numerical data manipulation, 2018. URL <https://www.cfd-online.com/Forums/openfoam-solving/117901-pimplefoam-full-transient-problems.html>.
- H.-P. Cheng, R. S. Berry, und R. L. Whetten. Electronic structure and binding energies of aluminum clusters. *Physical Review B*, 43(13):10647–10653, 1991. doi: 10.1103/PhysRevB.43.10647.
- L. J. Colley. *Microstructure-Property Models For Heat Treatment Of A356 Aluminum Alloy*. PhD thesis, The University of British Columbia (Vancouver), 2011.
- J. A. Dantzig und M. Rappaz. *Solidification*. EPFL Press, CH-1015 Lausanne, Switzerland, first edition, 2009.
- K. Dash, S. Sukumaran, und B. C. Ray. The behaviour of aluminium matrix composites under thermal stresses. *Sci Eng Compos Mater*, 23:1–20, 2016. doi: 10.1515/secm-2013-0185.

- I. Demirdžić und S. Muzaferija. Numerical method for coupled fluid flow, heat transfer and stress analysis using unstructured moving meshes with cells of arbitrary topology. *Computer Methods in applied mechanics and engineering*, 125:235–255, 1995. doi: [https://doi.org/10.1016/0045-7825\(95\)00800-G](https://doi.org/10.1016/0045-7825(95)00800-G).
- A DIN EN 1706:2013-12. Aluminium und aluminiumleraluminium - gussstücke - chemische zusammensetzung und mechanische eigenschaften (din en 1706:2013-12). Deutsche Institut für Normung, Dezember 2013.
- A DIN EN 573-1:2005-02. Aluminium und aluminiumlegierungen - chemische zusammensetzung und form von halbzeug - teil 1: Numerisches bezeichnungssystem. Deutsche Institut für Normung, Februar 2005.
- I. Dutta und S. M. Allen. A calorimetric study of precipitation in commercial aluminium alloy 6061. *Journal of Material Science Letters*, 10:323–326, 1991.
- B. Dybowski, B. Adamczyk-Cieślak, K. Rodak, I. Bednarczyk, A. Kielbus, und J. Mizera. The microstructure of als7mg alloy in as cast condition. *Solid State Phenomena*, 229:3–10, 2015.
- F. J. H. Ehlers, S. Wenner, S. J. Andersen, C. D. Marioara, W. Lefebvre, C. B. Boothroyd, und R. Holmestad. Phase stabilization principle and precipitate-host lattice influence for al-mg-si-cu alloy precipitates. *Journal of Material Science*, 49:6413–6426, 2014.
- E. Eiper, R. Resel, C. Eisenmenger-Sitter, M. Hafok, und J. Keckes. Thermal-induced stresses in thin aluminum layers grown on silicon. *Powder Diffraction*, 47, 2004.
- A. Falahati, E. Povoden-Karadeniz, P. Lang, und E. Kozeschnik. Thermo-kinetic computer simulation of differential scanning calorimetry curves of almg-si alloys. *Material Science and Technology*, 101:1089–1096, 2010. doi: 10.3139/146.110396.
- A. Falahati, M. R. Ahmadi, P. Lang, E. Povoden-Karadeniz, P. Warczok, und E. Kozeschnik. Thermo-kinetic computer simulation of precipitation and age-hardening effect in al-mg-si alloys with arbitrary heat treatment. *Materials Science and Technology*, pages 292–299, 2011.
- F. Fazeli und C. W. Sinclair. The role of excess vacancies on precipitation kinetics in an al-mg-sc alloy. *Metallurgical and Materials Transactions*, 2008.
- J. H. Ferziger und M. Perić. *Numerische Strömungsmechanik*. Springer-Verlag, Heidelberg, Berlin, second edition, 2008.
- F. D. Fischer, J. Svoboda, F. Appel, und E. Kozeschnik. Modeling of excess vacancy annihilation at different types of sinks. *Acta Materialia*, 59:3463–3472, 2011. doi: 10.1016/j.actamat.2011.02.020.
- F. D. Fischer, J. Svoboda, und E. Kozeschnik. Interstitial diffusion in systems with multiple sorts of traps. *Modelling and Simulation in Materials Science and Engineering*, 21, 2013. doi: 10.1088/0965-0393/21/2/025008.
- F. D. Fischer, J. Svoboda, T. Antretter, und E. Kozeschnik. Relaxation of precipitate misfit stress state by creep in the matrix. *International Journal of Plasticity*, 64:164–176, 2015.

- E. Flender. Magmasoft material datenbank, 2015. URL <http://www.magma-soft.de>.
- R. Fletcher und C. M. Reeves. Function minimization by conjugate gradients. *The Computer Journal*, 7:149–154, 1964. doi: 10.1093/comjnl/7.2.149.
- Community Foam Extend. The unofficial foam extend project, 2016. URL <https://github.com/Unofficial-Extend-Project-Mirror/foam-extend-foam-extend-4.0>.
- OpenFOAM® Foundation. C++ source code guide, 2018. URL <https://cpp.openfoam.org/v4/>.
- M. Galassi, J. Davies, J. Theiler, B. Gough, Ge. Jungman, P. Alken, M. Booth, und F. Rossi. *GNU Scientific Library Reference Manual*. Network Theory Ltd, third edition, January 2009.
- S. Ganeshan, S. L. Shang, H. Zhang, Y. Wang, M. Mantina, und Z. K. Liu. Elastic constants of binary mg compounds from first-principles calculation. *Intermetallics*, 17:313–318, 2009.
- P. L. George und E. Seveno. The advancing-front mesh generation method revised. *International Journal for Numerical Methods in Engineering*, 37:3605–3619, 1994.
- V. Gnielinski, S. Kabelac, M. Kind, H. Martin, D. Mewes, K. Schaber, und P. Stephan. *VDI - Wärmeatlas*. Verein Deutscher Ingenieure VDI-Gesellschaft Verfahrenstechnik und Chemieingenieurwesen (GVC), 10 edition, 2005.
- H. Goering, L. Tobiska, und H-G. Roos. *Die Finite-Elemente-Methode*. WILEY-VCH Verlag GmbH Weinheim, 2010.
- C. J. Greenshields. *OpenFOAM® Programmers Guide*. OpenFOAM Foundation Ltd., version 3.0.1 edition, December 2015. URL <http://foam.sourceforge.net/docs/Guides-a4/ProgrammersGuide.pdf>.
- C. J. Greenshields und H. G. Weller. A unified formulation for continuum mechanics applied to fluid-structure interaction in flexible tubes. *International Journal for Numerical Methods in Engineering*, 64:1575–1593, July 2005.
- Christopher J. Greenshields. The openfoam user guide, 2017. URL <http://foam.sourceforge.net/docs/Guides-a4/OpenFOAMUserGuide-A4.pdf>.
- H-J. Gudladt. Klassierung von aluminiumlegierungen. Bundeswehr Universität München, 2017. URL <https://www.unibw.de/lrt5/>.
- M. E. Gurtin, E. Fried, und L. Anand. *The Mechanics and Thermodynamics of Continua*. 2010.
- K. Gustafson und A. Takehisa. The third boundary condition - was it robin's? *Mathematical Intelligencer*, 20(1):63–71, March 1998. doi: 10.1007/BF03024402.
- H. Hajkowski, L. Bernat, und J. Hajkowski. Mechanical properties of al-si-mg alloy castings as a function of structure refinement and porosity fraction. *Archives of Foundry Engineering*, 12:57–64, 2012.
- D. Heiny und V. Dölle. Cfd, fea and thermal simulation in the cloud, 2018. URL <https://www.simscale.com/projects/>.

- T. Holzmann. Numerical schemes. Holzmann CFD, 2013. URL [www.Holzmann-CFD.de/index.php/en/numerical-schemes](http://www.Holzmann-CFD.de/index.php/en/numerical-schemes).
- T. Holzmann. *Mathematics, Numerics, Derivations and OpenFOAM*. Holzmann CFD, Leoben, fourth edition, October 2016a. doi: 10.13140/RG.2.2.27193.36960. URL [www.holzmann-cfd.de](http://www.holzmann-cfd.de).
- T. Holzmann. C++ laser boundary class. Montanuniversität Leoben, September 2016b. URL <https://bitbucket.org/shor-ty/laserconvectionbc>.
- T. Holzmann, A. Ludwig, und P. Raninger. Yield strength prediction in 3d during local heat treatment of structural a356 alloy components in combination with thermal-stress analysis. In G. Lambotte, J. A. Allanore, und S. Wagstaff, editors, *Materials Processing Fundamentals 2018*, pages 77–85. Springer, 2018. doi: 10.1007/978-3-319-72131-6\_7.
- M. A. Hopcroft, W. D. Nix, und T. W. Kenny. What is the young’s modulus of silicon? *Journal of Microelectromechanical Systems*, 2010.
- E. Hornbogen, A. K. Mukhopadhyay, und E. A. Starke. Nucleation of the diamond phase in aluminium-solid solutions. *Journal of Material Science*, 28(13):3670–3674, 1993. doi: 10.1007/BF01159852.
- G. Jakesch. Möglichkeiten zur steuerung der dauerfestigkeit in lokalen gussflächen bei aluminium. Diplomarbeit, Montanuniversität Leoben, Franz-Josef-Strasse 18, Montanuniversität Leoben, Austria, May 2015.
- H. Jasak. *Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows*. PhD thesis, University of London, 1996.
- H. Jasak und H. G. Weller. Application of the finite volume method and unstructured meshes to linear elastics. *International Journal for Numerical Methods in Engineering*, August 1998. doi: 10.1002/(SICI)1097-0207(20000520)48:23.0.CO;2-Q.
- H. Jasak, A. Jemcov, und Ž Tuković. Openfoam: A c++ library for complex physics simulations. In Zdravko Terze und Chris Lacor, editors, *Coupled Methods in Numerical Dynamics*, pages 47–67. Faculty of Mechanical Engineering and Naval Architecture, 2007.
- W. Jiang, Z. Faz, und D. Liu. Microstructure, tensile properties and fractography of a356 alloy under as-cast and t6 with expendable pattern shell casting process. *Transaction of Nonferrous Metals Society China*, 22:s7–s13, 2012.
- Z. Q. Jiang, Y. Huang, und A. Chandra. Thermal stresses in layered electronic assemblies. *Journal of Electronic PackagingMat*, 119, 1997. doi: 10.1115/1.2792218.
- M. Kahrom und S. M. Javadi. Application of single objective genetic algorithm to optimize heat transfer enhancement from a flat plate. *International Journal of Engineering Transactions C*, 25:67–78, 2012. doi: 10.5829/idosi.ije.2012.25.01c.08.
- N. Kasuya, T. Nakazawa, A. Matsushita, T. Okane, und M. Yoshida. Mechanical properties of a partial solidified cu-zn alloy. *Metallurgical and Materials Transactions A*, 47:1661–67, 2016. doi: 10.1007/s11661-016-3353-6.

- J. Kavosi, M. Saei, M. Kazeminezhad, und A. Daodangeh. Modeling of dislocation density and strength on rheoforged a356 alloy during multi-directional forging. *Computational Materials Science*, 81:284–289, 2014.
- D. Klobč und J. Tušek. Thermal stresses in aluminium alloy die casting dies. *Computational Materials Science*, 43:1147–54, 2008. doi: 10.1016/j.commatsci.2008.03.009.
- E. Kozeschnik. The scientific tool box matcalc 6, September 2017. URL <http://matcalc.tuwien.ac.at/>.
- E. Kozeschnik, J. Svoboda, P. Fratzl, und F. D. Fischer. Modelling of kinetics in multi-component multi-phase systems with spherical precipitates: Ii: Numerical solution and application. *Materials Science and Engineering, A: Structural Materials: Properties, Microstructure and Processing*, 385:157–168, November 2004. doi: 10.1016/j.msea.2004.06.016.
- E. Kozeschnik, J. Svoboda, und F. D. Fischer. Shape factors in modeling of precipitation. *Materials Science and Engineering A*, 441:68–72, 2006.
- H. Kuchling. *Taschenbuch der Physik*. Carl Hanser Verlag, 2007.
- P. Kuzmanov, R. Dimitrova, R. Lazarova, A. Cherepanov, S. Popov, R. Petrov, und V. Manolov. Investigation of the structure and mechanical properties of castings of alloy als7mg, cast irons gg15 and gg25 and steel gx120mn12, modified by nanosized powders. *Journal of Nanoengineering and Nanosystems*, 228(1):11–18, 2014.
- Sandia National Laboratories. Surrogate models. Sandia National Laboratories. URL <https://dakota.sandia.gov/training/dakota-training-materials>.
- P. Lang, E. Povoden-Karadeniz, W. Mayer, A. Falahati, und E. Kozeschnik. The bustling nature of vacancies in al alloys. In F. Marquis, editor, *Proceedings of the 8th Pacific Rim International Congress on Advanced Materials and Processing*, pages 3181–3188, 2013. doi: 10.1007/978-3-319-48764-9\_392.
- B. Li, H. C. Wang, J. C. Jie, und Z. J. Wei. Microstructure evolution and modification mechanism of the ytterbium modified al–7.5%si–0.045%mg alloys. *Journal of Alloys and Compounds*, 509: 3387–3392, 2011. doi: 10.1016/j.jallcom.2010.12.081.
- M. Li, Y. Li, X. Huang, Y. Ma, und R. Guan. Secondary solidification behavior of a356 aluminium alloy prepared by the self-inoculation method. *Metals*, 7, 2017.
- J. Lienig und H. Brümmer. *Elektronische Gerätetechnik*. Springer Verlag, 2014. doi: DOI10.1007/978-3-642-40962-2\_5.
- Y. L. Liu, S. B. Kang, und H. W. Kim. The complex microstructures in an as-cast al-mg-si alloy. *Materials Letters*, 41:267–272, 1999.
- A. Ludwig und A. Holzmann. Automatic optimization of localized heat treatment for al-si-mg alloys. *Material Science and Engineering*, 119, June 2016a. doi: doi:10.1088/1757-899X/119/1/012026. URL <http://iopscience.iop.org/article/10.1088/1757-899X/119/1/012026/pdf>.

- A. Ludwig und T. Holzmann. Localized strengthening of al-based casting alloys by automatized optimization of laser heat treatment. In *CFD Modeling and Simulation in materials Processing*, pages 125–132. Wiley Online Library, 2016b. doi: doi:10.1002/9781119274681.ch16.
- W. MacCormack, O. R. Tutty, E. Rogers, und P. A. Nelson. Stochastic optimisation based control of boundary layer transition. *Control Engineering Practice*, 10:243–260, 2002. doi: 10.1016/S0967-0661(01)00140-X.
- R. I. Mackay und J. E. Gruzleski. Quantification of magnesium in 356 alloy via thermal analysis. *International Journal of Cast Metals Research*, 10:255–265, 1998.
- M. Mantina, Y. Wang, Z. K. Chen L. Q. Liu, und C. Wolverton. First pprinciple impurity diffusion coefficients. *Acta Materialia*, 57:4102–4108, 2009. doi: 10.1016/j.actamat.2009.05.006.
- C. D. Marioara, H. Nordmark, S. J. Andersen, und R. Holmestad. Post- $\beta''$  phases and their influence on microstructure and hardness in 6xxx al-mg-si alloys. *Journal of Material Science*, 41:471–478, 2006.
- M. M. Marzouk. Large deformation behavior of cast a356 aluminium alloy under uniaxial tension, compression and v-bending. Master’s thesis, McMaster University, 2012.
- K. Matsuda, H. Gamada, K. Fujii, Y. Uetani, T. Sato, A. Kamio, und S. Ikeno. High-resolution electron microscopy on the structure of guinier–preston zones in an al-1.6 mass pct Mg<sub>2</sub>Si alloy. *Metallurgical and Materials Transactions A*, 29:1161–1167, 1998.
- A. Matsushita, T. Nakazawa, T. Okane, und M. Yoshida. Crack prediction for a partially solidified lead-free bronze casting using thermal stress analysis. *Journal of Materials Processing Tech*, 249:46–56, 2017. doi: 10.1016/j.jmatprotec.2017.05.022.
- A. M. A. Mohamed und F. Samuel. *Heat Treatment - Conventional and Novel Applications*, chapter 4, pages 55–72. INTech, 2012. doi: DOI:10.5772/79832.
- M. Moukalled, F. and Darwish und L. Mangani. *Fluid Mechanics and Its Applications: The Finite Volume Method in Computational Fluid Dynamics*, volume 113. Springer-Verlag, 2016. doi: 10.1007/978-3-319-16874-6.
- D. P. Myriounis, S. T. Hasan, und T. E. Matikas. Microdeformation behaviour of al-sic metal matrix composites. *Composite Interfaces*, 15(5):495–514, 2008.
- H. Möller, G. Govender, und W. E. Stumpf. Application of shortened heat treatment cycles in a356 automotive brake calipers with respective globular and dendritic microstructure. *Transactions of Nonferrous Metals Society of China*, pages 814–820, 2010.
- W. Nes und K. Marthinsen. Modeling the evolution in microstructure and properties during plastic deformation of f.c.c.-metals and alloys – an approach towards a unified model. *Materials Science and Engineering, A*, 322, 2002.
- T. T. H. Ng und G. S. B. Leng. Application of genetic aalgorithm to conceptual design of a micro-air vehicle. *Engineering Applications of Artificial Intelligence*, 15:439–445, 2002. doi: 10.1016/S0952-1976(02)00072-6.

- A. Olsson, G. Sandberg, and O. Dahlblom. On latin hypercube sampling for structural reliability analysis. *Structural Safety*, 25:47–68, 2003.
- S. Özbilen und H. M. Flower. Zirconium–vacancy binding and its influence on s’-precipitation in an al–cu–mg alloy. *Acta Metallurgica*, 37(11):2993–3000, 1989.
- J. Paz. *Heat Treatment and Precipitation in A356 Aluminium Alloy*. PhD thesis, McGill University, 2003.
- K. E. Petersen. Silicon as a mechanical material. *Proceedings of the IEEE*, 70(5):420–457, 1982. doi: 10.1109/PROC.1982.12331.
- S. Pogatscher, E. Kozeschnik, H. Antrekowitsch, M. Werinos, S. S. A. Gerstl, J. F. Löffler, and P. J. Uggowitzzer. Process-controlled suppression of natural aging in an al–mg–si alloy. *Scripta Materialia*, 89:53–56, 2014. doi: 10.1016/j.scriptamat.2014.06.025.
- E. Povoden-Karadeniz, P. Lang, P. Warczok, A. Falahati, W. Jun, and E. Kozeschnik. Calphad modeling of metastable phases in the al–mg–si system. *Computer Coupling of Phase Diagrams and Thermochemistry*, 43:94–104, 2013. doi: 10.1016/j.calphad.2013.03.004.
- N. Prabhu and J. M. Howe. The effect of ternary trace additions on the nucleation and growth of  $\gamma'$  precipitates in an al-4.2 at. pct ag alloy. *Metallurgical Transaction A*, 1990.
- A. K. Prasada Roa, K. Das, B. S. Murty, and M. Chakraborty. Microstructural features of as-cast a356 alloy inoculated with sr, sb modifiers and al-ti-c grain refiner simultaneously. *Materials Letters*, 62:273–275, 2008. doi: 10.1016/j.matlet.2007.05.020.
- W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes - The Art of Scientific Computing*. Cambridge University Press, third edition, 2007.
- A. Raji. A comparative analysis of grain size and mechanical properties of al-si alloy components produced by different casting method. Technical report, Department of Mechanical Engineering, Federal University of Tehchnology Yola, Nigeria, 2010.
- C. Ramey und B. Fox. Bash reference manual, 2016. URL <https://www.gnu.org/software/bash/manual/bash.pdf>.
- C. E. Rasmussen und C. K. I. Williams. *Gaussian PrProcess for Machine Learning*. MIT Press, 2003. URL <http://www.gaussianprocess.org/gpml/chapters/RW.pdf>.
- H. Rusche. *Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions*. PhD thesis, Imperial College of Science, December 2002.
- Ø. Ryen, B. Holmedal, K. Marthinsen, and T. Furu. Precipitation, strength and work hardening of age hardened aluminium alloys. *Materials Science and Engineering A*, 89, 2015. doi: 10.1088/1757-899X/89/1/012013.
- J. Sacks, W. J. Welch, T. J. Mitchell, and H. P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4:409–423, 1989. doi: 10.2307/3315868.
- L. Sagalowicz, G. Lapasset, and G. Hug. Thranmission electron microscopy study of precipitate which forms in the al-mg-si system. *Philosophical Magazine Letters*, 74(2):57–66, 2010.



- M. S. Salleh, M. Z. Omar, und J. Syarif. The effects of mg addition on the microstructure and mechanical properties of thixoformed al-5 *Journal of Alloys and Compounds*, 621:121–130, February 2015. doi: <http://dx.doi.org/10.1016/j.jallcom.2014.09.152>.
- T. J. Santner, B. J. Williams, und W. I. Notz. The design and analysis of computer experiments. *Journal of the American Statistical Association*, 99:1203–1204, 2003. doi: 10.1198/jasa.2004.s360.
- F.-J. Sayas. A gentle introduction to the finite element method, 2008.
- J. Schöberl. An advancing front 2d/3d-mesh generator based on abstract rules. *Computing and Visualization in Science*, (1):41–52, 1997. URL [http://www.asc.tuwien.ac.at/~schoeberl/wiki/publications/netgen\\_org.pdf](http://www.asc.tuwien.ac.at/~schoeberl/wiki/publications/netgen_org.pdf).
- R. Schwarze. *CFD Modellierung*. Springer-Vieweg, Heidelberg, Berlin, first edition, 2013.
- M. D. Shields und J. Zhang. The generalizaion of latin hypercube sampling. *Reliability Engineering and System Safety*, 2015. doi: 10.1016/j.res.2015.12.002.
- H. J. Skimin und J. P. Andreatch. Elastic moduli of silicon vs hydrostatic pressure at 25.0 °c and -195.8 °c. *Applied Physics A*, 35(7), 2004.
- B. Sonderegger und E. Kozeschnik. Generalized nearest-neighbor broken-bond analysis of randomly oriented coherent interfaces in multicomponent fcc and bcc structures. *Metallurgical and Materials Transactions A*, 40:499–510, March 2009a.
- B. Sonderegger und E. Kozeschnik. Size dependence of the interfacial energy in the generalized nearest-neighbor broken-bond approach. *Scripta Materialia*, 60:635–638, 2009b. doi: 10.1016/j.scriptamat.2008.12.025.
- B. Sonderegger und E. Kozeschnik. Interfacial energy of diffuse phase boundaries in the generalized broken-bond approach. *Metallurgical and Materials Transactions A*, 41:3262–3269, 2010. doi: 10.1007/s11661-010-0370-8.
- M. J. Starink, L. F. Cao, und P. A. Rometsch. A model for the thermodynamics of and strengt-hening due to co-clusters in al–mg–si-based alloys. *Acta Materialia*, 60:4194–4207, 2012. doi: 10.1016/j.actamat.2012.04.032.
- X. Sun, C. Han, und P. Chen. Instantaneous gnss attitude determination: A monte carlo sampling approach. *Acta Astronautica*, 133:124–129, 2017. doi: 10.1016/j.actaastro.2017.01.006.
- E. Suwartadi, S. Krogstad, und B. Foss. Adjoint-based surrogate optimization of oil reservoir water flooding. *Optimization and Engineering*, 16:441–481, 2015. doi: 10.1007/s11081-014-9268-4.
- J. Svoboda, F. D. Fischer, Peter Fratzl, und Ernst Kozeschnik. Modelling of kinetics in multi-component multi-phase systems with spherical precipitates: I: Theory. *Materials Science and Engineering A: Structural Materials: Properties, Microstructure and Processing*, 385:166–174, November 2004. doi: 10.1016/j.msea.2004.06.018.

- J. Svoboda, Y. V. Shan, E. Kozeschnik, und F. D. Fischer. A thermokinetic model for mg–si couple formation in al–mg–si alloys. *Modelling and Simulation in Materials Science and Engineering*, 24, 2016.
- Y.-H. Tan, S.-L. Lee, und H.-Y. Wu. Effects of beryllium on fatigue crack propagation of a357 alloys containing iron. *International Journal of Fatigue*, 18:137–147, 1996.
- K. Teichmann, C. D. Marioara, S. J. Andersen, und K. Marthinsen. Tem study of  $\beta'$  precipitate interaction mechanisms with dislocations and  $\beta'$  interfaces with the aluminium matrix in al–mg–si alloys. *Materials Characterization*, 75:1–7, 2013.
- J. F. Thompson, B. K. Soni, und N. P. Weatherill. *Handbook of grid generation*. CRC Press, 1 edition, December 1998.
- S. Timoshenko und J. N. Goodier. *Theory of elasticity*. McGraw-Hill Book, February 1951.
- M. Timpel, N. Wanderka, R. Schlesiger, T. Yamamoto, N. Lazarev, D. Isheim, G. Schmitz, S. Matsumura, und J. Banhart. The role of strontium in modifying aluminium–silicon alloys. *Acta Materialia*, 60:3920–3928, 2012. doi: 10.1016/j.actamat.2012.03.031.
- Ž. Tuković und H. Jasak. Updated lagrangian finite volume solver for large deformation dynamic response of elastic body. *Transaction of FAMENA*, 2007.
- M. A. van Huis, J. H. Chen, H. W. Zanderbergen, und Sluiter M. H. F. Phase stability and structural relations of nanometer-sized, matrix-embedded precipitate phases in al–mg–si alloys in the late stages of evolution. *Acta Materialia*, 54:2945–2955, 2006. doi: 10.1016/j.actamat.2006.02.034.
- M. A. van Huis, J. H. Chen, M. H. F. Sluiter, und H. W. Zandbergen. Phase stability and structural features of matrix-embedded hardening precipitates in al–mg–si alloys in the early stages of evolution. *Acta Materialia*, 55:2183–2199, 2007.
- A. R. Velasco. Boundary conditions for convective heat transfer in openfoam, 2015. URL <https://foamingtime2.files.wordpress.com/2015/11/convection-bc1.pdf>.
- R. Vilums. Implementation of transient robin boundary conditions in openfoam. Workshop - Multiphysical modelling in OpenFOAM, October 2011. URL [www.modlab.lv/docs/2011/OpenFOAM/Abstracts\\_R\\_Vilums\\_pdf.pdf](http://www.modlab.lv/docs/2011/OpenFOAM/Abstracts_R_Vilums_pdf.pdf).
- R. Vissers, J. Janssen M. A. Van Huis, C. D. Marioara H. W. Zandbergen, und S. J. Andersen. The crystal structure of the beta-prime phase in al–mg–si alloys. *Acta Materialia*, 55:3815–3823, 2007.
- G. Wang, Q. Sun, L. Feng, L. Hui, und C. Jing. Influence of cu content on ageing behavior of alsimgcu cast alloys. *Materials and Design*, 2007.
- H. Wang, C. J. Davidson, und D. H. StJohn. Semisolid microstructural evolution of alsi7mg alloy during partial remelting. *Material Science and Engineering A*, 368:159–167, 2004. doi: 10.1016/j.msea.2003.10.305.

- Q. G. Wang und C. J. Davidson. Solidification and precipitation behaviour of al-si-mg casting alloys. *Journal of Materials Science*, 36:739–750, 2001.
- P. Warczok. Matcalc approach, Dezember 2017. URL <https://www.matcalc-engineering.com/index.php/matcalc-software/tutorials/matcalc-approach>.
- P. Warczok, J. Ženíšek, und E. Kozeschnik. Atomistic and continuums modeling of cluster migration and coagulation in precipitation reactions. *Computational Material Science*, 60: 59–65, 2012.
- J. Warnatz, U. Maas, und R. W. Dibble. *Verbrennung*. Springer-Verlag, 3 edition, 2001.
- T. Weisz, T. Warczok, P. and Ebner, A. Falahati, und E. Kozeschnik. Simulation of natural aging in al-mg-si alloys. *Materials Science Forum*, 828:468–473, 2015.
- H. G. Weller, G. Tabor, H. Jasak, und C. Fureby. A tensorical approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, 1998. doi: 10.1063/1.168744.
- D. C. Wilcox. *Turbulence Modeling for CFD*. DCW Industries, 5354 Palm Drive, La Canada, California 91011, first edition, November 1994.
- Thomas Wlanis. Simulation der wärmebehandlung der aluminiumaluminium alsi7cu0,5mg. Doktorarbeit nicht beendet, 2015.
- J. Wolf. *Grundkurs C++: C++-Programmierung verständlich erklärt*. Galileo Computing, 2 edition, May 2013.
- C. Wolverton. Solute–vacancy binding in aluminium. *Acta Materialia*, 55:5867–5872, 2007. doi: 10.1016/j.actamat.2007.06.039.
- M. Wu, J. Li, A. Ludwig, und A. Kharicha. Modeling diffusion-governed solidification of ternary alloys – part 1: Coupling solidification kinetics with thermodynamics. *Computational Materials Science*, 79:830–840, 2013. doi: 10.1016/j.commatsci.2013.05.015.
- S. Xiao, K. Yi, und Y. Du. The microstructure evolution of u1 and u2 nanowires constrained in al matrix. *Computational Materials Science*, 117:180–187, 2016.
- R. Yu, J. Zhu, und H. Q. Ye. Calculations of single-crystal elastic constants made simple. *Computer Physics Communications*, 181:671–675, 2010.
- H. W. Zandbergen, S. J. Andersen, und J. Jansen. Structure determination of Mg<sub>5</sub>Si<sub>6</sub> particles in al by dynamic electron diffraction studies. *Science*, 277:1221–1225, 1997.
- L. Zhang, W. Zhou, und D. Li. Global convergence of a modified fletcher–reeves conjugate gradient method with armijo-type line search. *Numerische Mathematik*, 104:561–572, 2006.



# Anhang



# A. Thermische Spannungsmodellierung

Nachfolgend sind alle Ergänzungen zum Kapitel der thermischen Spannungsmodellierung gegeben, die vom Hauptteil dieser Arbeit ausgelagert wurden. Die im Anhang diskutierten Themenbereiche sind der Beweis zur Herleitung der Temperaturerhaltungsgleichung, die detaillierte Betrachtung der Temperaturrendbedingung, die Berechnung des temperaturabhängigen Wärmeübergangskoeffizienten — der sowohl freie Konvektion als auch Strahlung beinhaltet —, eine detaillierte Beschreibung der implementierten Wärmequellenbewegungsmechanismen, der Analyse des Quellcodes und eine detaillierte numerische Analyse der entwickelten FV Applikation; Stichwort Relaxation und Konvergenzmethoden.

## A.1. Herleitung der Temperaturerhaltungsgleichung

Der Beweis, dass Gleichung (2.13) mithilfe der Gleichung (2.14) und (2.15) in (2.16) überführt werden kann, ist nachfolgend ausführlich gegeben. Zum Verständnis der Herleitung wird die Kenntnis der mathematischen Operationen aus Anhang C vorausgesetzt. Formulierung (2.13) wird der Vollständigkeit halber erneut präsentiert:

$$\underbrace{\frac{\partial \rho h}{\partial t}}_{\text{zeitliche Änderung}} + \underbrace{\nabla \bullet (\rho \mathbf{U} h)}_{\text{konvektiver Fluss}} = \underbrace{-\nabla \bullet (\mathbf{q})}_{\text{diffusiver Fluss}} + \underbrace{S_h}_{\text{Quellterm}} . \quad (\text{A.1})$$

Zur Herleitung der Temperaturgleichung muss eine Annahme bezüglich der Enthalpie getroffen werden. Nachfolgend wird davon ausgegangen, dass die Enthalpie  $h$  von der Temperatur und dem Druck abhängig ist  $\rightarrow h(T, p)$ ; vergleiche Moukalled und Mangani (2016). Entsprechend folgt, für die substantielle Ableitung  $dh$ :

$$dh(T, p) = \left( \frac{\partial h}{\partial T} \right)_p dT + \left( \frac{\partial h}{\partial p} \right)_T dp . \quad (\text{A.2})$$

Der zweite Term auf der rechten Seite ist Null, da für Festkörper  $dp = 0$  angenommen wird. Für Fluide trifft diese Annahme nicht zu. Der Klammerausdruck des ersten Terms auf der rechten Seite repräsentiert die Wärmekapazität  $c_p$  bei konstantem Druck, da diese wie folgt definiert ist:

$$c_p = \left( \frac{\partial h}{\partial T} \right)_p . \quad (\text{A.3})$$

Entsprechend kann Gleichung (A.2) wie folgt ausgedrückt werden und wird lediglich der Vollständigkeit gegeben:

$$dh = c_p dT . \quad (\text{A.4})$$

Zur Herleitung der Temperaturgleichung werden zunächst beide Terme auf der linken Seite von Gleichung (A.1) mit der Produktregel aufgeteilt. Es folgt:

$$\rho \frac{\partial h}{\partial t} + h \frac{\partial \rho}{\partial t} + \rho \mathbf{U} \bullet \nabla h + h \nabla \bullet (\rho \mathbf{U}) = -\nabla \bullet (\mathbf{q}) + S_T , \quad (\text{A.5})$$

Anschließend können jeweils zwei Terme auf der linken Seite zusammengefasst werden:

$$\rho \left[ \frac{\partial h}{\partial t} + \mathbf{U} \bullet \nabla h \right] + h \left[ \frac{\partial \rho}{\partial t} + \nabla \bullet (\rho \mathbf{U}) \right] = -\nabla \bullet (\mathbf{q}) + S_T . \quad (\text{A.6})$$

wobei der zweite Term auf der linken Seite aufgrund der Kontinuitätsgleichung gleich null ist und herausfällt. Der erste Term im Klammerausdruck des ersten Terms auf der linken kann modifiziert, wodurch sich folgende Erhaltungsgleichung ergibt:

$$\rho \left[ \frac{\partial h}{\partial T} \frac{\partial T}{\partial t} + \mathbf{U} \bullet \nabla h \right] = -\nabla \bullet (\mathbf{q}) + S_T . \quad (\text{A.7})$$

Wird Gleichung (A.3) in (A.7) eingesetzt, folgt weiter:

$$\rho \left[ c_p \frac{\partial T}{\partial t} + \mathbf{U} \bullet \nabla h \right] = -\nabla \bullet (\mathbf{q}) + S_T . \quad (\text{A.8})$$

Anschließend wird der Gradient der spezifischen Enthalpie (zweiter Term des Klammerausdrucks auf der linken Seite) weiter umformuliert:

$$\nabla h = \begin{pmatrix} \frac{\partial h}{\partial x} \\ \frac{\partial h}{\partial y} \\ \frac{\partial h}{\partial z} \end{pmatrix} = \begin{pmatrix} \frac{\partial h}{\partial T} \frac{\partial T}{\partial x} \\ \frac{\partial h}{\partial T} \frac{\partial T}{\partial y} \\ \frac{\partial h}{\partial T} \frac{\partial T}{\partial z} \end{pmatrix} = \begin{pmatrix} c_p \frac{\partial T}{\partial x} \\ c_p \frac{\partial T}{\partial y} \\ c_p \frac{\partial T}{\partial z} \end{pmatrix} = c_p \begin{pmatrix} \frac{\partial T}{\partial x} \\ \frac{\partial T}{\partial y} \\ \frac{\partial T}{\partial z} \end{pmatrix} = c_p \nabla T , \quad (\text{A.9})$$

in Gleichung (A.8) eingesetzt:

$$\rho \left[ c_p \frac{\partial T}{\partial t} + \mathbf{U} \bullet c_p \nabla T \right] = -\nabla \bullet (\mathbf{q}) + S_T , \quad (\text{A.10})$$

wodurch die Wärmekapazität  $c_p$  herausgezogen werden kann, sodass folgende Formulierung erhalten wird:

$$c_p \rho \left[ \frac{\partial T}{\partial t} + \mathbf{U} \bullet \nabla T \right] = -\nabla \bullet (\mathbf{q}) + S_T . \quad (\text{A.11})$$

Der Klammerausdruck auf der linken Seite repräsentiert die substantielle/materielle Ableitung. Dementsprechend kann Gleichung (A.11) erneut umgeschrieben werden:

$$c_p \rho \frac{DT}{Dt} = -\nabla \bullet (\mathbf{q}) + S_T . \quad (\text{A.12})$$

Zur Implementierung in ein FVM Programm wird jedoch die eulersche Formulierung benötigt. Diesbezüglich wird Gleichung (A.11) erneut modifiziert. Zu Beginn wird die Dichte in den Klammerausdruck multipliziert und dieser anschließend mit der Kontinuitätsgleichung erweitert:

$$c_p \left\{ \rho \frac{\partial T}{\partial t} + \rho \mathbf{U} \bullet \nabla T + T \left[ \frac{\partial \rho}{\partial t} + \nabla \bullet (\rho \mathbf{U}) \right] \right\} = -\nabla \bullet (\mathbf{q}) + S_T . \quad (\text{A.13})$$



Ferner wird der innere Klammersausdruck auf der linken Seite ausmultipliziert. Mit der Produktregel können die Terme erneut zusammengefasst werden, womit folgender Ausdruck erhalten wird:

$$c_p \frac{\partial \rho T}{\partial t} + c_p \nabla \bullet (\rho \mathbf{U} T) = -\nabla \bullet \mathbf{q} + S_T . \quad (\text{A.14})$$

Für linear-elastische Spannungsprobleme sind die Verschiebungen sehr klein, wodurch der konvektive Term vernachlässigt werden darf:

$$c_p \nabla \bullet (\rho \mathbf{U} T) = c_p \nabla \bullet (\rho \underbrace{\dot{\mathbf{D}}}_{\approx 0} T) \approx 0 . \quad (\text{A.15})$$

Entsprechend lautet das Gleichungssystem für den Erhaltungssatz der Temperatur, der in dieser Arbeit verwendet wird:

$$c_p \frac{\partial \rho T}{\partial t} = \nabla \bullet (\lambda \nabla T) + S_h . \quad (\text{A.16})$$

Dies stellt somit den Beweis zur Temperaturgleichung dar. Wie bereits erwähnt, korreliert die Formulierung mit den Betrachtungen von Ferziger und Perić (2008) sowie Moukalled und Mangani (2016).

Es soll erwähnt werden, dass die Wärmekapazität  $c_p$  und Dichte  $\rho$  temperaturabhängig modelliert werden. Numerische Untersuchungen zeigen jedoch auf, dass beispielsweise das Herausziehen der Dichte aus dem Zeitintegral keine signifikanten Änderungen in den quantitativen Ergebnissen liefert.

## A.2. Herleitung der Temperaturrandbedingung

Die Energieeinbringung in das Material wie beispielsweise durch Verwendung eines LASER's und der simultanen Wärmeabfuhr aufgrund von Strahlung und freier Konvektion wird unter folgenden Annahmen hergeleitet:

- Das Profil des LASER's wird mit einer Gauss'schen Normalverteilung modelliert,
- Strahlung und freie Konvektion werden durch einen temperaturabhängigen Wärmeübergangskoeffizienten (Heat Transfer Coefficient — HTC) abgebildet; die Funktionsfindung des temperaturabhängigen Wärmeübergangskoeffizienten  $HTC(T)$  wird in Anhang A.3 auf Seite 134 diskutiert.

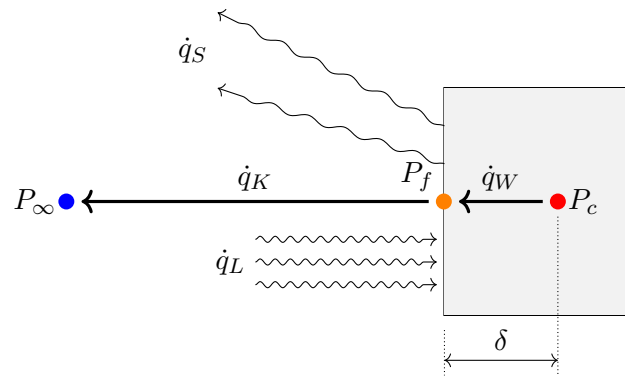


Abbildung A.1.: Schematische Darstellung aller Wärmeflüsse die auf eine Randfläche  $f$  des numerischen Netzes einwirken; zu erkennen sind vier Wärmeströme, welche mit  $\dot{q}_i$  markiert sind:  $\dot{q}_S$  = Strahlung,  $\dot{q}_K$  = freie Konvektion,  $\dot{q}_W$  = Wärmeleitung und  $\dot{q}_L$  = Energiequelle (LASER).

### A.2.1. Wärmeströme und Bilanzierung

Ausgangspunkt zur Herleitung der Randbedingung ist Gleichung (2.28), für die die Bedingung des Gradienten und dem Verhältnisfaktor  $\zeta$  gefunden werden muss. Mathematisch betrachtet müssen diesbezüglich alle Wärmeströme, die aus der numerischen Domäne hinein- und herausfließen, bilanziert werden. Abbildung A.1 zeigt alle Phänomene beziehungsweise Wärmeströme, die für die Herleitung berücksichtigt werden müssen.

Es ist deutlich erkennbar, dass ein Gleichgewicht aller im Flächenzentrum  $P_f$  zusammenkommenden Wärmeströme vorliegen muss. Daraus ergibt sich folgender mathematischer Zusammenhang für jeden Wärmestrom  $\dot{q}_i$  unter Berücksichtigung der Konvention, dass die Wärmeenergie stets dem Gradienten zur niedrigeren Temperatur folgt und diesbezüglich mit einem negativen Vorzeichen behaftet wird.

#### Wärmeleitung $\dot{q}_W$

Die Wärmeleitung kann durch das Fourier Gesetz ausgedrückt werden. Dabei wird Energie durch Diffusion von Punkt  $P_c$  nach  $P_f$  transportiert. Es folgt:

$$\dot{q}_W = -\lambda_{th} \nabla T = -\lambda_{th} \frac{(T_{P_c} - T_{P_f})}{\delta} . \quad (\text{A.17})$$

$\lambda_{\text{th}}$  stellt dabei die Wärmeleitfähigkeit des Materials und  $\delta$  den Normalenabstand zwischen Punkt  $P_c$  und  $P_f$  dar.

### Freie Konvektion $\dot{q}_K$

Der Tatsache geschuldet, dass die umgebende Luft nicht modelliert wird, wird die Abbildung der freien Konvektion durch einen Wärmeübergangskoeffizient (HTC) abgebildet — in der deutschen Literatur ist dieser mit dem griechischen Buchstaben  $\alpha$  notiert und wird zur besseren Unterscheidung und um Verwechslungen mit dem isobaren Wärmeausdehnungskoeffizient, welcher in der Verschiebungsgleichung auftritt, mit einem Subskript  $\text{HTC}$  erweitert.

Im Allgemeinen wird der HTC Koeffizient unter Zuhilfenahme von semi-empirischen Gleichungen bestimmt. Diese Formulierungen verwenden dimensionslose Kennzahlen wie beispielsweise der Reynolds, Grashof, Prandtl oder Nusselt Zahl und sind weiter abhängig von geometrischen Parametern. Explizit für Auslegungsberechnungen von Wärmeübertrager werden diese Gleichungen herangezogen; vergleiche Gnielinski et al. (2005)

Die Grundgleichung, die den Wärmefluss von den numerischen Rändern  $P_f$  zum sogenannten *Farfield*  $P_\infty$  beschreibt, ist gegeben durch:

$$\dot{q}_K = -\alpha_{\text{HTC}}(T_{P_f} - T_{P_\infty}) . \quad (\text{A.18})$$

Farfield bezeichnet dabei die Temperatur in unendlicher Entfernung zum numerischen Rand.

### Wärmestrahlung $\dot{q}_S$

Die Wärmestrahlung folgt dem Gesetz von *Stefan-Boltzmann* und beinhaltet im Wesentlichen zwei relevante Größen, die den Betrag des Wärmestroms definieren. Diese ist die niedrigste und höchste Temperatur im System; bezogen auf Abbildung A.1 ist dies die Temperatur auf dem numerischen Rand und dem Farfield. Der Grund weshalb die Temperaturen maßgeblich für den quantitativen Betrag des Wärmestroms verantwortlich sind, basiert auf der Formulierung, die diese Größen in der 4ten Potenz beinhaltet:

$$\dot{q}_S = \epsilon_0 \sigma_S (T_f^4 - T_\infty^4) . \quad (\text{A.19})$$

Die Größe  $\epsilon_0$  repräsentiert den Emissionsgrad, der Werte von  $0 \leq \epsilon_0 \leq 1$  annehmen kann. Bewegt man sich in der Nähe von null, verhält sich der Körper wie ein Spiegel und reflektiert die Wärmestrahlung, wobei der Wert von Eins einen schwarzen Körper darstellt (komplette Absorption der Energie).  $\sigma_S$  ist eine Konstante mit dem Wert  $5,670367 \text{ Wm}^{-2}\text{K}^{-4}$  und wird als *Stefan-Boltzmann* Konstante bezeichnet; vergleiche Kuchling (2007).

### Energieeintrag durch einen LASER

Der Energieeintrag in das Material wird über den funktionalen Zusammenhang einer Wahrscheinlichkeitsfunktion (PDF) beschrieben. Die Abbildung des LASER's erfolgt daher durch die Verwendung der zweidimensionalen Gauß'schen Normalverteilung, welche in der  $x - y$  Ebene appliziert wird; eine Transformationsregel auf anders orientierte Ebenen ist nicht gegeben. Daher muss die Fläche, welche einen Energieeintrag erfährt, stets in der  $x - y$  Ebene liegen; der LASER ist diesbezüglich parallel zur  $z$ -Achse ausgerichtet.

Die Grundgleichung für die zweidimensionale Gaußverteilung lautet:

$$\text{PDF}(x, y, \sigma_x, \sigma_y, \rho_L) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho_L^2}} \exp\left[-\frac{1}{2(1-\rho_L^2)}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2} - \frac{2\rho_L xy}{\sigma_x\sigma_y}\right)\right], \quad (\text{A.20})$$

und trägt die Einheit  $\text{m}^{-2}$ . Die Variablen  $x$  und  $y$  repräsentieren die Koordinaten der Flächen Mittelpunkte ( $P_f$ ),  $\sigma_x$  und  $\sigma_y$  die Varianz zur entsprechenden Ordinate und  $\rho_L$  den Korrelationsfaktor, der mit null definiert wird. Dies bedeutet, dass keine Verzerrung der Gauss-Funktion vorliegt  $\rightarrow$  symmetrisch.

Da der Energieeintrag durch Elektronen einer bestimmten Wellenlänge erfolgt, muss realistisch betrachtet ein Absorptionskoeffizient  $\psi$  beachtet werden — Analogie zur Strahlung. Diese gibt an, wie hoch der energetische Eintrag in das Material ist und wie viel Energie reflektiert wird. Für Aluminiumlegierungen variiert dieser Wert in der Literatur zwischen 0,1 und 0,7 (Oberflächenbeschaffenheit oder Oxidationsgrad) und ist zudem abhängig vom verwendeten LASER Typ; vergleiche Jakesch (2015). Der Wärmefluss  $\dot{q}_L$  ergibt sich folglich aus dem Produkt von LASER Leistung  $Q$ , der ausgewählten PDF und des Absorptionskoeffizienten  $\psi$ :

$$\dot{q}_L = Q \text{PDF}(x, y, \sigma_x, \sigma_y, \rho_L) \psi. \quad (\text{A.21})$$

Da Gleichung (A.20) die Normalverteilung im Koordinatenursprung  $\mathcal{O}(0\ 0\ 0)$  darstellt, die Energiequelle jedoch örtliche individuell positioniert werden kann, müssen die Koordinaten  $x$  und  $y$  modifiziert werden, um die Gauß'sche Normalverteilung zu verschieben. Es folgt:

$$x = x_{\text{Spot}} - x, \quad (\text{A.22})$$

$$y = y_{\text{Spot}} - y. \quad (\text{A.23})$$

Abschließend ist anzumerken, dass die Wahrscheinlichkeitsfunktion hochwertiger LASER verschiedene Profile aufweisen kann und generell nicht dem hier vorgestellten Gauß'schen Profil folgen. Andere werden im weiteren Verlauf der Arbeit allerdings nicht berücksichtigt, da es lediglich eine mathematische Neuformulierung des Problems darstellt. Als Beispiel eines modernen LASER Profils soll das sogenannte *Top-Hat-Profil* genannt werden.

Des Weiteren wird angenommen, dass der Wert des Absorptionskoeffizienten  $\psi = 1$  beträgt, da es in der numerischen Betrachtung willkürlich ist, wie der Wärmestrom  $\dot{q}_L$  zum Erreichen definierter Temperaturen berechnet wird. Das bedeutet, dass eine Reduzierung von  $\psi$  lediglich eine simultane Erhöhung der LASER Leistung  $Q$  bedeutet, wenn der gleiche Energieeintrag erzielt werden soll.

### A.2.2. Wärmefluss Bilanzierung (ohne Strahlung)

Zur Bestimmung der korrekten Randbedingung, die über Gleichung (2.28) wiedergegeben wird, müssen die Dirichlet und Neumann Anteile und die lineare Übergangsfunktion bestimmt werden.

Die erste Betrachtung bezieht sich auf die Wärmeleitung vom Zell- zum Flächenmittelpunkt und der freien Konvektion. Wird die Wärmestrahlung und der Energieeintrag zunächst vernachlässigt, kann aus Abbildung A.1 folgende Gleichgewichtsbedingung bestimmt werden:

$$\dot{q}_W = \dot{q}_K. \quad (\text{A.24})$$

Nach Einsetzen von Gleichung (A.17) und (A.18) folgt:

$$-\lambda_{\text{th}} \frac{(T_{P_c} - T_{P_f})}{\delta} = \alpha_{\text{HTC}}(T_{P_f} - T_{P_\infty}) . \quad (\text{A.25})$$

Wird diese Gleichung nach  $T_{P_f}$  aufgelöst und in die Form von Gleichung (2.28) gebracht, folgt:

$$T_{P_f} = \underbrace{\frac{\alpha}{\lambda_{\text{th}} \left( \frac{\alpha}{\lambda_{\text{th}}} + \frac{1}{\delta} \right)}}_{\zeta} T_{P_\infty} + \underbrace{\frac{1}{\delta \left( \frac{\alpha}{\lambda_{\text{th}}} + \frac{1}{\delta} \right)}}_{(1-\zeta)} T_{P_c} . \quad (\text{A.26})$$

Durch mathematische Manipulation kann bewiesen werden, dass die Vorfaktoren von  $T_{P_\infty}$  und  $T_{P_c}$ , den Ausdrücken von  $\zeta$  und  $(1-\zeta)$  entsprechen. Ein Vergleich zu Gleichung (2.28) liefert das Ergebnis, dass eine Randbedingung die lediglich freie Konvektion betrachtet, keine Gradienten Bedingung benötigt, allerdings eine Übergangsfunktion  $\zeta$  aufweist und daher mit einer Robinson Bedingung modelliert werden muss:

$$T_{P_f} = \zeta T_{P_\infty} + (1-\zeta) T_{P_c} . \quad (\text{A.27})$$

Wird im Gegenzug nur die Bilanzierung der zugeführten Wärmeenergie mithilfe eines LASER's betrachtet (ohne Konvektion und Strahlung), kann folgende Bilanz aufgestellt werden:

$$\dot{q}_W = \dot{q}_L . \quad (\text{A.28})$$

Daraus ergibt sich unter der Annahme, dass die gesamte Energie absorbiert wird, folgendes Gleichungssystem:

$$Q \text{ PDF}(x, y, \sigma_x, \sigma_y, \rho_L) = \frac{\lambda_{\text{th}}}{\delta} (T_{P_c} - T_{P_f}) . \quad (\text{A.29})$$

Erkennbar ist die Gradientenbedingung, die durch einfache Umstrukturierung folgende Form annimmt:

$$\frac{Q \text{ PDF}(x, y, \sigma_x, \sigma_y, \rho_L)}{\lambda_{\text{th}}} = \frac{(T_{P_c} - T_{P_f})}{\delta} . \quad (\text{A.30})$$

Werden die drei Wärmeströme zusammengefasst ( $\dot{q}_L - \dot{q}_K - \dot{q}_W = 0$ ), kann daraus der Ausdruck für Gleichung (2.28) erhalten werden. Der Beweis, dass die in Kapitel 2.7.2 auf Seite 21 gegebenen Terme korrekt sind, wird nachfolgend ausführlich gezeigt.

Die Bilanzierung aller Wärmeströme ergibt:

$$\begin{aligned} Q \text{ PDF}(x, y, \sigma_x, \sigma_y, \rho_L) - \alpha_{\text{HTC}}(T_{P_f} - T_{P_\infty}) - \frac{\lambda_{\text{th}}}{\delta} (T_{P_c} - T_{P_f}) &= 0 , \\ Q \text{ PDF}(x, y, \sigma_x, \sigma_y, \rho_L) - \alpha_{\text{HTC}} T_{P_f} + \alpha_{\text{HTC}} T_{P_\infty} - \frac{\lambda_{\text{th}}}{\delta} (T_{P_c} - T_{P_f}) &= 0 , \\ \frac{Q \text{ PDF}(x, y, \sigma_x, \sigma_y, \rho_L)}{\lambda_{\text{th}}} - \frac{\alpha_{\text{HTC}}}{\lambda_{\text{th}}} T_{P_f} + \frac{\alpha_{\text{HTC}}}{\lambda_{\text{th}}} T_{P_\infty} - \frac{1}{\delta} (T_{P_c} - T_{P_f}) &= 0 , \\ \frac{Q \text{ PDF}(x, y, \sigma_x, \sigma_y, \rho_L)}{\lambda_{\text{th}}} - \frac{\alpha_{\text{HTC}}}{\lambda_{\text{th}}} T_{P_f} + \frac{\alpha_{\text{HTC}}}{\lambda_{\text{th}}} T_{P_\infty} - \frac{1}{\delta} T_{P_c} + \frac{1}{\delta} T_{P_f} &= 0 . \end{aligned} \quad (\text{A.31})$$

Wird diese Gleichung nach  $T_{P_f}$  aufgelöst und der erst Terme um  $\delta$  erweitert, folgt:

$$\begin{aligned}
\left[-\frac{\alpha_{\text{HTC}}}{\lambda_{\text{th}}} + \frac{1}{\delta}\right] T_{P_f} &= -\frac{\alpha_{\text{HTC}}}{\lambda_{\text{th}}} T_{P_\infty} + \frac{1}{\delta} T_{P_c} - \frac{\delta}{\lambda_{\text{th}}} \frac{Q \text{ PDF}(x, y, \sigma_x, \sigma_y, \rho_L)}{\lambda_{\text{th}}}, \\
T_{P_f} &= \frac{-\frac{\alpha_{\text{HTC}}}{\lambda_{\text{th}}}}{-\frac{\alpha_{\text{HTC}}}{\lambda_{\text{th}}} + \frac{1}{\delta}} T_{P_\infty} + \frac{1}{\delta \left[-\frac{\alpha_{\text{HTC}}}{\lambda_{\text{th}}} + \frac{1}{\delta}\right]} T_{P_c} - \frac{\delta}{\lambda_{\text{th}} \left[-\frac{\alpha_{\text{HTC}}}{\lambda_{\text{th}}} + \frac{1}{\delta}\right]} \frac{Q \text{ PDF}(x, y, \sigma_x, \sigma_y, \rho_L)}{\lambda_{\text{th}}}, \\
T_{P_f} &= \underbrace{\frac{1}{1 + \frac{\lambda_{\text{th}}}{\delta + \alpha_{\text{HTC}}}}}_{\zeta} T_{P_\infty} + \underbrace{\frac{1}{\delta \left[-\frac{\alpha_{\text{HTC}}}{\lambda_{\text{th}}} + \frac{1}{\delta}\right]}}_{1-\zeta} \left\{ T_{P_c} - \frac{\delta}{\lambda_{\text{th}}} \frac{Q \text{ PDF}(x, y, \sigma_x, \sigma_y, \rho_L)}{\lambda_{\text{th}}} \right\}.
\end{aligned} \tag{A.32}$$

Bei der Gradientenbedingung ist darauf zu achten, dass der Abstand  $\delta$  in OpenFOAM<sup>®</sup> mit dem Reziprokwert  $\delta^*$  bezeichnet wird:  $\delta^* = \delta^{-1}$ . Eine Einheitenkontrolle aller Vorfaktoren bestätigt weiter die Herleitung. Des Weiteren korreliert  $\zeta$  mit ähnlichen Herleitungen wie beispielsweise von Vilums (2011) oder Velasco (2015).

### A.2.3. Wärmefluss Bilanzierung (mit Strahlung)

Wird Strahlung als eigener Quellterm in der Bilanz angesehen, induziert Gleichung (A.19) ein numerisches Problem, da die Temperatur der Fläche  $T_f$ , welche als Unbekannte vorliegt, zur 4ten Potenz steht. Wird die Wärmestrombilanzierung durchgeführt, werden zusätzliche Terme erzeugt, die es verhindern, die Form von Gleichung (2.28) zu erhalten. Die Quintessenz hieraus ist der wesentlich größere programmiertechnische Aufwand. Hinzu kommt die Linearisierung des nicht-linearen Terms, dass die Konvergenzrate deutlich herabsetzt (Potenz 4ter Ordnung).

Um dieses Problem zu umgehen wird die Strahlung in den Wärmeübergangskoeffizient  $\alpha_{\text{HTC}}$  integriert. Mit anderen Worten bedeutet dies, dass anstelle eines zusätzlichen Terms, der Wärmeübergangskoeffizient  $\alpha_{\text{HTC}}$  als *effektive* Quantität betrachtet wird, der sowohl Konvektion als auch Strahlung beinhaltet. Entsprechend wird in Gleichung (A.32) der Koeffizient für die Konvektion durch den Effektivwert  $\alpha_{\text{eff}}$  ersetzt, sodass gilt:

$$\alpha_{\text{eff}} = \alpha_{\text{HTC}} + \alpha_{\text{R}}. \tag{A.33}$$

$\alpha_{\text{R}}$  gibt dabei den Anteil der Strahlung an.

Die Bestimmung des Effektivwerts  $\alpha_{\text{eff}}$  wird durch eine numerische Modellierung durchgeführt, die in Anhang A.3 auf Seite 134 diskutiert wird. Es wird vorweggenommen, dass dies nur eine Abschätzung darstellt und für jedes geometrische Design anders ausfällt, da vor allem der Wärmeübergangskoeffizient  $\alpha_{\text{HTC}}$  stark von der geometrischen Form abhängig ist; vergleiche Gnielinski et al. (2005).

### A.2.4. Quellcode

Die verbale Analyse des Quellcodes der entwickelten Randbedingung wird nicht gegeben, da der C++ Code als Open Source publiziert wurde. Es wird daher nicht als notwendig erachtet, den Quellcode detailliert zu beschreiben. Jedoch werden die Hauptfunktionen durch Kommentare präsentiert. Weitere Informationen befinden sich in Holzmann (2016b).

```

1 // Member functions
2
3     //- Update the coefficients associated with the patch field
4     virtual void updateCoeffs ();
5
6     //- Update spot center
7     virtual void updateSpotCenter ();
8
9     //- Update value fraction
10    virtual void updateValueFraction ();
11
12    //- Update gauss distribution based on input file
13    virtual void updateGaussDistribution ();
14
15    //- Update the heat flux distribution (refGrad)
16    virtual void updateHeatFluxDistribution ();
17
18    //- Update the reference temperature (refValue)
19    virtual void updateRefTemperature ();

```

Quellcode 1: Funktionsdeklarationen der LASER Randbedingung in der Header Datei.

### A.3. Bestimmung des effektiven Wärmeübergangskoeffizienten

Wie in Anhang A.2 bereits erwähnt, ist die Modellierung des Strahlungsterms mit den Standardmethoden wesentlich aufwendiger und erfordert eine weitaus umfangreichere Programmierung, da auf Gleichung (2.28) — bezüglich der 4ten Potenz und der Nichtlinearität — nicht zurückgegriffen werden kann. Daher wird der Energieaustrag aufgrund von Strahlung in den konvektiven Term miteinbezogen. Dies kann erreicht werden, in dem der Wärmeübergangskoeffizient  $\alpha_{\text{HTC}}$  durch einen effektiven Koeffizienten  $\alpha_{\text{eff}}$  ersetzt wird. Die Berechnung des Effektivwerts wird nachfolgend diskutiert. Der Modellierungscase ist auf dem beiliegenden Datenspeicher unter `Doktorarbeit/thermalElasticStressFoam/BestimmungHTC` zu finden. Diesbezüglich werden im Verlaufe dieses Kapitels keine geometrischen, netz- oder solverspezifischen Einzelheiten gegeben.

Das Hauptproblem bei der Verwendung eines Effektivwerts  $\alpha_{\text{eff}}$ , der sowohl Konvektion als auch Strahlung beinhaltet, besteht bei der Bestimmung des Koeffizienten selbst. Bereits bei reiner Konvektion ist die quantitative Wahl von  $\alpha_{\text{HTC}}$  nicht offensichtlich und keine leichte Aufgabe. Grund hierfür ist die Komplexität des Energieaustauschs und die Abschätzung dieses Wertes mittels semi-empirischen Gleichungen, die von dimensionslosen Kennzahlen und somit von der Strömungscharakteristik, den Fluideigenschaften und der geometrischen Gegebenheit abhängig ist. Eine Vielzahl von Grundgleichungen für spezielle Apparaturen und Anwendungen sind in Gnielinski et al. (2005) zu finden. Diese beschränken sich allerdings auf allgemeine und/oder sehr spezielle Wärmeübergangsprobleme.

In der vorliegenden Arbeit werden keine dieser Gleichungen verwendet, da es unmöglich ist, eine genaue Abschätzung der Parameter, die für die semi-empirischen Gleichungen bekannt sein müssen, zu treffen. Daher wird eine einfache Simulation durchgeführt, bei der eine aus Aluminium bestehende Platte in Luft — durch Konvektion — abgekühlt wird. Da selbst diese Abschätzung nur ein grober Richtwert für den Wärmeübergangskoeffizienten  $\alpha_{\text{HTC}}$  darstellt, wird der Fall einfach gehalten und keine weiteren Analysen bezüglich Netzabhängigkeit, Löser, Diskretisierung einzelner Terme usw. durchgeführt.

Die Randbedingungen werden so gewählt, dass die Luft aufgrund der Druckverteilung, die bezüglich der Dichteänderung hervorgerufen wird, im Berechnungsraum entsprechend ein- und ausströmen kann. Des Weiteren werden temperaturabhängige Stoffdaten für das Fluid verwendet, die mit Polynomen abgebildet werden. Die Koeffizienten sind in Tabelle A.1 aufgelistet und werden mit Stoffdaten aus Gnielinski et al. (2005) bestimmt. Das Fitting wird mithilfe der *Least-Square* (kleinste Fehlerquadrate) Methode unter Verwendung von `gnuplot` durchgeführt. Alle Rohdaten sind gesondert in Anhang G auf Seite 221 gegeben.

Die thermodynamischen Größen werden über Polynome 4ter Ordnung abgebildet:

$$f(T) = DT^3 + CT^2 + BT + A . \quad (\text{A.34})$$

Die Koeffizienten A bis D besitzen jeweils die Einheit der gesuchten Größe, wobei diese stets mit der Einheit *pro Kelvin* bezüglich des Temperaturexponenten erweitert werden muss.

Die Fluidgeometrie beträgt 0,48 m x 0,36 m x 0,48 m. In der Mitte befindet sich eine nach unten versetzte Aluminiumplatte mit den Abmessungen 0,2 m x 0,1 m x 0,008 m welche eine Anfangstemperatur von 850 K aufweist. Die Wärmeleitfähigkeit des Aluminiums wird mit einem konstanten Wert von  $230 \text{ Wm}^{-1}\text{K}^{-1}$  festgelegt und basiert auf der Zusammensetzung



	Einheit	A	B	C	D
Dichte $\rho$	[kgm <sup>-3</sup> ]	3,1558e <sup>+1</sup>	-9,3827e <sup>-3</sup>	1,1261e <sup>-5</sup>	-4,6872e <sup>-9</sup>
dynamische Viskosität $\mu$	[m <sup>2</sup> s <sup>-1</sup> ]	1,3780e <sup>-6</sup>	6,7574e <sup>-8</sup>	-3,8719e <sup>-11</sup>	1,3067e <sup>-14</sup>
Wärmekapazität $c_p$	[Jkg <sup>-1</sup> K <sup>-1</sup> ]	1047,6600e <sup>+1</sup>	-0,3469e <sup>+1</sup>	8,0840e <sup>-4</sup>	-3,6807e <sup>-7</sup>
Wärmeleitfähigkeit $\lambda$	[Wm <sup>-1</sup> K <sup>-1</sup> ]	8,7067e <sup>-4</sup>	9,7469e <sup>-5</sup>	-4,6381e <sup>-8</sup>	1,5741e <sup>-11</sup>

Tabelle A.1.: Polynomkoeffizienten zur Modellierung der temperaturabhängigen Stoffdaten für Luft bei 1013,15 hPa; Fitting basiert auf den Werten aus Gnielinski et al. (2005).

von 93% Aluminium und 7% Silizium. Aufgrund der Tatsache, dass der temperaturabhängige effektive Wärmeübergangskoeffizient  $\alpha_{\text{eff}}$  von Interesse ist, wird weiter die Wärmekapazität des Festkörpers auf 200 Jkg<sup>-1</sup>K<sup>-1</sup> gesetzt. Die Verringerung des Wertes beeinflusst das Ergebnis nur marginal, beschleunigt jedoch die Berechnung erheblich, da die Abkühlung wesentlich schneller voranschreitet und dadurch der temperaturabhängige Wärmeübergangskoeffizient rascher berechnet wird. Mit anderen Worten: wird die Wärmekapazität auf einen anderen Wert gesetzt (beispielsweise des realen Wertes der Legierung), bedeutet dies nichts anderes, als dass die Abkühlung der Platte mehr Zeit in Anspruch nimmt, da mehr Energie dem System entzogen werden muss. Da sich das Strömungsprofil der Luft um die Platte hierbei aber nicht stark verändert, folgt, dass ein ähnliches Ergebnis mit einer kleineren Wärmekapazität erreicht wird; des Weiteren ist dies nur eine grobe Modellierung (Abschätzung), um einen Anhaltspunkt für den Wärmeübergangskoeffizient zu erhalten. Die Dichte wird auf 2685 kgm<sup>-3</sup> festgelegt.

Das umgebende Fluid beträgt zu Beginn 293 K und wird aufgrund der heißen Aluminiumplatte erwärmt. Diesbezüglich verändern sich die physikalischen Größen der Luft, die mit den oben genannten Polynomen bestimmt werden. Die Dichteänderung ergibt eine Auftriebskraft, wodurch sich ein Strömungsprofil aufgrund der freien Konvektion um die Platte ausbildet. Die neu einströmende Luft besitzt eine Temperatur von 293 K. Ein numerisches Strahlungsmodell wird nicht angewendet, da der Strahlungsanteil nach der Berechnung separat über Gleichung (A.19) bestimmt wird.

Der gemittelte, temperaturabhängige Wärmeübergangskoeffizient  $\alpha_{\text{HTC}}$  wird nach erfolgreicher Simulation mit folgenden Schritten, die für jeden Zeitschritt ( $\Delta t = 0,1$  s) durchgeführt werden, erhalten:

- Berechnung einer gemittelten Oberflächentemperatur der Aluminiumplatte (Kontaktfläche beziehungsweise Interface zwischen Festkörper und Fluid),
- Mithilfe des Post-Processing Programms `wallHeatFlux`, wird die gemittelte Wärmeenergie  $Q_K$  [W], die über die Oberfläche der Platte abgeführt wird, berechnet,
- Durch Kenntnis der oben genannten Werte und der Farfield Temperatur  $T_{P_\infty}$  kann der gemittelte Wärmeübergangskoeffizient  $\alpha_{\text{HTC}}$  über Gleichung (A.18) bestimmt werden; die Wärmeenergie  $Q_K$  muss durch die Oberfläche  $A$  geteilt werden, um den gemittelten Wärmefluss  $\dot{q}_K$  zu erhalten ( $A = 0,0448$  m<sup>2</sup>).

Da die Simulation nur den Einfluss der Konvektion berücksichtigt, die Strahlung aber nicht, wird anschließend über Gleichung (A.19) ein Wärmefluss  $\dot{q}_S$  berechnet. Die erhaltene Energieabfuhr wird nachfolgend dazu verwendet mittels Gleichung (A.18) einen Wärmeübergangskoeffizienten  $\alpha_S$  für die Strahlung zu bestimmen. Der Effektivwert  $\alpha_{\text{eff}}$  wird anschließend durch Addition beider (Konvektion und Strahlung) erhalten; analog ist es möglich beide Wärmeströme  $\dot{q}_K$  und

$\dot{q}_S$  zu addieren und danach den effektiven Wärmeübergangskoeffizient mittels Gleichung (A.18) zu berechnen. Beide Varianten sind äquivalent.

Die Ergebnisse der Berechnungen sind in Abbildung A.2 grafisch dargestellt. Deutlich erkennbar sind die gegenläufigen Eigenschaften zwischen Konvektion und Strahlung. Bei höheren Temperaturen kann für den konvektiven Anteil angenommen werden, dass sich ein asymptotischer Verlauf parallel zur Temperaturachse einstellt, die allerdings stark von der Umströmung abhängig ist und diese mit der geometrischen Formgebung zusammenhängt. Periodische Ablösungen können den Wärmeübergang deutlich verändern (eine bis mehrere Zehnerpotenzen). Im Gegensatz dazu nimmt der Strahlungsanteil mit höherer Temperatur zu, da die Grundgleichung mit der 4ten Potenz beschrieben wird.

Die Peaks die in der Nähe des rechten Rands ersichtlich sind, lassen sich auf die Numerik zurückführen. Die Anfangsbedingungen sowie die Strömungsbildung um die Platte liefern zu Beginn stark variierende Werte, die auf Ablösungen zurückzuführen sind.

Zur Modellierung des temperaturabhängigen Wärmeübergangskoeffizienten  $\alpha_{\text{eff}}$ , wird die in Abbildung A.2 dargestellte Funktion mit einem Polynom 2ten Grades angenähert; vergleiche Gleichung (A.34). Die ermittelten Koeffizienten sind unten angeben und werden für die Temperaturrandbedingung, die in Anhang A.2 diskutiert wird, verwendet. Dadurch ist es möglich einen temperaturabhängigen Wärmeübergangskoeffizienten zu verwenden und zugleich den Strahlungsterm mithilfe des Konvektionsterms auszudrücken.

Die Koeffizienten für das Polynom lauten:

$$\begin{aligned} A &= -9,79250 \quad \text{Wm}^{-2}\text{K}^{-1}, & B &= 0,03960 \quad \text{Wm}^{-2}\text{K}^{-2}, \\ C &= -1,70802e^{-5} \quad \text{Wm}^{-2}\text{K}^{-3}, & D &= 0,00000 \quad \text{Wm}^{-2}\text{K}^{-4}. \end{aligned}$$

Durch die hier vorgestellte numerische Handhabung des Strahlungsterms ist es möglich, weiterhin Gleichung (2.28) zu verwenden. Zudem können die Polynomkoeffizienten A, B, C und D für die Randbedingung beliebig abgeändert werden, um so diverse Szenarien abzubilden.

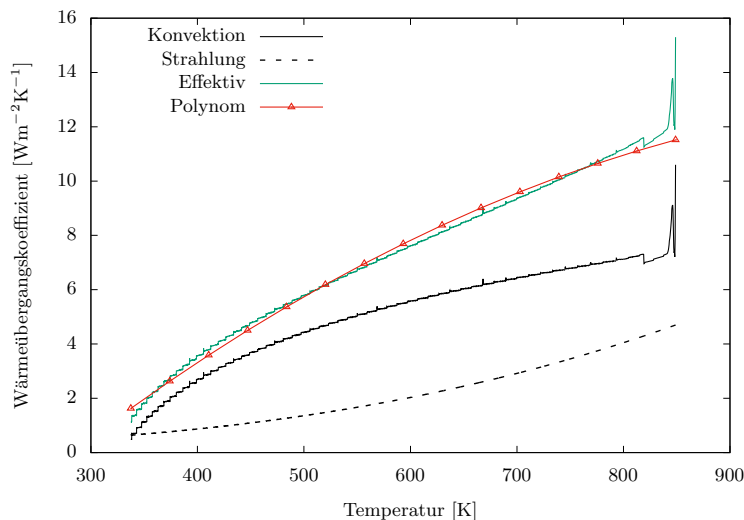


Abbildung A.2.: Auswertung der Wärmeübergangskoeffizienten für freie Konvektion und Strahlung. Der Effektivwert ist als Kombination beider zu sehen und wird durch ein Polynom angenähert.

## A.4. Dynamischer LASER Spot

Die Bewegung der Energiequelle ist in zwei Klassen eingeteilt. Dazu gehören die linearen und kreisförmigen Bewegungsformen. Für die lineare Translation werden einzelne Punkte definiert, die mithilfe der angegebenen Spotgeschwindigkeit abgefahren werden. Daher eignet sich diese Methodik für alle willkürlichen Bewegungsmuster. Dies bedeutet, dass die Energiequelle mithilfe einer Punktwolke in beliebiger Form bewegt werden kann. Die zweite Möglichkeit besteht darin, einen Kreismittelpunkt und Radius zu definieren. Unter Verwendung einer Winkelgeschwindigkeit kann der Energie-Spot entlang des Kreises bewegt werden.

Die nötige Änderung in der Temperaturrandbedingung beinhaltet lediglich den Austausch der  $x_{\text{Spot}}$  und  $y_{\text{Spot}}$  Koordinaten. Anders ausgedrückt werden die statischen Spotzentren durch zeitlich veränderbare ausgetauscht. Vergleiche hierzu Anhang A.2 auf Seite 129.

### A.4.1. Kreisbewegung

Die Kreisbewegung des Spots kann mithilfe von Abbildung A.3 hergeleitet werden. Die Bewegung ist definiert durch das Rotationszentrum (●) und den Radius  $r_L$ . Die Geschwindigkeit und die Drehrichtung, mit der sich die Energiequelle kreisförmig um das Zentrum bewegt, wird mit der Winkelgeschwindigkeit  $\omega$  definiert. Unter Verwendung eines LASER's wird der Startpunkt des Gauß Spots, welcher bei  $t_0$  mit einem Startwinkel  $\omega_{t_0}$  vorgegeben werden muss, angegeben. Wird die Verdrehung ( $\omega_{t_0}$ ) vernachlässigt, würde das bedeuten, dass der Energieeintrag für  $t_0$  im blau markierten Punkt (●) beginnt. Die Angabe der Position des Spots zum Zeitpunkt  $t_0$  erfolgt über  $\omega_{t_0}$  in Radian.

Durch das Rotationszentrum, der Anfangsposition und der Winkelgeschwindigkeit  $\omega$  ist es möglich, für jeden Zeitpunkt die genaue Position des Gauß Spots zu bestimmen. Dazu wird zuerst die An- und Gegenkathete mit den trigonometrischen Funktionen bestimmt und anschließend

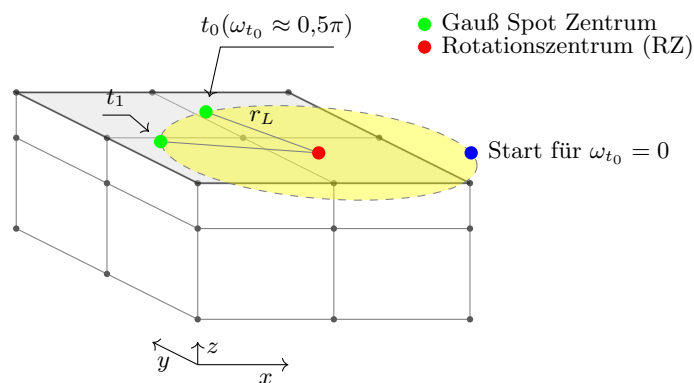


Abbildung A.3.: Darstellung der Kreisbewegung einer Energiequelle (LASER Spot) auf der Oberfläche (in grau dargestellt) für zwei verschiedene Zeitpunkte. In blau ist der Startpunkt für  $\omega_{t_0} = 0$  zu erkennen.

die neuen Koordinaten  $x$  und  $y$  für den Gauß Spot berechnet.

$$\Delta x = \cos(\omega + \omega_{t_0})r_L , \quad (\text{A.35})$$

$$\Delta y = \sin(\omega + \omega_{t_0})r_L , \quad (\text{A.36})$$

$$x_{\text{Gauss}} = x_{\text{RZ}} + \Delta x , \quad (\text{A.37})$$

$$y_{\text{Gauss}} = y_{\text{RZ}} + \Delta y . \quad (\text{A.38})$$

Die Variablen  $x_{\text{Gauss}}$  und  $y_{\text{Gauss}}$  repräsentieren die aktualisierten Koordinaten  $x_{\text{Spot}}$  und  $y_{\text{Spot}}$  die in Anhang A.2 in Gleichung (A.23) diskutiert werden. Die Koordinaten mit dem Subskript RZ verweisen auf das Rotationszentrum.

#### A.4.2. Linear Bewegung

Für beliebige Bewegungen kann eine Punktliste  $P_i$  verwendet werden, die mit einer konstanten Bewegungsgeschwindigkeit  $v_L$  durchlaufen wird. Daraus ergibt sich die Möglichkeit, jede Bewegungsform zu modellieren. Die Energiequelle wird für  $t_0$  auf den ersten Punkt gelegt. Beim Voranschreiten der Zeit wird die Energiequelle entsprechend neu positioniert.

Die Ermittlung der Länge  $l_n$  für jeden akkumulierten Längenabschnitt kann wie folgt bestimmt werden und wird beim Starten des Lösers berechnet und abgespeichert:

$$l_n = \sum_{i=0}^n |P_{i+1} - P_i| . \quad (\text{A.39})$$

Aufgrund der konstanten Bewegungsgeschwindigkeit ist es möglich die Zeitpunkte zu ermitteln, in denen die Energiequelle sich zwischen zwei Punkten bewegt; beispielsweise zwischen  $P_1$  und  $P_2$ . Demnach ist bekannt, ab welcher Zeit sich der LASER Spot in einem neuen Längenintervall befindet. Die Berechnung lautet:

$$t_{P_n, P_{n+1}} = \frac{l_n}{v_L} . \quad (\text{A.40})$$

Daraus lassen sich während der Simulation die  $x$  und  $y$  Koordinaten einfach bestimmen, sodass das Zentrum der Energiequelle aktualisiert werden kann. Daraus ergibt sich zu jedem Zeitpunkt die örtliche Position der Energiequelle.

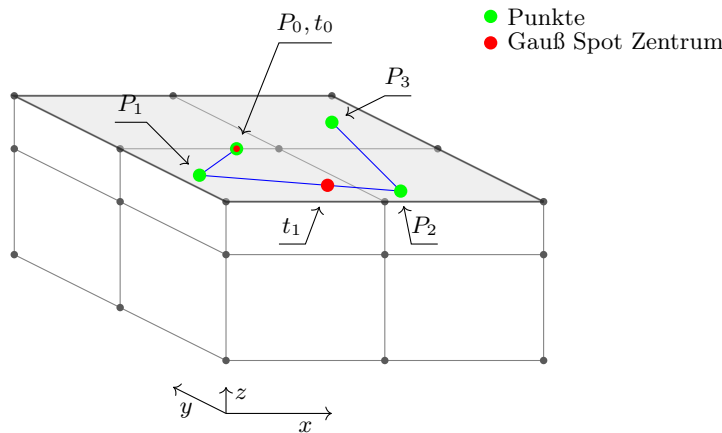


Abbildung A.4.: Darstellung der Linearbewegung der Energiequelle (LASER Spot) auf einer Oberfläche (grau).

Abbildung A.4 gibt weiteren Aufschluss über die lineare Bewegung der Energiequelle und vervollständigt die zuvor genannten Aussagen. Für eine detailliertere Übersicht der Umsetzung wird auf den C++ Quellcode verwiesen; vergleiche Holzmann (2016b). Die Funktion, welche die Position der Energiequelle aktualisiert, ist im Quellcode 1 unter dem Namen `updateSpotCenter()` gelistet.

## A.5. Herleitung der Neumann Bedingung für den Verschiebungsvektor $\mathbf{D}$

Nachfolgend wird die Herleitung der Neumann Randbedingung für spannungsfreie, Kraft beziehungsweise Druck beaufschlagte Oberflächen, die in Kapitel 2.7.1 auf Seite 20 erwähnt wird, gegeben. Ausgangspunkt für die Herleitung ist Gleichung (2.25).

Nach Einsetzen des Hooke'schen Gesetzes und der Definition des Verschiebungstensors  $\mathbf{D}^S$ , Gleichung (2.6), sowie der Beachtung der Duhamel-Neumann Formulierung und der Stabilisierungserweiterung, folgt:

$$\mathbf{S}_S = [(2\mu + \lambda)(\nabla \otimes \mathbf{D}) + \mu(\nabla \otimes \mathbf{D})^T + \lambda \operatorname{tr}(\nabla \otimes \mathbf{D})\mathbf{I} - (\mu + \lambda)(\nabla \otimes \mathbf{D}) - (3\lambda + 2\mu)\alpha(T - T_0)\mathbf{I}] \bullet \mathbf{n} . \quad (\text{A.41})$$

Die Zug- oder Druckkräfte, die mit dem Formelzeichen  $\mathbf{S}_S$  definiert sind, können wie folgt ausgedrückt werden:

$$\mathbf{S}_S = \mathbf{F} - p\mathbf{n} , \quad (\text{A.42})$$

wobei  $\mathbf{F}$  eine Zugkraft (Vektor in Newton) darstellt und  $p$  einen Druck (in Pascal) repräsentiert, der senkrecht auf die Oberfläche wirkt. Wird die neue Formulierung eingesetzt, der Klammerausdruck auf der rechten Seite ausmultipliziert und die gesamte Gleichung nach dem Gradienten aufgelöst, folgt:

$$(\nabla \otimes \mathbf{D}) \bullet \mathbf{n} = [(\mathbf{F} - p\mathbf{n}) - \mu(\nabla \otimes \mathbf{D})^T \bullet \mathbf{n} - \lambda \operatorname{tr}(\nabla \otimes \mathbf{D})\mathbf{I} \bullet \mathbf{n} + (\mu + \lambda)(\nabla \otimes \mathbf{D}) \bullet \mathbf{n} + (3\lambda + 2\mu)\alpha(T - T_0)\mathbf{I} \bullet \mathbf{n}] (2\mu + \lambda)^{-1} . \quad (\text{A.43})$$

Nach genauer Betrachtung kann festgestellt werden, dass einige Terme auf der rechten Seite in einen *negativen, expliziten* Teil zusammengefasst werden können:

$$-\sigma_{\text{exp}} = -\mu(\nabla \otimes \mathbf{D})^T \bullet \mathbf{n} - \lambda \operatorname{tr}(\nabla \otimes \mathbf{D})\mathbf{I} \bullet \mathbf{n} + (\mu + \lambda)(\nabla \otimes \mathbf{D}) \bullet \mathbf{n} . \quad (\text{A.44})$$

Der explizite Term  $\sigma_{\text{exp}}$  kann wie folgt definiert werden; vergleiche Gleichung (2.24):

$$\sigma_{\text{exp}} = \sigma - \underbrace{(2\mu + \lambda)(\nabla \otimes \mathbf{D})}_{\text{impliziter Anteil}} . \quad (\text{A.45})$$

Gleichung (A.43) kann damit vereinfacht werden und ergibt unter Berücksichtigung der Vorzeichen:

$$(\nabla \otimes \mathbf{D}) \bullet \mathbf{n} = \frac{[(\mathbf{F} - p\mathbf{n}) - \sigma \bullet \mathbf{n} + (2\mu + \lambda)(\nabla \otimes \mathbf{D}) \bullet \mathbf{n} + \sigma_{\text{th}}]}{(2\mu + \lambda)} . \quad (\text{A.46})$$

Der Term  $\sigma_{th}$  repräsentiert den Verschiebungsquellterm aufgrund von Temperaturänderungen und stellt den letzten Term in Gleichung (A.43) dar.

Zur Wiederholung: Der Term auf der linken Seite stellt den Gradienten normal zur Oberfläche dar und wird in OpenFOAM<sup>®</sup> wie folgt dargestellt; vergleiche Foundation (2018):

$$(\nabla \otimes \mathbf{D}) \bullet \mathbf{n} = \frac{\nabla_{\perp} \mathbf{D}}{\Delta} . \quad (\text{A.47})$$

$\Delta$  stellt dabei den inversen Abstand zwischen Flächenmittelpunkt und Zellmittelpunkt dar und ist identisch zu der bereits eingeführten Variable  $\delta^*$ .

Gleichung (A.46) wird als Ausgangspunkt für die C++ Implementierung in OpenFOAM<sup>®</sup> herangezogen. Es ist durchaus möglich die Formulierung (A.43) zu verwenden; allerdings ist dies aufgrund des bereits vorhandenen Tensors  $\sigma$  unvorteilhaft, da weitere Rechenoperationen bei der Evaluation der Randbedingungsinformationen anfallen.

Gleichung (A.46) stellt eine eigene Randbedingungsklasse dar und ist auf dem mitgelieferten Datenträger beigefügt. Die Implementierung der Gleichung ist im nachstehenden Codeausschnitt gegeben. Der Quellcode ist in `Doktorarbeit/Programming/boundaryConditions` zu finden.

```

190 gradient() =
191 (
192     (traction_ - pressure_ * n)
193     + twoMuLambda*fvPatchField<vector>::snGrad()
194     - (sigma&n)
195     + threeK*alpha*(T-Tref)*n
196 )/twoMuLambda;

```

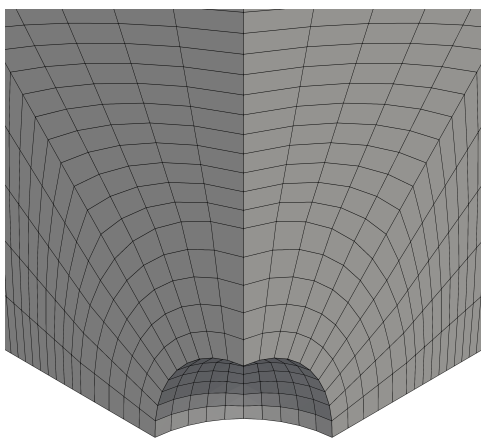
Quellcode 2: Berechnung des Gradienten für die Neumann Randbedingung für spannungsfreie oder kraft- bzw. druckbeaufschlagte Oberflächen; Quellcode entnommen aus der `tractionDisplacementFvPatchVectorField.C` Datei.

## A.6. Ergänzendes zu den Validierungsfälle

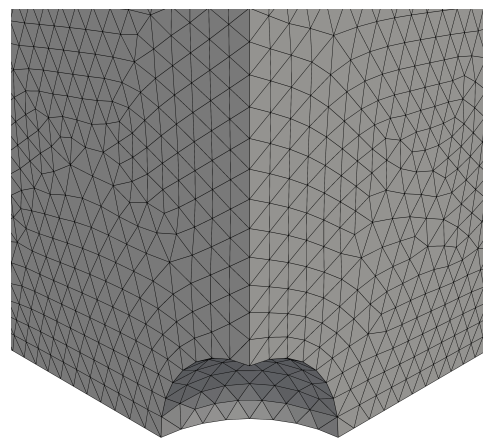
Nachfolgend sind visuelle Einzelheiten bezüglich den erstellten numerischen Netzen und der qualitative Validierung gegeben. Die Darstellung der erstellten Netze zeigt deutlich den Unterschied der verwendeten Generierungsalgorithmen auf. Im Bezug auf die quantitative Ergebnisbeurteilung verdeutlicht vor allem das Spannungsfeld die Komplexität des hier fokussierten Themas. Des Weiteren ist ersichtlich, dass eine qualitative Analyse zwischen Numerik und Analytik, keine nennenswerten Unterschiede aufzeigt.

### A.6.1. Erster Validierungsfall - Grenzfall Würfel

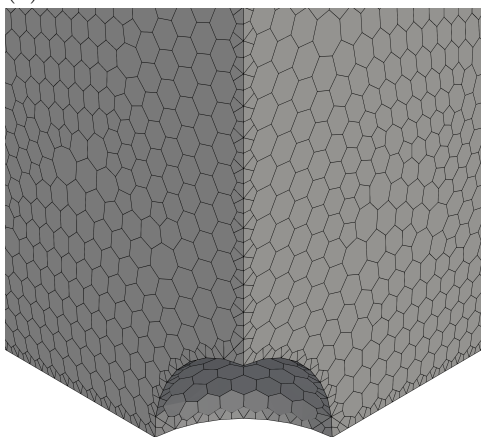
Die nachstehenden Darstellungen geben die erstellten Netze der unterschiedlichen Volumentypen des ersten Validierungsfalls wieder. Zur besseren Darstellung wird nur ein Ausschnitt gezeigt. Anschließend werden die qualitativen Ergebnisse präsentiert.



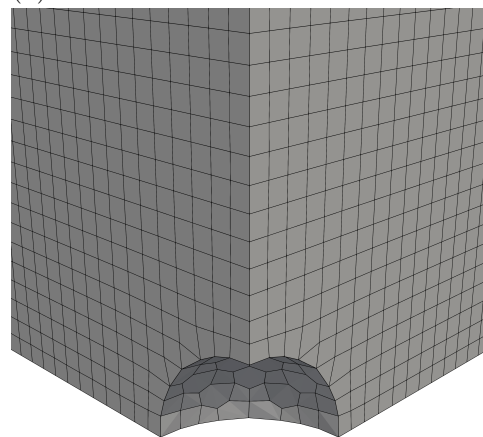
(a) Hexaeder



(b) Tetraeder

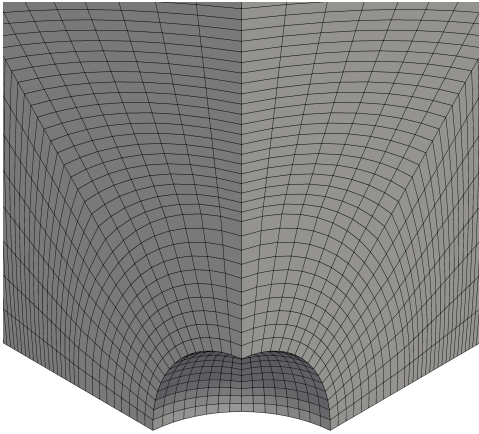


(c) Polyeder

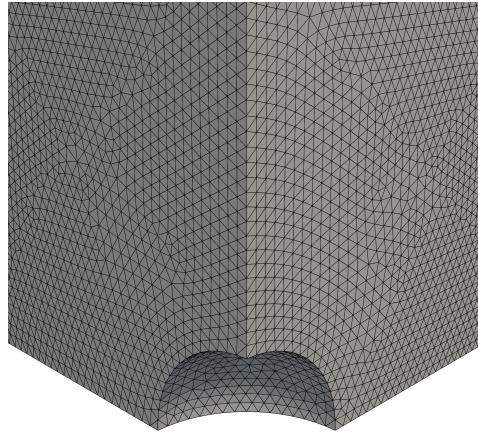


(d) Hexadominant

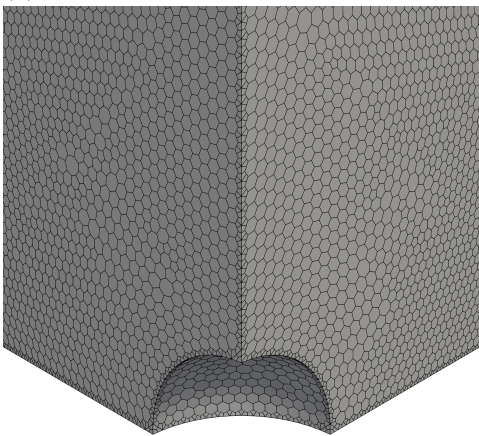
Abbildung A.5.: Numerische Netztypen für den ersten Validierungsfall; Vernetzungsdichte grob.



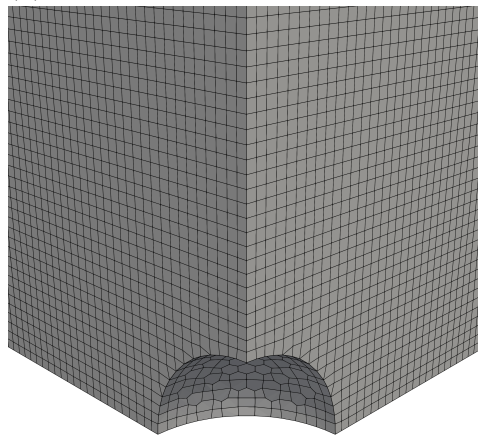
(a) Hexaeder



(b) Tetraeder

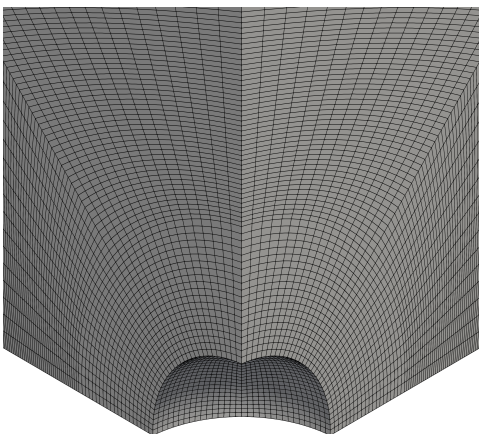


(c) Polyeder

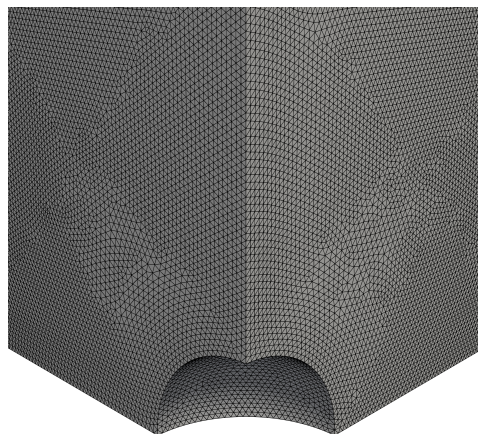


(d) Hexadominant

Abbildung A.6.: Numerische Netztypen für den ersten Validierungsfall; Vernetzungsdichte mittel.

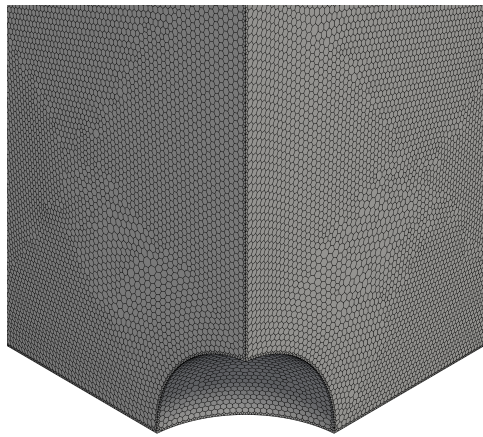


(a) Hexaeder

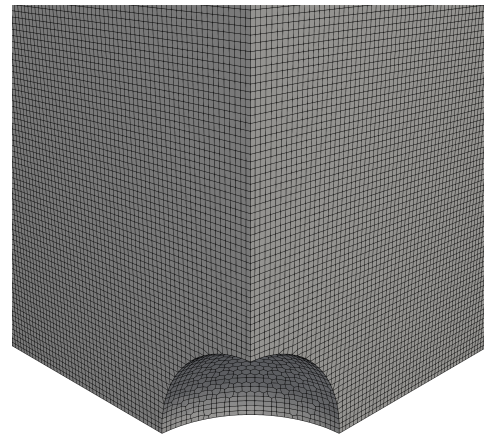


(b) Tetraeder



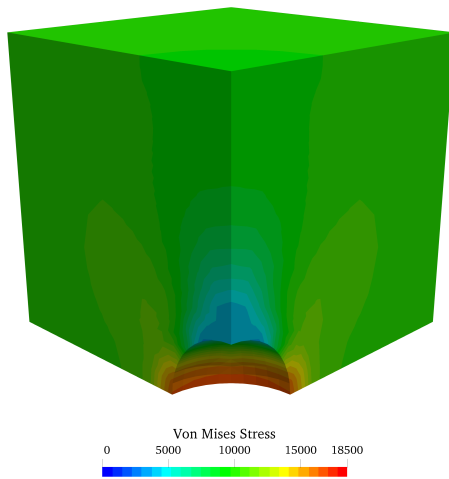


(c) Polyeder

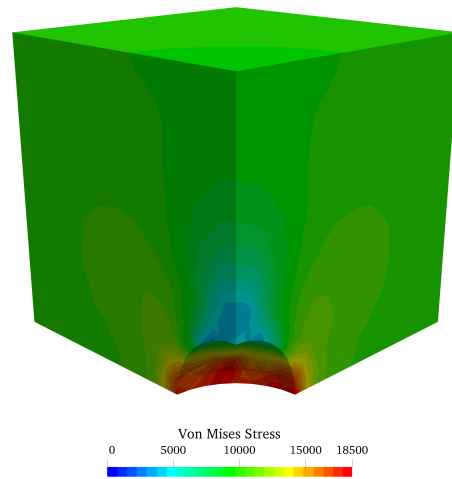


(d) Hexadominant

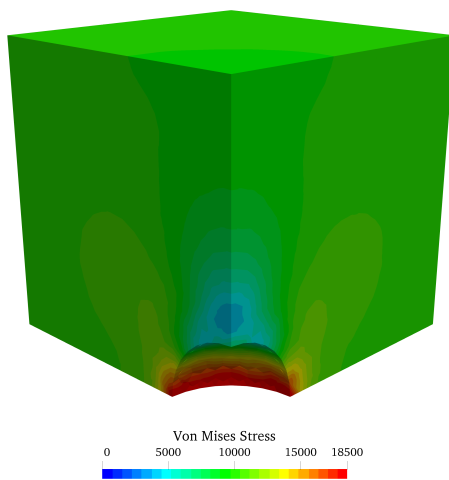
Abbildung A.7.: Numerische Netztypen für den ersten Validierungsfall; Vernetzungsdichte fein.



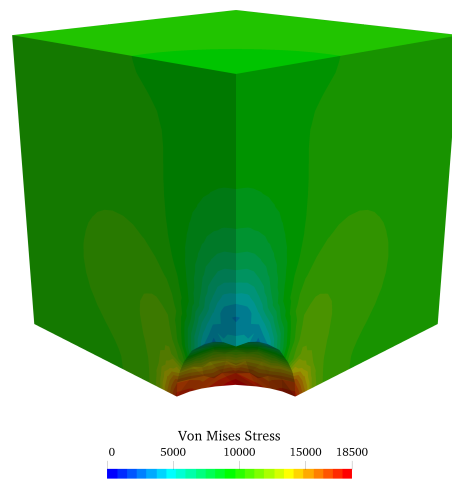
(a) Hexaeder



(b) Tetraeder

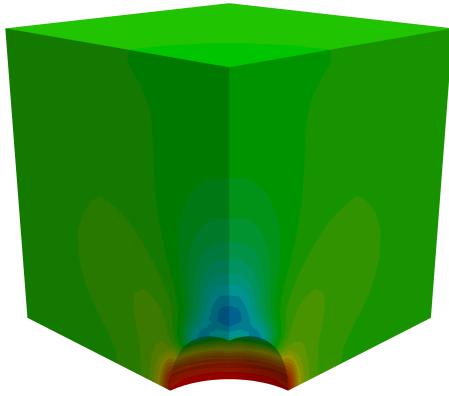


(c) Polyeder

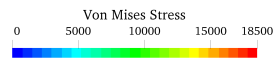
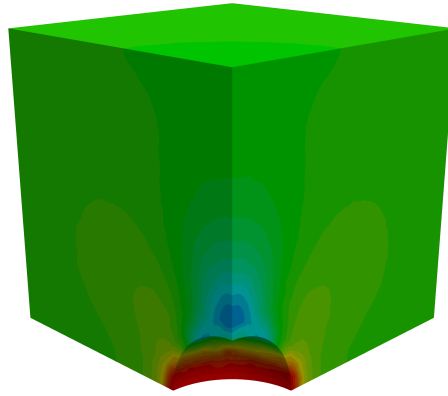


(d) Hexadominant

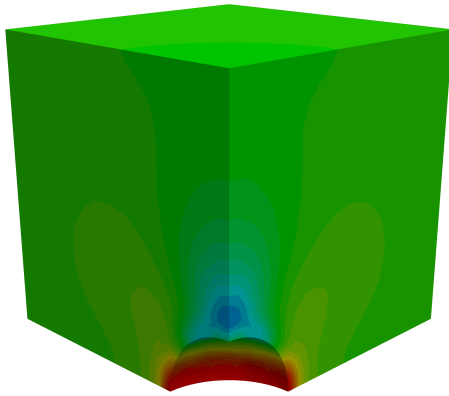
Abbildung A.8.: Visualisierung der numerischen Berechnungsdaten für die vier verschiedenen Netztypen (grobe Vernetzung) des ersten Validierungsfalls.



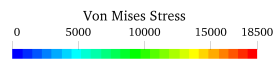
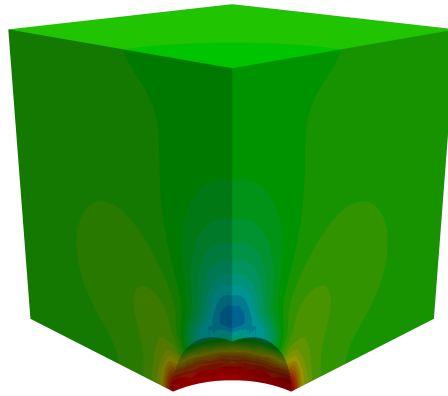
(a) Hexaeder



(b) Tetraeder

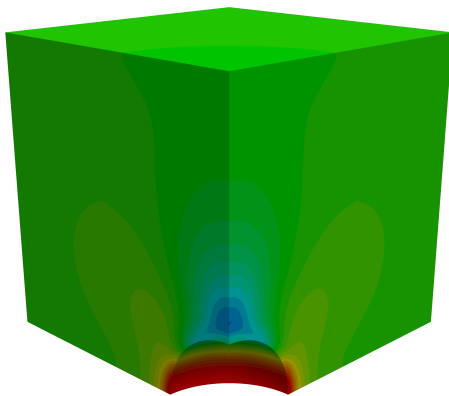


(c) Polyeder

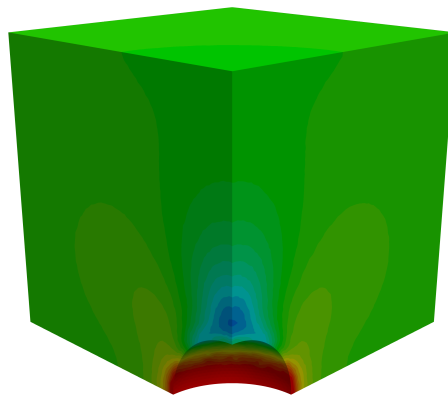


(d) Hexadominant

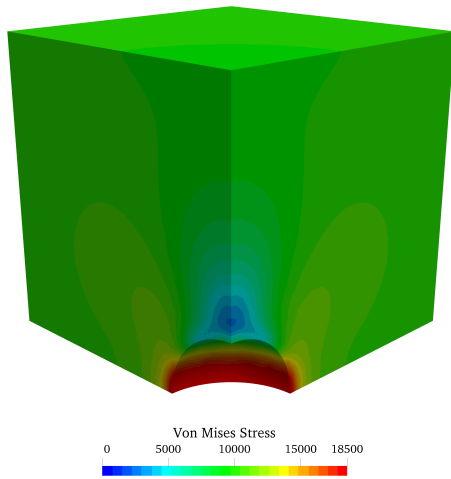
Abbildung A.9.: Visualisierung der numerischen Berechnungsdaten für die vier verschiedenen Netztypen (mittlere Vernetzung) des ersten Validierungsfalls.



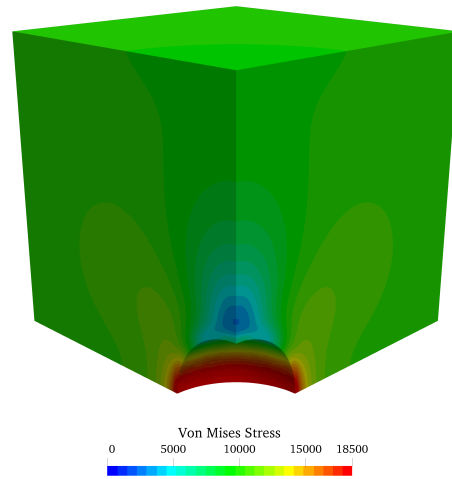
(a) Hexaeder



(b) Tetraeder



(c) Polyeder



(d) Hexadominant

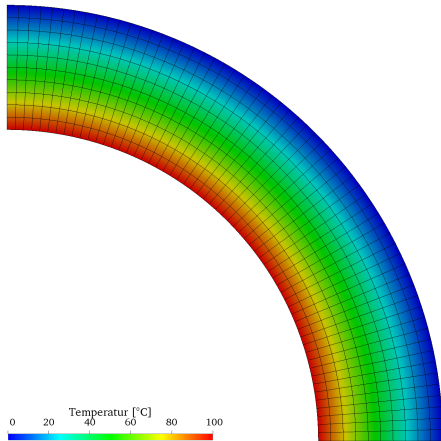
Abbildung A.10.: Visualisierung der numerischen Berechnungsdaten für die vier verschiedenen Netztypen (feine Vernetzung) des ersten Validierungsfalls.

### A.6.2. Zweiter Validierungsfall - Hohlrohr

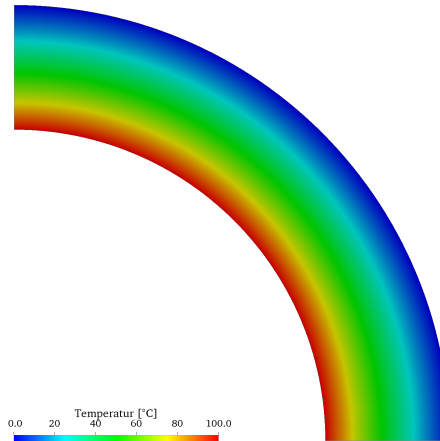
Die Thermospannung aus Abbildung A.11c und A.11d zeigt den Effektivwert des Spannungstensors  $\sigma$ . Die Berechnung erfolgt nach folgender Gesetzmäßigkeit:

$$\|\sigma\| = \sqrt{\sigma : \sigma} . \quad (\text{A.48})$$

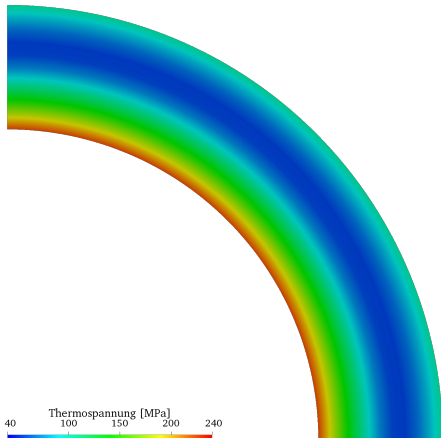
Der Wurzelausdruck repräsentiert das *double inner product* der zwei Tensoren; vergleiche Greenshields (2015) oder Holzmann (2016a).



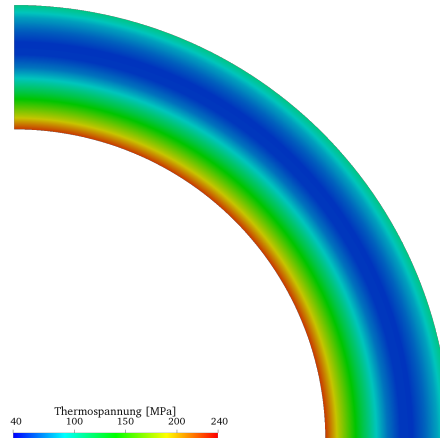
(a) Temperaturverteilung (Analytik)



(b) Temperaturverteilung (Numerik)



(c) Thermospannungen (Analytik)

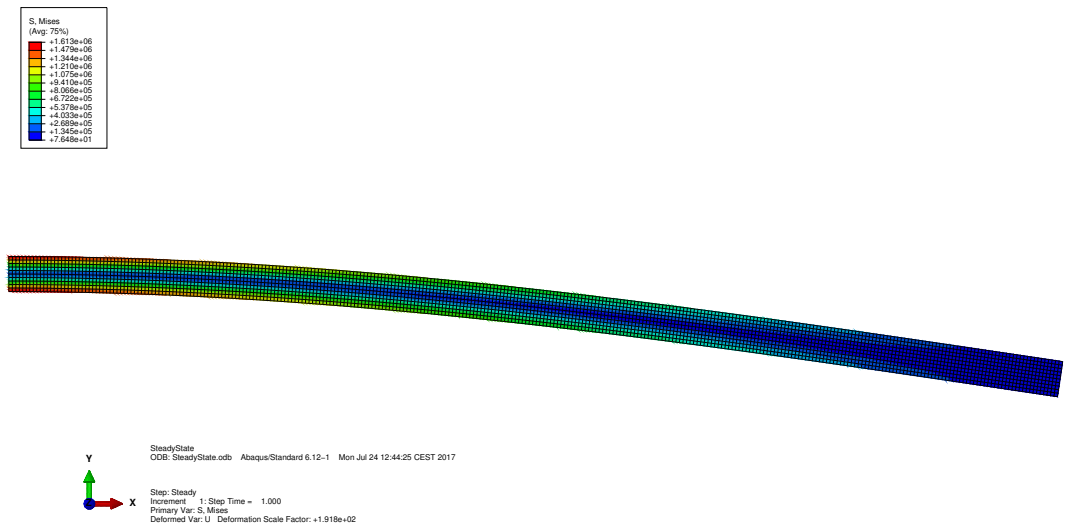


(d) Thermospannungen (Numerik)

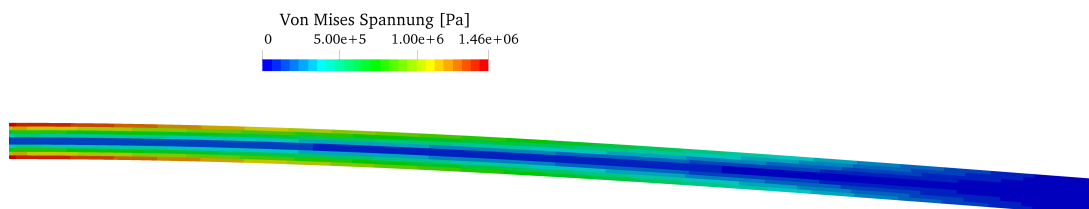
Abbildung A.11.: Visualisierung der analytischen und numerischen Berechnungsdaten; (a) zeigt die räumliche Diskretisierung. Numerische Ergebnisse zum Zeitpunkt  $t = 300$  s (stationäre Lösung).

### A.6.3. Dritter Validierungsfall - Biegebalken

Die Gegenüberstellung der visuellen Ergebnisse des Biegebalkens werden mit einem *Wrap-Vektor Filter* dargestellt. Das bedeutet, dass der Verschiebungsvektor  $\mathbf{D}$  verwendet wird, um das Netz visuell zu deformieren. Es soll angemerkt werden, dass die Auslenkungseinstellungen in Abaqus<sup>®</sup> und OpenFOAM<sup>®</sup> unterschiedlich gewählt sind. Dies ist der Grund für die Differenz der Auslenkung des Biegebalkens für die unten dargestellten Fälle. Das numerische Netz ist für beide Softwaretools identisch.



(a) FEM Berechnung



(b) FVM Berechnung

Abbildung A.12.: Visualisierung der numerischen Ergebnisse aus der FEM und FVM Berechnung unter Verwendung eines *Wrap-Vektor Filter* zur Visualisierung der Verformung im stationären Zustand; Dargestellt sind die Von Mises Spannungen.

## A.7. thermalElasticStressFoam C++ Code Beschreibung

Die Erläuterung des C++ Codes für den entwickelten Löser, welcher thermische Spannungsprobleme berechnen kann, wird nachfolgend gegeben. Es werden auf die wesentlichen Programmierzeilen eingegangen, da die C++ Applikation zu umfangreich ist. Der Quellcode kann auf dem beigefügten Datenträger unter `Doktorarbeit/ Programming/thermalElasticStressFoam` eingesehen werden.

### A.7.1. Der Strukturbaum

Die Datenstruktur des Programms ist im Quellcode 3 ersichtlich. Wie jede C++ Applikation besteht diese aus einer Quelldatei (Source) und mehreren Headerdateien. Im `Make` Order befinden sich die Datensätze für den Compiler.

```
thermalElasticStressFoam
|-- calcSigmaEpsilon.H
|-- createFields.H
|-- createStressFields.H
|-- DEqn.H
|-- laserControl.H
|-- readDSolutions.H
|-- readLaserControl.H
|-- readPolyCoeffs.H
|-- readTimeControls.H
|-- readTSolutions.H
|-- TEqn.H
|-- thermalElasticStressFoam.C
|-- updateThermodynamicFields.H
|-- write.H
|-- writePimple.H
|-- Make
    |-- files
    |-- options

1 directory , 17 files
```

Quellcode 3: Struktur der Applikation `yieldStrengthLaplacianFoam`.

### A.7.2. Code Beschreibung

Die Diskussion des C++ Codes wird kurz gehalten, da zum einen der Quellcode mit sehr vielen Kommentaren versehen ist und zum anderen die Applikation mehr als 2000 Codezeilen aufweist. Daher werden nur die Kernfunktionen beschrieben.

Die `thermalElasticStressFoam.C` Datei stellt das Programm dar. Quellcode 4 zeigt die Zeitschleife, in der die Energie- und Verschiebungsgleichung gelöst wird. Die meisten Kommentare sowie die Codezeilen zur Zeitschrittanpassung werden aus Gründen der Übersichtlichkeit entfernt.

Ersichtlich ist die entkoppelte Lösung zwischen Energie- und Verschiebungsgleichung. In den Headerdateien `TEqn.H` und `DEqn.H` werden die Matrixsysteme erstellt und gelöst.

```

9  while (runTime.run())
10 {
11     //- Some features for time-step adjustments
12     {
13         ...
14     }
15
16     runTime++;
17
18     Info << "Time_=" << runTime.timeName() << nl << endl;
19     {
20         //- Solve Temperature equation
21         #include "TEqn.H"
22
23         //- Solve Momentum equation for displacements and calculate the
24         // Cauchy stress tensor as well as the Von Mises Stress
25         #include "DEqn.H"
26
27         //- Write fields
28         #include "write.H"
29     }
30
31     Info << "ExecutionTime_=" << runTime.elapsedCpuTime() << "s"
32         << "ClockTime_=" << runTime.elapsedClockTime() << "s"
33         << nl << endl;
34 }
35
36 Info << "End\n" << endl;

```

Quellcode 4: Ein Ausschnitt des Quellcodes aus der `yieldStrengthLaplacianFoam.C` Datei.

### Die `TEqn.H` Header Datei

Die Energiegleichung 2.16 wird in OpenFOAM<sup>®</sup>, wie aus Quellcode 5 und 6 ersichtlich ist, umgesetzt und gelöst. Zu Beginn wird geprüft, ob der Benutzer das Lösen der Energiequelle betrachten möchte. Anschließend werden Eingabeparameter gelesen und einige Objekte, die in der weiteren Berechnung benötigt werden, erstellt. Im Zeilenabschnitt von 29–80 wird die Energiegleichung gelöst. Diverse Konditionen werden überprüft und die korrekten Relaxationsfaktoren gesetzt. In Zeile 54 wird die Matrix erstellt und ferner relaxiert und gelöst. Nach dem Lösen des linearen Gleichungssystems wird das Temperaturfeld relaxiert. Es folgen einige Ausgaben für den Benutzer und der Vorfaktor  $\psi$  der Energiequelle wird, basierend auf der Maximaltemperatur, justiert. Hinzu kommt die Aktualisierung der thermodynamischen Größen. Ab Zeile 80 werden weitere Ausgaben auf den `stdout` geleitet (Bildschirm) und die Simulation abgebrochen, sofern die maximale Temperatur in der numerischen Domäne größer als die Solidustemperatur der Legierung ist.

```

9  if (thermalStress)
10 {
11     //- Run time modifiable solution control
12     #include "readTSolutions.H"
13
14     //- Residual for temperature equation
15     SolverPerformance<scalar> solPerformance;
16
17     //- Iteration counter and bool
18     int nCorrT = 0;
19     bool lastIter = false;
20
21     //- Relaxation values for field and equation (change in last iteration)
22     scalar TRelField = TRelaxField;
23     scalar TRelEqn = TRelaxEqn;
24
25     //- Solution settings for solver (change in last iterations)
26     dictionary solverSettingsT = solverSettingsTRelaxed;
27
28     //- Solve temperature equation
29     do
30     {
31         T.storePrevIter();
32
33         //- Minimum and maximum temperature in the domain
34         Tmin = gMin(T);
35         Tmax = gMax(T);
36
37         //- Check if last iteration
38         if ((solPerformance.initialResidual() < Ttol) && (nCorrT > 0))
39         {
40             lastIter = true;
41             TRelEqn = TRelaxEqnFinal;
42             TRelField = TRelaxFieldFinal;
43             solverSettingsT = solverSettingsTFinal;
44         }
45
46         if (monitorRelaxT)
47         {
48             Info<< "T_relax_field_=" << TRelField << "\n"
49                 << "T_relax_eqn_=" << TRelEqn << endl;
50         }
51
52         nCorrT++;
53
54         fvScalarMatrix TEqn
55         (
56             cp*fvm::ddt(rho, T) == fvm::laplacian(k, T)
57         );

```

Quellcode 5: Teil 1 des C++ Codes aus der TEqn.H Datei.



```

59     //- Relax Equation - Solve Equation - Relax Field
60     TEqn.relax(TRelEqn);
61     solPerformance = TEqn.solve(solverSettingsT);
62     T.relax(TRelField);
63
64     //- Output the behavior of the linear system
65     {
66         Info<< "Corr_" << nCorrT << " :_"
67             << solPerformance.solverName() << "_T_field ,_Initial_residual_"
68             << solPerformance.initialResidual() << " ,_Final_residual_"
69             << solPerformance.finalResidual() << " ,_Iterations_"
70             << solPerformance.nIterations() << "\n";
71     }
72
73     //- Adjust the laser power coefficient
74     #include "laserControl.H"
75
76     //- Update the thermodynamic fields
77     #include "updateThermodynamicFields.H"
78
79 }
80 while (!lastIter && (nCorrT < outerCorrT));
81
82 if (lastIter)
83 {
84     Info<< "TEqn_converged_after_" << nCorrT << "_outer_loops\n";
85 }
86 else
87 {
88     Info<< "TEqn_not_converged_after_" << outerCorrT << "_outer_loops\n";
89 }
90
91     //- Minimum and maximum temperature in the domain
92     Tmin = gMin(T);
93     Tmax = gMax(T);
94
95     Info<< "T_min/max:_" << Tmin << "/" << Tmax << endl;
96
97     //- Check if we remelt
98     if (Tmax > Tsolidus)
99     {
100         FatalErrorIn("TEqn.H") << "Temperature_is_higher_than_solidus ..."
101             << abort(FatalError);
102     }
103 }

```

Quellcode 6: Teil 2 des C++ Codes aus der TEqn.H Datei.

## Die DEqn.H Header Datei

Der nachfolgende Quellcode ist stark gekürzt; Zeilennummern beachten. Es werden nur die wichtigen Programmierzeilen dargestellt und erläutert.

Nachdem die Energiegleichung gelöst ist, wird die Verschiebungsgleichung (2.10) berechnet. Hierzu wird zu Beginn geprüft, ob die Spannungsberechnung aktiviert ist. Anschließend werden diverse Objekte, die für die Berechnung notwendig sind, erstellt. In Zeile 58–67 wird der explizite Quellterm (2.9) berechnet. Ab Zeile 70 beginnt die äußere Iterationsschleife des Lösungsalgorithmus. Anschließend folgen einige Kontrollstrukturen, die überprüfen, ob das lineare Gleichungssystem konvergiert ist. Es folgt die Erstellung der Vektormatrix, welche die Duhamel-Neumann Gleichung repräsentiert. In Zeile 98 wird geprüft, ob der Anwender bestimmte Zellen fixiert hat. Trifft dies zu, werden alle Einträge der Matrix, die betroffen sind, auf einen Nullvektor gesetzt. Anschließend folgt die Relaxierung der Matrix, dem Lösen des Gleichungssystems und dem Relaxieren des Feldes. Nachdem die neuen Werte des Verschiebungsvektors  $\mathbf{D}$  an allen Positionen bekannt sind, wird der Cauchy Spannungstensor  $\boldsymbol{\sigma}$  berechnet, der für die Berechnung des expliziten Teils herangezogen wird. Ab Zeile 120 werden einige Kontrollen bezüglich Konvergenz und weitere Terminalausgaben durchgeführt. Dieses Prozedere — zwischen Zeile 70 und 128 — wird solange wiederholt, bis die Konvergenz des linearen Systems erfüllt ist. Abschließend werden die Von Mises Spannungen in der Datei `calcSigmaEpsilon.H` berechnet.

```
38 if (solveD)
39 {
40     //- Erstellen diverser Objekte
41     ...

56
57     //- Pre-Calculate the explicit thermal stress part (not changing)
58     if (planeStress)
59     {
60         divSigmaThermal =
61             fvc::grad(E/(1-nu)*alpha*(T-Tref), "grad(sigmaThermal)");
62     }
63     else
64     {
65         divSigmaThermal =
66             fvc::grad((3*lambda + 2*mu)*alpha*(T-Tref), "grad(sigmaThermal)");
67     }
68
69     //- Calculation based on derivation in thesis
70     do
71     {
72         ++nCorrD;
73
74         D.storePrevIter();
75
76         //- Konvergenz checks
77         ...
```

Quellcode 7: Teil 1 des C++ Codes aus der `DEqn.H` Datei.

```

86
87     //- Constructing equation
88     fvVectorMatrix DEqn
89     (
90         fvm::d2dt2(rho, D)
91         ==
92         fvm::laplacian((2*mu+lambda), D, "laplacian(twoMuLambda,D)")
93         + divSigmaExplicit
94         - divSigmaThermal
95     );

96     //- If cells are fixed, set the value and exclude the cells
97     // from the matrix
98     if (setValuesD)
99     {
100         //labelUList test = cellIDs;
101
102         DEqn.setValues
103         (
104             cellIDs ,
105             List<vector>(cellIDs->size() ,
106             setValuesValue)
107         );
108     }

109
110     //- Relax Equation - Solve Equation - Relax Field
111     DEqn.relax(DRelEqn);
112     solPerformance = DEqn.solve(solverSettingsD);
113     D.relax(DRelField);
114
115     //- Update explicit fields with new displacement field
116     gradD = fvc::grad(D);
117     sigma = 2 * mu * symm(gradD) + lambda * tr(gradD)* I;
118     divSigmaExplicit = fvc::div(sigma-(2*mu+lambda)*gradD, "div(sigma)");
119
120     //- Berechnung des relativen Residuals
121     ...
122
123     //- Ausgabe der Performance des linearen Loesers
124     ...
125
126     //- Ausgabe weiterer Information im Terminal
127     ...
128 }

238     //- Calculate the Von Mises Stress and Strain
239     #include "calcSigmaEpsilon.H"

```

Quellcode 8: Teil 2 des C++ Codes aus der DEqn.H Datei.

## Die calcSigmaEpsilon.H Header Datei

Die Berechnung des Spannungstensors  $\sigma$  und Verschiebungstensors  $\mathbf{D}^S$  wird nach den Gesetzmäßigkeiten aus Gleichung (2.6) und (2.3) bestimmt. Es soll angemerkt werden, dass `epsilon` dem Verschiebungsvektor  $\mathbf{D}^S$  entspricht und der Ausdruck über die Funktion `symm(gradD)` erhalten wird. Im Spannungstensor  $\sigma$  muss der thermische Anteil mitberücksichtigt werden. In Zeile 18 und 19 werden die äquivalenten von Mises Spannungen und Dehnungen mittels Gleichung (2.11) und (2.12) bestimmt.

```
3  //- Update strain and stress field with converged
4  gradD = fvc::grad(D);
5  epsilon = symm(gradD);
6
7  if (planeStress)
8  {
9      sigma = 2*mu*epsilon+lambda*tr(epsilon)*I-(E/(1-nu))*alpha*(T-Tref)*I;
10 }
11 else
12 {
13     sigma =
14         2*mu*epsilon+lambda*tr(epsilon)*I-(3*lambda+2*mu)*alpha*(T-Tref)*I;
15 }
16
17 //- Calculate the von Mises equivalent strain and stress
18 sigmaVonMises = sqrt((3./2.)*magSqr(dev(sigma)));
19 epsilonVonMises = sqrt((2./3.)*magSqr(dev(epsilon)));
20
21 //- Output the Von Mises equivalent strain and stress
22 Info<< "Von_Mises_Strain_[-]_min/max:_" << gMin(epsilonVonMises) << "/"
23     << gMax(epsilonVonMises) << "\n";
24 Info<< "Von_Mises_Stress_[MPa]_min/max:_" << gMin(sigmaVonMises)/1e6 << "/"
25     << gMax(sigmaVonMises)/1e6 << "\n" << endl;
26
27
28 //- Update fields of interest; based on the problematic of stress calculation
29 if (volumeOfInterest != "none")
30 {
31     //- Bestimmung der Cells Of Interest und Bildschirmausgaben
32     ...
33 }
```

Quellcode 9: Gekürzter C++ Code der `calcSigmaEpsilon.H` Datei.

## B. Numerik

Das folgende Kapitel diskutiert einige wichtige Aspekte der Numerik. Diesbezüglich wird aufgezeigt, wie ein lineares Gleichungssystem aussieht, welche Lösungsprozedur in dieser Arbeit verwendet wird und wie sich die Feld- und Matrixrelaxation auf transiente und stationäre Problemstellungen auswirkt. Ferner wird der Einfluss des finalen Relaxationsfaktors aufgezeigt sowie zwei Berechnungsmöglichkeiten des Residuums diskutiert und die Einflussnahme auf die Ergebnisse erörtert. Es werden Grundlagen im Bereich der numerischen Simulation vorausgesetzt.

### B.1. Das lineares Gleichungssystem

In der numerischen Simulation wird ein kontinuierliches Kontinuum in ein diskretes überführt. Für jede numerische Zelle wird eine Erhaltungsgleichung gelöst. Allgemein liegen diese Gleichungen als Integralform vor. Entsprechend muss jeder Term der Gleichungen diskretisiert werden. Diesbezüglich wird an jeder numerischen Zelle ein lineares Gleichungssystem (LGS) erhalten. Dieses System kann wie folgt repräsentiert werden:

$$\mathbf{Ax} = \mathbf{b} . \tag{B.1}$$

Dabei stellt  $\mathbf{A}$  die Koeffizientenmatrix,  $\mathbf{x}$  den Lösungsvektor und  $\mathbf{b}$  den Quellvektor dar. Dieses LGS muss mithilfe numerischer Methoden berechnet werden. Hierfür werden lineare Löser verwendet. In der FVM liegt die Koeffizientenmatrix als Bandmatrizen vor. Diese besitzen die Eigenschaft, dass zur Hauptdiagonalen nur eine bestimmte Anzahl von Nebendiagonalen existieren; vergleiche Ferziger und Perić (2008). Diesbezüglich werden iterative Lösungsverfahren verwendet, um das LGS zu lösen.

### B.2. Der lineare Löser

Der lineare Löser berechnet das LGS  $\mathbf{Ax} = \mathbf{b}$  bis zu einer gewissen Genauigkeit. Das LGS wird als gelöst betrachtet, wenn die Abweichung von  $\epsilon = \mathbf{Ax} - \mathbf{b}$  sehr klein ist.  $\epsilon$  bezeichnet dabei das Residuum und kann auf verschiedene Arten berechnet werden. In OpenFOAM<sup>®</sup> wird beispielsweise die L1-Norm herangezogen, um diesen Wert zu berechnen. Die Definition dieser Norm wird später vorgestellt.

Es ist hierbei jedoch zu unterscheiden, ob der Quellterm  $\mathbf{b}$  vom Lösungsvektor  $\mathbf{x}$  abhängig ist oder nicht. Generell werden Terme die im Quellterm enthalten sind und von der Lösung selbst abhängig sind als *explizite* Terme bezeichnet. In diesem Fall ist es nicht ausreichend, dass LGS  $\mathbf{Ax} = \mathbf{b}$  nur einmal zu berechnen, da die Änderung des Lösungsvektors  $\mathbf{x}$  mit einer Änderung des Quellvektors  $\mathbf{b}$  einhergeht. Zur korrekten Berechnung dieser Problemstellungen müssen erweiternde Algorithmen erstellt werden. Der in dieser Arbeit verwendete Algorithmus wird im nächsten Kapitel erklärt.

### B.3. Lösungsalgorithmus zur Berechnung des linearen Gleichungssystems

Der Lösungsalgorithmus, welcher in dieser Arbeit zum Einsatz kommt, wird nachfolgend vorgestellt und bezieht sich sowohl auf die Erhaltungsgleichung der Energie als auch auf die Duhamel-Neumann Formulierung.

Da beide Erhaltungsgleichungen explizite Terme aufweisen — in der Energiegleichung ist dies aufgrund von Randbedingungen der Fall —, muss das LGS, für jeden Zeitschritt, mehrmals berechnet werden.

Der vorgestellte Algorithmus arbeitet ähnlich wie der bekannte PIMPLE-Algorithmus, welcher für die Druck-Korrektur verwendet wird; der Name der Methode ist auf die Kombination aus den bekannten PISO- und SIMPLE-Algorithmen zurückzuführen. PISO steht dabei für *Pressure Implicit with Splitting of Operator* und SIMPLE für *Semi-Implicit Method for Pressure Linked-Equations*.

Zur korrekten Berechnung der Erhaltungsgleichungen wird das Schema, welches in Abbildung B.1 gezeigt ist, verwendet. Hier sind zwei Iterationsschleifen ersichtlich. Zum einen wird eine

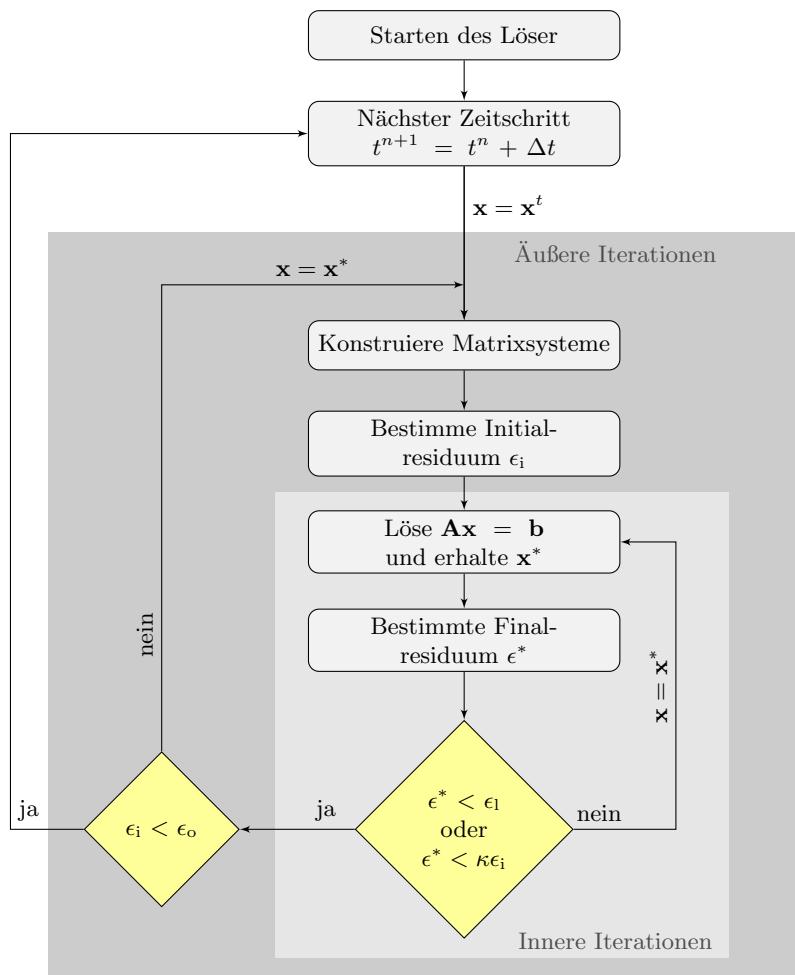


Abbildung B.1.: Schematische Darstellung der Lösungsprozedur des Algorithmus mit innerer und äußerer Schleife für das Gleichungssystem. Die Subskripte der Residuumsvariable  $\epsilon$  stehen für: i Initial, l Linearer Löser und o Outer-Loop (äußere Schleife). Die letzten beiden sind Eingabeparameter.  $\kappa$  steht für einen Vorfaktor und ist vom Benutzer vorzugeben. Dieser Wert ist stets kleiner als Eins.

*innere* Schleife und zum anderen eine *äußere* Schleife durchlaufen. Dieses Schema ist allgemeingültig und kann für willkürliche Problemstellungen herangezogen werden.

### Innere Iterationsschleife

Die innere Iterationsschleife berechnet das LGS  $\mathbf{Ax} = \mathbf{b}$ , wobei die Matrixkoeffizienten  $\mathbf{A}$  und der Quellterm  $\mathbf{b}$  mit den Werten  $\mathbf{x}$  der vorherigen Iteration konstruiert werden. Das LGS wird vom linearen Löser solange gelöst, bis entweder das Konvergenzkriterium des linearen Löfers  $\epsilon_1$ , oder eine relative Toleranz gegenüber dem Initialresiduum  $\epsilon_1$  des Matrixsystems erreicht wird. Die relative Toleranz wird durch das Multiplizieren des Faktors  $\kappa$  mit dem Initialresiduum ermittelt und ist besonders für stark explizite Gleichungen von Vorteil;  $\kappa < 1$ . Der Grund hierfür liegt in der starken Kopplungsbeziehung zwischen dem Lösungsvektor  $\mathbf{x}^*$  und dem Quellterm  $\mathbf{b}$ .

### Äußere Iterationsschleife

Die äußere Iterationsschleife dient dazu, das LGS  $\mathbf{Ax} = \mathbf{b}$  bis zur endgültigen Konvergenz zu lösen. Das bedeutet, dass der Lösungsvektor  $\mathbf{x}$  und der Quellvektor  $\mathbf{b}$  sich nicht weiter verändern und damit das Initialresiduum dementsprechend klein ist.

In der äußeren Schleife wird der neu errechnete Lösungsvektor  $\mathbf{x}^*$  verwendet, um die Koeffizientenmatrix  $\mathbf{A}$  und den Quellvektor  $\mathbf{b}$  zu aktualisieren.

Diese Prozedur wird solange durchgeführt, bis, wie bereits erwähnt, das Initialresiduum des LGS kleiner ist als das vom Benutzer geforderten Konvergenzkriterium  $\epsilon_0$ .

### Schlüsselworte in OpenFOAM<sup>®</sup>

Die oben aufgeführten Toleranzeinstellungen für den linearen Löser werden in OpenFOAM<sup>®</sup> in der Datei `system/fvSolutions` eingestellt. Es ist zudem möglich eine minimale und maximale Anzahl der Iterationen des linearen Löfers vorzugeben. Nachstehender Codeausschnitt zeigt die Parameter, die für den linearen Löser eingestellt werden können.

```
1 D
2 {
3     solver          PBiCGStab;          // Linearer Loeser
4     preconditioner  DILU;              // Vorkonditionierer
5     tolerance       1e-7;              // Abbruchkriterium Innere Schleife
6     relTol          0.1;                // Abbruchkriterium Faktor kappa
7     maxIter         100;                // Maximale Anzahl an Inneren Iterationen
8     minIter         1;                  // Minimale Anzahl an Inneren Iterationen
9 }
```

Quellcode 10: Codeausschnitt zur Bestimmung der Abbruchkriterien für die innere Schleife in `system/fvSolutions`.

Die Einstellungen für das Abbruchkriterium der äußeren Schleife ist in der Unterstruktur der Schlüsselwörter `PIMPLE` und `residualControl` in derselben Datei zu finden. Es wird angemerkt, dass wie beim linearen Löser, die äußere Schleife zudem beendet werden kann, wenn eine Maximalanzahl von äußeren Iterationen durchgeführt worden ist. Im Quellcode 11 liegt diese Grenze beispielsweise bei 50000 Iterationen.

```

1 PIMPLE
2 {
3     //- Outer corrections for TEqn
4     outerCorrT      600;
5
6     //- Outer corrections for DEqn (used after calcSigma time is reached)
7     outerCorrD      50000;
8
9     //- Outer corrections for DEqn (used in every time step)
10    // + if set to zero - DEqn is only solved after calcSigma
11    outerCorrD2      0;
12 }
13
14 residualControl
15 {
16     TTolerance      1e-6;
17     DTolerance      1e-6;
18
19     //- Use initial residual from matrix (init)
20     // or relative residual from previous iteration (relative)
21     DToleranceMethod  init;
22 }

```

Quellcode 11: Codeausschnitt zur Bestimmung der Abbruchkriterien für die äußeren Schleifen und der verwendeten Methodik in `system/fvSolutions`.

### Anmerkung

Das Lösen der Verschiebungsgleichung ist rechenaufwendig und nicht zu jedem Zeitpunkt notwendig. Daher sind in oben stehender Ausführung zwei Angaben zu den äußeren Schleifen der Verschiebungsgleichung zu sehen. Die Berechnung der Verschiebungsgleichung erfolgt mit einem anderen Zeitschritt als die Lösung der Temperaturgleichung — sub-time-stepping. Es steht dem Anwender jedoch frei, das Zeitintervall für die Berechnung des Verschiebungsvektors so zu wählen, dass dieser für jeden Zeitschritt ausgewertet wird. Dies ist aber aufgrund der Problemstellung nicht nötig und reduziert die Geschwindigkeit des Lösungsprozesses. Optional kann mit der Angabe `outerCorrD2` eine definierte Anzahl an äußeren Iterationen angegeben werden, die die Verschiebungsgleichung nach jedem Zeitschritt (bezogen auf das Lösen der Temperatur) berechnet. Weitere Details befinden sich im Quellcode des Lösers, welcher auf dem Datenträger unter `Doktorarbeit/Programming/thermalElasticStressFoam` zu finden ist.

## B.4. Konvergenzmethoden

Beim Lösen linearer Gleichungssysteme, wird über das zu lösende algebraische Gleichungssystem  $\mathbf{Ax} = \mathbf{b}$  so lange iteriert, bis der Restfehler  $\epsilon$  — das Residuum — eine gewisse Schwelle unterschreitet. Das Residuum  $\epsilon$  kann dabei mit unterschiedlichen Methoden bestimmt werden.

### B.4.1. L1-Norm

In OpenFOAM<sup>®</sup> wird das Residuum mit der L1-Norm bestimmt, wobei das Residuum zwischen 0 und 1 skaliert wird. In OpenFOAM<sup>®</sup> wird hierzu zuerst der Residuumsvektor  $\mathcal{E}$  berechnet; vergleiche Foundation (2018). Es folgt:

$$\mathcal{E} = \mathbf{b} - \mathbf{Ax}^n . \tag{B.2}$$



$\mathbf{x}^n$  stellt dabei die Lösung der aktuellen Iteration dar;  $n$  bezeichnet den Iterationsindex. Um das Residuum  $\epsilon$  zu erhalten, wird anschließend eine Normalisierung durchgeführt. Der Normalisierungsfaktor  $\mathcal{F}$  wird wie folgt bestimmt:

$$\mathcal{F} = \sum (|\mathbf{A}\mathbf{x}^n - \mathbf{A}\bar{\mathbf{x}}^n| + |\mathbf{b} - \mathbf{A}\bar{\mathbf{x}}^n|) , \quad (\text{B.3})$$

wobei  $\bar{\mathbf{x}}^n$  den Durchschnittswert von  $\mathbf{x}^n$  im Lösungsgebiet darstellt. Abschließend wird das skalierte Residuum  $\epsilon$  berechnet:

$$\epsilon = \frac{1}{\mathcal{F}} \sum |\mathbf{b} - \mathbf{A}\mathbf{x}^n| . \quad (\text{B.4})$$

Der Term in der Summe stellt die Abweichung zwischen Quellterm  $\mathbf{b}$  und der derzeitigen Lösung  $\mathbf{A}\mathbf{x}^n$  in jeder numerischen Zelle dar. Anschließend wird der Betrag davon aufsummiert (L1-Norm) und durch den Normalisierungsfaktor  $\mathcal{F}$  geteilt.

Der Fehler  $\epsilon$  beschreibt damit die Abweichung des Matrixsystems  $\mathbf{A}\mathbf{x} - \mathbf{b} = 0$ . Ist das Residuum zu groß, wird das Lösungssystem erneut gelöst; vergleiche *Jacobi*-Algorithmus, *LU*-Zerlegung und andere Methoden; siehe auch Press et al. (2007); Ferziger und Perić (2008) sowie Galassi et al. (2009). Es soll angemerkt werden, dass durch die Normalisierung des Fehlers in speziellen Fällen eine weitaus kleinere Toleranzgrenze vorgegeben werden muss.

Die Bestimmung des Restfehlers mit dieser Methodik wird in vorliegender Arbeit als *initial* bezeichnet (L1-Norm). Der Name rührt daher, da nach jedem Lösen des Gleichungssystems Gleichung (B.4) evaluiert wird und der erhaltene Fehler den Initialfehler für den nächsten Iterationsschritt darstellt.

#### B.4.2. Relative Methode – Feldbetrachtung

Im Gegensatz zur L1-Norm ist es möglich für bestimmte Erhaltungsgleichungen, wie beispielsweise der Duhamel-Neumann Formulierung, die Bestimmung des Residuums mit anderen Methoden durchzuführen. Nachfolgend soll auf eine *relative* Methode, die den Restfehler  $\epsilon_{\text{rel}}$  nicht durch das Matrixsystem bestimmt, sondern durch die Abweichung des Lösungsvektors  $\mathbf{x}$  selbst (des zu lösenden Feldes), eingegangen werden. Die Berechnung erfolgt nach folgender Gesetzmäßigkeit; vergleiche Cardiff (2012) und Foam Extend (2016):

$$\epsilon_{\text{rel}} = \max \left| \frac{\phi^n - \phi^{n-1}}{\Delta\phi_{\text{max}}} \right| , \quad (\text{B.5})$$

wobei  $\Delta\phi_{\text{max}}$  eine skalare Größe darstellt und definiert ist als:

$$\Delta\phi_{\text{max}} = \max |\phi_t^n - \phi_{t-1}| . \quad (\text{B.6})$$

$\phi$  repräsentiert dabei die gesuchte Größe, hier den Verschiebungsvektor  $\mathbf{D}$ . Diese Methode wird in dieser Arbeit als *relative* bezeichnet. Zu beachten ist, dass die Skalierungsgröße  $\Delta\phi_{\text{max}}$  durch den Wert der derzeitigen Iteration  $\phi_t^n$  und dem letzten Zeitschritt  $\phi_{t-1}$  bestimmt wird. Diese Methode wird in der FVM im Bereich der Spannungsberechnung angewendet. Vor allem im Bereich der stationären Berechnungen. Eine weitere Ausführung ist in Cardiff (2012) gegeben.

Beide Methoden liefern gleiche Ergebnisse, sofern die Einstellungen für den linearen Löser sowie für die innere und äußere Iterationsschleife korrekt eingestellt sind.

## B.5. Feld- und Matrixrelaxation

Für numerische Berechnungen in der FVM gibt es zwei Möglichkeiten die Konvergenzrate eines Algorithmus zu verbessern. Dies ist immer dann nötig, wenn beispielsweise bestimmte Stabilitätskriterien wie die Courant oder Fourier Zahl überschritten werden. Generell müssen diese Stabilitätskriterien eingehalten werden, um die Konvergenz des Lösungsalgorithmus zu garantieren; vergleiche Ferziger und Perić (2008). Allerdings treten häufig die Probleme auf, dass aufgrund hoher Strömungsgeschwindigkeiten oder der Größe und Topologie der numerischen Zellen der Zeitschritt zur Einhaltung der Kriterien in Größenordnungen von  $1e^{-4}$  bis  $1e^{-8}$  — oder kleiner — fällt. Abhängig vom betrachteten System und den interessierenden Zeitskalen kann das Lösen dieses Problems aufgrund des zu hohen Zeitaufwandes nicht vertretbar sein. Diesbezüglich wurden Algorithmen entwickelt, die es ermöglichen, den Zeitschritt deutlich zu erhöhen, ohne die Stabilitätskriterien zu verletzen. Zugleich werden diese Algorithmen häufig für sehr steife Problemstellungen angewendet. Es sei darauf hingewiesen, dass diese Algorithmen korrekt verwendet werden müssen, um die Berechnung der physikalischen Größen fehlerfrei abzubilden. Anderenfalls können stark quantitative Änderungen der Größen auftreten, die sich gegenseitig verstärken, wodurch die Lösung divergiert oder unphysikalische Ergebnisse entstehen.

Der populärste Algorithmus, welcher auch in ähnlicher Form in dieser Arbeit verwendet wird, ist der PIMPLE-Algorithmus. Dieser Algorithmus berechnet zu jedem Zeitschritt die stationäre Lösung, muss aber aufgrund des größeren Zeitschrittes mit der sogenannten Feld- und Matrixrelaxation gekoppelt werden. Beide sind nachfolgend detailliert erläutert.

Weitere Details bezüglich Funktionsweise und Anwendung sind in Ferziger und Perić (2008) und Holzmann (2016a) zu finden. Des Weiteren soll angemerkt werden, dass der PIMPLE-Algorithmus für die Druck-Impulsgleichung hergeleitet wurde aber im Bereich der Festkörperberechnung in ähnlicher Form eingesetzt werden kann.

### B.5.1. Feldrelaxation

Unter Feldrelaxation wird die Einflussnahme der Änderung des neu berechneten Wertes der gesuchten Größe  $\phi$  verstanden. Wird beispielsweise das Gleichungssystem  $\mathbf{Ax} = \mathbf{b}$  iterativ gelöst, wobei  $\mathbf{x}$  der gesuchte Lösungsvektor der Größe  $\phi$  darstellt und mit einem Startvektor  $\mathbf{x}^0$  begonnen, wird nach dem ersten Lösen des Matrixsystems der neue Lösungsvektor  $\mathbf{x}^1$  erhalten. Die Feldrelaxation bestimmt in wieweit sich der neue Lösungsvektor gegenüber dem alten verändert:

$$\mathbf{x}^n = \mathbf{x}^{n-1} + \alpha_R (\mathbf{x}^n - \mathbf{x}^{n-1}) . \quad (\text{B.7})$$

Der Faktor  $\alpha_R$  stellt den Relaxationsfaktor und  $n$  die Anzahl der Iterationen dar. Für  $\alpha_R = 1$  erfolgt keine Relaxation. Der neu errechnete Wert wird direkt als neue Lösung herangezogen. Für  $\alpha_R < 1$  erfolgt eine Unterrelaxation. Das bedeutet, dass die neuen Iterationswerte aus den alten und neuen Werten berechnet wird. Hier wird jedoch die Gewichtung vom neuen Wert herabgesetzt. Eine Überrelaxation ( $\alpha_R > 1$ ) kann nur in gesonderten Fällen eingesetzt werden. Es ist offensichtlich, dass ein Wert von null die gesuchte Größe  $\phi$  nicht ändert, da der neu evaluierte Wert stets dem alten gleichgesetzt wird.

Die Implementierung der oben genannten Formulierung (B.7) ist in OpenFOAM® in der Klasse `GeometricField` definiert. Der entsprechende Codeausschnitt ist nachfolgend gegeben.

```

923 template<class Type, template<class> class PatchField, class GeoMesh>
924 void Foam::GeometricField<Type, PatchField, GeoMesh>::relax(const scalar alpha)
925 {
926     if (debug)
927     {
928         InfoInFunction
929             << "Relaxing" << endl << this->info() << "_by_" << alpha << endl;
930     }
931
932     operator==(prevIter() + alpha>(*this - prevIter()));
933 }

```

Quellcode 12: C++ Implementierung der Feldrelaxation OpenFOAM®.

### B.5.2. Matrixrelaxation

Anders als bei der Feldrelaxation dient die Matrixrelaxation vor allem dazu, eine diagonal dominante Matrix zu erstellen. In den meisten Fällen ist die erstellte Matrix  $\mathbf{A}$  von Natur aus diagonal dominant. Es kann aber bei Problemstellungen mit Schockwellen und ähnlichen Phänomenen vorkommen, dass eine Matrix vorliegt, die nicht-diagonal dominant ist. Da für lineare Löser, die das Matrixsystem berechnen, eine diagonal dominante Matrix erforderlich sein muss — vergleiche Rusche (2002) — wird durch die Matrixrelaxation das Lösungsverhalten für Problemstellungen, bei denen eine nicht-diagonal dominante Matrix erstellt wird, erst ermöglicht oder signifikant verbessert. Eine diagonal dominante Matrix besitzt die Eigenschaft, dass der Wert des Diagonalelements in jeder Zeile der Matrix stets größer oder gleich der Summe des Betrags der neben-Diagonalelemente ist. Die schwächste Form dieser Matrix ist dann gegeben, wenn alle Diagonalelemente gleich der Summe der neben-Diagonalelemente sind, eines davon aber größer ist.

Ein Fall, in dem eine diagonal dominante Matrix nicht mehr gegeben sein muss, kann beispielsweise unter Zuhilfenahme von kompressiblen Strömungen erörtert werden. Werden zwei numerische Zellen betrachtet, die Nachbarn zueinander darstellen, und sich eine Schockwelle in der numerischen Domäne ausbreitet und angenommen wird, dass diese sich in einer der beiden Zellen befindet, so folgt, dass für die Nachbarzelle — in der der Druck noch klein ist — die Diagonaldominanz unter keinen Umständen gegeben ist; vergleiche Abbildung B.2 Zelle  $C_1$  und  $C_2$ . Daher ist es notwendig die Diagonaldominanz herzustellen.

Dies wird in OpenFOAM® wie folgt erreicht:

- Manipuliere die Matrix so, dass zumindest eine diagonal gleichwertig Matrix entsteht,
- Relaxiere die Diagonalelemente (Teile durch den Relaxationsfaktor  $\alpha_R$ ),
- Manipuliere den Quellvektor  $\mathbf{b}$ , so dass die durchgeführten Operationen äquivalent sind.

Es ist anzumerken, dass der Relaxationsfaktor  $\alpha_R = 1$  die Diagonalelemente nicht beeinflusst — zweiter Schritt in obiger Aufzählung ändert die Matrix nicht — und dadurch die Diagonaldominanz nicht verbessert wird. Allerdings kann es aufgrund des ersten Schrittes dazu kommen, dass die Matrix modifiziert wird. Deshalb ist zu beachten, dass die Verwendung der Matrixrelaxation mit dem Wert  $\alpha_R = 1$  durchaus die Matrix verändern kann. Dies tritt nur dann in Kraft, wenn

nicht-diagonal dominante Einträge vorliegen. In diesem Fall wird von einer schlecht konditionierten Matrix gesprochen. Zur Demonstration eines nicht-dominanten Diagonaleintrags in einer Matrix wird Abbildung B.2 herangezogen. Das Beispiel zeigt eine sich sphärisch ausbreitende Schockfront und einen Rechenstern ( $C_1$  bis  $C_5$ ) der nachfolgend weiter betrachtet wird. Wird die Matrix  $\mathbf{A}$  veranschaulicht und speziell auf den Zeileneintrag der Zelle  $C_2$  eingegangen, kann festgestellt werden, dass der Diagonalwert ( $C_2$ ) kleiner ist als die Summe der Beträge der Nachbarzellen. Unter Zuhilfenahme von Abbildung B.2 kann dies einfach begründet werden. Während in  $C_1$  bereits die Schockfront vorliegt, wobei  $p_1 \gg p_2$ , ist ohne weiteren Beweis offensichtlich, dass der Wert  $C_2 \ll C_1$  ist. Da zusätzlich die Zellen der Nebendiagonalen ( $C_3$ ,  $C_4$  und  $C_5$ ) zu  $C_1$  aufsummiert werden, wird die nicht-diagonale Dominanz weiter verstärkt. Siehe nachfolgende Matrixerklärung:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{1m} \\ & \ddots & & & \\ \mathbf{C}_1 & C_4 & \mathbf{C}_2 & C_3 & C_5 \\ & & & \ddots & \\ a_{n1} & a_{n2} & a_{n3} & a_{n4} & a_{nm} \end{bmatrix} . \quad (\text{B.8})$$

Daraus resultiert folgender Beweis:

$$C_2 \ll |C_1| + |C_3| + |C_4| + |C_5| . \quad (\text{B.9})$$

Die Matrix  $\mathbf{A}$  wird bei der Verwendung eines Relaxationsfaktors von  $\alpha_R = 1$  wie folgt abgeändert:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{1m} \\ & \ddots & & & \\ \mathbf{C}_1 & C_4 & \mathbf{C}_2^* & C_3 & C_5 \\ & & & \ddots & \\ a_{n1} & a_{n2} & a_{n3} & a_{n4} & a_{nm} \end{bmatrix} , \quad (\text{B.10})$$

wobei  $C_2^* = |C_1| + |C_3| + |C_4| + |C_5|$ . Die Manipulation der Diagonalelemente ist streng genommen nicht konsistent. Um diesen Schritt auszugleichen, muss der Quellterm entsprechend modifiziert werden; vergleiche die Codezeilen am Ende vom Quellcode 13.

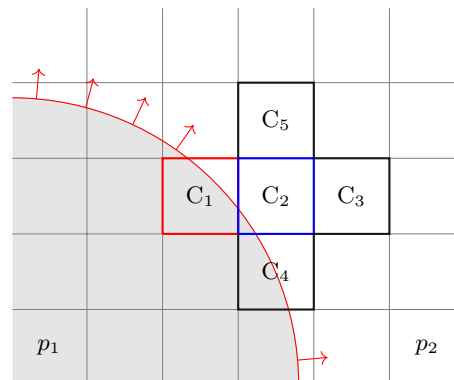


Abbildung B.2.: Skizze einer Schockfront Ausbreitung zur Darstellung eines nicht-diagonal dominanten Matrixeintrags.  $p_1 \gg p_2$ .

Die Implementierung der Matrixrelaxation ist in der Klasse `fvMatrix` vorzufinden und ist nachfolgend mit den wichtigsten Syntaxstellen gegeben. Die eingefügten Kommentare verdeutlichen weiter die durchgeführten Operationen.

```

1  template<class Type>
2  void Foam::fvMatrix<Type>::relax(const scalar alpha)
3  {
4      if (alpha <= 0)
5      {
6          return;
7      }
8
9      if (debug)
10     {
11         InfoInFunction
12             << "Relaxing_" << psi_.name() << "_by_" << alpha << endl;
13     }
14
15     Field<Type>& S = source();
16     scalarField& D = diag();
17
18     // Store the current unrelaxed diagonal for use in updating the source
19     scalarField D0(D);
20
21     // Calculate the sum-mag off-diagonal from the interior faces
22     scalarField sumOff(D.size(), 0.0);
23     sumMagOffDiag(sumOff);
24
25     // Spezialbehandlung fuer gekoppelte Randbedingungen und Debug Mode
26     .
27     .
28
29     // Ensure the matrix is diagonally dominant...
30     // Assumes that the central coefficient is positive and ensures it is
31     forAll(D, celli)
32     {
33         D[celli] = max(mag(D[celli]), sumOff[celli]);
34     }
35
36     // ... then relax
37     D /= alpha;
38
39     // Erneute Behandlung von gekoppelten Randbedingungen
40     .
41     .
42
43     // Finally add the relaxation contribution to the source.
44     S += (D - D0)*psi_.primitiveField();
45 }

```

Quellcode 13: C++ Implementierung der Matrixrelaxation in OpenFOAM®.

### B.5.3. Finaler Relaxationsfaktor

OpenFOAM® bietet die Möglichkeit, zwei verschiedene Relaxationsfaktoren zu definieren. Die Definition ist wie folgt: der generelle Faktor wird solange verwendet, bis das Matrixsystem das Konvergenzkriterium erreicht. Anschließend wird erneut ein letzter *finaler* Iterationsschritt durchgeführt, bei dem die *finalen* Relaxationsfaktoren verwendet werden. Diese müssen stets den Zahlenwert von Eins betragen und werden von OpenFOAM®, sofern nicht anders vorgegeben, auf diesen gesetzt. Die Verwendung eines kleineren Wertes kann die Lösung verfälschen, dass in Anhang B.6 aufgezeigt wird. Jedoch kann die Konvergenz des Lösungsalgorithmus auch verbessert werden.

## B.6. Einfluss der Relaxations- und Konvergenzmethoden

Das folgende Kapitel befasst sich mit den Relaxationsmethoden die in Anhang B.4 beschrieben werden und charakterisiert deren Einfluss auf die Genauigkeit, Konvergenzrate und Berechnungseffizienz in der FVM. Zudem werden einige Sachverhalte bezüglich der Lösungstoleranz für den Matrixlöser, d.h. wie oft das Matrixsystem gelöst wird (innere Iteration), um in die nächste äußere Iteration zu kommen, diskutiert. Ferner wird der Einfluss des *finalen* Relaxationsfaktors für beide Methoden, die in Anhang B.5.3 beschrieben sind, auf das Ergebnis und das Konvergenzkriterium untersucht.

Für die Untersuchung wird der Biegebalken, welcher in Abbildung 2.7 auf Seite 34 dargestellt ist, herangezogen. Das Lösen des linearen Gleichungssystems basiert auf dem bi-konjugierten Gradientenverfahren, das zusätzlich über eine Vorkonditionierung und Stabilisierung der Matrix verfügt — PBiCGStab. Die Konditionierung wird mit der diagonalen unvollständigen Cholesky Zerlegung durchgeführt — FDIC. Jede innere Iteration wird bis zu einem Restfehler von  $\epsilon_1 = 1e^{-10}$  gelöst. Die relative Toleranz  $\kappa$  für den linearen Löser wird auf null gesetzt. Eine detaillierte Übersicht über die Lösungsprozedur des linearen Löser und die einzelnen Toleranzgrößen ist in Anhang B.3 auf Seite 156 gegeben.

Es ist anzumerken, dass bei stark expliziten Gleichungssystemen, wie der hier verwendeten Verschiebungsgleichung  $\mathbf{D}$ , das Lösen des Matrixsystems  $\mathbf{Ax} = \mathbf{b}$  bis zur relativen Toleranzgrenze  $\kappa$  weitaus sinnvoller ist, als die Berechnung bis zum Erreichen des absoluten Toleranzkriteriums  $\epsilon_1$  des linearen Löser. Grund hierfür ist die signifikante Abhängigkeit der Matrix  $\mathbf{A}$  und des Quellterms  $\mathbf{b}$  bezüglich der neuen Lösung  $\mathbf{x}^*$ . Daher ist es sinnvoller, die innere Schleife des Lösungsalgorithmus frühzeitig zu verlassen, um die nächste äußere Iteration einzuleiten und so das System schneller zu konvergieren. Die Bestätigung dieser Aussage ist in der Arbeit von Herrn Jasak und Weller (1998) zu finden; sie wird hier jedoch nochmals explizit an einem Beispiel aufgezeigt.

Die für diese Analyse verwendeten Fallbeispiele sind auf dem beigelegten Speichermedium unter `Doktorarbeit/thermalElasticStressFoam/` zu finden. Das Reproduzieren der Ergebnisse ist vollständig automatisiert; dies beinhaltet das numerische Berechnen aller Beispiele und die zusätzliche Datenmanipulation plus Grafikerstellung.

### B.6.1. Feldrelaxation in transienten Berechnungen

Zur Untersuchung der Feldrelaxation auf das Lösungsverhalten in transienten Anwendungsfällen, werden vier verschiedene Beispiele aufgesetzt. Das Setup der einzelnen Berechnungsfälle ist in Tabelle B.1 gegeben. Die Konvergenzgrenze stellt das Rest-Residuum dar, das die Beendigung der äußeren Iterationen signalisiert und den nächsten Zeitschritt einleitet. Für das Lösen der inneren Iterationen werden die oben genannten Einstellungen verwendet.

Die quantitativen Ergebnisverläufe der vier Berechnungen sind in Abbildung B.3 dargestellt und werden nachfolgend diskutiert. Ergänzend zu den transienten Verläufen, sind die Ergebnisse der stationären Lösungen beider Methoden (FVM und FEM) eingetragen.

	Case 1	Case 2	Case 3	Case 4
Relaxationsfaktor $\alpha$	0,8	0,8	0,8	0,8
Relaxationsfaktor $\alpha^{\text{Final}}$	1	0,8	1	0,8
Konvergenzgrenze $\epsilon_o$	$1e^{-6}$	$1e^{-6}$	$1e^{-6}$	$1e^{-6}$
Konvergenzmethode	Initial	Initial	Relative	Relative
Courant Zahl	0,5	0,5	0,5	0,5
Diskretisierungsschemen				
Term: $(\partial\rho\partial\mathbf{D})/\partial t^2$	Euler 1.Ordnung			
Term: $\nabla \otimes \mathbf{D}$	Gauss linear 2. Ordnung			
Term: $\nabla \bullet \boldsymbol{\sigma}$	Gauss linear 2. Ordnung			
Term: $\nabla \bullet [(2\mu + \lambda)\nabla \otimes \boldsymbol{\sigma}]$	Gauss linear orthogonal 2. Ordnung			

Tabelle B.1.: Numerisches Set-up der einzelnen Testfälle. Einstellungen sind für Feld- und Matrixrelaxation identisch.

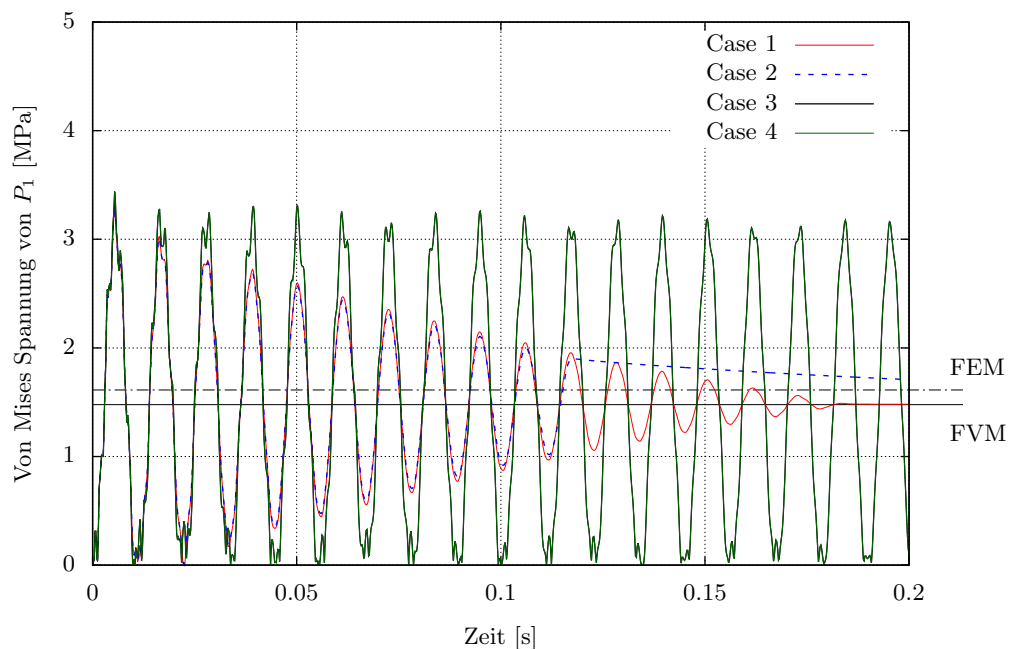


Abbildung B.3.: Einfluss der Feldrelaxation auf transiente Problemstellungen.

## Diskussion der Konvergenzmethode Initial und Relative

Werden die Berechnungen der zwei Konvergenzmethoden gegenübergestellt, sind klare Unterschiede der Ergebnisse erkennbar. Die Verwendung der *relative* Konvergenzmethode liefert für beide Fallbeispiele — Case 3 und Case 4 — identische Ergebnisse; die Kurven 3 und 4 liegen übereinander. Die Dämpfung der Amplitude aufgrund der ersten Zeitableitung ist für den zeitlich dargestellten Ausschnitt nicht erkennbar. Des Weiteren bewegt sich die Schwingung ausschließlich zwischen der minimalen (0 MPa) und maximalen (3,4 MPa) Auslenkung. Der höchste Amplitudenwert nimmt während des Zeitintervalls von 0,2 s nur marginal ab.

Die Dämpfung der Amplitude tritt bei der Verwendung der *initial* Methode wesentlich stärker ein und ist deutlich an Case 1 und Case 2 ersichtlich. In beiden Fällen werden die Informationen an den Extrempunkten stark geglättet. Dadurch wird die Schwingung im zeitlichen Verlauf gedämpft. Dies bedeutet, dass sich der transiente Charakter der Problemstellung dem stationären Zustand annähert; er kann durch das Pendeln um diesen bestätigt werden. Die Charakteristik beider Kurvenverläufe verhalten sich bis zum Zeitpunkt  $t \approx 0,118$  s ähnlich. Case 2 wird bis zu diesem Moment etwas intensiver gedämpft als Case 1 und verlässt den transienten Charakter nach  $t \approx 0,118$  s vollständig. Anschließend konvergiert dieser langsam asymptotisch gegen den stationären Endwert.

## Diskussion des finalen Relaxationsfaktors

Der Einfluss des *finalen* Relaxationsfaktors ist bereits vielfach in diversen Foren, wie beispielsweise auf CFD-Online (2018), kontrovers diskutiert worden, wobei stets die Unwissenheit in Bezug auf die Numerik und die angewendeten Algorithmen diese Diskussionen unstimmig gestalten. Aus Sicht der Numerik ist die Verwendung von Relaxationsparametern in der Feldrelaxation eine Reduzierung oder Limitierung der erhaltenen neuen Lösung; vergleiche Gleichung (B.7) in Anhang B.5.1. Anders ausgedrückt, die neu berechnete Lösung wird beschränkt. Deshalb ist zur korrekten Abbildung von transienten Problemstellungen der *finale* Relaxationsfaktor auf Eins zu setzen, da sonst Teile der Lösung weggeschnitten werden.

Dieses Phänomen äußert sich beispielsweise in Case 2. Hier werden aufgrund des Abschneidens von Informationen der transiente Verlauf ab  $t = 0,118$  s signifikant beeinflusst. Widersprüchlich ist allerdings die Fallpaarung Case 3 und Case 4, da hier mit Unterrelaxation des letzten Iterationsschrittes keine Ergebnisänderung eintritt. Dies ist jedoch nicht korrekt, da auch in diesen Fällen Informationen abgeschnitten werden. Die Ursache, weshalb die Unterrelaxation im finalen Iterationsschritt einen Einfluss im Fallbeispiel Case 2 hat, ist in der verwendeten Konvergenzmethode begründet.

Aufgrund des stark expliziten Charakters der Verschiebungsgleichung, wirkt die *relative* Konvergenzmethode wesentlich sensibler auf Änderungen der gesuchten Größe  $\mathbf{D}$ . Daher muss die Konvergenzgrenze, die für alle vier Beispiele auf  $\epsilon_o = 1e^{-6}$  gesetzt ist, bei den genannten Methoden unterschiedlich beurteilt werden. Die *relative* Methode kann eine betragsmäßig größere Toleranz verwenden als die *initial* Methode, ohne einen Informationsverlust zu zeigen. Dies ist in der Berechnung des Rest-Residuums beider Methoden begründet; vergleiche Anhang B.4. Die getroffene Aussage kann mithilfe der Berechnungszeiten der einzelnen Cases untermauert werden. Da bei der *relative* Methode das System wesentlich weiter — im Sinne des Residuen Verlaufes — konvergiert ist, müssen diese Berechnungen länger andauern. Die Berechnungszeit



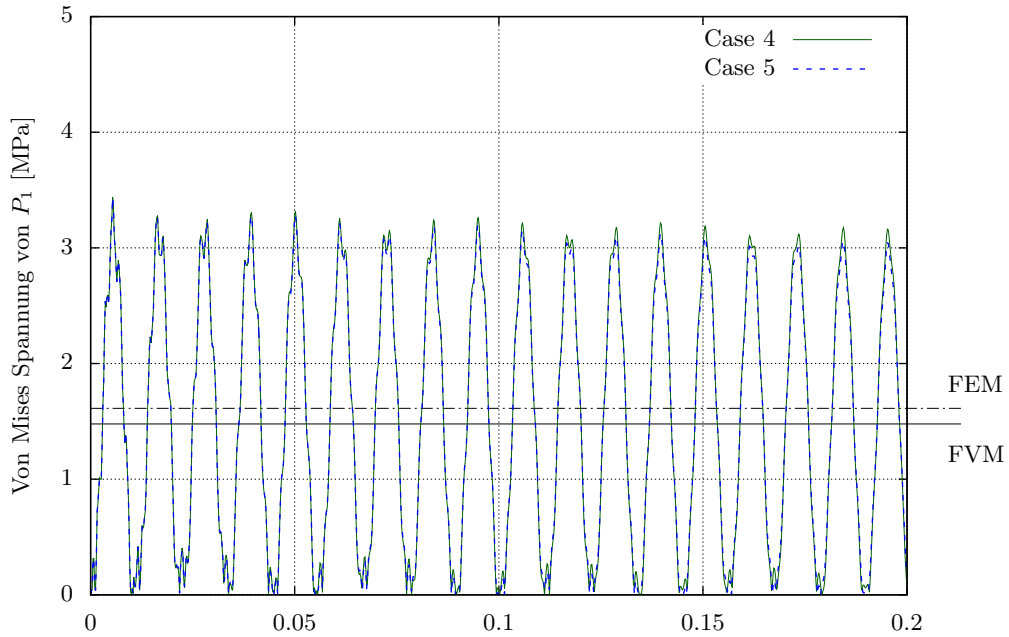


Abbildung B.4.: Feldrelaxation mit Unterrelaxation des letzten Iterationsschrittes bei unterschiedlichen Konvergenzgrenzen.

für Case 1, 2, 3 und 4 sind 16688 s, 13653 s, 65501 s und 65661 s. Es ist deutlich ersichtlich, dass die Berechnungszeit bei der Verwendung der *initial* Methode um ein Vielfaches geringer ist (Faktor  $\approx 3,8$ ). Des Weiteren ist der Einfluss des finalen Relaxationsfaktors bezogen auf die zeitliche Berechnung in Case 1 und Case 2 zu erkennen. Da das Lösen der letzten äußeren Schleife mit einem Relaxationsfaktor von Eins einen Mehraufwand für den linearen Löser darstellt (höhere Anzahl an inneren Iterationen), benötigt Case 1 circa 3000 s länger als Case 2. Der Mehraufwand zur Lösung der letzten äußeren Schleife mit einem Relaxationsfaktor von Eins gibt zudem Auskunft darüber, dass das System noch nicht weitgehend konvergiert ist. Dies kann durch den Vergleich mit Case 3 und Case 4 belegt werden, da hier kein signifikanter Mehraufwand bei der Verwendung eines finalen Relaxationsfaktors von Eins ersichtlich ist. Dies ist dadurch begründet, dass das Lösen des Matrixsystems in der letzten äußeren Schleife bereits ausreichend konvergiert ist, sodass das Lösen mit einem Relaxationsfaktor von Eins keinen zusätzlichen Mehraufwand darstellt. Daher kann in diesem Fall ohne Bedenken die letzte äußere Schleife mit einem Faktor kleiner Eins durchgeführt werden, ohne das große Abschneidefehler auftreten. Zur Bestätigung der oben genannten Aussagen ist in Abbildung B.4 eine Gegenüberstellung von Case 4 und Case 5 gegeben. Case 5 ist identisch zu 2, wird aber mit einer kleineren Konvergenzgrenze versehen;  $\epsilon_o = 1e^{-8}$ . Die Darstellung verdeutlicht die oben genannten Ausführungen und bestätigen diese. Zusätzlich untermauert die wesentlich längere Berechnungszeit von Case 5 mit 47345 s — bezogen auf Case 2 — die oben erläuterten Ausführungen weiter.

### B.6.2. Matrixrelaxation in transienten Berechnungen

Zur Darstellung des Einflusses der Matrixrelaxation werden, wie bei der Feldrelaxation, vier Testfälle diskutiert. Die Einstellungen die in Tabelle B.1 gegeben sind, werden beibehalten. Einziger Unterschied besteht in der Auswahl der Relaxationsmethode; die Relaxationsfaktoren werden auf die Matrix angewendet. Die quantitativen Verläufe sind in Abbildung B.5 zusam-

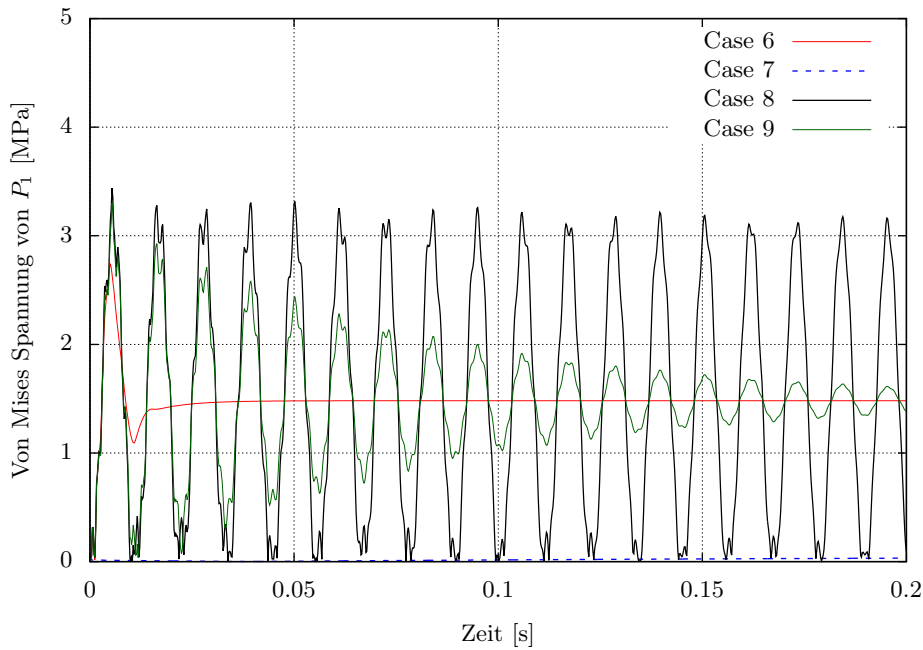


Abbildung B.5.: Einfluss der Matrixrelaxation auf transiente Problemstellungen.

mengetragen und werden sequentiell weiter nummeriert: Case 6, 7, 8 und 9.

Die Gegenüberstellung von Abbildung B.3 und B.5 zeigt auf, dass die Matrixrelaxation im Vergleich zur Feldrelaxation einen anderen Einfluss auf das zu lösende Gleichungssystem bewirkt und die Relaxationsfaktoren einen wesentlichen prägnanteren Einfluss auf das Ergebnis haben. Daher muss stets gründlich zwischen beiden Relaxationsmethoden unterschieden werden, da Fehleinstellungen die transienten Ergebnisse maßgeblich beeinflussen können.

### Diskussion der Konvergenzmethode Initial und Relative

Die Verwendung unterschiedlicher Konvergenzmethoden liefert, wie bei der Feldrelaxation, verschiedene Ergebnisse. Die Verwendung der *relative* Methode (Case 8 und Case 9) schwingt, wie bei der Feldrelaxation, mit der gleichen Frequenz um den stationären Endwert. Bei der Verwendung der *initial* Methode wird bereits nach dem Erreichen des ersten Maximalpunktes bei  $t \approx 0,008$  s die Schwingung so stark abgedämpft, dass die zweite Schwingung kaum eintritt und der transiente Charakter rapide abfällt. Bereits nach 0,025 s wird der stationäre Endwert erreicht. Case 7 liefert im gesamten Berechnungszeitraum keine vernünftigen Lösungen und verweilt nahezu unverändert an der Zeitachse.

Die einzelnen Verläufe sind auf die mathematische Behandlung der Matrix während der Relaxation zurückzuführen. Anders als bei der Feldrelaxation werden hier keine Werte abgeschnitten, sondern vielmehr die Diagonaldominanz der Matrix erhöht, wobei zugleich alle Manipulationen, die an der Matrix durchgeführt werden, im mathematischen Gleichgewicht sind. Das bedeutet, dass die Verbesserung der Diagonaldominanz der Matrix  $\mathbf{A}$  das gesamte Gleichungssystem verändert. Es folgt:  $\mathbf{A}^* = \mathcal{O}\mathbf{A}$ , wobei  $\mathcal{O}$  eine Operation auf die Diagonalelemente der Matrix darstellt und kommutativ ist. Zur Erhaltung der Konsistenz des Lösungssystems  $\mathbf{A}^*\mathbf{x} = \mathbf{b}$  muss dementsprechend die äquivalente Operation mit dem Quellterm  $\mathbf{b}$  durchgeführt werden:

$$\mathcal{O}\mathbf{A}\mathbf{x} = \mathcal{O}\mathbf{b} \longrightarrow \mathbf{A}^*\mathbf{x} = \mathbf{b}^* . \quad (\text{B.11})$$

Weitere Einzelheiten zu den während der Matrixrelaxation durchgeführten Operationen  $\mathcal{O}$  sind in Anhang B.5.2 auf Seite 161 gegeben.

### Diskussion des finalen Relaxationsfaktors

Der *finale* Relaxationsfaktor beeinflusst das Lösungssystem und den transienten Verlauf der Lösung signifikant und kann nicht mit der Feldrelaxation gleichgesetzt werden. Wird der Relaxationsfaktor bei der Verwendung der *relative* Konvergenzmethode kleiner als Eins gewählt (Case 9), wird das System gedämpft. Dies ist dadurch begründet, dass bei der Verwendung der Matrixrelaxation das lineare Gleichungssystem *verexpliziert* wird. Das bedeutet, dass die Verbesserung der Diagonaldominanz der Matrix  $\mathbf{A}$  mit einer Änderung des Lösungsvektor  $\mathbf{b}$  einhergeht; vergleiche Gleichung (B.11). Bei der Verwendung eines unterrelaxierten finalen Relaxationsfaktors  $\alpha_{R_{\text{Final}}} < 1$  bedeutet dies, dass der explizite Teil des Lösungssystems  $\mathbf{Ax} = \mathbf{b}$  verstärkt wird. Bildlich gesprochen werden Teile der Lösung in den Lösungsvektor  $\mathbf{b}$  geschoben. Dadurch ergibt sich bei der Verwendung eines Faktors kleiner Eins in der letzten äußeren Iteration, die mit dem finalen Relaxationsfaktor durchgeführt wird, stets eine Art nicht konvergiertes Lösungssystem aufgrund der *Verexplizierung*. Diesbezüglich wird der transiente Charakter der Lösung gedämpft; vergleiche Case 9 in Abbildung B.5. Der Verlauf der gedämpften Werte strebt gegen den stationären Endwert. Wird der *finale* Relaxationsfaktor hingegen auf Eins gesetzt, ist die letzte äußere Schleife vollständig konvergiert und der transiente Charakter bleibt erhalten.

Wird mit ein finaler Relaxationsfaktor kleiner Eins gewählt, muss das lineare Gleichungssystem bereits weit genug konvergiert sein. Um dies zu erreichen, muss die Konvergenzgrenze strenger gewählt werden. Dies wirkt sich jedoch wiederum nachteilig auf die Berechnungszeit aus.

Im Gegensatz dazu bewirkt eine Unterrelaxation des letzten Iterationsschrittes bei Anwendung der *initial* Methode ein Versagen des numerischen Algorithmus. Die erhaltenen Spannungen am

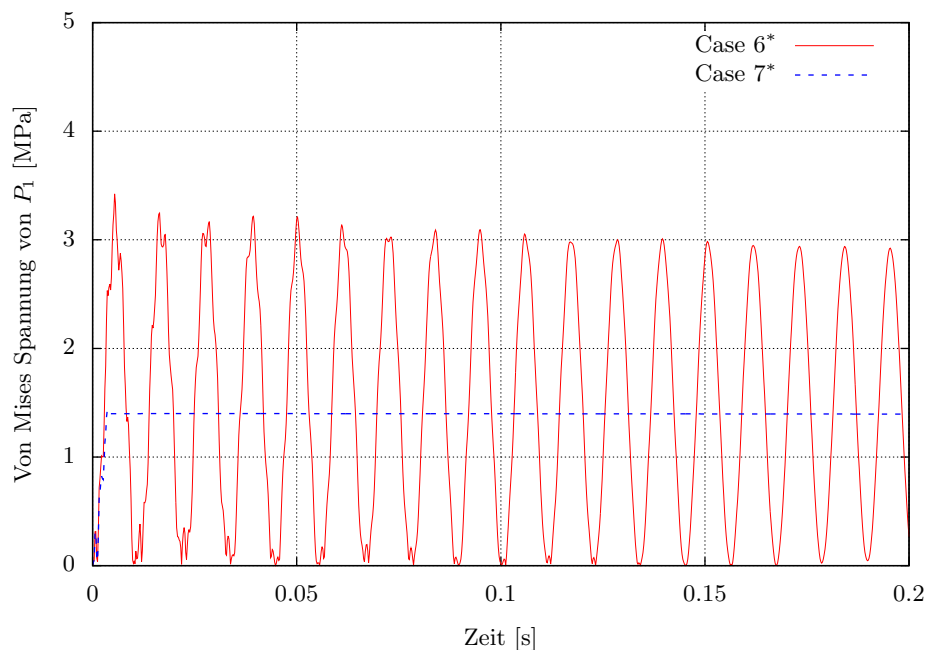


Abbildung B.6.: Wiederholung der Matrixrelaxationsberechnung für Case 6 und 7 mit einer verschärften Konvergenzgrenze von  $\epsilon_o = 1e^{-8}$ .

Punkt  $P_1$  ändern sich während der numerischen Berechnung kaum. Dies ist zum einen durch die Unterrelaxation des *finalen* Relaxationsfaktors der letzten äußeren Schleife — analog zur *relative* Methode — und zum anderen durch das zu hohe Toleranzkriterium  $\epsilon_o = 1e^{-6}$  der äußeren Schleife — analog zur Feldrelaxation — bedingt. Beide Umstände verstärken sich und liefern das dargestellte Ergebnis. Es wird vorweggenommen, dass es im Allgemeinen kein Versagen des Algorithmus darstellt.

Wird der *finale* Relaxationsfaktor auf Eins gesetzt und mit der *initial* Methode gearbeitet (Case 6), wird das transiente Problem kurzzeitig abgebildet. Nach  $t \approx 0,01$  s ist der instationäre Charakter jedoch abgeklungen und die stationäre Lösung liegt vor.

Die oben genannten Verhaltensmuster spiegeln sich erneut in den Berechnungszeiten wider. Die Berechnungszeiten für Case 6, 7, 8 und 9 sind 3176 s, 1598 s, 105659 s und 100754 s. Daraus können die gleichen Aussagen wie die bei der Feldrelaxation abgeleitet werden. Für die *initial* Methode ist das gesamte Lösungssystem nicht ausreichend konvergiert, das sich durch das Unterrelaxieren der letzten äußeren Schleife weiter verstärkt (Case 7). Zur teilweisen Untermauerung dieser Aussage werden Case 6 und Case 7 erneut mit einer strengeren Konvergenzgrenze von  $\epsilon_o = 1e^{-8}$  durchgeführt und subsequent als Case 6\* und 7\* notiert. In Abbildung B.6 ist deutlich zu erkennen, dass die Verringerung der Konvergenzgrenze — für die äußere Schleife — eine signifikante Verbesserung zur Abbildung des transienten Charakters für Case 6\* darstellt. Für den Fall das die äußere Schleife stets unterrelaxiert wird, kann festgehalten werden, dass kein transienter Charakter abgebildet wird. Allerdings erzielt Case 7, welcher zuvor keine plausiblen Werte lieferte, die stationäre Lösung, sofern die Simulation lange genug durchgeführt wird ( $t \gg 0,2$  s); in Abbildung B.6 ist der Endwert der Spannung von Case 7 unterhalb der stationären Endlösung.

### B.6.3. Feld- und Matrixrelaxation in stationären Berechnungen

Abschließend wird der Einfluss der Feld- und Matrixrelaxation in stationären Fallbeispielen analysiert. Die Gleichung für den Verschiebungsvektor (2.10) ändert damit ihren Charakter von hyperbolisch zu elliptisch.

Es werden drei Fallbeispiele betrachtet, die mit der Nummerierung Case 10, 11 und 12 auf dem Datenträger vorzufinden sind. Der erste Case wird klassisch ohne Relaxationsmethode berechnet. Die Konvergenzgrenze des linearen Lösers  $\epsilon_1$  wird auf  $1e^{-6}$  gesetzt und die *initial* Methode zum Voranschreiten in den nächsten Zeitschritt verwendet. Der zweite und dritte Case wird identisch zum ersten aufgesetzt, jedoch wird entweder die Feld- oder Matrixrelaxation mit einem Relaxationsfaktor von  $\alpha_R = 0,8$  angewendet.

Darstellung B.7 zeigt die quantitativen Ergebnisse während der stationären Berechnung. Des Weiteren ist in Abbildung B.8 das Konvergenzverhalten dargestellt.

Es ist deutlich ersichtlich, dass für stationäre Probleme die Matrixrelaxation kein plausibles Ergebnis liefert, obwohl die Initialresiduen bis auf  $1e^{-6}$  zurückgegangen sind; vergleiche Abbildung B.8. In dieser Grafik ist weiter die Residuenberechnung mithilfe der *relative* Methode dargestellt; diese zeigt keine erreichte Konvergenz an. Daraus folgt, dass die Matrixrelaxation für dieses Beispiel und den hier verwendeten Einstellungen falsche Ergebnisse liefert — eine Untersuchung mit schärferen Konvergenzgrenzen oder der *relative* Methode wird nicht betrachtet.

Die Feldrelaxation hingegen konvergiert zur stationären Lösung. Werden keine Relaxationsmethoden verwendet, wird das erwartete Ergebnis mit den wenigsten Iterationen erhalten. Die

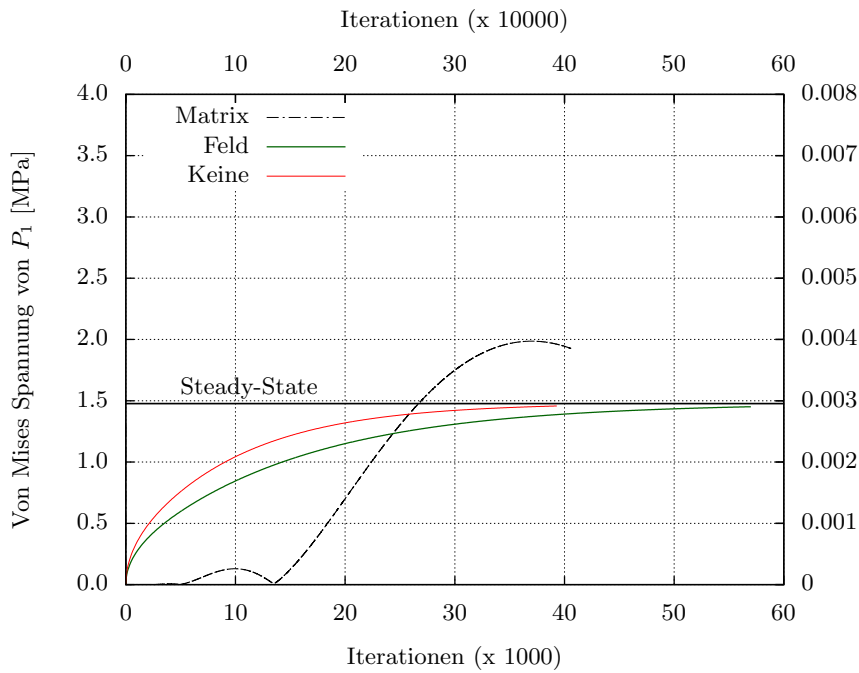


Abbildung B.7.: Einfluss der Feld- und Matrixrelaxation auf stationäre Problemstellungen.

Berechnungszeiten der einzelnen Testfälle — Case 10, 11 und 12 — sind 1063 s, 1488 s und 877 s. Hierbei zeigt vor allem die Matrixrelaxation eine fast um die Hälfte kürzere Berechnungszeit, obwohl über 350000 zusätzliche äußere Iterationen durchlaufen werden. Der Grund hierfür ist die *Verexplizitierung* des Problems: Die Matrix wird unterrelaxiert und Informationen in den Quellterm verschoben. Dadurch wird zwar das Lösen des linearen Gleichungssystems verbessert, jedoch wird das Voranschreiten der Lösung limitiert und kann durch die Residuenverläufe der *relative* Methode untermauert werden.

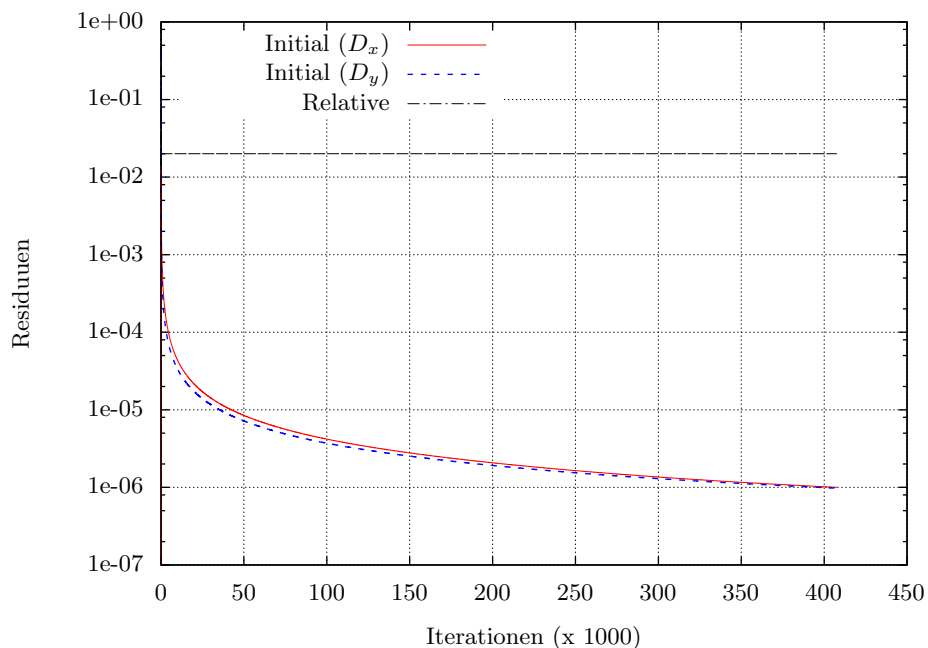


Abbildung B.8.: Vergleich der Residuen der *initial* und *relative* Methode unter Verwendung der Matrixrelaxation in stationären Problemstellungen.

## Anzahl der äußeren Iterationen

Im detaillierten Vergleich schneidet der Fall ohne Relaxation mit den wenigsten äußeren Iteration ab (Case 10). Hier werden 39234 äußere Iterationen ausgeführt, bis das System  $\mathbf{Ax} = \mathbf{b}$  konvergiert ist. Diesbezüglich wird eine Gesamtzeit von 17,7 Minuten benötigt.

Im Gegensatz dazu benötigt die Feldrelaxation weitaus mehr Iterationen (56910), genauso wie die Matrixrelaxation (407689). Bei der Feldrelaxation wird nach 24,8 Minuten der stationäre Zustand erreicht. Die Matrixrelaxation wird nach 14,6 Minuten abgebrochen.

Entsprechend kann festgehalten werden, dass die schnellste Variante zur Berechnung stationärer, strukturmechanischer Probleme mithilfe der Duhamel-Neumann Formulierung ohne numerische Relaxation durchgeführt werden kann. Es ist allerdings darauf hinzuweisen, dass die hier verwendete Geometrie in 2D vorliegt und die Problemstellung sehr einfach ist. Hinzu kommt die Verwendung optimaler numerischer Zellen. Für komplexere Geometrien kann eine Unterrelaxation sinnvoll sein, um das lineare Gleichungssystem besser zu konditionieren.

## B.7. Einfluss der inneren Schleife des linearen Löser auf die Konvergenzrate

Der Einfluss des Abbruchfehlers  $\epsilon^*$  der inneren Schleife wird nachfolgend mithilfe des Biegebalkenbeispiels untersucht. Alle Grundeinstellungen sind in Anhang B.6 gegeben, wobei keine Relaxation zum Einsatz kommt.

Das Matrixsystems  $\mathbf{Ax} = \mathbf{b}$  wird bis zu einer Konvergenzgrenze von  $\epsilon_0 = 1e^{-12}$  gelöst (äußere Schleife). Die innere Iterationsschleife wird mit drei verschiedenen relativen Residuen-Einstellungen ( $\kappa$ -Faktoren) untersucht und die Ergebnisse anschließend gegenübergestellt. Bezüglich der Nomenklatur wird auf Abbildung B.1 verwiesen.

Da im vorliegenden Fall das Gleichungssystem stark explizit ist, ist der Quellvektor  $\mathbf{b}$  von der letzten Iteration abhängig  $\rightarrow \mathbf{b} = f(\mathbf{x}^{n-1})$ . Veranschaulicht bedeutet dies:

$$\mathbf{Ax}^n = \mathbf{b}(\mathbf{x}^{n-1}) . \quad (\text{B.12})$$

Der Quellterm  $\mathbf{b}$  enthält damit alle expliziten Terme, die von der Lösung der vorherigen Iteration stammen. Dementsprechend ist es nicht ratsam das lineare Lösungssystem akkurat in der inneren Schleife zu berechnen, da der Quellterm wiederum von der neuen Lösung  $\mathbf{x}^n$  abhängig ist. Daher kann die innere Schleife inakkurat gelöst werden, wodurch die äußere Schleife schneller durchlaufen werden kann. Generell erhöht sich dadurch die Anzahl der äußeren Schleifendurchgänge, allerdings reduziert sich die Berechnungszeit der inneren Schleife.

Zur Untersuchung dieser Aussage werden drei Testfälle herangezogen, bei denen die innere Schleife mit einem relativen Residuen ( $\kappa$ -Faktor) von 0,01, 0,1 und 0,8 durchgeführt wird. Die Ergebnisse sind in Abbildung B.9 gegeben. Es ist deutlich ersichtlich, dass je *lockerer* (ungenauer) das Matrixsystem in der inneren Schleife gelöst wird, desto größer ist die benötigte Anzahl von äußeren Iterationen; vergleiche 0,8 zu 0,01 und 0,1. Die Faktoren 0,01 und 0,1 ergeben fast identische Iterationsdurchläufe. Daher wird als zweiter Anhaltspunkt die Berechnungszeit herangezogen. Diese liegt bei 554 s, 537 s und 451 s und bezieht sich auf die  $\kappa$ -Faktoren 0,01, 0,1 und 0,8. Wie ersichtlich ist, benötigt das Lösen mit einem Faktor 0,8 fast doppelt so viele äußere Iteration, ist aber, aufgrund der Tatsache das die innere Schleife schneller durchlaufen

wird, um fast ein Fünftel schneller.

Daher konvergiert der Löser beim Verwenden einer *lockeren* Konvergenzgrenze für den linearen Löser schneller und ist auf das stark explizite Gleichungssystem zurückzuführen.

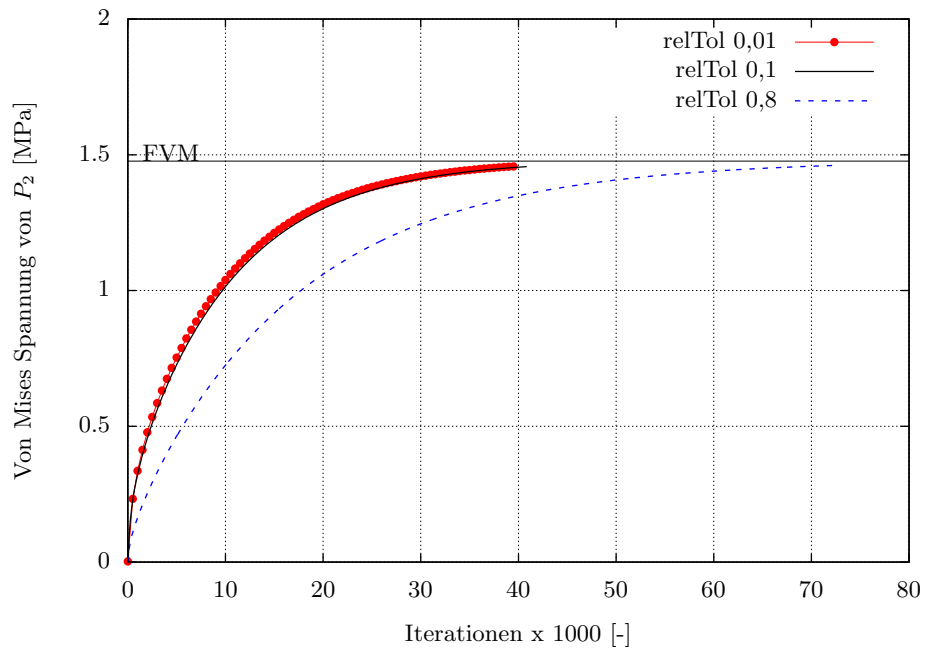


Abbildung B.9.: Einfluss des Abbruchfehlers  $\epsilon^*$  der inneren Schleife auf die Konvergenzrate des Löser. Die Konvergenzgrenze wird mittels dem  $\kappa$ -Faktor beeinflusst.





## C. Vektor- und Tensoralgebra

Das nachfolgende Kapitel enthält die wichtigsten Vektor- und Tensoroperationen, die in dieser Arbeit verwendet werden. Weitere Informationen zu diesem Themenbereich sind in Greenshields (2015) und Holzmann (2016a) zu finden.

Zu Beginn werden zwei Vektoren  $\mathbf{a}$  und  $\mathbf{b}$ , zwei Tensoren  $\mathbf{T}$  und  $\mathbf{S}$ , die Einheitsvektoren  $\mathbf{e}_1, \mathbf{e}_2$  und  $\mathbf{e}_3$ , die Identitätsmatrix (Einheitsmatrix)  $\mathbf{I}$  und der Nabla-Operator  $\nabla$  definiert:

$$\mathbf{a} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix},$$

$$\mathbf{T} = \begin{bmatrix} T_{xx} & T_{xy} & T_{xz} \\ T_{yx} & T_{yy} & T_{yz} \\ T_{zx} & T_{zy} & T_{zz} \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix},$$

$$\mathbf{S} = \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix},$$

$$\mathbf{e}_1 = \mathbf{e}_x = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{e}_2 = \mathbf{e}_y = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{e}_3 = \mathbf{e}_z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},$$

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \nabla = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix} = \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_3} \end{pmatrix}.$$

Mit diesen Definitionen werden nachfolgend die grundlegenden mathematischen Operationen veranschaulicht. Weitere Informationen können in den zwei oben genannten Referenzen eingeholt werden. Des Weiteren wird die Nomenklatur so gewählt, sodass diese mit OpenFOAM<sup>®</sup> bezogener Literatur korreliert. Des Weiteren steht die Größe  $\phi$  für eine willkürlich skalare Größe oder Quantität. Ferner werden nur 3D Räume ( $\mathcal{R}^3$ ) im Kartesischen Koordinatensystem betrachtet. Die Indizes  $i$  und  $j$  laufen somit von eins bis drei beziehungsweise von  $x$  bis  $z$ .

## C.1. Das innere Produkt

Das innere Produkt zweier Vektoren  $\mathbf{a}$  und  $\mathbf{b}$  ergibt eine skalare Größe  $\phi$ . Diese Operation ist kommutativ (vertauschbar) und wird als Skalarprodukt bezeichnet.

$$\phi = \mathbf{a} \bullet \mathbf{b} = \mathbf{a}^T \mathbf{b} = \sum_{i=1}^3 a_i b_i . \quad (\text{C.1})$$

Ferner existiert das innere Produkt zwischen einem Vektor  $\mathbf{a}$  und einem Tensor  $\mathbf{T}$ . Diese Operation ist kommutativ sofern der Tensor  $\mathbf{T}$  symmetrisch ist; Symmetrie bedeutet hier, dass  $T_{ij} = T_{ji}$  ist. Das Ergebnis dieser Operation ist ein Vektor  $\mathbf{b}$ .

$$\mathbf{b} = \mathbf{T} \bullet \mathbf{a} = \sum_{i=1}^3 \sum_{j=1}^3 T_{ij} a_j \mathbf{e}_i = \begin{pmatrix} T_{11}a_1 + T_{12}a_2 + T_{13}a_3 \\ T_{21}a_1 + T_{22}a_2 + T_{23}a_3 \\ T_{31}a_1 + T_{32}a_2 + T_{33}a_3 \end{pmatrix} , \quad (\text{C.2})$$

$$\mathbf{b} = \mathbf{a} \bullet \mathbf{T} = \sum_{i=1}^3 \sum_{j=1}^3 a_j T_{ji} \mathbf{e}_i = \begin{pmatrix} a_1 T_{11} + a_2 T_{21} + a_3 T_{31} \\ a_1 T_{12} + a_2 T_{22} + a_3 T_{32} \\ a_1 T_{13} + a_2 T_{23} + a_3 T_{33} \end{pmatrix} . \quad (\text{C.3})$$

Ist der Tensor  $\mathbf{T}$  symmetrisch folgt:  $\mathbf{b} = \mathbf{a} \bullet \mathbf{T} = \mathbf{T} \bullet \mathbf{a}$ .

Das doppelte innere Produkt (double inner product) wird auf zwei Tensoren  $\mathbf{T}$  und  $\mathbf{S}$  angewendet und ergibt eine skalare Größe  $\phi$ . Die Operation ist kommutativ und wird mit dem Operator  $:$  gekennzeichnet.

$$\begin{aligned} \phi = \mathbf{T} : \mathbf{S} &= \sum_{i=1}^3 \sum_{j=1}^3 T_{ij} S_{ij} = T_{11}S_{11} + T_{12}S_{12} + T_{13}S_{13} + T_{21}S_{21} \\ &+ T_{22}S_{22} + T_{23}S_{23} + T_{31}S_{31} + T_{32}S_{32} + T_{33}S_{33} . \end{aligned} \quad (\text{C.4})$$

## C.2. Das äußere Produkt

Unter dem äußeren Produkt — auch als dyadisches Produkt bekannt — wird die Operation zweier Vektoren  $\mathbf{a}$  und  $\mathbf{b}$  bezeichnet, die als Ergebnis einen Tensor  $\mathbf{T}$  liefert. Oftmals wird das dyadische Symbol  $\otimes$  vernachlässigt, das jedoch aus mathematischer Sicht nicht korrekt ist.

$$\mathbf{T} = \mathbf{a} \otimes \mathbf{b} = \mathbf{a} \mathbf{b}^T = \begin{bmatrix} a_x b_x & a_x b_y & a_x b_z \\ a_y b_x & a_y b_y & a_y b_z \\ a_z b_x & a_z b_y & a_z b_z \end{bmatrix} . \quad (\text{C.5})$$

### C.3. Gradientenoperator

Der Gradient einer skalaren Größe wird mithilfe des Nabla-Operators  $\nabla$  dargestellt. Das Ergebnis dieser Operation ist ein Vektor  $\mathbf{b}$ .

$$\mathbf{b} = \text{grad } \phi = \nabla \phi = \begin{pmatrix} \frac{\partial \phi}{\partial x} \\ \frac{\partial \phi}{\partial y} \\ \frac{\partial \phi}{\partial z} \end{pmatrix}. \quad (\text{C.6})$$

Ferner kann der Gradient einer Vektorgröße  $\mathbf{a}$  bestimmt werden. Die Nomenklatur in der Open-Source-Software OpenFOAM<sup>®</sup> wird mit dem Nabla-Operator  $\nabla$  und dem dyadischen Symbol  $\otimes$  gekennzeichnet. Das Resultat dieser Operation ist ein Tensor  $\mathbf{T}$ :

$$\mathbf{T} = \text{grad } \mathbf{a} = \nabla \otimes \mathbf{a} = \begin{bmatrix} \frac{\partial}{\partial x} a_x & \frac{\partial}{\partial x} a_y & \frac{\partial}{\partial x} a_z \\ \frac{\partial}{\partial y} a_x & \frac{\partial}{\partial y} a_y & \frac{\partial}{\partial y} a_z \\ \frac{\partial}{\partial z} a_x & \frac{\partial}{\partial z} a_y & \frac{\partial}{\partial z} a_z \end{bmatrix} \quad (\text{C.7})$$

Der Gradient erhöht den Rang der Größe stets um eins. Das heißt, ein Skalar wird zu einem Vektor, ein Vektor zu einem Tensor und so weiter. Die Bestimmung des Gradienten eines Tensors  $\mathbf{T}$  wird in dieser Arbeit nicht benötigt, wodurch hier nicht weiter darauf eingegangen wird.

### C.4. Divergenzoperator

Der Divergenzoperator kann auf Vektoren und Tensoren angewendet werden und verringert den Rang der Größe stets um eins. In OpenFOAM<sup>®</sup> wird der Divergenzoperator mit dem Nabla-Operator  $\nabla$  und dem Punkt  $\bullet$  dargestellt.

Die Divergenz eines Vektors  $\mathbf{a}$  ergibt eine skalare Größe  $\phi$ :

$$\phi = \text{div } \mathbf{a} = \nabla \bullet \mathbf{a} = \sum_{i=1}^3 \frac{\partial}{\partial x_i} a_i = \frac{\partial a_1}{\partial x_1} + \frac{\partial a_2}{\partial x_2} + \frac{\partial a_3}{\partial x_3}. \quad (\text{C.8})$$

Im Sinne der FVM ist die Divergenz eines Vektorterms (Konvektions- oder Advektionsterm), die Summe der Flüsse durch die einzelnen Oberflächen eines jeden Kontrollvolumens.

Die Divergenz eines Tensors  $\mathbf{T}$  ergibt einen Vektor  $\mathbf{b}$ :

$$\mathbf{b} = \text{div } \mathbf{T} = \nabla \bullet \mathbf{T} = \sum_{i=1}^3 \sum_{j=1}^3 \frac{\partial}{\partial x_j} T_{ji} \mathbf{e}_i = \begin{bmatrix} \frac{\partial T_{11}}{\partial x_1} + \frac{\partial T_{21}}{\partial x_2} + \frac{\partial T_{31}}{\partial x_3} \\ \frac{\partial T_{12}}{\partial x_1} + \frac{\partial T_{22}}{\partial x_2} + \frac{\partial T_{32}}{\partial x_3} \\ \frac{\partial T_{13}}{\partial x_1} + \frac{\partial T_{23}}{\partial x_2} + \frac{\partial T_{33}}{\partial x_3} \end{bmatrix}. \quad (\text{C.9})$$

### C.5. Produktregel mit dem Divergenzoperator

In der numerischen Mathematik können Terme, welche den Divergenzoperator enthalten, mithilfe der Produktregel aufgeteilt werden. Wird der Divergenzoperator auf das Produkt eines Vektors  $\mathbf{a}$

und eines Skalars  $\phi$  angewendet, kann dieser Term wie folgt gesplittet werden:

$$\nabla \bullet (\mathbf{a}\phi) = \mathbf{a} \bullet \nabla \phi + \phi \nabla \bullet \mathbf{a} . \quad (\text{C.10})$$

Wird der Divergenzoperator auf das äußere Produkt zweier Vektoren  $\mathbf{a}$  und  $\mathbf{b}$  angewendet, lautet die Produktregel:

$$\nabla \bullet (\mathbf{a} \otimes \mathbf{b}) = \mathbf{a} \bullet \nabla \otimes \mathbf{b} + \mathbf{b} \nabla \bullet \mathbf{a} . \quad (\text{C.11})$$

Der Vollständigkeit wird die Produktregel für den Divergenzoperator auf ein inneres Produkt eines Tensors  $\mathbf{T}$  und Vektor  $\mathbf{a}$  gegeben:

$$\nabla \bullet (\mathbf{T} \bullet \mathbf{a}) = \mathbf{T} : \nabla \otimes \mathbf{a} + \mathbf{a} \bullet \nabla \bullet \mathbf{T} . \quad (\text{C.12})$$

## C.6. Spuroperator (Trace-Operator)

Der Spuroperator wird mit  $\text{tr}()$  gekennzeichnet und stellt die Summe der Diagonalelemente einer Matrix dar. Das Ergebnis ist eine skalare Größe. Bezogen auf den Tensor  $\mathbf{T}$  folgt:

$$\phi = \text{tr}(\mathbf{T}) = \sum_{i=1}^3 T_{ii} = T_{11} + T_{22} + T_{33} . \quad (\text{C.13})$$

## C.7. Zusammenhang zwischen Spur- und Divergenzoperator

In der numerischen Modellierung von Spannungen in einem Festkörper, kann der Zusammenhang zwischen dem Verschiebungsvektor  $\mathbf{D}$  und dem Verzerrungstensor  $\mathbf{D}^S$  verwendet werden, um folgende Formulierung zu erhalten:

$$\nabla \bullet \mathbf{D} = \text{tr}(\mathbf{D}^S) . \quad (\text{C.14})$$

Die Definition von  $\mathbf{D}^S$  ist in Gleichung (2.6) auf Seite 12 gegeben.

## C.8. Substantielle Ableitung (materielle Ableitung)

Mithilfe der substantiellen Ableitung — auch bekannt unter der materiellen Ableitung — können Erhaltungsgleichungen umgeschrieben werden. Die Formulierung ist wie folgt:

$$\frac{D\phi}{Dt} = \frac{\partial \phi}{\partial t} + \mathbf{U} \bullet \nabla \phi . \quad (\text{C.15})$$

Die Größe  $\mathbf{U}$  stellt den Geschwindigkeitsvektor und  $\phi$  eine beliebige Größe dar. Generell wird die substantielle Form für die Lagrange Betrachtungsweise verwendet. Jede konservative Erhaltungsgleichung — Euler Betrachtung — kann mithilfe der substantiellen Ableitung in eine nicht-konservative Erhaltungsgleichung überführt werden (Lagrange Betrachtung).

## C.9. Deviatorischer und hydrostatischer Anteil einer Matrix

Jeder quadratische Matrix  $\mathbf{A}$  kann in einen deviatorischen  $\mathbf{A}^{\text{dev}}$  und hydrostatischen  $\mathbf{A}^{\text{hyd}}$  Anteil aufgeteilt werden. Als quadratische Matrix wird jene Matrix bezeichnet, die eine identische Anzahl von Zeilen und Spalten aufweist; im Folgenden wird erneut die generellen Matrizen aus der FVM herangezogen welche im Allgemeinen eine  $3 \times 3$  Matrix repräsentieren. Daraus folgt:

$$\mathbf{A} = \mathbf{A}^{\text{dev}} + \mathbf{A}^{\text{hyd}} . \quad (\text{C.16})$$

Der hydrostatische Anteil  $\mathbf{A}^{\text{hyd}}$  ist definiert als:

$$\mathbf{A}^{\text{hyd}} = \frac{1}{3} \text{tr}(\mathbf{A})\mathbf{I} = \frac{1}{3} \sum_{i=1}^3 A_{ii} \mathbf{e}_i . \quad (\text{C.17})$$

Mithilfe der Definition des hydrostatischen Anteils ist es möglich, die deviatorische Matrix  $\mathbf{A}^{\text{dev}}$  zu bestimmen:

$$\mathbf{A}^{\text{dev}} = \mathbf{A} - \mathbf{A}^{\text{hyd}} = \mathbf{A} - \frac{1}{3} \text{tr}(\mathbf{A})\mathbf{I} . \quad (\text{C.18})$$

Es ist erwähnenswert, dass die Spur der deviatorischen Matrix  $\mathbf{A}^{\text{dev}}$  gleich null ist.



# D. Verfahren zur automatischen Netzerzeugung

Die Diskretisierung des Kontinuums ist eine der wichtigsten Aufgaben der numerischen Simulation. Durch die Qualität der Volumina — das bedeutet die Abweichung der Topologie zum Ideal (Würfel in 3D oder Rechteck in 2D) — wird die Numerik maßgeblich beeinflusst, weshalb nachfolgend ein Überblick über verschiedene Netztypen und deren Generierung gegeben wird. Am Ende wird aufgezeigt, welche numerischen Schwierigkeiten auftreten, wenn die Form der Volumina (Topologie) beispielsweise durch Verzerrung abgeändert wird.

## D.1. Das Lösungsgebiet

Zur numerischen Berechnung jeder Problemstellung bedarf es der Notwendigkeit das Lösungsgebiet, welches das interessierende Kontinuum darstellt, in einen diskreten Volumenkörper zu unterteilen. In der FVM können beliebige Volumina eingesetzt werden, wodurch sich die Möglichkeit ergibt, komplexe Geometrien akkurat abzubilden. Grund hierfür sind die Integralgleichungen, die eine Bilanzierung für jedes Volumen darstellt — somit ist die Topologie im Sinne der räumlichen Diskretisierung nicht relevant. Als Gegenbeispiel wird kurz die Methode der Finiten Differenzen herangezogen. Aufgrund der mathematischen Gegebenheit ist es nur möglich, Quader oder dessen Sonderform — den Würfel —, als Volumenzellen zu verwenden. Komplexe Konturen können diesbezüglich nur mit Stufenprofilen abgebildet werden.

Das diskretisierte Lösungsgebiet wird als numerisches Netz bezeichnet und kann, wie bereits erwähnt, in der FVM aus unterschiedlichen Volumina bestehen, die untereinander vermischt werden können. Die bekanntesten Netztypen sind reine Hexaeder, Tetraeder, Polygon und hexadominante Netze. Letzteres kombiniert die numerischen Vorteile von Hexaedern mit anderen Volumentypen zur Abbildung komplexer Geometrien. Im Allgemeinen werden bei hexadominanten Netzen im Inneren des Kontinuums Hexaedervolumina und zur akkuraten Auflösung der Ränder eine Mischung aus anderen verwendet.

## D.2. Volumentypen

In der numerischen Simulation mittels der FVM sind drei Haupttypen von Volumina vertreten, die in Abbildung D.1 dargestellt sind. Jedes Volumen hat dabei spezielle Eigenschaften, auf die nachfolgend kurz eingegangen wird.

Im Bereich der wissenschaftlichen Analysen werden am häufigsten die Hexaedervolumen verwendet, da diese die besten numerischen Eigenschaften besitzen. Ein Beweis hierzu wird nicht gegeben, ist aber in einer Vielzahl von Grundlagenbüchern zu finden; vergleiche Ferziger und Perić (2008). Des Weiteren sind die zu untersuchenden Phänomene oft mit einer 2D Modellie-

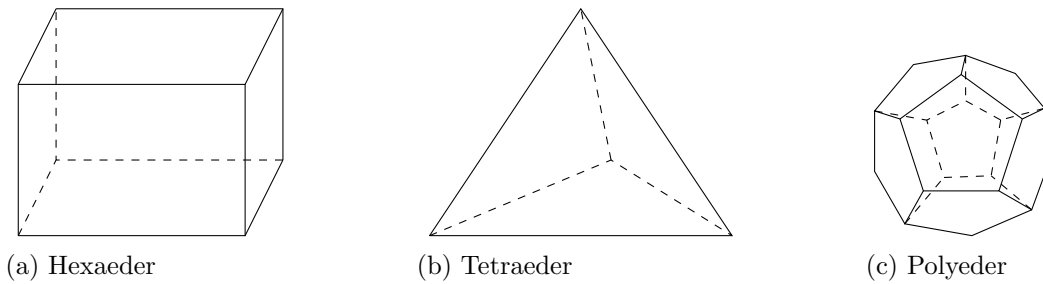


Abbildung D.1.: Darstellung der am meist verwendeten Volumina in der FVM. Das Polyeder stellt den allgemeinsten Körper dar.

rung abbildbar und weisen das Merkmal der Orthogonalität auf. Daher ist die Vernetzung einer solchen Geometrie mithilfe von Hexaedern einfach durchführbar.

Das Vernetzen von komplexen Geometrien kann mittels Hexaedervolumen nicht automatisch durchgeführt werden, wodurch andere Zelltypen interessant werden. Beispielsweise können mittels Tetraeder oder Polyeder automatische Algorithmen entwickelt werden, die ein komplexes Design vollautomatisch diskretisieren. Auf die Generierung wird später gesondert eingegangen, allerdings soll erwähnt werden, dass die Abwägung zwischen automatischer Netzerstellung und gegebenenfalls einer späteren numerischen Problematik gegen die Berechnungsgenauigkeit abgewogen werden muss.

### D.3. Generierung der Netztypen

Wie bereits erwähnt, werden in der numerischen Simulation verschiedene Netztypen verwendet. Die Erstellung dieser Netze wird nachfolgend kurz diskutiert. Thompson et al. (1998) und Jasak (1996) differenzieren Netze durch zwei Merkmale. Diese sind entweder strukturiert oder nicht-strukturiert. Erstere zeichnen sich dadurch aus, dass strukturell gesehen sich periodisch einfache Volumina wiederholen, wie beispielsweise Würfelzellen. Daraus ergibt sich eine reguläre Topologie und die Eigenschaft, dass Punkte eine definierte Anordnung im Raum besitzen. Nicht-strukturierte Netze weisen hingegen eine willkürliche Anordnung der Netzelemente auf. Eine sich periodisch wiederholende Struktur ist nicht festzustellen.

Ein Beispiel für ein nicht-strukturiertes Netz ist in Abbildung D.3 gegeben. Wie zu erkennen ist, sind keine Strukturen der Trajektorien erkennbar; strukturierte Netze weisen dieses Merkmal auf (Ferziger und Perić (2008)).

Ein programmiertechnisches Merkmal beider Varianten stellt die Speicheranforderung dar. Für strukturierte Netze kann der Programmieraufwand wesentlich kleiner gehalten werden, da Nachbarzellen einfach bestimmbar sind. Bei nicht-strukturierten Gittern müssen zusätzlich Informationen zum Nachbar gespeichert werden. Die Suche nach diesen ist mit einem Mehraufwand verknüpft. Im heutigen digitalen Zeitalter ist dieser Unterschied jedoch nicht mehr diskussionsrelevant und wird daher nur erwähnt.

#### D.3.1. Hexaedernetze

Bezüglich der numerischen Diskretisierung der Erhaltungsgleichungen und den daraus resultierenden Diskretisierungsfehlern sind reine Hexaedernetze — numerisch betrachtet — am besten. Dies liegt in der Natur der numerischen Behandlung der einzelnen Terme.



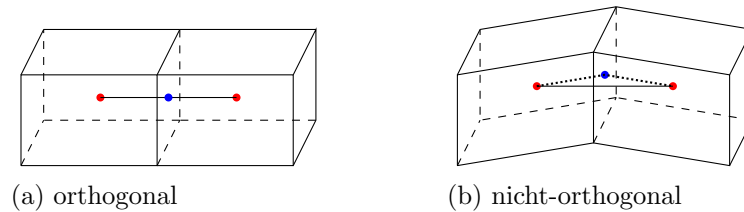


Abbildung D.2.: Interkonnektivität der Zellmittelpunkte und der dazwischen liegenden Fläche zweier Hexaeder Zellen für orthogonale und nicht-orthogonale Konfiguration.

Allgemein sind Hexaedernetze orthogonal; bestehen also aus Quader oder Würfel. Dies bedeutet, dass die Verbindungslinie zweier benachbarter Volumenzentren durch den Mittelpunkt der geteilten Fläche geht und daher senkrecht zu dieser steht; vergleiche Abbildung D.2a. Diese Konfiguration ist aus Sicht der Numerik optimal, da Berechnungsverfahren, die die Werte zwischen zwei Volumenzellen auf die Fläche interpolieren müssen, keinen Diskretisierungsfehler beinhalten und damit der Genauigkeit 2. Ordnung entsprechen. Im Gegensatz dazu bedarf es einer numerischen Korrektur, wenn strukturierte Netze verzerrt sind. Abbildung D.2b zeigt ein solches Szenario, bei dem die Verbindungslinie zwischen den beiden Zellmittelpunkten nicht durch den Flächenmittelpunkt geht. Diese Charakteristik wird als nicht-Orthogonalität bezeichnet und verschlechtert die Konvergenzrate des Lösungsalgorithmus, sofern keine numerische Behandlung (Korrektur) erfolgt.

Angemerkt werden soll, dass alle Netztypen nicht-Orthogonalitäten oder Verzerrungen (skewness) aufweisen können und diese numerisch behandelt werden müssen, wenn Grenzwerte — im Sinne der Netzqualität — überschritten werden. Jasak (1996) gibt hierzu eine detaillierte Diskussion in seiner Arbeit an.

Die Erzeugung eines Hexaedernetzes kann mit der in OpenFOAM<sup>®</sup> enthaltenen Applikation `blockMesh` erzeugt werden.

### D.3.2. Tetraedernetze

Das Tetraedernetz ist eines der verbreitetsten Gitter in der numerischen Simulation. Vor allem in der FEM werden diese Netze oft verwendet. Auch für die FVM können diese Netze verwendet werden, jedoch müssen hierbei einige numerische Aspekte intensiver beachtet werden, wie beispielsweise die Verzerrung oder Nicht-Orthogonalität.

Bei der Generierung können unterschiedliche Methoden verwendet werden. Eine der bekanntesten ist das *Advancing Front Meshing* Verfahren. Diese Methode erzeugt zu Beginn ein Oberflächennetz und generiert, basierend darauf, die ersten Tetraeder Volumina im Inneren des Körpers. Dazu werden neue Punkte definiert, die mit der Oberflächeneinteilung Tetraeder ergeben. Die Generierung ins Innere des Körpers kann gedanklich als Front, die sich ins Innere des Kontinuums bewegt, beschrieben werden. Eine überarbeitete Version des Verfahrens wird von George und Seveno (1994) vorgestellt. Der Algorithmus *NETGEN*, der dieses Verfahren verwendet, wird von Schöberl (1997) diskutiert.

Dieser Netztyp zeichnet sich vor allem dadurch aus, dass die Gittererzeugung einfach automatisierbar ist und komplexe Geometrien ohne großen Aufwand diskretisieren werden können. Aus numerischer Sicht behält sich dieser Netztyp jedoch einige Schwachstellen vor. Durch die nicht-Orthogonalität und Schrägheit (skewness) werden numerische Fehler angefacht, die vor allem bei Standard Diskretisierungsmethoden (bezogen auf die numerischen Terme) Proble-

me verursachen; vergleiche Holzmann (2013). Es ist anzumerken, dass hauptsächlich die nicht-Orthogonalität ein der wichtigsten Netzqualitätsgrößen darstellt und ist in der Numerik begründet.

Dieser Netztyp kann beispielsweise mithilfe des CAD Softwareprogramms Salome<sup>®</sup>, das in CASCADE (2017) zu finden ist, erstellt werden.

### D.3.3. Polygonnetze

Ein Polygonnetz kann mithilfe der *Voronoi* Methode erstellt werden. Dabei wird ein vorhandenes Tetradernetz mit einer einfachen Methodik, die von Berger et al. (2008) vorgestellt wird, umgewandelt. Das Verfahren transformiert die vorhandenen Volumenzentren der Tetraeder zu Eckpunkte des Polyedernetzes und erstellt aufgrund dieser Information neue Volumina. In Abbildung D.3 ist das Verfahren für ein 2D Netz veranschaulicht dargestellt. Die Erstellung eines solchen Netzes kann mit OpenFOAM<sup>®</sup> über das Programm `polyDualMesh` erfolgen.

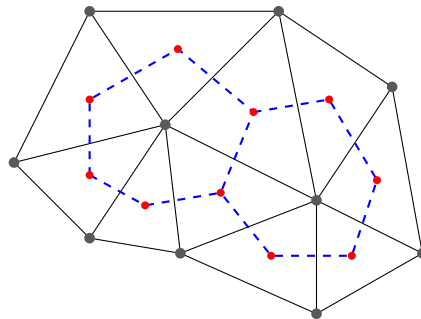


Abbildung D.3.: Veranschaulichung der Transformation von Tetraeder (—) in Polygone (---) in 2D. Die grauen Punkte (•) repräsentieren die Eckpunkte der Tetraeder und die roten (•) die Tetraederzentren. Die Typen (Eckpunkte oder Volumenzentrum) werden bei der Voronoi Methode vertauscht, wodurch Polygone erhalten werden.

### D.3.4. Hybride Netze

Netze, die aus einer Mischung aus strukturierten und nicht-strukturierten Volumenzellen darstellen, sind unter dem Namen der Hybride bekannt. Damit ist es möglich, die Vorteile der einzelnen Netztypen zu kombinieren. Hybridnetze können unterschiedlich aufgebaut sein. Eine Liste verschiedener Netzkombination ist in Ferziger und Perić (2008) gegeben. Ein beliebtes Beispiel ist die Kombination aus strukturierten und nicht-strukturierten Netzen bei Umströmungsproblemen. Hier wird das Profil beziehungsweise die Grenzschicht mithilfe eines strukturierten, das Farfield hingegen unter Verwendung eines unstrukturierten Netzes abgebildet.

Mithilfe des OpenFOAM<sup>®</sup> Programms `snappyHexMesh` können hexadominante Netze vollautomatisch erstellt werden. Diese können als Kombination beider (strukturierter und nicht-strukturierter Zellen) betrachtet werden. Die Diskretisierung einer Geometrie erfolgt dabei taktisch intelligent und wird wie folgt durchgeführt; zur Verdeutlichung wird das Vernetzen eines Kreises diskutiert.

Zu Beginn wird ein Hexaedernetz, das die zu vernetzende Geometrie umschließt, erstellt. Dieses Netz wird als *Hintergrundnetz* bezeichnet. Zur optimalen Nutzung des Algorithmus sollte das Hintergrundnetz äquidistante Gitterabstände aufweisen (Würfel). Ausgehend von einem

Oberflächennetz, das den zu vernetzenden Körper darstellt und meistens als *stereolithografische* (STL) Datei vorliegt, wird die Diskretisierung des Kontinuums gestartet. Abbildung D.4a zeigt das Hintergrundnetz und den zu vernetzenden Kreis.

Zu Beginn werden alle Zellen des Hintergrundnetzes gesucht, die die Oberfläche (STL) der Geometrie schneiden. Diese Zellen werden über eine sogenannte *Engine* — ein C++ Programm — markiert und verfeinert; sofern dies gewünscht ist. Dies bedeutet, dass ein Volumenwürfel, der die Oberfläche schneidet, in acht kleinere unterteilt wird. Die Aufteilung erfolgt im Schwerpunkt. Der Benutzer hat die Möglichkeit anzugeben, wie oft diese Teilung stattfindet. In Abbildung D.4b ist eine maximale Verfeinerung um zwei Grade dargestellt; erster Grad ist in Blau und der zweite Grad in Rot hervorgehoben.

Nachdem dieser Prozess beendet ist, werden alle Volumina gelöscht, die nicht zur Geometrie gehören; vergleiche Abbildung D.4c. Dieses Netz stellt die Oberflächen des Kontinuums gestuft dar. Anschließend wird versucht, unter Berücksichtigung diverser Netz-Qualitätsparameter, die Eckpunkte der äußersten Volumina auf die Oberfläche zu bewegen. Der Verschiebungsvektor von jedem Punkt wird anschließend dazu verwendet, das innen liegende Netz mittels einer Laplace-Gleichung zu verschieben, um ein einheitlicheres Netz zu erzeugen. Dieser Schritt ist nötig, da das

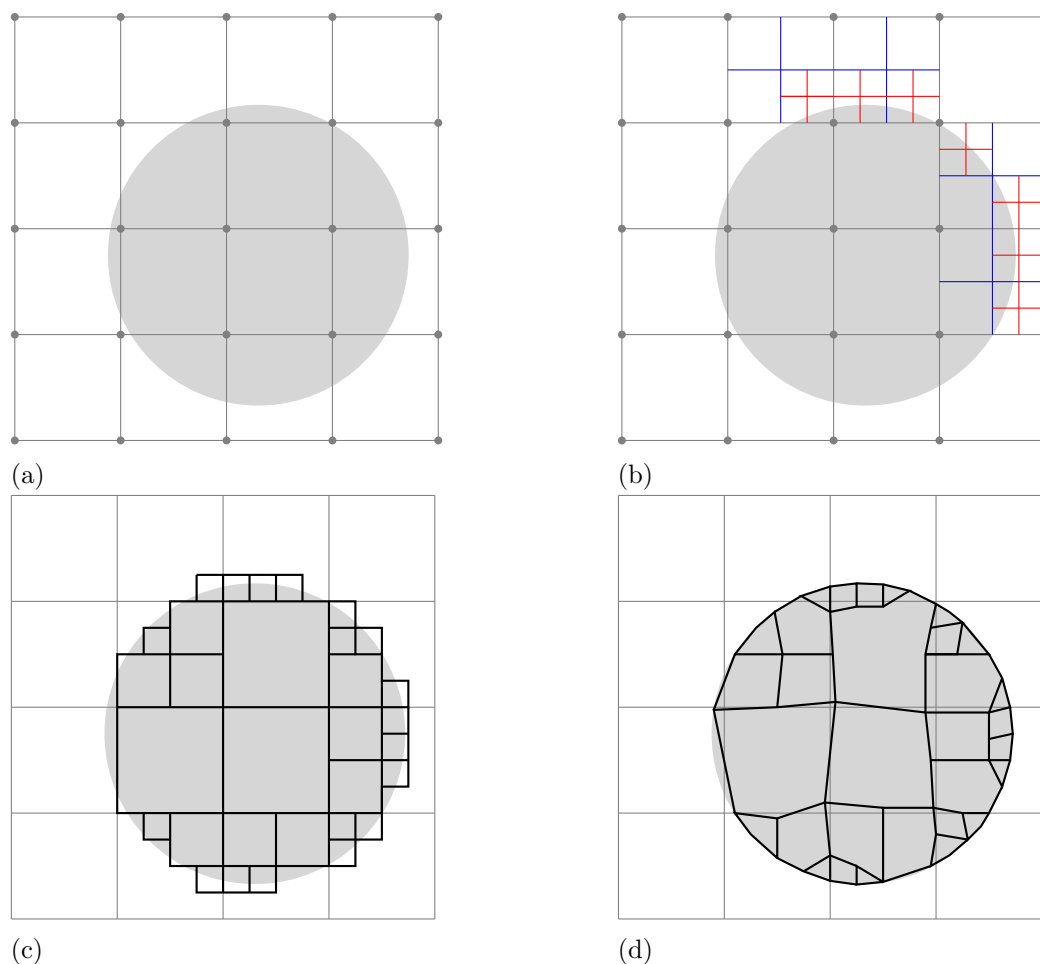


Abbildung D.4.: Generierung eines willkürlichen hexadominanten Gitters. (a) Hintergrundnetz und zu vernetzende Geometrie. (b) Zellverfeinerung der Volumina, welche die Oberfläche der Geometrie schneiden. (c) Wegschneiden der nicht relevanten Volumina. (d) Verschiebung der Außenpunkte zur akkuraten Oberflächenabbildung.

Verschieben der Eckpunkte auf die Oberfläche die Zellen stark verzerren kann. Zudem entscheidet der Algorithmus selbständig über die Zusammenlegung einzelner Punkte und entsprechend über den Typ der Zelle. Es entstehen dadurch Tetraeder, Prismen, Pyramiden, Polygone oder andere Volumina. Darstellung D.4d zeigt diesen Schritt schematisch auf.

Es soll angemerkt werden, dass der Benutzer eine Vielzahl an Einstellungsmöglichkeiten für die automatische Vernetzung abändern kann, um ein optimales Ergebnis zu erzielen. Zudem dient das hier präsentierte Beispiel der Veranschaulichung; weitere Informationen zu diesem Vernetzungsalgorithmus befinden sich in Greenshields (2017).

# E. Ergänzendes zur Materialberechnung

Nachfolgende Kapitel enthalten ausführliche Diskussionen über die Ermittlung der Phasenverteilung der Silizium- und  $\beta$ -Phase, welche nach der Erstarrungsberechnung bekannt sein müssen. Ferner wird eine detaillierte Erläuterung zur Bestimmung der temperaturabhängigen Transformationsradien getätigt, die in der kinetischen Modellierung benötigt werden.

## E.1. Bestimmung der Phasenverteilung

In der Kinetikberechnung werden die Ausscheidungsverteilungen der Phasen, die während der Scheil-Gulliver Berechnung erhalten werden, benötigt. Die relevanten Phasen sind das Silizium im Eutektikum und die  $\beta$ -Phase. Die Bestimmung dieser Verteilungen wird anschließend diskutiert.

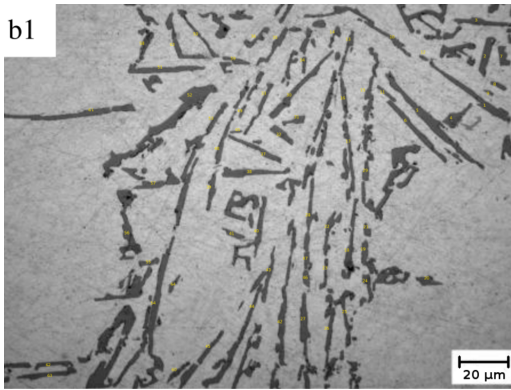
### E.1.1. Bestimmung der Siliziumverteilung

Zur Bestimmung der Siliziumverteilung im Eutektikum werden — wie bereits im Hauptteil dieser Arbeit erwähnt — die Arbeiten von Jakesch (2015) und Hajkowski et al. (2012) herangezogen. Die in diesen Arbeiten enthaltenen Gefügebildungen sind in Abbildung E.1 gegeben und werden für die weitere Auswertung herangezogen. Ersichtlich ist das nadelförmige Silizium im Eutektikum (schwarze Regionen). Zur Bestimmung der Radiusverteilung für die Siliziumphase wird diesbezüglich jede einzelne Nadel nummeriert und vermessen. Dabei werden die Längen und Breiten der Nadeln in Pixel durch die freie Software GIMP<sup>®</sup> bestimmt. Anschließend werden die erhaltenen Daten mithilfe des Maßstabs in SI-Einheiten umgerechnet. Die detaillierte Vermessung beider Eutektikas ist in Tabelle E.1 und E.2 gegeben. Es soll angemerkt werden, dass die gemessene Länge stets den betragsmäßig größeren Wert darstellt und beide Autoren ähnliche Abgussbedingungen und gleiche Legierungen verwendet haben.

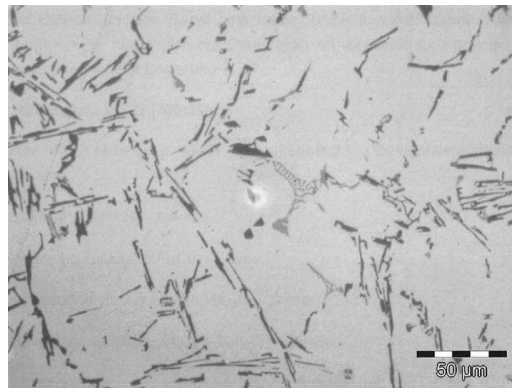
Nachdem die Länge und Breite jeder Nadel vorliegt, wird das Volumen bestimmt. Dies wird mit zwei verschiedene Annahmen durchgeführt. Die erste beruht darauf, dass sich die Siliziumphase nadelförmig ausbildet. Entsprechend wird zur Berechnung des Volumens die Formulierung für einen Zylinder verwendet, wobei die gemessene Länge die Höhe und die gemessene Breite den Durchmesser des Zylinders repräsentiert.

Aufgrund der Tatsache, dass die Schlifffbilder nur einen 2D Einblick gewähren, kann zudem angenommen werden, dass sich die Siliziumphase im dreidimensionalen Raum als Platte abbildet. Diesbezüglich wird in der zweiten Betrachtung eine Plattencharakteristik des Siliziums angenommen. Die Berechnung des Volumens lautet:  $\text{Volumen} = \text{gemessene Länge} \cdot \text{gemessene Länge} \cdot \text{gemessene Breite}$ .

Abschließend wird ein äquivalenter Kugelradius mit den berechneten Volumina bestimmt. Da MatCalc eine Größenverteilung benötigt und diese als Wahrscheinlichkeitsfunktion angegeben werden muss, ist es erforderlich, die berechneten Äquivalentradien in ein Histogramm



(a) Hajkowski et al. (2012)



(b) Jakesch (2015)

Abbildung E.1.: Verwendete Mikroskopieaufnahmen zur Ermittlung der Größenverteilung der Siliziumphase im Eutektikum.

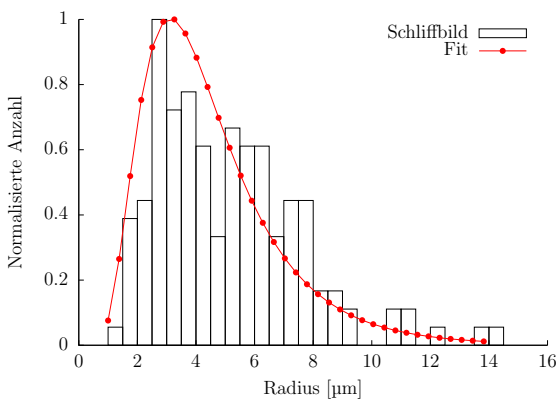
überzuführen und mittels einer Wahrscheinlichkeitsfunktion zu fitten. Die Bestimmung des Histogramms wird mithilfe der Software `gnuplot` durchgeführt. Das Fitten der Funktion wird iterativ durchgeführt und beinhaltet das Erstellen einer Ausscheidungsverteilung in MatCalc, dem Exportieren und Normalisieren dieser und dem anschließenden Vergleich mit dem erstellten Histogramm. Die Eingabeparameter in MatCalc werden solange angepasst, bis eine Übereinstimmung gefunden wird.

Die in dieser Arbeit verwendeten Verteilungen und Fittingkurven sind in Abbildung E.2 dargestellt. Unter der Annahme einer Platte wird in MatCalc die logarithmische Normalverteilung mit folgenden Werten generiert:

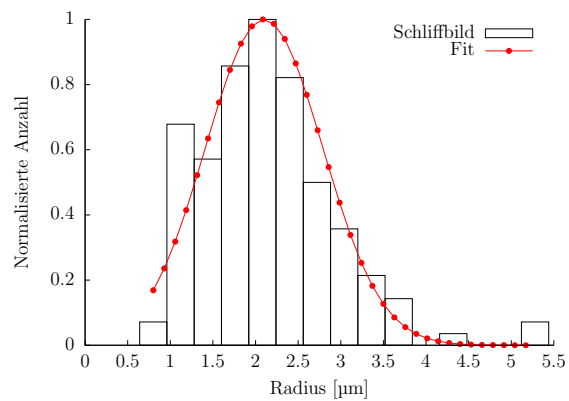
$$r_{\min} = 1,00 \quad r_{\text{mean}} = 4,00 \quad r_{\max} = 14,20 \quad \text{std} = 0,50 \quad [-] .$$

Die Radien sind in  $\mu\text{m}$  angegeben und `std` bezeichnet die Standardabweichung. Unter der Annahme einer Nadel wird erneut die logarithmische Normalverteilung mit nachfolgenden Einstellungen gefunden:

$$r_{\min} = 0,8 \quad r_{\text{mean}} = 2,10 \quad r_{\max} = 5,30 \quad \text{std} = 0,13 .$$



(a) Annahme Platte



(b) Annahme Nadel

Abbildung E.2.: Ermittelte Ausscheidungsverteilung der Siliziumphase im Eutektikum mit den Daten von Hajkowski et al. (2012) und Jakesch (2015) nach dem Gießen.

Die Werte der Radien sind in  $\mu\text{m}$  angegeben.

Es ist anzumerken, dass die Erstellung des Histogramms sehr sensible auf die gewählte Bandbreite ist. Die Erstellung dieser wird mit einer iterativ-visuellen Methode durchgeführt. Das bedeutet, dass die Bandbreite iterativ geändert, das Ergebnis visuell dargestellt und das qualitativ beste Resultat verwendet wird. Ferner ist darauf zu achten, dass die Einteilung der Abszisse nicht identisch ist; dadurch wird eine detailliertere Darstellung ermöglicht.

### E.1.2. Bestimmung der $\beta$ -Verteilung

Der Ablauf zur Bestimmung der  $\beta$ -Verteilung wird analog zur Siliziumverteilung durchgeführt. Dazu werden die Mikroskopieaufnahmen von Liu et al. (1999) herangezogen, die in Abbildung E.4 auf nächster Seite gegeben sind. Tabelle E.3 enthält die detaillierte Zusammenfassung der berechneten Werte, die zur Histogrammerstellung herangezogen werden.

Daraus ergibt sich unter der Annahme einer plattenförmigen Ausbildung der  $\beta$ -Phase eine logarithmische Normalverteilung; vergleiche Darstellung E.3a. Die dazugehörigen Parameter lauten — Angaben in  $\mu\text{m}$ :

$$r_{\min} = 0,6 \quad r_{\text{mean}} = 2,90 \quad r_{\max} = 7,00 \quad \text{std} = 0,50 .$$

Hingegen wird bei der Analyse der Verteilungsgröße für den Fall, dass die  $\beta$ -Phase nadelförmig wächst, eine Gauß'sche Normalverteilung verwendet; siehe Grafik E.3b. Für die MatCalc relevanten Parameter folgt:

$$r_{\min} = 0,5 \quad r_{\text{mean}} = 1,90 \quad r_{\max} = 5,61 \quad \text{std} = 0,30 .$$

Die Einheiten der Radien sind erneut in  $\mu\text{m}$  gegeben.

Erneut ist anzumerken, dass die Abszisseneinteilung beider Grafiken, E.3a und E.3b, zur Steigerung des Informationsgehalts, unterschiedlich eingeteilt ist.

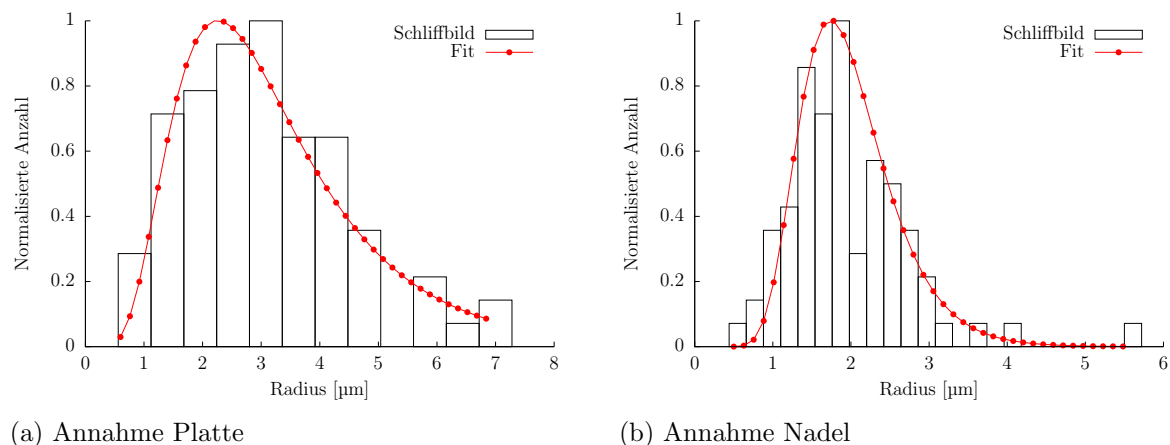
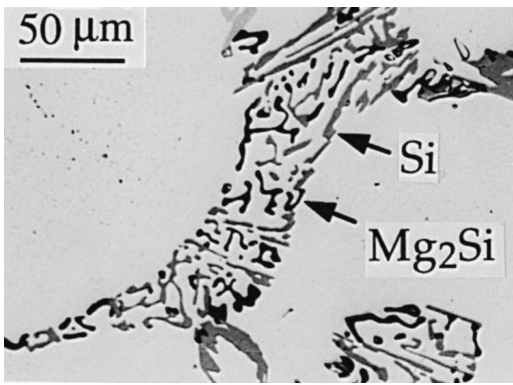
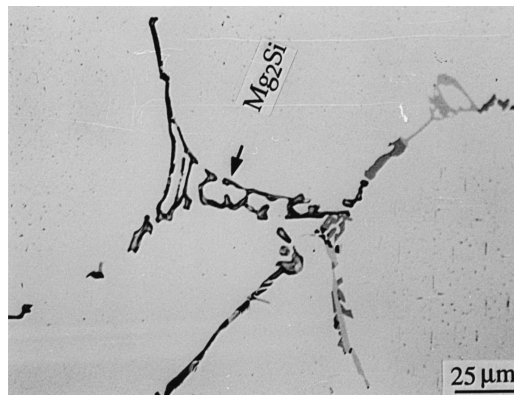


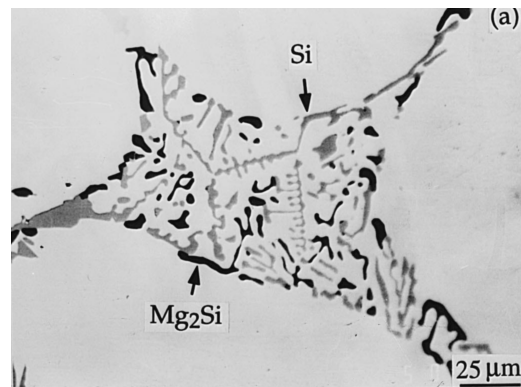
Abbildung E.3.: Ermittelte Größenverteilung der  $\beta$ -Phase mit den Daten aus Liu et al. (1999) nach dem Gießen..



(a) Liu et al. (1999) Figure 1



(b) Liu et al. (1999) Figure 4



(c) Liu et al. (1999) Figure 6

Abbildung E.4.: Verwendete Mikroskopieaufnahmen zur Ermittlung der Größenverteilung für die  $\beta$ -Phase ( $Mg_2Si$ ).











## E.2. Ermittlung der Phasentransformationsradien

Aufgrund der Tatsache, dass bis heute kein Modell zur Phasenumwandlung in Aluminiumlegierungen postuliert wurde, kann die Kinetikberechnung nicht physikalisch abgebildet werden. Allerdings erlaubt MatCalc eine Phasentransformation mithilfe einer linearen Wahrscheinlichkeitsfunktion abzubilden — dies entspricht keiner physikalischen Modellierung. Der Grundgedanke ist wie folgt: Es werden zwei Transformationsradien  $r_{\text{trans},1}$  und  $r_{\text{trans},2}$  vorgegeben, die die Wahrscheinlichkeit einer Transformation von Phase 1 in Phase 2 von 0% und 100% darstellt und ist als Linearfunktion festgeschrieben.

Ist eine Phase als transformierbar eingestellt, werden alle einzelnen Radienklassen dieser Phase bezüglich unten stehender Abbildung E.5 geprüft. Das bedeutet, dass sobald eine Radienklasse im Transformationsgebiet vorliegt, stets geprüft wird, ob eine Umwandlung erfolgt oder nicht. Wird entschieden, dass eine Radienklasse umgewandelt wird, entfällt diese von Phase 1 und wird zur Phase 2 hinzugefügt. Entsprechend können numerische Probleme auftauchen, da aufgrund der unverzögerten und unphysikalischen Umwandlung, sprunghafte Änderung der Materialeigenschaften erfolgen können.

Die Bestimmung der temperaturabhängigen Transformationsradien für die Umwandlungsreaktionen von Cluster  $\rightarrow \beta''$ ,  $\beta'' \rightarrow \beta'$  und  $\beta' \rightarrow \beta$  wird mithilfe der Messdaten von Colley (2011) bestimmt. Dazu werden alle Streckgrenzkurven  $R_{p,0.2}$  mit MatCalc modelliert und die Transformationsradien so angepasst, dass die Messwerte bestmöglich approximiert werden.

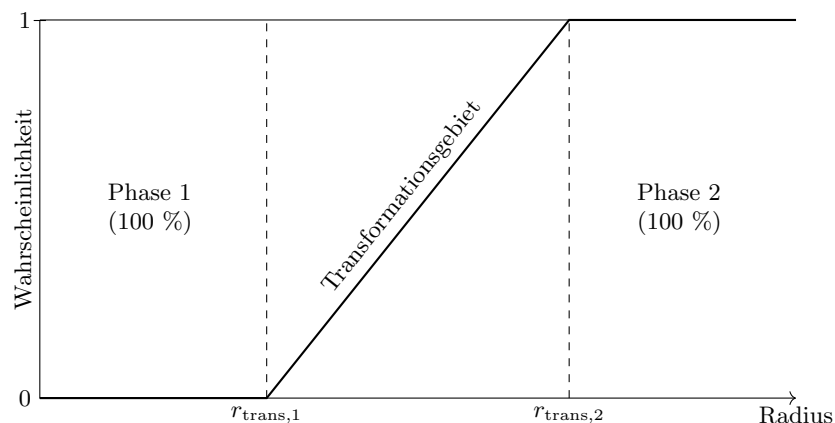


Abbildung E.5.: Darstellung des in MatCalc implementierten Pseudo-Phasenumwandlungsmodells. Die Wahrscheinlichkeit einer Transformation steigt linear von 0 auf 1 (0 % auf 100 %) an.

### Kaltauslagern 25 °C

Beim Kaltauslagern der Aluminiumlegierungen nach dem Lösungsglühen (540 °C für 30 Minuten) und anschließendem Abschrecken in Wasser, wird das Material bei 25 °C für 1000 h ausgelagert. Mit steigender Auslagerungszeit erhöht sich die Streckgrenze  $R_{p,0.2}$  des Materials. Dies ist durch das Wachsen der Cluster-Phase begründet. Daher werden keine anderen Phasen erwartet, sodass der Transformationsradius  $r_{\text{trans},1}$  für die Cluster  $\rightarrow \beta''$  Umwandlung so gewählt wird, dass keine  $\beta''$ -Phase erzeugt wird. Mit anderen Worten bedeutet das, dass der Transformationsradius  $r_{\text{trans},1}$  stets größer sein muss, als der maximal auftretende Radius der Cluster. Da sich nur Cluster ausbilden, ist es nicht möglich die Transformationsradien anderer

Phasen zu bestimmen; vergleiche Tabelle E.6.

Das Ergebnis der Berechnung ist in Abbildung E.6 zu sehen. Es ist ersichtlich, dass die Messwerte durch die numerische Modellierung abgebildet werden.

	Cluster $\rightarrow \beta''$	$\beta'' \rightarrow \beta'$	$\beta' \rightarrow \beta$
$r_{\text{trans},1}$ [nm]	1,82	-	-
$r_{\text{trans},2}$ [nm]	-	-	-

Tabelle E.6.: Eingestellte Transformationsradien für die unterschiedlichen Phasenreaktionen für das Kaltauslagern bei 25 °C.

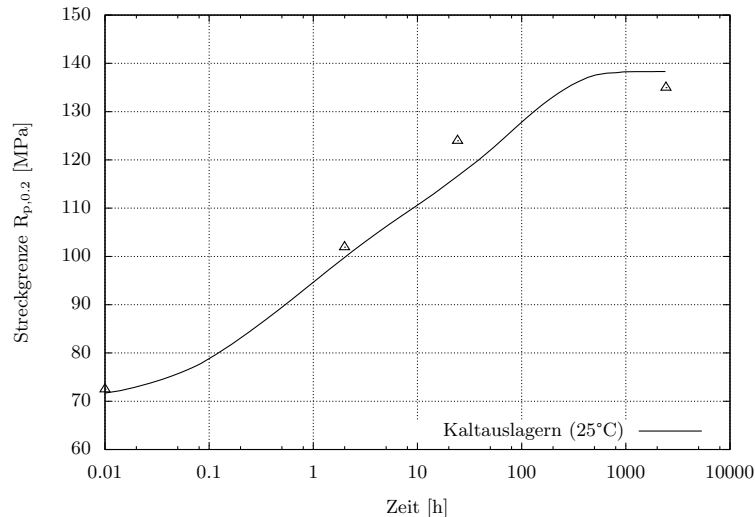


Abbildung E.6.: Berechnete Streckgrenze  $R_{p,0.2}$  mithilfe der Festkörper-Kinetikberechnung für das Kaltauslagern bei 25 °C nach dem Lösungsglühen. Die eingezeichneten Symbole stellen die Messwerte von Colley (2011) dar.

### Warmauslagern 150 °C

Das Warmauslagern bei 150 °C bedarf der Transformation verschiedener Phasen. Hierbei werden die Cluster zunächst in  $\beta''$  umgewandelt, diese anschließend in  $\beta'$  und zuletzt wird die  $\beta$ -Phase wiederum in  $\beta$ -Phase transformiert. Hier wird von Pre-Aging, Peak-Aging und Over-Aging gesprochen. Diese sind direkt mit den vorliegenden Phasen korrelierbar. Während beim Pre-Aging hauptsächlich die Cluster- und  $\beta''$ -Phasen vorliegen, sind beim Peak-Aging vor allem die  $\beta''$ - und  $\beta'$ -Phase vorzufinden. Beim Overaging werden die metastabilen Phasen in die stabile  $\beta$ -Phase umgewandelt. Da diese inkohärent zur Matrix ist, sinkt die Streckgrenze  $R_{p,0.2}$ . Die für den Fit verwendeten Transformationsradien für die Aluminiumlegierung beim Auslagern von 150 °C sind in Tabelle E.7 gegeben. Die quantitativen Ergebnisse der Streckgrenzkurve sind in Abbildung E.7 dargestellt. Erneut ist eine sehr gute Übereinstimmung zwischen den experimentellen Werten und der Numerik gegeben. Die Fluktuationen treten aufgrund der Phasenumwandlung auf. Bei jedem Peak wird eine Klasse einer Phase in die nächste verschoben. Die Phasenanteile sind in Abbildung E.8 gegeben.

Es ist auf die unterschiedliche Abszisseneinteilung acht zu geben.

	Cluster $\rightarrow \beta''$	$\beta'' \rightarrow \beta'$	$\beta' \rightarrow \beta$
$r_{\text{trans},1}$ [nm]	1,20	1,25	2,10
$r_{\text{trans},2}$ [nm]	1,65	2,05	3,00

Tabelle E.7.: Eingestellte Transformationsradien für die unterschiedlichen Phasenreaktionen für das Warmlagern bei 150 °C.

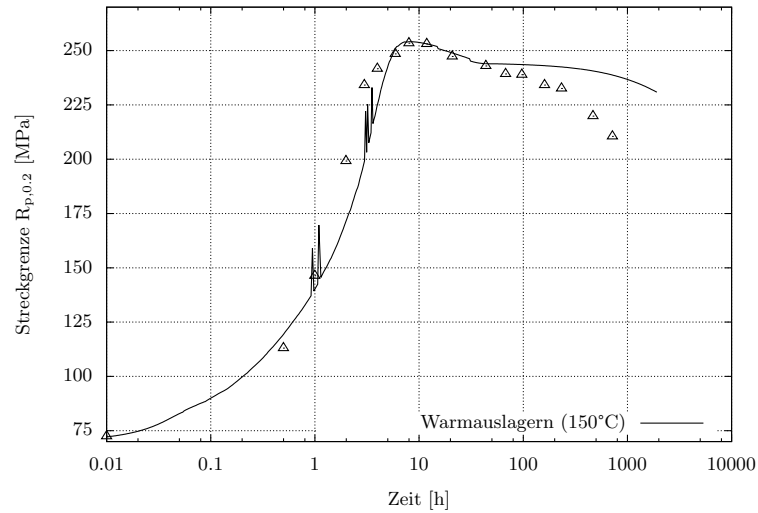


Abbildung E.7.: Berechnete Streckgrenze  $R_{p,0.2}$  mithilfe der Festkörper-Kinetikberechnung für das Warmauslagern bei 150 °C nach dem Lösungsglühen. Die eingezeichneten Symbole stellen die Messwerte von Colley (2011) dar.

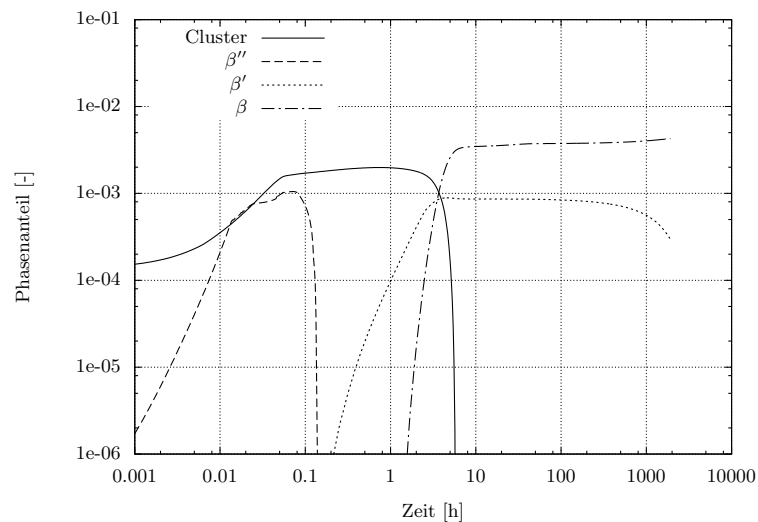


Abbildung E.8.: Zeitliche Entwicklung der einzelnen Phasen während des Warmauslagerns bei 150 °C.

## Warmauslagern 180 °C

Werden die Auslagerungstemperaturen erhöht, werden Diffusionsvorgänge beschleunigt. Das bedeutet, dass Phasen schneller wachsen, schneller den stabilen Zustand verlassen und — physikalisch gesehen — sich in andere Phasen umwandeln. Wird der stabile Phasenzustand in MatCalc verlassen, *verschwindet* diese Phase. Mit anderen Worten, die Anzahl der Nuklei und der Radius der einzelnen Partikel nimmt ab. Daher muss mit dem Transformationsradiusmodell dafür gesorgt werden, dass sich die Cluster früh genug in eine andere Phase transformieren, bevor diese nicht mehr existiert. Diesbezüglich werden die Transformationsradien im Vergleich zu 150 °C weiter — hin zu kleineren Werten — korrigiert. Für die  $\beta$ -Phase werden die Radien vergrößert, um ein besseres Ergebnis der Streckgrenzkurve  $R_{p,0.2}$  zu erhalten. Abbildung E.11 zeigt den zeitlichen Verlauf der quantitativen Werte. Zu Beginn ( $\approx 0,1$  h) ist ein Peak der Materialfestigkeit ersichtlich, der auf das schnelle Wachstum der Cluster zurückzuführen ist. Anschließend fällt die Festigkeit ab, da die Cluster-Phase instabil wird. Die bis zu  $t = 0,1$  h transformierte, vorliegende  $\beta''$ -Phase besitzt einen zu kleinen Radius, um eine signifikante Verbesserung der Streckgrenze  $R_{p,0.2}$  zu erzielen. Mit Voranschreiten der Zeit wachsen diese, wodurch zum einen der Phasenanteil in der Matrix zunimmt und dadurch eine Verfestigung durch die  $\beta''$ -Phase erkennbar wird und zum anderen eine weitere Transformation von  $\beta'' \rightarrow \beta'$  vorliegt. Da beide Phasen kohärent zur Matrix sind, erfolgt vor allem basierend auf dem sogenannten volumetrischen Misfit (Gitterlänge der Ausscheidung ist unterschiedlich zur Matrix) eine Festigkeitssteigerung.

Des Weiteren ist ersichtlich, dass fast keine Überalterung eintritt. Dies ist darin begründet, dass zum Erreichen der Maximalfestigkeit ein schnelles Umwandeln von  $\beta'' \rightarrow \beta'$  benötigt wird. Im vorliegenden Fall ist bei  $t = 2,5$  h keine  $\beta''$ -Phase mehr im System enthalten, wodurch nur noch die Transformation von  $\beta' \rightarrow \beta$  erfolgt. Der Festigkeitsabfall wird hier nicht korrekt wiedergegeben; vergleiche Abbildung E.9 mit E.10. Diese Eigenschaft ist der nicht physikalischen Transformationsmodellierung geschuldet.

Erneut wird auf die verschiedenen Zeitachsen hingewiesen.

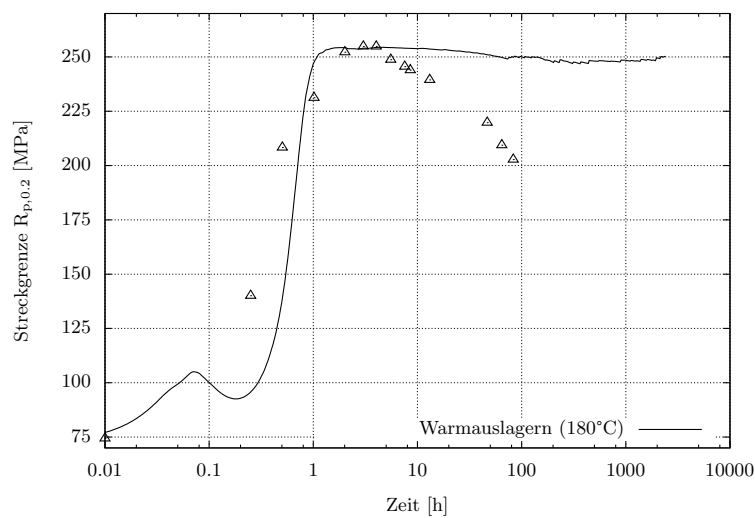


Abbildung E.9.: Berechnete Streckgrenze  $R_{p,0.2}$  mithilfe der Festkörper-Kinetikberechnung für das Warmauslagern bei 180 °C nach dem Lösungsglühen. Die eingezeichneten Symbole stellen die Messwerte von Colley (2011) dar.



	Cluster $\rightarrow \beta''$	$\beta'' \rightarrow \beta'$	$\beta' \rightarrow \beta$
$r_{\text{trans},1}$ [nm]	0,98	1,20	2,20
$r_{\text{trans},2}$ [nm]	1,28	3,10	4,50

Tabelle E.8.: Eingestellte Transformationsradien für die unterschiedlichen Phasenreaktionen für das Warmlagern bei 180 °C.

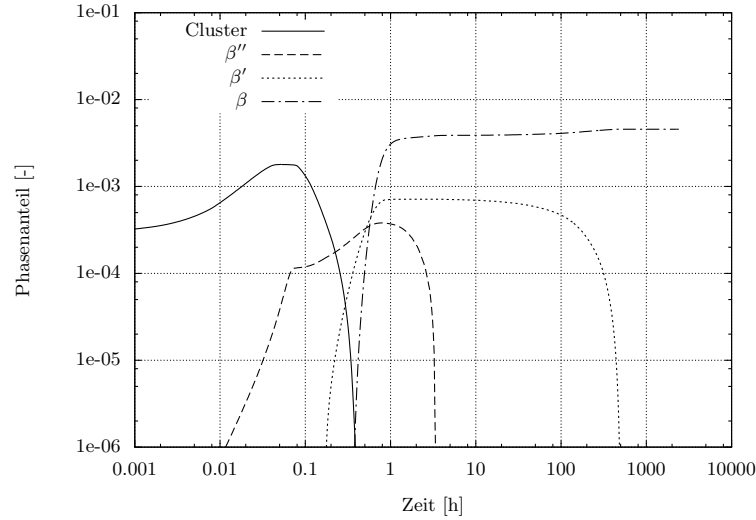


Abbildung E.10.: Zeitliche Entwicklung der einzelnen Phasen während dem Warmauslagern bei 180 °C.

### Warmauslagern 200 °C

Das Warmauslagern bei 200 °C bedarf eine weitere Anpassung der Transformationsradien; vergleiche Tabelle E.9. Wie bei der Wärmebehandlung mit 180 °C ist der erste Festigkeitspeak auf das rasche Wachstum der Cluster zurückzuführen; hier jedoch wesentlich ausgeprägter. Während die Cluster wachsen, wird ein Großteil dieser in die  $\beta''$ -Phase transformiert, wodurch ein drastischer Anstieg der Festigkeit auf bis 150 MPa zu verzeichnen ist. Anschließend verschwindet die Cluster-Phase vollständig, während die  $\beta''$ -Phase wächst, sich ein Teil dieser in die  $\beta'$ -Phase umwandelt und anschließend verschwindet. Durch das Fehlen der Cluster- und  $\beta''$ -Phase fällt die Festigkeit auf circa 80 MPa herab. Anschließend wächst die  $\beta'$ -Phase, wodurch die  $\beta$ -Phase entsteht. Das Wachstum der  $\beta'$ - $\beta$ -Phase erhöht die Festigkeit erneut. Die Überalterung wird auch hier nicht korrekt abgebildet.

	Cluster $\rightarrow \beta''$	$\beta'' \rightarrow \beta'$	$\beta' \rightarrow \beta$
$r_{\text{trans},1}$ [nm]	0,80	0,85	2,50
$r_{\text{trans},2}$ [nm]	1,10	3,10	3,80

Tabelle E.9.: Eingestellte Transformationsradien für die unterschiedlichen Phasenreaktionen für das Warmlagern bei 200 °C.

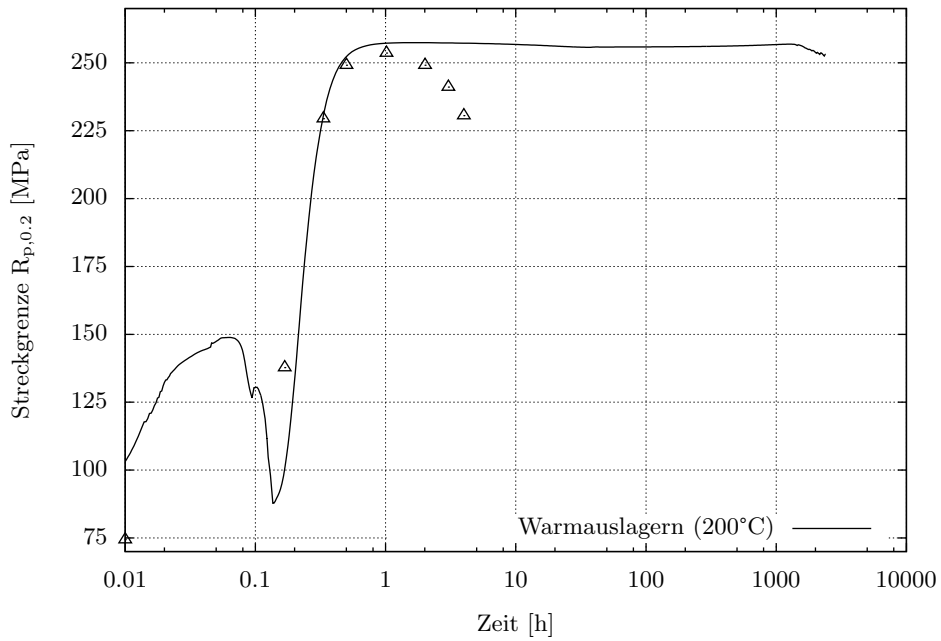


Abbildung E.11.: Berechnete Streckgrenze  $R_{p,0.2}$  mithilfe der Festkörper-Kinetikberechnung für das Warmauslagern bei 200 °C nach dem Lösungsglühen. Die eingezeichneten Symbole stellen die Messwerte von Colley (2011) dar.

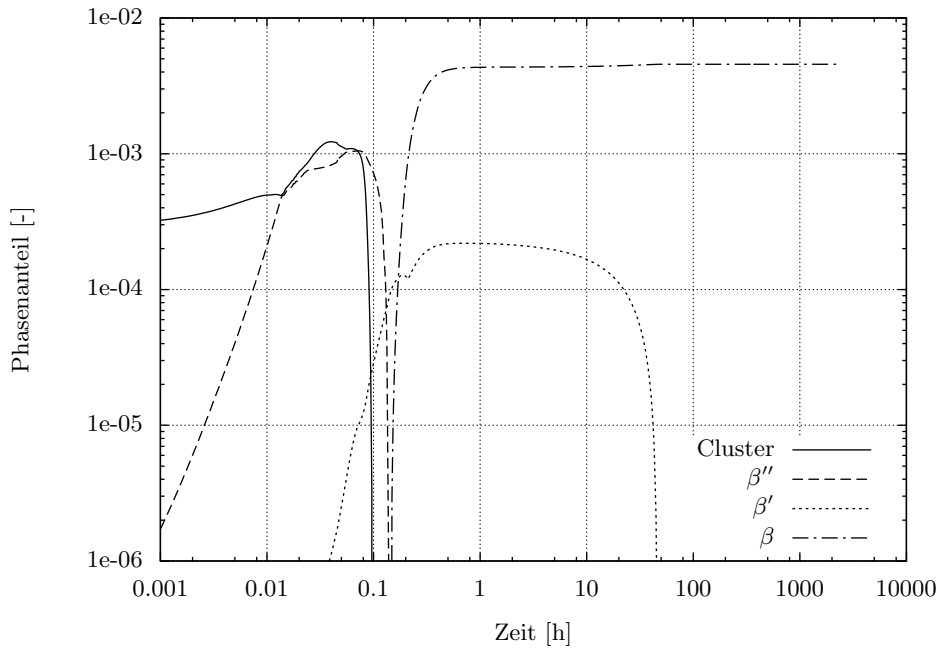


Abbildung E.12.: Zeitliche Entwicklung der einzelnen Phasen während dem Warmauslagern bei 200 °C.

## Abschließend

Durch die Nachbildung der verschiedenen experimentellen Auslagerungswerte kann geschlussfolgert werden, dass die Verwendung des Radien-Transformationsmodells eine Möglichkeit zur numerischen Abbildung verschiedener Warmauslagerungsszenarien darstellt. Jedoch wird ersichtlich, dass je höher die Auslagerungstemperaturen sind, dass Modell unphysikalische Ergebnisse erzielt; beispielsweise der starke Anstieg auf 150 MPa bei einer Auslagerungstemperatur von 200 °C in den ersten Minuten.

## Temperaturabhängige Phasentransformationsradien

Die justierten Transformationsradien sind nur für die entsprechenden Auslagerungstemperaturen gültig. Für das Kaltauslagern (25 °C) und Warmauslagern (150 °C) korrelieren die gefitteten Kurvenverläufe mit den Messungen sehr gut. Dies ist auf das Clusterwachstum und die thermodynamischen Zustände zurückzuführen; bei niedrigen Temperaturen sind die Cluster länger stabil, sodass eine realistische Phasenumwandlung mithilfe des Transformationsmodells abgebildet werden kann.

Zur Erstellung der temperaturabhängigen Transformationsradien wird versucht, einen funktionalen Zusammenhang zwischen den einzelnen Temperaturen und Radien zu erhalten. Da nur für den minimalen Transformationsradius von Cluster  $\rightarrow \beta''$  alle Temperaturen vorliegen, wird dieser als Startpunkt herangezogen. Abbildung E.13 und E.14 zeigen alle verwendeten Transformationsradien in Abhängigkeit der Temperatur.

Der minimale Transformationsradius von Cluster  $\rightarrow \beta''$  wird mit einem Polynom 2. Ordnung unter Verwendung der Methode der kleinsten Fehlerquadrate approximiert. Die Koeffizienten des Polynoms  $r(T) = AT^2 + BT + C$  bestimmen sich zu:

$$\begin{aligned} A &= -1,74959e^{-14} \quad [\text{m}^\circ\text{C}^{-2}] , \\ B &= -1,87199e^{-12} \quad [\text{m}^\circ\text{C}^{-1}] , \\ C &= 1,87756e^{-09} \quad [\text{m}] . \end{aligned}$$

Anschließend werden die maximalen Transformationsradien der Phasenumwandlung von Cluster  $\rightarrow \beta''$  herangezogen und die Differenz zwischen minimalem und maximalem Radius bestimmt; vergleiche Tabelle E.10. Diese liegen zwischen 0,30 nm und 0,45 nm. Es wird angenommen, dass dieses Intervall im gesamten Temperaturbereich konstant ist. Aufgrund zahlreicher MatCalc Tests wird die Differenz  $\Delta$  mit 0,45 nm festgelegt. Entsprechend kann der zweite Transformationsradius für die Obergrenze  $r_{\text{trans},2}$  mit dem gleichen Polynom, das für die Untergrenze verwendet wird, beschrieben werden; mit einem Offset von  $\Delta = 0,45$  nm.

Die Phasenumwandlungen von  $\beta'' \rightarrow \beta'$  und  $\beta' \rightarrow \beta$  werden ähnlich behandelt. Da hier die Werte für 25 °C nicht vorliegen, ist eine Approximation der Kurvenverläufe für niedrige Temperaturen nicht möglich. Hinzukommt der unphysikalische Radienverlauf, wie beispielsweise bei der

	25 °C	150 °C	180 °C	200 °C
$\Delta_1 =  r_{\text{trans},2} - r_{\text{trans},1} $ [nm]	-	0,45	0,30	0,30

Tabelle E.10.: Differenz zwischen dem maximalen und minimalen Transformationsradius der Phasenumwandlung Cluster  $\rightarrow \beta''$ .

$\beta$ -Phase. Ein Fit mittels einer Parabel durch die gegebenen Punkte — vergleiche Abbildung E.14 — würde bedeuten, dass bei Temperaturen kleiner als 120 °C ein negativer Transformationsradius vorliegt. Aufgrund der Anomalien im Radienverlauf für beide Radien wird angenommen, dass alle Transformationsintervalle dem Polynom 2. Ordnung, wie es bei der Sequenz Cluster  $\rightarrow \beta''$  verwendet wird, folgen.

Die Bestimmung des Offsets zwischen  $r_{\text{trans},1}$  und  $r_{\text{trans},2}$  erfolgt wie zuvor. Tabelle E.11 gibt die Zusammenfassung aller Intervalle der einzelnen Phasenumwandlungen an. Der Mittelwert dieser wird verwendet, um einen Anhaltspunkt für die MatCalc Berechnungen zu generieren. Die zum Schluss verwendeten Werte sind extra aufgeführt und stellen einen Kompromiss dar, bei dem die Genauigkeit zwischen Raumtemperatur und dem Warmauslagern bei 150 °C akzeptabel ist.

Die identifizierten Polynome lauten:

$$\text{Cluster} \rightarrow \beta'' : r_{\text{trans},1} = AT^2 + BT + C , \quad (\text{E.1})$$

$$\text{Cluster} \rightarrow \beta'' : r_{\text{trans},2} = AT^2 + BT + C + \Delta_1 , \quad (\text{E.2})$$

$$\beta'' \rightarrow \beta' : r_{\text{trans},1} = AT^2 + BT + C + \delta_2 , \quad (\text{E.3})$$

$$\beta'' \rightarrow \beta' : r_{\text{trans},2} = AT^2 + BT + C + \delta_2 + \Delta_2 , \quad (\text{E.4})$$

$$\beta' \rightarrow \beta : r_{\text{trans},1} = AT^2 + BT + C + \delta_3 , \quad (\text{E.5})$$

$$\beta' \rightarrow \beta : r_{\text{trans},2} = AT^2 + BT + C + \delta_3 + \Delta_3 . \quad (\text{E.6})$$

Die Größe  $\delta_i$  stellt dabei den Abstand zwischen der Phasenreaktion Cluster  $\rightarrow \beta''$  und  $\beta'' \rightarrow \beta'$  dar. Diese werden iterativ in MatCalc erzeugt und sind als  $\delta_2 = 0,05$  nm und  $\delta_3 = 0,70$  nm identifiziert.

Diese Grundidee ist in MatCalc implementiert und dient der Ersetzung eines physikalischen Modells. Die entsprechenden Skriptertklärungen sind in Kapitel 3.6 auf Seite 58 diskutiert und befinden sich auf dem Datenträger unter `Doktorarbeit/MatCalc/`.

Phasenreaktion	Intervall	25 °C	150 °C	180 °C	200 °C	Verwendet
Cluster $\rightarrow \beta''$	$\Delta_1 =  r_{\text{trans},2} - r_{\text{trans},1} $ [nm]	-	0,45	0,30	0,30	0,45
$\beta'' \rightarrow \beta'$	$\Delta_2 =  r_{\text{trans},2} - r_{\text{trans},1} $ [nm]	-	0,70	2,80	2,25	0,80
$\beta' \rightarrow \beta$	$\Delta_3 =  r_{\text{trans},2} - r_{\text{trans},1} $ [nm]	-	0,90	2,23	1,30	0,80

Tabelle E.11.: Differenz zwischen dem minimalen und maximalen Transformationsradius der einzelnen Phasenumwandlungen.

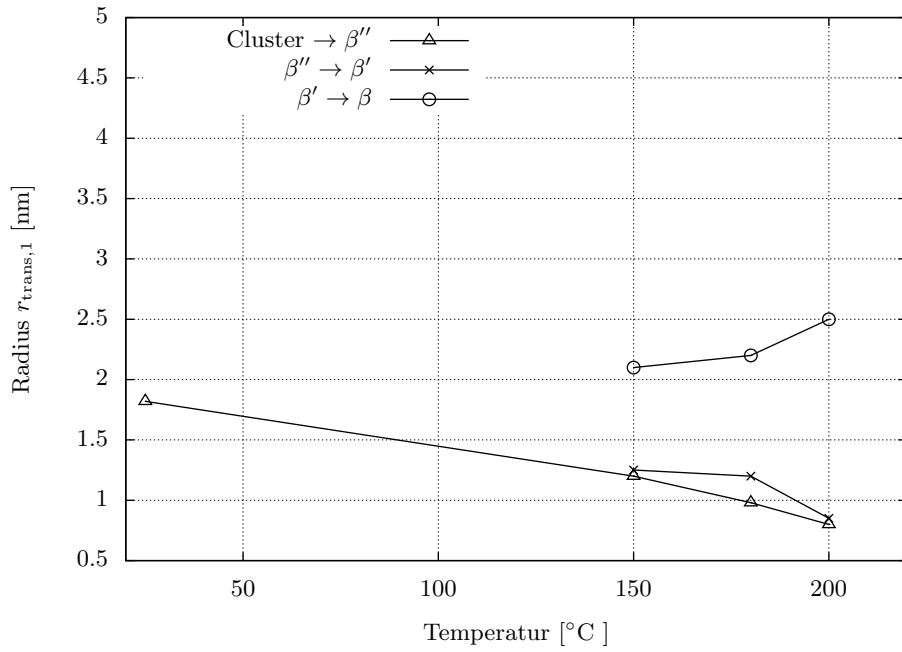


Abbildung E.13.: Visualisierung der verwendeten Transformationsradien (min) zum Fit der experimentellen Daten mithilfe des Radien-Transformationsmodells.

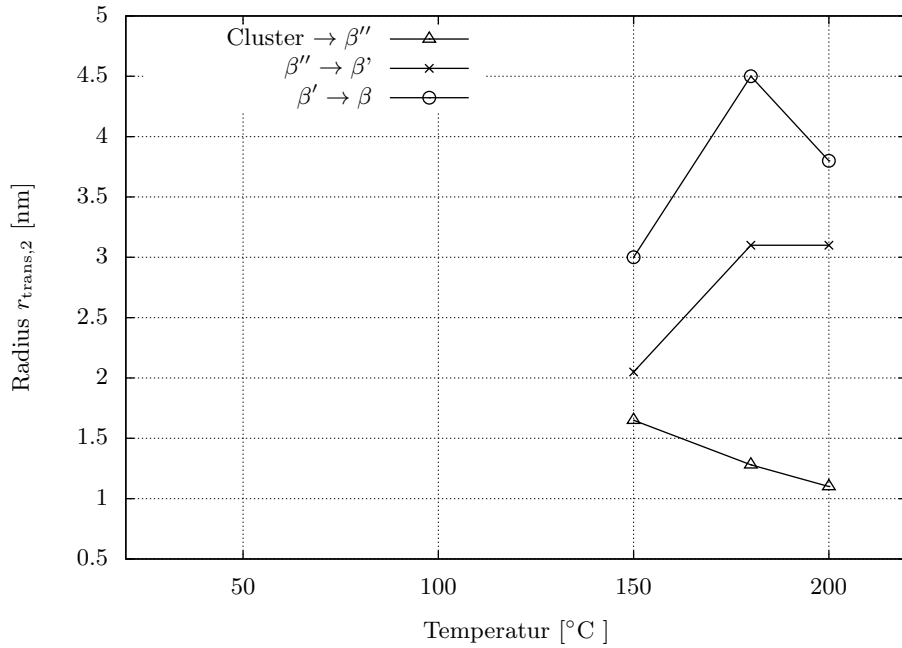


Abbildung E.14.: Visualisierung der verwendeten Transformationsradien (max) zum Fit der experimentellen Daten mithilfe des Radien-Transformationsmodells.

### E.3. Bestimmung der Komposition des reduzierten Systems

Die in der Scheil-Gulliver Analyse entstehenden Phasen, explizit der  $\alpha$ -AlFeSi und  $\pi$ -AlFeMgSi, werden für die weitere Kinetik Berechnung nicht weiter betrachtet. Allerdings entziehen diese Phasen, wenn auch nur geringfügig existent, dem globalen System, Silizium, Magnesium und Eisen. Daher ist es notwendig, die Zusammensetzung des reduzierten Systems zu kennen. Die Bestimmung des Massen- und Molenbruchs für dieses System ist in Tabelle Tabelle E.12 aufgezeigt. Für die Berechnung werden die Gleichungen aus Warnatz et al. (2001) verwendet und lauten für den Massen- ( $w$ ) und Molenbruch ( $x$ ):

$$w_i = \frac{M_i x_i}{\sum M_j x_j}, \quad (\text{E.7})$$

$$x_i = \frac{w_i}{M_i} \bar{M}. \quad (\text{E.8})$$

$M_i$  repräsentiert die molare Masse der einzelnen Legierungselemente bezogen auf 1 mol und ist mit der relativen Atommasse identisch — siehe Periodensystem der Elemente.  $\bar{M}$  stellt die mittlere molare Masse des Systems dar und kann durch nachfolgende Gleichung bestimmt werden:

$$\bar{M} = \frac{1}{\sum \frac{w_i}{M_i}}. \quad (\text{E.9})$$

Des Weiteren kann der Massenbruch mit den Massen direkt bestimmt werden:

$$w_i = \frac{m_i}{m_{\text{ges}}}. \quad (\text{E.10})$$

Tabelle E.12 gibt einen Überblick über die Berechnung der Zusammensetzung des reduzierten Systems. Es ist zu beachten, dass die angegebenen Zahlenwerte gerundete Endwerte darstellen. Die Berechnung selbst wird mit höherer Genauigkeit durchgeführt. Die Bezugsmasse des Systems beträgt  $m_{\text{ges}} = 1000$  g.

Weitere Details sind aus `Doktorarbeit/MatCalc/ScheilReducedSystem.ods` zu entnehmen.

	Einheit	Al	Si	Mg	Fe	Ti	Summe	Sonstiges
Molekulargewicht aus MatCalc	[g mol <sup>-1</sup> ]	26,98154	28,08550	24,30500	55,84700	47,88000		
Legierung aus Colley (2011)	[wt%]	92,00000	7,42000	0,30000	0,17000	0,11000	100	
Masse des Elements im System	[g]	920,00000	74,20000	3,00000	1,70000	1,10000	1000	
Massenbruch des Elements im System ( $w_i$ )	[-]	0,92000	0,07420	0,00300	0,00170	0,00110	1	→ mittlere molare Masse [g mol <sup>-1</sup> ]: 27,08840
Molenbruch des Elements ( $x_i$ )	[-]	0,92364	0,07157	0,00334	0,00084	0,00062	1	
Ergebnisse der Scheil Berechnung aus MatCalc und Bestimmung des reduzierten Systems								
Zusammensetzung von $\alpha$ -AlFeSi ( $x_i$ )	[-]	0,66520	0,18280	0,00000	0,15200	0,00000	1	Molenbruch der Phase im System: 0,00475
Zusammensetzung von $\pi$ -AlFeMgSi ( $x_i$ )	[-]	0,48649	0,27027	0,18919	0,05405	0,00000	1	Molenbruch der Phase im System: 0,00167
Eingebaute Elemente in $\alpha$ -AlFeSi ( $x_i$ )	[-]	0,00316	0,00088	0,00000	0,00072	0,00000	0,00475	
Eingebaute Elemente in $\alpha$ -AlFeMgSi ( $x_i$ )	[-]	0,00081	0,00045	0,00032	0,00009	0,00000	0,00167	
Molenbruch beider Phasen ( $x_i$ )	[-]	0,00398	0,00132	0,00032	0,00081	0,00000	0,00643	
Molenbruch des reduzierten Systems ( $x_i$ )	[-]	0,91967	0,07025	0,00302	0,00001	0,00062	0,99357	
Massenbruch beider Phasen ( $w_i$ )	[-]	0,00396	0,00137	0,00028	0,00168	0,00000	0,00729	
Massenbruch des reduzierten Systems ( $w_i$ )	[-]	0,91604	0,07281	0,00272	0,00002	0,00110	0,99271	
Masse des Elements in den Phasen	[g]	3,95984	1,36948	0,28385	1,67595	0,00000	7,28913	
Masse des Elements im reduzierten Systems	[g]	916,04016	72,83052	2,71615	0,02405	1,10000	992,71087	← neue Bezugsmasse des Systems
Summe der letzten beiden Einträge	[g]	920,00000	74,20000	3,00000	1,70000	1,10000	1000	← muss Original ergeben (Zeile 3)
Neubestimmung der Massen- und Molenbrüche im reduzierten System								
Masse des Element	[g]	916,04016	72,83052	2,71615	0,02405	1,10000	992,71087	← neue Bezugsmasse des Systems
Massenbruch des Elements	[-]	0,92277	0,07337	0,00274	0,00002	0,00111	1	→ molare Masse [g mol <sup>-1</sup> ]: 27,06486
Molenbruch des Elements	[-]	0,92562	0,07070	0,00305	0,00001	0,00063	1	
Molenbruch via MatCalc	[-]	0,92561	0,07070	0,00305	0,00001	0,00063	1	

Tabelle E.12.: Zusammenfassung der Berechnung zur Ermittlung der Massen- und Molenbrüche für das reduzierte System. Alle Werte sind gerundet; die Berechnung selbst ist mit höherer Genauigkeit durchgeführt.

## E.4. Bestimmung der Abkühlkurve von AlSi7Mg0.3 nach dem Erstarren im Sandguss

Für die Kinetikberechnung — die aus Phase 2 hervorgeht — wird angenommen, dass die Aluminiumlegierung im Sandgussverfahren abgegossen wird. Die Abkühlkurve wird mithilfe der kommerziellen Software MAGMASoft<sup>®</sup> bestimmt. Hierzu wird eine Sandform mit den Abmaßen 0,465 m x 0,340 m x 0,200 m erzeugt. Im Schwerpunkt der Form wird eine runde Aluminiumplatte mit dem Durchmesser von 0,2 m und einer Wandstärke von 0,007 m platziert. Die Platte besteht aus der in dieser Arbeit verwendeten AlSi7Mg0.3 Legierung und wird zum Zeitpunkt  $t = 0$  s von 670 °C auf 35 °C heruntergekühlt. Alle Stoffwerte sind temperaturabhängige Größen; es wird die Datenbank von MAGMASoft<sup>®</sup> verwendet. Während der Simulation wird die Temperatur an einem Punkt mit den Koordinaten  $x = 0,285$  m,  $y = 0,170$  m und  $z = 0,104$  m aufgezeichnet. Die Originaldaten dieses Temperaturprofils sind auf dem beigegeführten Datenträger unter Doktorarbeit/MatCalc/Kinetics/TemperatureKinetic\_1 einsehbar.

Für die Abkühlkurve die in MatCalc verwendet wird, werden die Originaldaten manipuliert, sodass lediglich der Temperaturbereich von Solidus bis RT vorliegt. Diese Daten sind in Tabelle E.13 einsehbar. Anschließend werden diese Daten mittels der Applikation namens MatCalcPreparation in ein MatCalc lesbares Format überführt. Die Erläuterung des Quellcodes ist in Anhang E.6 auf Seite 212 gegeben.

Index [-]	Zeit [s]	Temperatur [K]	Index [-]	Zeit [s]	Temperatur [K]	Index [-]	Zeit [s]	Temperatur [K]
1	93,074	559,846	41	420,987	269,161	81	3732,492	73,546
2	97,314	553,395	42	452,800	258,494	82	3814,640	72,599
3	103,852	541,805	43	480,992	249,933	83	3900,785	71,641
4	108,328	532,058	44	514,958	240,561	84	3990,929	70,676
5	112,899	522,443	45	544,994	233,009	85	4137,418	69,183
6	117,565	513,052	46	581,113	224,709	86	4342,000	67,236
7	122,325	503,884	47	618,391	216,910	87	4450,289	66,265
8	127,180	495,032	48	651,267	210,589	88	4562,575	65,298
9	134,640	482,407	49	690,698	203,606	89	4743,130	63,819
10	139,731	474,320	50	725,418	197,931	90	4799,142	63,379
11	144,917	466,476	51	767,002	191,648	91	5122,429	60,984
12	150,198	458,849	52	803,568	186,531	92	5183,977	60,554
13	155,574	451,418	53	847,304	180,853	93	5256,858	60,056
14	161,044	444,169	54	885,715	176,217	94	5320,251	59,631
15	169,426	433,579	55	931,604	171,061	95	5395,285	59,137
16	175,133	426,737	56	1062,004	158,292	96	5684,134	57,332
17	180,934	420,043	57	1208,493	146,514	97	5834,555	56,440
18	186,830	413,492	58	1413,076	133,292	98	6073,779	55,076
19	192,820	407,082	59	1633,650	121,993	99	6265,850	54,077
20	201,983	397,741	60	1870,217	112,246	100	6466,995	53,053
21	208,210	391,683	61	2063,074	105,630	101	6680,951	52,014
22	214,532	385,778	62	2327,934	97,991	102	6865,279	51,157
23	220,948	380,019	63	2608,786	91,352	103	7116,334	50,041
24	227,459	374,402	64	2835,708	86,811	104	7349,203	49,057
25	234,065	368,921	65	2931,004	85,075	105	7564,528	48,186
26	244,151	360,936	66	2940,942	84,900	106	8085,393	46,220
27	250,993	355,777	67	2960,630	84,555	107	8417,269	45,062
28	257,930	350,739	68	3011,813	83,679	108	8696,890	44,138
29	264,961	345,819	69	3056,104	82,940	109	8992,504	43,212
30	272,087	341,014	70	3073,842	82,650	110	9376,633	42,066
31	282,953	334,017	71	3127,830	81,782	111	9756,906	41,005
32	290,316	329,483	72	3169,690	81,127	112	10121,355	40,046
33	297,773	325,052	73	3241,896	80,031	113	10510,838	39,080
34	305,325	320,721	74	3293,905	79,267	114	10964,289	38,026
35	312,972	316,488	75	3319,258	78,901	115	11418,741	37,038
36	320,713	312,349	76	3381,725	78,021	116	11923,251	36,012
37	332,502	306,312	77	3443,882	77,172	117	12462,686	35,004
38	340,479	302,399	78	3510,038	76,297			
39	364,980	291,161	79	3580,191	75,398			
40	394,641	278,939	80	3654,343	74,480			

Tabelle E.13.: Rohdaten der Abkühlung einer Aluminiumplatte nach dem Abguss in der Sandform; Daten stammen von MAGMASoft<sup>®</sup>.



## E.5. cellListGeneration C++ Code Beschreibung

Der Quellcode der Applikation `cellListGeneration` wird nachfolgend diskutiert. Der Source-Code ist in `Doktorarbeit/Programming` zu finden.

### E.5.1. Der Strukturbaum

Der Aufbau des Programms ist im unten angegebenen Strukturbaum ersichtlich. Es besteht aus der Quelldatei und dem `Make` Ordner, welcher Informationen für den Compiler bereitstellt.

```
cellListGeneration
|-- cellListGeneration.C
|-- Make
    |-- files
    |-- options

2 directories , 3 files
```

Quellcode 14: Struktur der Applikation `cellListGeneration`.

### E.5.2. Code Beschreibung

Die folgende Beschreibung bezieht sich auf den Quellcode 15 und 16.

Zu Beginn werden einzelne Header-Dateien eingebunden, um Klassen aus der Namespace `FOAM` verwenden zu können. Anschließend erfolgt ab Zeile 36 die Definition der Hauptfunktion des Programms. Diese Funktion generiert beim Aufruf eine willkürliche Raumkoordinate und prüft, ob an dieser Position eine numerische Zelle vorliegt. Falls dies zutrifft, wird diese in ein sogenanntes `cellSet` gespeichert; sofern die Zelle noch nicht vorhanden ist. Tritt der Fall ein, dass keine numerische Zelle gefunden wird, oder die ausgewählte Zelle bereits registriert ist, wird die Funktion ohne weitere Tätigkeiten verlassen. Die genaue Arbeitsweise dieser Funktion kann im Selbststudium erhalten werden.

Es ist erwähnenswert, dass diese Applikation einen `trueRandomMode` Schalter besitzt. Wird dieser aktiviert, kann eine bereits erhaltene Verteilung nicht reproduziert werden.

```

29  \*-----*/
30
31  #include "fvCFD.H"
32  #include "cellSet.H"
33
34  // * * * * * //
35
36  void findAndAppend
37  (
38      label& i ,
39      labelList& cells ,
40      List<point>& points ,
41      const point& ptMin ,
42      const point& ptMax ,
43      label& seed ,
44      const fvMesh& mesh ,
45      const bool trueRandMode ,
46      Random& randObj
47  )
48  {
49      // Get a random position
50      const vector randPos = randObj.position(ptMin, ptMax);
51
52      // Checking if point is inside the mesh
53      const label cellLabel = mesh.findCell(randPos);
54
55      if (cellLabel == -1)
56      {
57          //- Position is not in the mesh. Reduce search label
58          i--;
59      }
60      else
61      {
62          // Position is in the mesh.
63          // Append cell label to list if not already there
64          bool exist = false;
65
66          // Check if the cell is already in the set
67          forAll(cells, j)
68          {
69              if (cells[j] == cellLabel)
70              {
71                  // Position is already available, leave and decrement i
72                  i--;
73                  exist = true;
74                  break;
75              }
76          }

```

Quellcode 15: Teil 1: Definition der Helferfunktion findAndAppend.

```

77     // Point is not in the list so add it
78     if (!exist)
79     {
80         // Reset the X vector after one point was found. A bit tricky
81         // to explain. After we build randObj with seed, we have a
82         // defined X vector which is random but fixed. Tests showed
83         // that with time(NULL) we almost get the same distribution.
84         // If we put it after the forAll loop, it takes too long
85         // because initializing the osRandomSeed(seed) object which
86         // actually calls the srand48(seed) function takes a while
87         if (trueRandMode)
88         {
89             seed = time(NULL);
90             osRandomSeed(seed);
91         }
92         cells.append(cellLabel);
93         points.append(randPos);
94     }
95 }
96 }

```

Quellcode 16: Teil 2: Definition der Helferfunktion `findAndAppend`.

Der Kommentar in Zeile 80 gibt hierzu weitere Informationen. Die Helferfunktion wird vom Hauptprogramm `int()` ausgeführt, welches ab Zeile 105 startet. Beginnend mit der Einbindung einiger Header Dateien wird die *BoundingBox* ermittelt, welche geometrischen Abmessungen enthält. Das Objekt `globalBd` enthält alle Informationen über die Größe des numerischen Netzes und wird verwendet, um die Koordinaten der Punkte `pMin` und `pMax` zu ermitteln.

```

105 int main(int argc, char *argv [])
106 {
107     #include "setRootCase.H"
108     #include "createTime.H"
109     #include "createMesh.H"
110
111     // * * * * * //
112     const boundingBox& globalBb = mesh.bounds();
113     point ptMin = globalBb.min();
114     point ptMax = globalBb.max();
115     ~

```

Quellcode 17: Teil 1: Hauptprogramm der `cellListGeneration` Applikation.

Anschließend wird ein Objekt, das Zugriff auf die Datei `cellListGenerationDict` besitzt, erstellt. Die Variable stellt das Interface zwischen Programm und Benutzer dar, da mit dessen Hilfe die notwendigen Variablen, die zur Steuerung des Programms benötigt werden, aus der Datei `cellListGenerationDict` gelesen werden. Die Initialisierung der für das Programm benötigten Größen erfolgt ab Zeile 128 bis 144. Im Anschluss werden einige Kontrollstrukturen durchlaufen, die Falscheingaben behandeln. Ab Zeile 158 beginnt die Definition und Initialisierung weiterer Objekte, die für das Programm erforderlich sind. Der Aufruf der oben gezeigten Helferfunktion

erfolgt in den Zeilen 171–245, die nicht abgebildet sind.

Abschließend werden die Indexnummern und Kartesischen Koordinaten aller Zellen, welche im *cellSet* gespeichert sind, in einer Datei abgespeichert und das Programm beendet; vergleiche Quellcode 19.

```
116 IOdictionary cellListGenerationProperties
117 (
118     IOobject
119     (
120         "cellListGenerationDict",
121         runTime.system(),
122         mesh,
123         IOobject::MUST_READ,
124         IOobject::NO_WRITE
125     )
126 );
127
128 const label nPoints =
129     readLabel(cellListGenerationProperties.lookup("nPoints"));
130
131 const label nAdditionalPoints =
132     readLabel(cellListGenerationProperties.lookup("nAdditionalPoints"));
133
134 label seed =
135     readLabel(cellListGenerationProperties.lookup("seed"));
136
137 const bool trueRandMode =
138     readBool(cellListGenerationProperties.lookup("trueRandomMode"));
139
140 const bool extendedPositions =
141     readBool(cellListGenerationProperties.lookup("extendedPositions"));
142
143 bool specialRegion =
144     readBool(cellListGenerationProperties.lookup("ownRegion"));
145
146 //- For user fail input
147 if (extendedPositions)
148 {
149     specialRegion = false;
150 }
151
152 if (specialRegion)
153 {
154     ptMin = cellListGenerationProperties.lookup("min");
155     ptMax = cellListGenerationProperties.lookup("max");
156 }
157
158 //- Definitionen von weiteren Objekten
159 ...
```

Quellcode 18: Teil 2: Hauptprogramm der *cellListGeneration* Applikation.

```

250 // Construct a cellSet
251 cellSet arbitraryCellSet
252 (
253     mesh,
254     "arbitraryPoints",
255     cells,
256     IOobject::AUTO_WRITE
257 );
258
259 // Write positions to a file
260 {
261     ofstream os("constant/arbitraryPositions");
262
263     os.write("//_Arbitrary_positions\n");
264     os.write("//_Cell_ID_____Position_(x_y_z)\n");
265     os.write("//=====\\n\\n");
266
267     forAll(points, pt)
268     {
269         const scalar x = points[pt].x();
270         const scalar y = points[pt].y();
271         const scalar z = points[pt].z();
272         const label cellID = cells[pt];
273
274         os.write(cellID);
275         os.write("_");
276         os.write(x);
277         os.write("_");
278         os.write(y);
279         os.write("_");
280         os.write(z);
281         os.write(")\n");
282     }
283 }
284
285 // Explicit writing
286 runTime.writeNow();
287
288 Info<< nl << "ExecutionTime_=_ " << runTime.elapsedCpuTime() << "_s"
289     << "_ClockTime_=_ " << runTime.elapsedClockTime() << "_s"
290     << nl << endl;
291
292 Info<< "End\n" << endl;
293
294 return 0;
295 }

```

Quellcode 19: Teil 3: Hauptprogramm der cellListGeneration Applikation.

## E.6. MatCalcPreparation C++ Code Beschreibung

Nachfolgend wird auf die wichtigsten Aspekte der `MatCalcPreparation` Applikation eingegangen. Das Programm wandelt `OpenFOAM`<sup>®</sup> generierte Temperaturdaten in ein für `MatCalc` lesbares Format um. Der Quellcode ist auf dem Datenträger unter `Doktorarbeit/Programming` zu finden.

### E.6.1. Der Strukturbaum

Die Struktur der `MatCalcPreparation` Applikation unterscheidet sich zu den bereits diskutierten Programmen, da hier eine eigene Namespace `AOPC` entworfen wurde. Diesbezüglich wird dieses Programm mit dem Standard C++ Compiler kompiliert, wodurch die `makefile` Datei erforderlich wird.

Des Weiteren werden eigene Typ-Definitionen und Funktionen programmiert, die zur Datenmanipulationen herangezogen werden.

```
MatCalcPreparation
|-- makefile
|-- MatCalcPreparation.cpp
|-- typedef
    |-- typedef.cpp
    |-- typedef.hpp
|-- functions
    |-- functions.cpp
    |-- functions.hpp

2 directory , 6 files
```

Quellcode 20: Struktur der Applikation `MatCalcPreparation`.

### E.6.2. Code Beschreibung

Nachfolgend wird der Quellcode — gekürzt — diskutiert. Die Quelldatei `MatCalcPreparation.C` beginnt mit dem Einbinden der eigenen Funktionen und einer Standard C++ Bibliothek. Anschließend wird die Verwendung der `AOPC` Namespace definiert. In Zeile 29 wird die `Header()` Funktion aufgerufen, welche eine Terminalausgabe erzeugt. Anschließend werden die an die Applikation übergebenen Argumente geprüft.

In Zeile 38 wird die zu konvertierende Datei gelesen, diverse Manipulationen durchgeführt und in einem String-Array abgespeichert. In Zeile 54 werden die neuen Daten in eine Datei abgespeichert. Anschließend kann diese Datei in `MatCalc`— als Temperaturprofil — importiert werden. Die genaue Datenmanipulation ist in der Datei namens `function.cpp` zu finden. Aufgrund ihrer Größe wird diese nicht angegeben. Der interessierte Leser wird auf den Quellcode, welcher auf dem Datenträger zu finden ist, verwiesen.



## E.7. YieldStrengthLaplacianFoam C++ Code Beschreibung

Die Applikation `yieldStrengthLaplacianFoam` löst im Prinzip eine Poisson-Gleichung der Form  $\mathbf{Ax} = \mathbf{b}$ , wobei die Matrix  $\mathbf{A}$  so manipuliert wird, dass sich das System auf eine Laplace-Gleichung vereinfachen lässt ( $\mathbf{Ax} = 0$ ). Der Löser wird verwendet, um die Streckgrenz-Informationen, welche nur an diskreten Punkten berechnet werden, auf ein 3D Feld zu extrapolieren. Die wichtigsten Codeabschnitte werden nachfolgend diskutiert.

### E.7.1. Der Strukturbaum

Der Aufbau des Programms ist im Strukturbaum 22 angegeben. Er besteht aus der Quell- und mehreren Header-Dateien sowie dem `Make` Ordner.

```
yieldStrengthLaplacianFoam
|-- createFields.H
|-- residualControl.H
|-- write.H
|-- yieldStrengthLaplacianFoam.C
|-- Make
    |-- files
    |-- options

1 directory , 6 files
```

Quellcode 22: Struktur der Applikation `yieldStrengthLaplacianFoam`.

### E.7.2. Code Beschreibung

Nach dem Aufruf der Applikation werden verschiedene Header-Dateien geladen, die verschiedene Objekte erstellen, welche für die spätere Berechnung benötigt werden. Die vom Programm einbezogenen Dateien in Zeile 42–43 erzeugen diverse Objekt, die Informationen zur Zeitintegration und dem numerischen Netz enthalten. In Zeile 44 werden weitere Objekte erzeugt, die für die Arbeitsweise des Algorithmus verantwortlich sind.

Anschließend werden weitere Objekte und Kontrollstrukturen durchlaufen.

```
38 int main(int argc, char *argv[])
39 {
40     #include "setRootCase.H"
41
42     #include "createTime.H"
43     #include "createMesh.H"
44     #include "createFields.H"
45
46     //- Weitere Objekte und Kontrollstrukturen
47     ...
```

Quellcode 23: Erstellung der für die Simulation notwendigen Objekte.



Ab Zeile 71 wird die Laplace-Gleichung, die der Form

$$\frac{\partial \phi}{\partial t} - \nabla \bullet (D \nabla \phi) = 0, \quad (\text{E.11})$$

entspricht, in ein lineares Gleichungssystem (LG) überführt. In Zeile 76 wird die Matrix **A** manipuliert und anschließend relaxiert — sofern eingestellt — und das LG gelöst. Da hier die Lösung des stationären Zustands gesucht wird, erfolgt der iterative Prozess, bis dieser gefunden ist.

```

64  // - Outer corrector loop
65  while (!steadyState && (nCorr < nCorrTYS))
66  {
67      nCorr++;
68
69      SolverPerformance<scalar> solvPerf;
70
71      fvScalarMatrix TYSEqn
72      (
73          fvm::ddt(TYS) - fvm::laplacian(D, TYS)
74      );
75
76      TYSEqn.setValues(cellIDs, cellValues);
77
78      TYSEqn.relax();
79
80      solvPerf = TYSEqn.solve();
81
82      #include "residualControl.H"
83
84      if (solvPerf.initialResidual() < TYSRes)
85      {
86          Info<< "Steady_state_solution_found.\n" << endl;
87          steadyState = true;
88          TYS.write();
89          break;
90      }
91  }

```

Quellcode 24: Iteratives Lösen der Laplace Gleichung bis zur Stationarität.

Abschließend, sofern gewünscht, wird eine volumengewichtete integrale Festigkeit einer vordefinierten Zone berechnet. Alle Volumina die sich in dieser Zone befinden werden für die Berechnung des Festigkeitswerts  $\bar{R}_{p,0.2}$  herangezogen. Die Berechnungsformulierung lautet:

$$\bar{R}_{p,0.2} = \frac{\sum_{j=1}^m \phi[\text{ID}_j] V[\text{ID}_j]}{\sum_{j=1}^m V[\text{ID}_j]}. \quad (\text{E.12})$$

Die Laufvariable  $j$  läuft von Eins bis  $m$  und definiert dadurch die Anzahl der an der Integration beteiligten Zellen.  $[\text{ID}]_j$  gibt die Zellen ID an. Der C++ Code ist nachfolgend gegeben. Die Berechnung erfolgt mit einem eigenen Programm namens **averageYS**. Aufgrund der Trivialität wird dieses Programm nicht weiter erläutert. Der interessierte Leser wird auf die den Quellcode

auf dem Datenträger hingewiesen.

```
121     scalar volTimesYS = 0;
122     scalar sumV = 0;
123     const scalarField& V = mesh.V();
124
125     forAll(cellID , i)
126     {
127         const label ID = cellID [i];
128
129         sumV += V[ID];
130
131         volTimesYS += V[ID] * TYS[ID];
132     }
133
134     scalar volAveYS = volTimesYS / sumV;
```

Quellcode 25: Implementierung der Berechnung der volumengewichteten integralen Festigkeit.

## F. Zusatzinformationen zu den Ergebnissen

Die Analyse der thermischen Spannungsmodellierung zeigt auf, dass die Spannungsberechnung abgeschaltet werden kann, sofern dafür Sorge getragen wird, dass die thermisch induzierten Spannungen während der lokalen Wärmebehandlung stets unterhalb der Streckgrenze  $R_{p,0.2}$  liegen. Diesbezüglich wird der Intensitätsfaktor, welcher den Energieeintrag steuert, im Falle des Motorengehäuses für die ersten 20 s mit einer Punktliste gesteuert. Die verwendeten Datensätze sind in den Beispielen auf dem Datenträger unter `Doktorarbeit/processChain/` zu finden. Es ist erwähnenswert, dass eine lineare Interpolation für den Intensitätsfaktor zwischen zwei diskreten Punkten stattfindet.

Die Generierung basiert auf einem Polynom zweiten Grades,  $\psi(t) = Ct^2 + Bt + A$ , mit den Vorfaktoren:

$$C = -0,104 \quad [\text{s}^{-2}] , \quad B = 0,420 \quad [\text{s}^{-1}] , \quad A = 0,000 \quad [-] ,$$

und fünf weiteren Punkten:

$$t = 4 \text{ s} \longrightarrow \psi = 0,200 ,$$

$$t = 6 \text{ s} \longrightarrow \psi = 0,195 ,$$

$$t = 8 \text{ s} \longrightarrow \psi = 0,190 ,$$

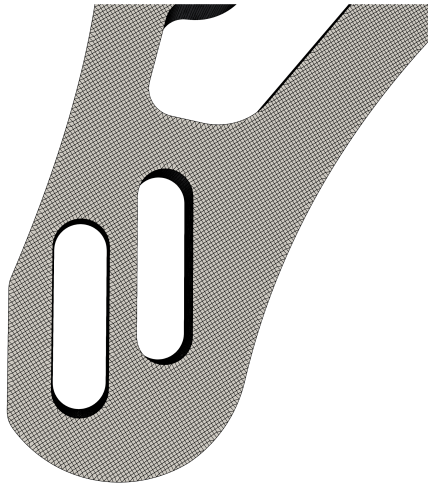
$$t = 20 \text{ s} \longrightarrow \psi = 0,185 ,$$

$$t = 21 \text{ s} \longrightarrow \psi = 0,185 .$$

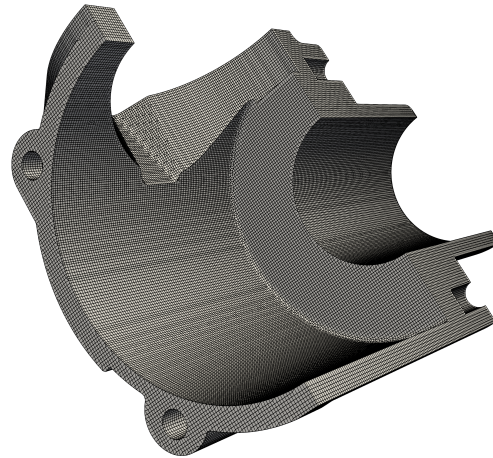
Die Punktliste wird anschließend durch die Diskretisierung des Polynoms zweiten Grades, mit  $\Delta t = 0,01 \text{ s}$ , bis zur Zeit  $t = 2 \text{ s}$  erhalten. Anschließend werden die fünf weiteren Punkte angehängt.

## F.1. Numerische Netze des Motorengehäuses und der Dämpferbrücke

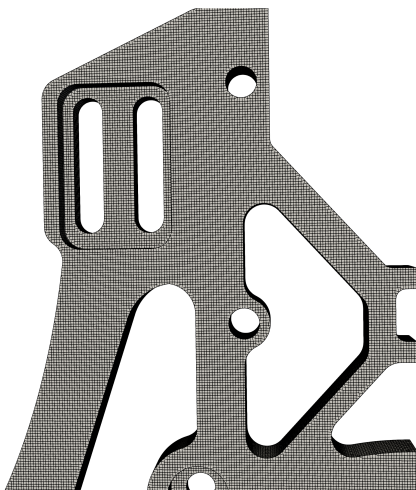
Die numerische Diskretisierung ist in Abbildung F.1 gegeben. Da die Anzahl der Volumina mit über 600000 für diese Strukturbauteile sehr groß ist, die Darstellungen der numerischen Zellen nicht eindeutig ersichtlich ist, wird der interessierte Leser zur Betrachtung der Netze auf eine direkte Betrachtung in ParaView<sup>®</sup> hingewiesen.



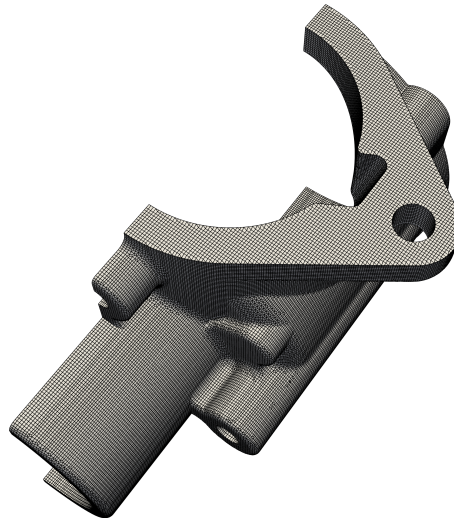
(a) Dämpferbrücke Ausschnitt 1



(b) Motorengehäuse Innenleben



(c) Dämpferbrücke Ausschnitt 2

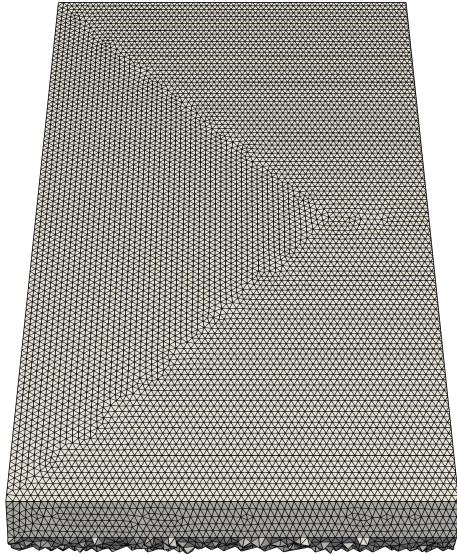


(d) Motorengehäuse Draufsicht

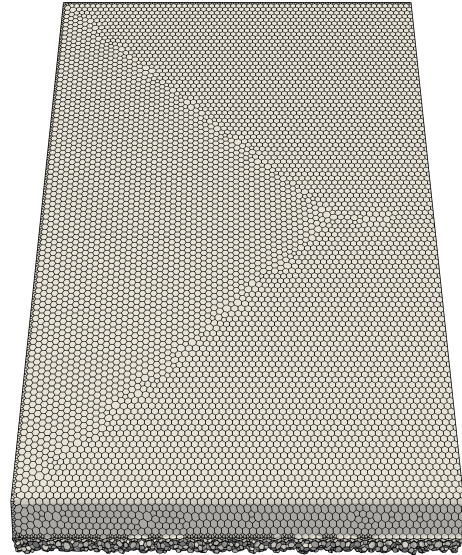
Abbildung F.1.: Visualisierung der numerischen Diskretisierung der Dämpferbrücke und des Motorgehäuses; eine genaue Analyse der Netzstruktur kann nur mit ParaView<sup>®</sup> getätigt werden.

## F.2. Numerische Netze der Stahlplatten

Für die Simulation der Stahlplatten wird ein Tetraedernetz mit der Software Salome® erstellt. Ein Ausschnitt ist in Abbildung F.2a gegeben. Mittels der Applikation `polyDualMesh` wird dieses Netz anschließend in ein Polyedernetz transformiert; vergleiche Abbildung F.2b.



(a) Tetraedernetz



(b) Polyedernetz

Abbildung F.2.: Netzausschnitt. Dargestellt ist das erstellte Tetraedernetz welches in ein Polyedernetz mithilfe der Applikation `polyDualMesh` überführt wird..



## G. Stoff- und Materialdaten

Für die Berechnung der Polynomkoeffizienten für die Stoffwerte von Luft und den Materialeigenschaften für AlSi7Mg0.3 sind nachfolgend die Rohdaten gegeben. Die Stoffwerte der Luft sind aus Gnielinski et al. (2005) bei 1 bar entnommen. Die Materialwerte für die Aluminiumlegierung stammen aus Flender (2015).

### G.1. Stoffwerte für Luft

Temperatur [°C]	Dichte [kgm <sup>-3</sup> ]	Wärmekapazität x 10 <sup>3</sup> [Jkg <sup>-1</sup> K <sup>-1</sup> ]	Viskosität x 10 <sup>-6</sup> [m <sup>2</sup> s <sup>-1</sup> ]	Wärmeleitfähigkeit x 10 <sup>3</sup> [Wm <sup>-1</sup> K <sup>-1</sup> ]
-50	1,5632	1,0061	14,614	20,416
-30	1,4340	1,0058	15,680	22,023
-20	1,3771	1,0057	16,201	22,811
-10	1,3245	1,0058	16,714	23,590
0	1,2758	1,0059	17,218	24,360
10	1,2306	1,0061	17,715	25,121
20	1,1885	1,0064	18,205	25,873
30	1,1492	1,0067	18,689	26,618
40	1,1124	1,0071	19,165	27,354
50	1,0779	1,0077	19,635	28,082
60	1,0455	1,0082	20,099	28,804
70	1,0150	1,0089	20,557	29,518
80	0,9862	1,0097	21,009	30,225
90	0,9590	1,0105	21,455	30,925
100	0,9333	1,0115	21,896	31,620
120	0,8858	1,0136	22,763	32,989
140	0,8428	1,0160	23,610	34,336
160	0,8039	1,0188	24,439	35,660
180	0,7684	1,0218	25,251	36,964
200	0,7359	1,0252	26,046	38,248
250	0,6655	1,0347	27,970	41,382
300	0,6075	1,0454	29,811	44,417
350	0,5587	1,0568	31,579	47,367
400	0,5172	1,0688	33,284	50,240
450	0,4815	1,0808	34,932	53,047
500	0,4503	1,0927	36,530	55,795
550	0,4230	1,1043	38,084	58,490
600	0,3988	1,1154	39,597	61,139
650	0,3772	1,1260	41,073	63,745
700	0,3578	1,1361	42,517	66,312
750	0,3403	1,1455	43,931	68,846

Tabelle G.1.: Stoffwerte der Luft bei 1013,15 hPa. Entnommen aus Gnielinski et al. (2005).

## G.2. Materialwerte für AlSi7Mg0.3

Temperatur [°C]	Dichte [kgm <sup>-3</sup> ]	Wärmekapazität [Jkg <sup>-1</sup> K <sup>-1</sup> ]	Wärmeleitfähigkeit [Wm <sup>-1</sup> K <sup>-1</sup> ]
1	2670	870	151
100	2650	898	152
200	-	941	154
400	-	1058	157
500	-	1203	169
542	2560	1279	166
552	2556	-	-
561	2551	-	-
562	2547	-	-
563	2521	-	-
565	2508	-	-
570	2504	-	-
578	2495	-	-
591	2482	-	-
592	2478	-	-
602	2469	-	-
608	2460	-	-
609	2456	-	-
611	2443	-	-
613	2430	1168	68
700	-	1149	69
2000	2092	1149	70

Tabelle G.2.: Temperaturabhängige Materialwerte für AlSi7Mg0.3. Werte entnommen aus Flen-  
der (2015).